

UNIVERSITÉ DE MONTRÉAL

OPTIMISATION DE LA GESTION DU COMBUSTIBLE DANS LES
RÉACTEURS CANDU REFROIDIS À L'EAU LÉGÈRE

RICHARD CHAMBON
DÉPARTEMENT DE GÉNIE PHYSIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE NUCLÉAIRE)
AOÛT 2006

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée:

OPTIMISATION DE LA GESTION DU COMBUSTIBLE DANS LES
RÉACTEURS CANDU REFROIDIS À L'EAU LÉGÈRE

présentée par: CHAMBON Richard

en vue de l'obtention du diplôme de: Philosophiæ Doctor

a été dûment acceptée par le jury d'examen constitué de:

M. HÉBERT Alain, D.Ing, président

M. ROZON Daniel, Ph.D., membre et directeur de recherche

Mme. VARIN Élisabeth, D.Sc, membre et codirecteur de recherche

M. ROY Robert, Ph.D., membre

M. NICHITA Eleodor, Ph.D., membre

REMERCIEMENTS

Je voudrais en premier lieu remercier mes directeurs de thèse Daniel Rozon, PhD. et Dr. Elisabeth Varin de m'avoir encadré dans ma recherche. Je tiens à souligner tout particulièrement le soutien que m'a fourni Dr. Elisabeth Varin à travers toutes les réunions et discussions que nous avons eues. Je la remercie également pour sa grande disponibilité.

Je voudrais également remercier les membres du jury qui ont bien voulu prendre le temps de lire ce mémoire.

Je tiens aussi à remercier Guy Marleau, PhD ainsi que Romain Letellier, étudiant au doctorat, tout d'abord pour leur aide avec DRAGON, Latex et "linuxerie" en tous genres.

Je voudrais aussi remercier tous les gens qui ont rendu mon séjour à l'institut agréable, dont notamment Elisabeth Varin, Guy Marleau, Romain Letellier, Paul Dufour, Armando Nava, ... pour des discussions lors du lunch sur des sujets scientifiques et surtout autres.

Un remerciement va aussi à Alberto Teyssedou, PhD. et Daniel Rozon, PhD. pour les excellents cours que j'ai suivi avec eux, et la curiosité qu'ils ont su faire développer en moi pour le domaine du nucléaire en général.

Je tiens également à souligner le support financier de la chaire Hydro-Québec en génie nucléaire et le CRD-AECL sans qui ce travail n'aurait pas vu le jour.

Enfin, je voudrais adresser tous mes remerciements à tous mes amis qui m'ont motivé à continuer et rester au Canada, et qui supportent encore mes idées folles d'avenues de recherche, même s'ils ne sont pas toujours scientifiques dans l'âme.

RÉSUMÉ

Le travail accompli dans le cadre de ce projet de doctorat comporte deux aspects. Premièrement, il consiste à introduire dans le code diffusion DONJON des capacités d'optimisation, en vue de les appliquer à la gestion du combustible. Ensuite, le deuxième aspect, plus pratique, a pour but d'optimiser la gestion du combustible à l'équilibre du rechargement de différents réacteurs CANDU.

Deux types d'approches ont été envisagées et implantées pour résoudre des problèmes d'optimisation dans le code DONJON. Les premières méthodes sont basées sur les gradients et la programmation mathématique quasilinéaire. La méthode développée initialement dans le code OPTTEX a été implantée comme l'approche de base pour les méthodes de gradients. Cependant, cette approche présente une restriction majeure, à savoir que l'estimé initial doit toujours être un point réalisable. Plusieurs approches plus générales ont été proposées pour outrepasser cette limitation. Parmi celles-ci, les méthodes dites multi-étapes et mixte se sont avérées très efficaces.

Le deuxième type d'approches envisagées sont les méthodes méta-heuristiques. La recherche tabou a été implantée. Initialement celle-ci a été initialement développée pour des variables combinatoires. Nous l'avons cependant implantée pour des variables continues avec succès. En opposition aux méthodes de gradients, cette méthode permet de sortir de minima locaux.

L'optimisation de la distribution du burnup moyen de sortie a été envisagée pour les réacteurs CANDU-6 et ACR-700. L'enrichissement initial a également été optimisé pour le ACR-700. Les résultats obtenus s'accordent très bien avec la physique des réacteurs. De plus, la comparaison des deux types de méthodes a permis de valider les optimums obtenus.

ABSTRACT

This research has two main goals. First, we wanted to introduce optimization tools in the diffusion code DONJON, mostly for fuel management. The second objective is more practical. The optimization capabilities are applied to the fuel management problem for different CANDU reactors at refueling equilibrium state.

Two kinds of approaches are considered and implemented in this study to solve optimization problems in the code DONJON. The first methods are based on gradients and on the quasi-linear mathematical programming. The method initially developed in the code OPTEX is implemented as a reference approach for the gradient based methods. However, this approach has a major drawback. Indeed, the starting point has to be a feasible point. Then, several approaches have been developed to be more general and not limited by the initial point choice. Among the different methods we developed, two were found to be very efficient: the multi-step method and the mixte method.

The second kind of approach are the meta-heuristic methods. We implemented the tabu search method. Initially, it was designed to optimize combinatory variable problems. However, we successfully used it for continuous variables. The major advantage of the tabu method over the gradient methods is the capability to exit from local minima.

Optimisation of the average exit burnup has been performed for CANDU-6 and ACR-700 reactors. The fresh fuel enrichment has also been optimized for ACR-700. Results match very well what the reactor physics can predict. Moreover, a comparison of the two totally different types of optimization methods validated the results we obtained.

TABLE DES MATIÈRES

REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vi
TABLE DES MATIÈRES	vii
LISTE DES FIGURES	viii
LISTE DES TABLEAUX	ix
LISTE DES ANNEXES	x
CHAPITRE 1 INTRODUCTION	1
1.1 Optimisation de la gestion du combustible	1
1.2 Cadre de la recherche	5
1.3 Organisation du rapport	7
CHAPITRE 2 APPLICATION DE LA THÉORIE DES PERTURBATIONS GÉNÉRALISÉES À LA GESTION DU COMBUSTIBLE DANS LES RÉACTEURS CANDU	8
2.1 Introduction	8
2.2 Définition des calculs moyennés dans le temps	9
2.3 Calculs de références	10
2.4 Objectif des opérateurs	12
2.5 Optimisation de la référence	13
2.6 Paramètres d'optimisation	14
2.7 But de l'optimisation	17

2.8	Contraintes	20
2.8.1	Criticité du réacteur	20
2.8.2	Puissance canal maximum	21
2.8.3	Réactivité des barres de compensation	22
2.9	Méthodes d'optimisation utilisées	23
2.9.1	Introduction	23
2.9.2	Théorie des perturbations du premier ordre	26
2.9.3	Théorie des perturbations généralisées	28
2.9.4	Méthodes de programmation quasi-linéaire simples	32
CHAPITRE 3	IMPLANTATION DES MÉTHODES DE GRADIENTS DANS DONJON	35
3.1	Algorithme général d'optimisation avec la théorie des perturbations généralisées	36
3.1.1	Description de l'algorithme	36
3.1.2	Validation du nouveau point et convergence	39
3.1.3	Vérification successive des étapes	42
3.2	Précision du calcul des gradients	43
3.2.1	Introduction	43
3.2.2	Théorie	44
3.2.3	Comparaison et résultats	46
3.3	Vérification de la convergence des résultats	51
3.3.1	Première optimisation d'un cas simple	51
3.3.2	Influence du nombre de zones de combustion optimisées	54
CHAPITRE 4	IMPLANTATION DE MÉTHODES DE GRADIENTS AL- TERNATIVES	58
4.1	Introduction	58
4.2	Méthode multi-étapes	59

4.2.1	Contrainte sur la réactivité	59
4.2.2	Contraintes de puissance maximum	61
4.2.3	Conclusions	67
4.3	Méthodes duales	69
4.3.1	Théorie	69
4.3.2	Algorithme implanté	71
4.4	Comparaison des méthodes	81
4.4.1	Cas à 2 zones	81
4.4.2	Cas à N zones de combustion	83
4.4.3	Influence de la répartition des zones	87
4.4.4	Influence du poids des contraintes et effort de calcul	91
4.4.5	Influence du point de départ	94
4.4.6	Conclusions	96
4.5	Méthode mixte	98
4.5.1	Définition	98
4.5.2	Résultats de la fonction coût	99
4.5.3	Effort de calcul	102
4.5.4	Résultats de la distribution de burnup	103
4.6	Conclusion de l'implantation	105

CHAPITRE 5 GESTION DU COMBUSTIBLE DU ACR-700 PAR MÉTHODES DE GRADIENTS 107

5.1	Description du “Advanced CANDU Reactor”	107
5.2	Réactivité du vide	110
5.2.1	Définition	112
5.2.2	Calcul moyenné dans le temps	113
5.2.3	Approximation du calcul	116
5.2.3.1	Théorie	116

5.2.3.2	Résultats	122
5.2.3.3	Conclusions	124
5.3	Optimisation en burnup	125
5.3.1	Influence des paramètres d'optimisation	125
5.3.2	Effort de calcul	130
5.3.3	Analyse de sensibilité de la réactivité du vide	132
5.3.4	Influence de la puissance canal maximum permise	134
5.3.5	Conclusions	135
5.4	Dépendance en enrichissement des sections efficaces	137
5.4.1	Implantation	137
5.4.2	Vérification de l'implantation	139
5.4.2.1	Description du cas envisagé	139
5.4.2.2	Résultats	139
5.4.3	Influence du poids des variables	141
5.5	Optimisation du burnup moyen de sortie et de l'enrichissement	147
5.5.1	Influence du poids des variables	149
5.5.2	Fonction objectif en fonction de l'enrichissement	153
5.5.2.1	Résultats intermédiaires	153
5.5.2.2	Théorie	156
5.5.3	Résultats optimaux	160
5.5.4	Contrainte sur la réactivité du vide	163
5.5.4.1	Méthode MS	164
5.5.4.2	Méthode MM	166
5.6	Conclusion	168
CHAPITRE 6 MÉTHODES MÉTAHEURISTIQUES		170
6.1	Principe général des méthodes métaheuristiques	171
6.1.1	Algorithmes génétiques	171

6.1.2	Recuit simulé	173
6.1.3	Recherche tabou	175
6.1.4	Considérations générales	176
6.2	Applications à la gestion du combustible	176
6.2.1	Algorithmes génétiques	177
6.2.2	Recuit simulé	178
6.2.3	Recherche tabou	179
6.3	Recherche tabou en variables continues	180
6.4	Implantation de la méthode de recherche tabou	182
6.4.1	Initialisation	185
6.4.2	Exploration	185
6.4.3	Intensification	187
6.4.4	La méthode de Nelder-Mead	187
6.5	Banc d'essais de la méthode recherche tabou	190
6.5.1	Description détaillée des approches	191
6.5.2	Résultats et conclusions	195
CHAPITRE 7 APPLICATION DE LA MÉTHODE TABOU AU ACR-700200		
7.1	Choix des paramètres	200
7.1.1	Point initial	200
7.1.2	Poids relatif des variables	201
7.1.3	Compromis statistique	202
7.1.4	Poids des contraintes	202
7.2	Optimisation en burnup	204
7.3	Optimisation en burnup et en enrichissement	210
CHAPITRE 8 CONCLUSIONS ET RECOMMANDATIONS		
8.1	Globalité de l'implantation	212
8.2	Généralité des algorithmes développés	213

8.3	Application aux réacteurs CANDU	215
8.4	Limitations du problème original	217
8.5	Recommandations	218
8.6	Travaux futurs	221
	RÉFÉRENCES	223
	ANNEXES	232

LISTE DES FIGURES

Figure 2.1	Schéma d'un réacteur CANDU-6	11
Figure 2.2	Exemple de rechargement d'un canal, 8 grappes à la fois (<i>8 bundle shift</i>)	12
Figure 3.1	Algorithme général pour l'optimisation avec la méthode des gradients.	40
Figure 3.2	Variation de F_C avec le burnup moyen de sortie de la zone centrale.	49
Figure 3.3	Variation de k_{eff} avec le burnup moyen de sortie de la zone centrale.	50
Figure 3.4	Variation de q_{19} avec le burnup moyen de sortie de la zone centrale.	50
Figure 3.5	Répartition radiale des zones de combustion pour différents nombres de zones.	53
Figure 3.6	Impact de la contrainte de puissance canal maximum sur l'optimisation du burnup moyen de sortie à 2 zones dans un réacteur CANDU-6 sans barres de compensation.	53
Figure 3.7	Variation de la distribution du burnup moyen de sortie à n zones pour différentes contraintes de puissance canal maximum (L11 centre - L22 périphérie).	56
Figure 3.8	Burnup moyen de sortie du cœur en fonction du nombre de zones de combustion pour différentes contraintes de puissance canal.	57
Figure 3.9	Distribution radiale du burnup moyen de sortie obtenue par Wight.	57
Figure 4.1	Puissance canal maximum en fonction du rapport des burnups moyens de sortie	63

Figure 4.2	Norme $2N$ de la distribution de puissance canal en fonction du ratio des burnups moyen de sortie, méthode 1	64
Figure 4.3	Norme $2N$ de la distribution de puissance canal en fonction du ratio des burnups moyen de sortie, méthode 2	65
Figure 4.4	Norme $2N$ de la distribution de puissance canal en fonction du ratio des burnups moyen de sortie, méthode 3	66
Figure 4.5	Norme $2N$ de la distribution de puissance canal en fonction du ratio des burnups moyen de sortie, méthode 4	67
Figure 4.6	Algorithme général pour l'optimisation avec la méthode du lagrangien augmenté ou de pénalisation.	73
Figure 4.7	Calcul du pas d'avance avec la méthode du lagrangien augmenté ou de pénalisation.	77
Figure 4.8	Fonction objectif et puissance canal maximum vs. rapport $R = B_2^e/B_1^e$ pour $k_{eff} = 1$	84
Figure 4.9	Distribution des zones de burnup pour différents nombres de zones	87
Figure 4.10	Distribution du burnup et de la puissance canal pour 11 de zones	90
Figure 4.11	Distribution du burnup et de la puissance canal pour 11x2 de zones	90
Figure 4.12	Burnup moyen de sortie \overline{B}^e suivant le nombre de zones pour différentes méthodes d'optimisation	93
Figure 4.13	Variation de $\rho = 1 - 1/k_{eff}$ et $P_{C,mac}$ avec les itérations internes de la méthode LA.	99
Figure 4.14	Distribution du burnup pour le cas à 190 zones.	105
Figure 4.15	Distribution de la puissance canal pour le cas à 190 zones.	106
Figure 5.1	Modèle du ACR-700 à 284 canaux.	108
Figure 5.2	Modèle du ACR-700 à 300 canaux.	109

Figure 5.3	Exemple de répartition de zones de burnup de sortie [1], symétrie quart de réacteur	111
Figure 5.4	Distribution des 7 zones de burnup pour le ACR-700.	126
Figure 5.5	Distributions de la puissance canale et du burnup obtenues avec la MM pour 7 zones.	128
Figure 5.6	Distributions de la puissance canale et du burnup obtenues avec la MM pour 150 zones.	129
Figure 5.7	Réactivité du vide cœur complet en fonction des itérations pour la méthode MS.	133
Figure 5.8	Fonction coût vs. Puissance canal maximum	135
Figure 5.9	Profil de la puissance canal vs. Puissance canal maximum	136
Figure 5.10	Schéma du réacteur fictif	139
Figure 5.11	Valeur des fonctionnelles pour le réacteur fictif.	143
Figure 5.12	Valeur de la fonction objectif pour le réacteur fictif dans le domaine réalisable seulement.	144
Figure 5.13	Section efficace totale dans le groupe thermique.	148
Figure 5.14	Fonction objectif optimale vs. Enrichissement optimal.	155
Figure 5.15	Coût du combustible C_F , des matériaux C et fabrication C_{FAB} en fonction de l'enrichissement.	159
Figure 5.16	Rapport R_B et R_C en fonction de l'enrichissement.	160
Figure 5.17	Méthode MS avec la contrainte de réactivité du vide.	167
Figure 6.1	Exemples de répartition des éléments de N dans le voisinage de la solution courante \vec{X}	183
Figure 6.2	Algorithme général pour l'optimisation avec la méthode de recherche tabou.	184
Figure 6.3	Transformations géométriques élémentaires de la méthode Nelder-Mead, pour un cas à 3 variables (Chelouah)	188
Figure 7.1	Fonction objectif f_{TS} selon les itérations	209

LISTE DES TABLEAUX

Tableau 3.1	Option par défaut pour l'optimisation dans le module QLPUTL :	39
Tableau 3.2	Perturbations utilisés pour le calcul numérique des gradients	48
Tableau 3.3	Gradient des fonctionnelles suivant les deux méthodes de calcul et l'amplitude de la variation entre $\pm 20\%$ de perturbation	49
Tableau 3.4	Variation du burnup moyen de sortie du cœur (MWj/t) optimisé pour un réacteur CANDU-6 avec et sans barres de compensation.	57
Tableau 4.1	Burnup moyen de sortie pour que $k_{eff} = k_{ref}$	61
Tableau 4.2	Minimum de la puissance canal maximum obtenue par la méthode 3	68
Tableau 4.3	Résultats pour 2 zones de combustion pour un CANDU-6 avec toutes les barres de contrôle ($\delta = 0.0001$, $S_{min} = 10/5/1$)	85
Tableau 4.4	Résultats pour de plus grands nombres de zones de combustion	88
Tableau 4.5	Burnups pour des zones séparées haut/bas.	89
Tableau 4.6	Function objectif F_C suivant le nombre de zones pour différentes méthodes d'optimisation (sans dimension).	93
Tableau 4.7	Efforts de calculs pour les résultats du tableau 4.6.	94
Tableau 4.8	Distribution du burnup pour 11 zones pour la méthode MS 'hybride' et 'complète'.	96
Tableau 4.9	Distribution du burnup pour 11 zones pour un autre point initial.	96
Tableau 4.10	Fonction coût pour les cas à 2, 11 et 190 zones.	100
Tableau 4.11	Fonction coût pour le cas à 190 zones à partir de $8000 MWj/t$.	101
Tableau 4.12	Nombre d'itérations requises pour le cas à 190 zones.	103
Tableau 5.1	Réactivité du vide pour différents modes de calculs	115
Tableau 5.2	Calcul de la réactivité du vide par quatre méthodes	122

Tableau 5.3	Effort de calcul pour la réactivité du vide par trois méthodes	124
Tableau 5.4	Réactivité inverse du vide par trois méthodes	124
Tableau 5.5	Résultats pour différentes distributions initiales du burnup et différentes méthodes d'optimisation	126
Tableau 5.6	Efforts de calculs pour la résolution du problème d'optimisation.	131
Tableau 5.7	k_{eff} pour différents modes de calculs et enrichissements . . .	141
Tableau 5.8	Résultats d'optimisation pour le réacteur fictif avec la méthode MS	146
Tableau 5.9	Résultats d'optimisation pour le réacteur fictif avec la méthode LA	146
Tableau 5.10	Paramètres pour le calcul du coût des grappes.	148
Tableau 5.11	Résultats à partir de la distribution "AECL".	151
Tableau 5.12	Résultats à partir de la distribution "flat".	151
Tableau 5.13	Effort de calcul	153
Tableau 5.14	Résultats des optimisations complètes ($w_\varepsilon = 10^{12}$).	161
Tableau 5.15	Effort de calcul pour les optimisations complètes.	162
Tableau 5.16	Réactivité du vide pour les configurations optimales obtenues avec MS (mk).	163
Tableau 5.17	Résultats avec le lagrangien augmenté.	168
Tableau 6.1	Résultats de la recherche tabou pour des fonctions analytiques.	198
Tableau 6.2	Moyenne estimée et minimum global pour les fonctions ana- lytiques.	199
Tableau 7.1	Valeur générale des paramètres de la méthode TS selon [2]. . .	205
Tableau 7.2	Optimisation en burnup avec $N_{itmax} = 50$	207
Tableau 7.3	Distribution de burnup obtenue avec la méthode MM pour le cas à 7 zones à partir de la distribution "flat".	207
Tableau 7.4	Optimisation en burnup avec $N_{itmax} = 200$	208

Tableau 7.5	Optimisation en burnup et en enrichissement avec $N_{npd,max} =$	
	16	211
II.1	Structure FOBJCT:	241
II.2	Structure (descfobjct)	242
II.3	Structure (czdf_data)	243
II.4	Structure (fcdf_data)	244
II.5	Structure (cstzdf_data)	245
II.6	Structure (eval_data)	247
II.7	Structure (vardef_data)	250
II.8	Structure (seq_data)	253
II.9	Structure (data)	254
II.10	Structure QLPUTL:	256
II.11	Structure (descqlputl)	257
II.12	Structure (def_data)	260
II.13	Structure (PERTUR:)	264
II.14	Structure (pertur_data)	265
II.15	Structure GPTSRC:	267
II.16	Structure gptsrc_data	268
II.17	Structure GPTFLU:	269
II.18	Structure gptflu_data	270
II.19	Structure GPTGRD:	271
II.20	Structure direct_data	272
II.21	Structure gptgrd_data	273
II.22	Structure QLP:	274
II.23	Structure qlp_data	274
II.24	Structure TABU:	276
II.25	Structure (desctabu)	276
II.26	Structure (def_data)	278

II.27	Structure (nelder_data)	281
II.28	Structure ADDOBJ :	282
II.29	Structure (addmac_data)	283
II.30	Structure (addflu_data)	284
II.31	Structure XSFUEL :	286
II.32	Structure (descxsfuel)	287
II.33	Structure (descintfo)	287
II.34	Structure MATLAB :	290
II.35	Structure (descmatlgrd)	291
II.36	Structure (descmatlfu)	292
II.37	Structure GPTVRF :	294
II.38	Structure gptvrf_data	294
II.39	Structure XSCONS :	297
II.40	Structure (descxsc)	297
II.41	Structure (descint)	298
II.42	Main records and sub-directories in /optimize/	303
II.43	Main records and sub-directories in //OLD-VALUE//	308
II.44	Main records and sub-directories in //stepdir//	309
II.45	/optimize/ in the particular case of module GPTVRF :	310
II.46	The sub-directory /varpdir/ in /optimize/	311
II.47	Main records and sub-directories in /tabu/	314
II.48	Main records and sub-directories in //stepdir//	315
II.49	Main records and sub-directories in /table/	317
II.50	Main records in a fuel type sub-directory	318
II.51	Main records and sub-directories in /map/	321

LISTE DES ANNEXES

ANNEXE I	ALGORITHMES DÉTAILLÉS DES MÉTHODES DE GRADIENTS D'OPTEX	232
	I.1 SIMPLEX	233
	I.2 MAP	236
	I.3 Lemke	237
	I.4 Limitations	240
ANNEXE II	GUIDE DE L'USAGER	241
	II.1 Optimization modules	241
	II.1.1 The FOBJCT: module	241
	II.1.1.1 Data input for module FOBJCT:	242
	II.1.1.2 Data input for module functions definition	247
	II.1.1.3 Examples of function definition	255
	II.1.2 The QLPUTL: module	256
	II.1.2.1 Data input for module QLPUTL:	257
	II.1.3 The PERTUR: module	264
	II.1.3.1 Data input for module PERTUR:	265
	II.1.4 The GPTSRC: module	267
	II.1.4.1 Data input for module GPTSRC:	268
	II.1.5 The GPTFLU: module	269
	II.1.5.1 Data input for module GPTFLU:	270
	II.1.6 The GPTGRD: module	271
	II.1.6.1 Data input for module GTPGRD:	272
	II.1.7 The QLP: module	274
	II.1.7.1 Data input for module PQL:	274
	II.1.8 The TABU: module	276

II.1.8.1	Data input for module TABU :	276
II.2	Other modules	282
II.2.1	The ADDOBJ : module	282
II.2.1.1	Data input for module ADDOBJ :	283
II.2.2	The XSFUEL : module	286
II.2.2.1	Data input for module XSFUEL :	286
II.2.3	The MATLAB : module	290
II.2.3.1	Data input for module MATLAB :	291
II.2.4	The GPTVRF : module	294
II.2.4.1	Data input for module GPTVRF :	294
II.2.5	The XSCONS : module	297
II.2.5.1	Data input for module XSCONS :	297
II.3	Data structure description	301
II.3.1	Contents of a <code>/optimize/</code> data structure	301
II.3.1.1	The sub-directory <code>/OLD-VALUE/</code> in <code>/optimize/</code>	308
II.3.1.2	The sub-directory <code>/stepdir/</code> in <code>/optimize/</code>	309
II.3.1.3	Contents of a <code>/optimize/</code> data structure in the particular case of the module GPTVRF :	310
II.3.2	Contents of a <code>/tabu/</code> data structure	313
II.3.2.1	The sub-directories in <code>/optimize/</code>	315
II.3.3	Contents of a <code>/table/</code> data structure	317
II.3.4	The modifications in previous data structures	321
ANNEXE III	INTERPOLATION DES SECTIONS EFFICACES	322
III.1	Méthode d'Hermite pour une interpolation à une variable	323
III.2	Méthode d'Hermite pour une interpolation à deux variables	325
ANNEXE IV	GÉNÉRATION DE NOMBRES ALÉATOIRES	330

ANNEXE V FONCTIONS ANALYTIQUES TESTS 332

CHAPITRE 1

INTRODUCTION

1.1 Optimisation de la gestion du combustible

La gestion du combustible consiste à bien disposer le combustible dans un réacteur nucléaire pour en maximiser l'énergie extraite. Cela se traduit par différentes variables de décision tels le choix du type de combustible (dont l'enrichissement et la concentration en poison), la fréquence de rechargement, la quantité et l'emplacement du nouveau combustible à chaque rechargement. Deux contraintes primordiales doivent être respectées par le choix approprié de la configuration du combustible.

Premièrement, le réacteur doit pouvoir maintenir un *état critique* pour que la réaction en chaîne soit stable. En effet, si la réaction en chaîne est instable, la puissance dans le réacteur va alors soit augmenter jusqu'à destruction du réacteur, soit diminuer jusqu'à zéro. Les choix faits pour la gestion du combustible doivent donc faire en sorte que le système de régulation du réacteur puisse maintenir l'état critique pendant toute la durée d'opération souhaitée. Deuxièmement, la gestion du combustible doit tenir compte du changement de comportement du cœur (évolution à long terme) tout en assurant la sûreté, c'est à dire tout en respectant les contraintes thermohydrauliques, mécaniques et neutroniques, afin qu'aucune dégradation prématurée des composantes n'advienne. Cependant, pour pouvoir évaluer ces contraintes, le comportement du réacteur doit être calculé, ce qui revient à connaître la distribution du flux de neutrons dans le réacteur. La distribution macroscopique du flux ϕ dans le réacteur est généralement décrite par

l'équation de diffusion stationnaire, donnée par [3]:

$$\begin{aligned}
& -\vec{\nabla} \cdot D(r, E) \vec{\nabla} \phi(r, E) + \Sigma_t(r, E) \phi(r, E) \\
& \quad - \int_0^\infty dE' \Sigma_s(r, E' \rightarrow E) \phi(r, E') \\
& = \frac{1}{k_{eff}} \chi_p(E) \int_0^\infty dE' \nu_p \Sigma_f(r, E', t) \phi(r, E')
\end{aligned} \tag{1.1}$$

où r et E représentent respectivement la position et l'énergie des neutrons, $\phi(r, E)$ le flux neutronique scalaire, $D(r, E)$ le coefficient de diffusion, Σ les sections efficaces macroscopiques, avec les indices t , s et f qui indiquent respectivement les sections efficaces macroscopiques totales, de diffusion et de fission, $\chi(E)$ la densité de probabilité qu'un neutron libéré lors d'une fission ait l'énergie E , ν le nombre moyen de neutrons libérés lors d'une fission, k_{eff} le coefficient de multiplication des neutrons.

Généralement, la dépendance temporelle du flux neutronique n'est pas considérée car de larges intervalles de temps (quelques jours) sont pris en compte pendant lesquels chaque configuration du réacteur est stable (approche quasistatique). Une fois discrétisée spatialement et en énergie, l'équation de diffusion doit rester valide en tout point du réacteur et pour toutes les énergies. Elle est donc résolue pour n inconnues en espace fois ngp inconnues selon l'énergie, et le flux neutronique se présente donc sous la forme d'un vecteur $\vec{\phi}$ de dimension $(n \times ngp)$. Afin d'alléger la notation, nous utiliserons indifféremment ϕ pour la forme continue du flux et ainsi que pour sa représentation discrétisée. L'équation (1.1) se présente alors comme un ensemble d'équations linéaires, qui peuvent être regroupées sous la forme matricielle suivante [3]:

$$(\mathbf{A} - \lambda \mathbf{F}) \phi = 0 \tag{1.2}$$

où \mathbf{A} regroupe l'ensemble des termes de perte de neutrons et \mathbf{F} celui de leur production,

$-\lambda$ est la valeur propre du système. Elle est l'inverse du coefficient de multiplication des neutrons ($= 1/k_{eff}$). Cette constante s'ajuste pour assurer un état stationnaire, i.e. équilibre entre la production et l'élimination des neutrons.

La connaissance de ϕ permet le calcul de la distribution de puissance (en W) dans le réacteur par la formule suivante [4]:

$$P = \int_V H \cdot \phi \, dV \quad (1.3)$$

où H représente le coefficient de production d'énergie par fission ($H = \kappa \Sigma_f$, proportionnelle à la section efficace de fission).

Dans la forme discrète, la définition des intégrales continues de fonctionnelles du flux neutronique sera quant à elle remplacée par une somme sur les groupes d'énergie et en espace, qui est symbolisée par \langle, \rangle . L'équation (1.3) appliquée à la puissance de grappe est alors donnée par [3]:

$$P_{grappe} = \langle H, \phi \rangle_{grappe} \quad (1.4)$$

Les sections efficaces changent avec le temps puisque la composition du combustible varie. Ce processus de changement des propriétés nucléaires du combustible fait changer tous les coefficients de l'équation de diffusion stationnaire (1.1). On traite cette dépendance temporelle grâce au paramètre: *burnup* du combustible, qui représente l'énergie produite par unité de masse. Le burnup (B) d'une grappe

à un temps t est donc donné par:

$$B(t) = \frac{1}{m_{grappe}} \int_0^t P_{grappe}(t') \cdot dt' \quad (1.5)$$

$$\text{donc} \quad \Sigma(t) = \Sigma(B(t)) = \Sigma(B)$$

où m_{grappe} est la masse de la grappe.

Avec cette représentation, l'état du cœur est alors donné par la position de tous les contrôleurs (pour la criticité) et la distribution du burnup (pour le temps).

Pour optimiser la gestion du combustible, la première étape consiste à se donner un ou des objectifs, et d'y associer une fonction de coût F_C . Celle-ci peut être d'ordre économique ou technologique suivant le but visé. La difficulté numérique associée à la résolution de l'équation de diffusion sera alors d'autant plus importante dans l'atteinte d'un optimum que l'optimisation de l'objectif nécessite un nombre important de calculs de réacteur.

Le problème de la gestion du combustible d'un réacteur revient à optimiser les options de rechargement pour minimiser la fonction de coût en respectant plusieurs types de contraintes sur le fonctionnement du réacteur. Ce problème peut alors être représenté conceptuellement sous la forme suivante:

$$\min F_C \quad \text{avec} \quad \begin{cases} k_{eff} = 1 & \text{réacteur critique} \\ P_j \leq P_{j,lim} & \text{contraintes thermohydrauliques} \end{cases} \quad (1.6)$$

où l'indice j représente la distribution de la puissance par région j .

Même si les deux contraintes du problème précédent sont respectées, le choix précis des options de rechargement (enrichissement, fréquence, ...) représente un problème d'une part difficile à résoudre, et d'autre part avec un grand nombre de possibilités

voire une infinité. De plus, le temps de calcul pour la résolution de l'équation (1.6) vient de la taille du problème pour l'évaluation de l'équation de diffusion. En effet, pour les réacteurs CANDU, au moins un point par grappe doit être pris pour la discrétisation spatiale. En tenant compte du réflecteur, cela représente au strict minimum 5000 points. En ce qui concerne l'énergie des neutrons, généralement 2 groupes sont utilisés. Le nombre d'inconnues est donc pour les plus petits cas de l'ordre de 10000, ce qui représente un temps de résolution de l'ordre de 2 à 3 secondes avec un processeur de 3GHz environ. Quand la discrétisation spatiale est augmentée à 4 ou 8 points par grappes le temps de calcul d'un flux avec l'approche moyennée dans le temps est d'environ 30 secondes à 1 minute. Cela ne semble pas très long en soit, mais il faut se souvenir que le problème d'optimisation est résolu de manière itérative, et peut exiger de nombreux calculs de flux.

Le problème d'optimisation de la gestion du combustible peut être résolu sur des intervalles de temps T relativement grand. Deux approches sont alors envisageables. Tout d'abord pour un phénomène transitoire, l'approche quasistatique décrite précédemment peut être suivie. Deuxièmement, si l'évolution du cœur suit un phénomène cyclique, comme par exemple à l'équilibre du rechargement ou les fréquences de rechargement sont constantes, une moyenne des propriétés peut être faite. Cela correspond à l'approche moyennée dans le temps utilisée en général à l'équilibre du rechargement que nous avons choisi ici.

1.2 Cadre de la recherche

Le but général de cette recherche est d'optimiser la gestion du combustible à l'équilibre du rechargement d'un nouveau réacteur de type CANDU: le ACR-700, développé actuellement par AECL. L'acronyme ACR signifie *Advanced CANDU Reactor*. Afin de réduire le coût de construction et d'augmenter la sûreté inhérente

des réacteurs CANDU, plusieurs modifications ont été apportées pour le design du ACR-700 par rapport au réacteur CANDU-6. Premièrement, le caloporteur est de l'eau légère, le modérateur reste de l'eau lourde, mais le pas de réseau est diminué. L'impact sur la physique du réacteur, et donc sur la gestion du combustible, est très important. En effet, l'eau légère est beaucoup plus absorbante que l'eau lourde. De plus, avec la diminution de la proportion du volume d'eau lourde, le réacteur passe d'un comportement sur-modéré à sous-modéré. Pour qu'une réaction en chaîne stable puisse avoir lieu, du combustible légèrement enrichi ($\sim 2\%^{235}U$) est utilisé. Enfin, un poison consommable (Dy) est placé dans une partie du combustible pour assurer une réactivité du vide négative dans le réacteur en tout temps (i.e. lors d'une perte de caloporteur).

Étant donné ses similarités avec les autres réacteurs de type CANDU (rechargement en marche, quelques grappes par jour), plusieurs des techniques de gestion du combustible utilisées jusqu'à présent pour ce type de réacteur le sont aussi pour le développement du ACR-700. L'optimisation de la gestion du combustible dans les réacteurs CANDU utilise des méthodes heuristiques et déterministes. [5,6] Mais, certaines particularités du design du ACR-700 pourraient remettre en question quelques-uns des modèles utilisés jusqu'à présent. D'autres approches pour la gestion du combustible devront donc être développées pour le ACR-700 pour vérifier si les méthodes employées jusqu'à présent sont suffisantes et sinon pour avoir des outils plus adéquats. L'implantation des méthodes est faite dans le code de diffusion DONJON [7]. Une des contraintes de notre recherche est de garder l'aspect modulaire de ce code pour la nouvelle partie que forme l'optimisation de la gestion du combustible.

1.3 Organisation du rapport

Dans une première partie, nous décrirons le cadre dans lequel la gestion du combustible à l'équilibre du rechargement s'applique. Le problème d'optimisation sera alors défini, et les méthodes de résolution existantes basées sur les gradients seront détaillées. L'implantation de ces méthodes sera expliquée dans le chapitre suivant. Une vérification sera également faite pour un cas connu, le réacteur CANDU-6, dans ce chapitre. De nouvelles méthodes de gradients plus générales seront développées et testées pour le réacteur CANDU-6 dans le chapitre suivant. Ensuite, la partie suivante de cette recherche se concentrera sur l'applicabilité des meilleures méthodes de gradients développées à l'optimisation de la gestion du combustible du ACR-700. Deux spécificités du ACR-700 seront détaillées: la réactivité du vide, et l'enrichissement du combustible neuf comme variable de décision. La dernière partie sera consacrée à un type de méthodes non basées sur les gradients: les méthodes métaheuristiques. Leurs définitions et leurs applications à la gestion du combustible des réacteurs à eau légère seront détaillées. Les changements nécessaires à la méthode de recherche tabou en particulier pour la gestion du combustible du ACR-700 seront également examinés. Cette méthode sera alors appliquée au cas de la gestion du combustible du ACR-700. Enfin, nous conclurons sur l'efficacité relative des différentes méthodes envisagées, et nous tracerons un portrait des travaux nécessaires pour améliorer et étendre les capacités d'optimisation du code DONJON.

CHAPITRE 2

APPLICATION DE LA THÉORIE DES PERTURBATIONS GÉNÉRALISÉES À LA GESTION DU COMBUSTIBLE DANS LES RÉACTEURS CANDU

2.1 Introduction

La vie d'un réacteur CANDU peut être séparée en trois étapes: premièrement, *le cœur initial* où tout le combustible est neuf, la réactivité excédentaire du combustible neuf fait en sorte qu'il n'est pas nécessaire d'effectuer de rechargement. Durant une deuxième période, dite *l'approche à l'équilibre*, le combustible retiré du réacteur possède un taux de combustion (burnup) de plus en plus élevé, à mesure que le temps de résidence du combustible augmente. Enfin, après quelques mois (typiquement 6 [5]), le burnup de sortie devient presque constant dans chaque zone et *l'équilibre du rechargement* est atteint. Ainsi, durant la plus grande partie de sa vie utile, un réacteur CANDU est essentiellement à l'équilibre du rechargement en autant que la même stratégie de rechargement soit maintenue: à chaque canal correspond un type de combustible neuf et une fréquence de rechargement. Au contraire, pour un réacteur neuf, au moment de la mise en service, cet équilibre n'est pas encore atteint. Pour compenser pour l'excédant de réactivité et pour simuler l'effet d'aplatissement qu'aura le rechargement à l'équilibre, on utilise alors des poisons dans le modérateur et l'on positionne du combustible d'uranium appauvri pour le cœur neuf pour les CANDU-6 [5].

La plupart des études sur la gestion du combustible porte sur l'équilibre du rechargement. Très peu d'études ont été faites de manière mathématique sur la charge ini-

tiale du cœur neuf et l'approche à l'équilibre [8]. La gestion du combustible pour ces deux étapes est généralement faite de manière heuristique [5]. Le but dans ces cas est d'obtenir l'équilibre le plus rapidement / économiquement possible. La gestion du combustible par les concepteurs et les opérateurs a souvent été basée sur l'expérience acquise soit lors du design soit lors de l'utilisation d'autres réacteurs du même type. L'étude envisagée portant sur la gestion du combustible du ACR-700, un nouveau design, aucune expérience pratique d'exploitation n'existe. Nous avons choisi de cibler notre étude sur la gestion du combustible à l'équilibre. En effet celle-ci représente la majeure partie de la vie utile d'un réacteur d'une part, et influence beaucoup les choix pour un réacteur neuf et en approche à l'équilibre d'autre part.

Nous décrirons dans ce chapitre les techniques utilisées jusqu'à présent pour les calculs de gestion du combustible dans les CANDU à l'équilibre du rechargement. Des améliorations seront également proposées.

2.2 Définition des calculs moyennés dans le temps

Lorsque le cœur d'un réacteur atteint l'équilibre, la fréquence de rechargement des canaux devient constante, l'intervalle de temps entre les rechargements successifs d'un même canal représente un cycle de rechargement. Pour décrire le comportement d'un réacteur CANDU à l'équilibre, nous pourrions procéder à un suivi du cœur pendant une très longue période (> 1 an) en simulant chaque rechargement et en calculant les changements de composition du combustible. La distribution de puissance moyennée dans le temps s'obtient alors en faisant simplement la moyenne des distributions de puissance obtenues tout le long de la simulation. Pour suivre l'évolution du comportement du réacteur avec cette approche, l'équation de diffusion statique (1.1) devrait être réévaluée plusieurs fois puisque les propriétés du

combustible changent beaucoup sur une si longue période. Pour éviter de nombreux calculs de réacteurs, l'approche moyennée dans le temps (*time average* TA) est utilisée. Celle-ci consiste à remplacer dans l'équation de diffusion les propriétés des matériaux par des propriétés moyennées dans le temps qui sont données par la formule suivante [9]:

$$\bar{\Sigma}_{jk} = \frac{1}{B_{jk}^{eoc} - B_{jk}^{boc}} \int_{B_{jk}^{eoc}}^{B_{jk}^{boc}} \Sigma(B).dB \quad (2.1)$$

où B_{jk}^{boc} et B_{jk}^{eoc} représentent les burnups respectivement au début et à la fin du cycle de la grappe k du canal j.

Ainsi en prenant des propriétés moyennées, le calcul du flux neutronique pour une seule configuration du réacteur est nécessaire pour connaître la distribution de puissance en moyenne dans le temps dans le réacteur. Même si l'approche TA requiert quelques itérations (typiquement <10) pour converger sur la forme axiale du flux, elle est beaucoup plus rapide qu'un suivi de cœur avec une itération par rechargement (typiquement un à deux rechargement par jour, soit environ 500 itérations pour un suivi d'un an).

2.3 Calculs de références

Le design particulier des réacteurs CANDU avec des tubes de forces et avec un modérateur et un caloporteur en eau lourde, fait qu'une de leurs caractéristiques principales est le rechargement en marche. La figure (2.1) présente un schéma d'un réacteur CANDU-6. Une petite partie du combustible (typiquement 16 grappes sur les 4560 que contient un réacteur CANDU-6) est remplacée chaque jour. Les nouvelles grappes poussent axialement les anciennes hors du réacteur tel qu'illustré par la figure (2.2). Cette opération se fait sur un seul canal à la fois et dans un

mode bi-directionnel, c'est-à-dire qu'un canal sur deux est rechargé de l'avant vers l'arrière et l'autre dans le sens inverse. Un calcul de diffusion 3D doit donc être fait pour tenir compte de la non symétrie du problème. Lors du design du réacteur CANDU-6, le concepteur a déterminé un état de référence ([5] partie B). Cette configuration correspond à un réacteur critique, où l'intégrité du combustible est assurée et la distribution de la puissance reste dans les limites requises, et pour lequel l'efficacité énergétique des grappes est maximisée. Cette configuration de référence a été obtenue à la suite de plusieurs calculs de diffusion TA, celle-ci étant optimisée par les concepteurs à chaque itération de manière heuristique.

2.4 Objectif des opérateurs

La gestion du combustible des CANDU-6 par les exploitants est basée sur leur expérience pratique et les directives principales du concepteur ([5], partie A). Ainsi, le choix du canal à recharger est fait non seulement selon certains critères de sécurité, mais aussi pour maintenir le réacteur le plus proche possible de l'état de référence tel qu'optimisé par le concepteur. Dans l'industrie, des logiciels comme HQ-SIMEX [10] permettent de choisir automatiquement les canaux à recharger préférentiellement à chaque jour. La décision finale lors de l'opération des centrales est cependant prise par les ingénieurs nucléaires.

Un autre code pour le choix automatique du canal à recharger a récemment été développée par Jeong [11]: OPTIMA. La puissance mesurée par les détecteurs sert à calculer la distribution de puissance estimée pour le réacteur. Le principe du code OPTIMA vise à choisir les canaux à recharger afin de minimiser la différence des puissances zonales par rapport à la distribution de puissance de référence. Une simulation sur 700 jours, en utilisant les données d'une centrale en opération pour la puissance démontrent une légère amélioration des conditions de sûreté avec une

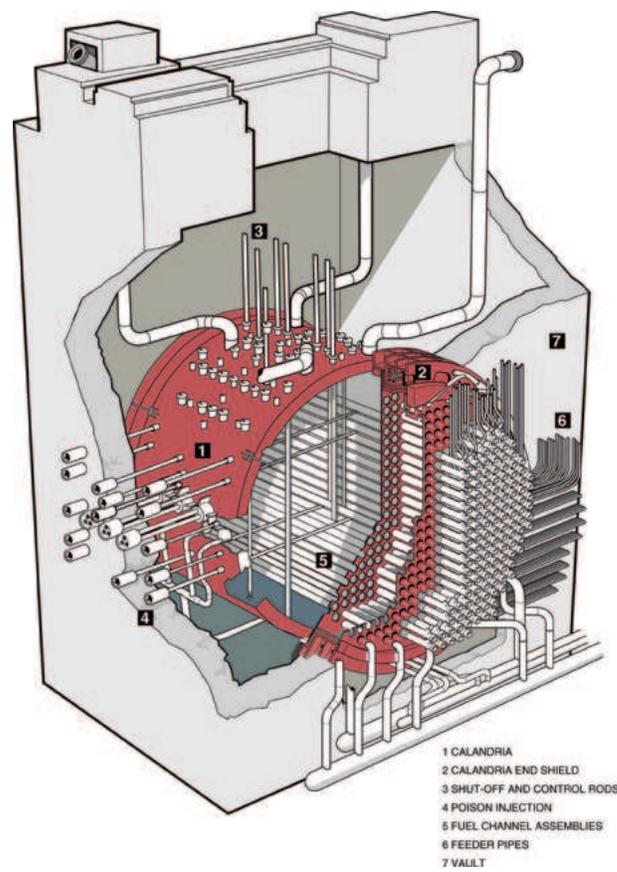


Figure 2.1 Schéma d'un réacteur CANDU-6

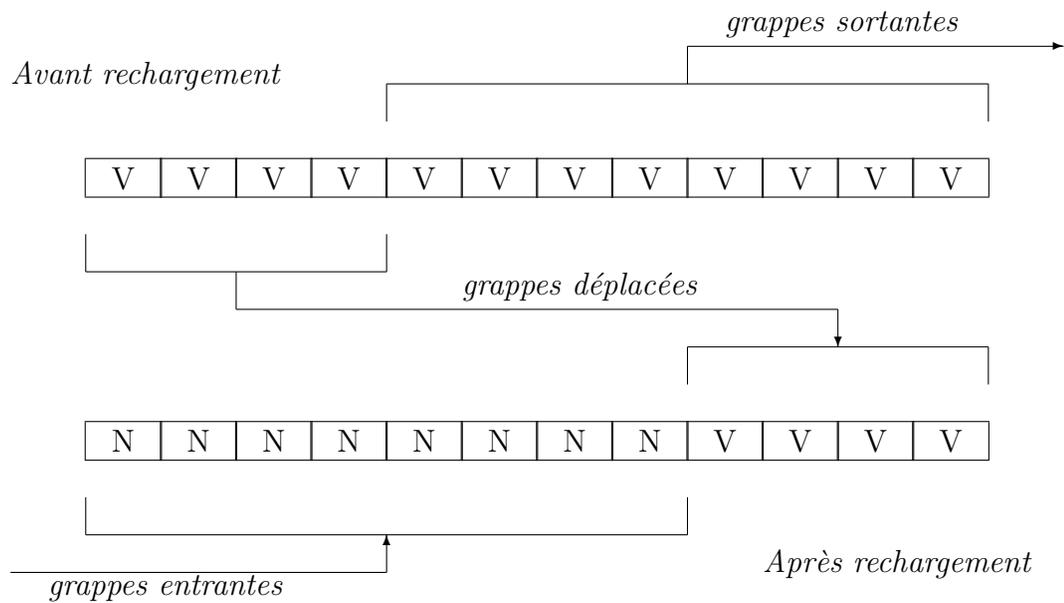


Figure 2.2 Exemple de rechargement d'un canal, 8 grappes à la fois (*8 bundle shift*)

rentabilité du combustible comparable entre le rechargement fait et celui simulé. La méthode proposée par Jeong est également basée sur une optimisation au jour le jour par rapport à une référence. Elle ne garantit donc la meilleure gestion du combustible à long terme que si la référence est elle-même optimisée.

2.5 Optimisation de la référence

Une approche mathématique a été développée par Rozon et Beaudet [6, 12] pour la gestion du combustible à l'équilibre du rechargement. Le but est d'optimiser les paramètres de rechargement pour un cœur moyenné dans le temps. Ces travaux serviront de base à ce projet. Ils sont décrits ici en spécifiant leurs particularités propres aux réacteurs CANDU actuels et les limites dans leur application au réacteur ACR-700.

L'approche est basée sur la programmation mathématique. Le principe général de cette méthode pour l'optimisation de la gestion du combustible est d'améliorer par itérations un ensemble de paramètres dont le burnup du combustible (1.5) qui définissent l'état du réacteur à l'équilibre. La résolution de l'équation de diffusion est donc faite avec la méthode TA. Le but de cette optimisation est de réduire les coûts de combustible du réacteur tout en respectant des conditions sécuritaires d'utilisation. Ce problème est représenté par une fonction objectif et des contraintes qui sont des fonctionnelles du flux neutronique. Cependant, les sections efficaces dépendent du burnup et donc du flux neutronique (1.5 et 1.3). Ainsi l'équation de diffusion est non-linéaire, et donc le problème d'optimisation de la gestion du combustible aussi. Afin d'utiliser les différentes méthodes de programmation quasi-linéaire, les gradients de ces fonctionnelles sont calculés en utilisant la théorie des perturbations classique (TPC) et la théorie des perturbations généralisées du premier ordre (TPG) à chaque itération. Ces techniques sont utilisées depuis plus de 40 ans pour diverses applications en physique des réacteurs. Mélice [13] a été le premier à utiliser la TPG pour calculer les gradients nécessaires aux calculs d'optimisation. L'implantation de ces méthodes a été faite dans le code de calcul OPTTEX développé par Rozon et al. [14]. Nous décrirons par la suite cet algorithme en commençant par la définition du problème d'optimisation. Ensuite, les méthodes de TPC et TPG seront expliquées, avant de décrire les méthodes de programmation quasi-linéaire utilisées pour la résolution de la partie optimisation du problème.

2.6 Paramètres d'optimisation

L'approche utilisée par Rozon [15], Beudet [16, 17] et Tajmouati [9] consiste à représenter les options qui définissent le rechargement à l'équilibre par quelques

paramètres. La gestion du combustible consiste donc au choix approprié de ces paramètres qui permet d'optimiser un objectif tout en respectant les contraintes. Les moyens de contrôler le réacteur à long terme sont entre autres le choix du type de combustible, la fréquence de rechargement des canaux, le nombre de grappes à chaque rechargement.

Plus un canal est rechargé souvent, moins les grappes ont le temps de fournir de l'énergie. La notion de *burnup moyen de sortie* B_j^e est alors définie pour représenter l'utilisation du combustible. Elle représente la moyenne de la densité d'énergie produite dans chaque grappe (MWj/t) extraite pour un canal j à chaque rechargement. Le burnup moyen de sortie est alors donné par:

$$B_j^e = \frac{1}{n_j} \sum_k \Delta B_{jk} = \frac{1}{n_j} \sum_k T_j P_{jk} = \frac{T_j}{n_j} P_j \quad (2.2)$$

où n_j représente le nombre de grappes rechargées pour le canal j , T_j la durée entre rechargements à l'équilibre, P_j la puissance de fission du canal j , P_{jk} la puissance moyenne des grappes pendant le temps T_j et ΔB_{jk} l'incrément de burnup des grappes pendant le temps T_j (tel que défini à l'équation (1.5)).

La fréquence des rechargements est alors fixée par la distribution du burnup moyen de sortie B_j^e de chaque canal j , et est donnée par:

$$\frac{n_j}{T_j} = \frac{P_j}{B_j^e} \quad (2.3)$$

Ainsi, pour une puissance moyenne canal donnée (P_j), spécifier le burnup moyen de sortie équivaut à spécifier la fréquence des rechargements. Le choix approprié des burnups moyens de sortie B_j^e de chaque canal j permet de maintenir une répartition de la puissance thermique la plus uniforme possible, et donc de diminuer la puissance maximum de canal dans le réacteur. Cependant, pour simplifier le problème,

les canaux sont regroupés en quelques zones de burnup à l'intérieur desquelles B_j^e est constant pour tous les canaux. Le nombre de zones sera notée par la suite par n_{zb} , et les burnups de chaque zone par B_{jzb}^e . Dans un réacteur CANDU-6 contenant 380 canaux, les études présentées par Rouben [5] et Rozon [15] n'ont utilisées que 2 ou 3 zones. Cela suffit à obtenir une puissance canal maximum petite.

Le combustible neuf doit être représenté par deux paramètres. Premièrement, nous définissons l'enrichissement initial (en ^{235}U) du combustible, qui est fourni à chaque canal j . D'une manière similaire au burnup, nous définirons des zones d'enrichissement identique qui seront au nombre de n_{ze} . L'enrichissement de chaque zone sera noté ε_{jze} . D'un point de vue pratique, nous préférons avoir un seul type de combustible par zone de combustion, ce qui entraîne que $n_{ze} \leq n_{zb}$. Deuxièmement, la concentration initiale en poison consommable contenu dans le combustible neuf peut varier. Là encore nous pouvons définir un nombre de zone de même concentration de poison n_{zp} , avec une concentration définie par x_{ijzp} . Ce paramètre adéquat pour le cas du ACR-700 n'a pas été utilisé pour les études réalisées avec OPTEX puisque l'utilisation de poison consommable n'a pas été envisagée pour les CANDU-6. Cependant, Rozon [15] souligne l'importance d'un tel paramètre lorsque de l'uranium légèrement enrichi (LEU) est utilisé dans un réacteur. D'un point de vue mathématique, la concentration de poison peut être traitée de la même manière que l'enrichissement pour son optimisation.

Le nombre de grappes rechargées pour le canal j est noté n_j . Nous regroupons les n_j en n_{zr} zones de rechargement, chacune étant notée par n_{jzr} . Nous prendrons comme convention que dans une zone de combustion un seul nombre de grappes par rechargement est permis, et donc $n_{zr} \leq n_{zb}$. Ce paramètre n'a cependant pas été utilisé par Beaudet et Tajmouati [9, 17]. Seul Rozon [15] en a étudié les influences théoriques sur une partie du comportement du réacteur (les facteurs de formes de la puissance). Il a cependant fixé ce paramètre lors des calculs d'optimisation. Son

caractère entier ne facilite pas son utilisation.

Dans les réacteurs CANDU, des barres de compensation (barres absorbantes en acier ou cobalt) sont utilisées pour constituer une réserve de réactivité qui permettra de compenser (partiellement) les variations de ^{135}Xe dans le combustible suite aux manœuvres de puissance ([4], annexe A). La présence de ces barres exercera une influence sur la distribution du flux neutronique. Leur influence sur le flux et donc indirectement sur la gestion du combustible dépend de leur épaisseur (α). Cette dernière a donc été également optimisée lors des différents travaux cités précédemment. Ce type de barre n'est cependant pas prévu dans le design du ACR-700. Elles ne seront donc pas prises en compte dans cette recherche.

En regroupant tous les paramètres susceptibles d'être utiles à l'optimisation de la gestion du combustible du ACR-700, nous obtenons le vecteur d'état \vec{X}_e :

$$\vec{X}_e = \left(B_1^e, \dots, B_{n_{zb}}^e, \varepsilon_1, \dots, \varepsilon_{n_{ze}}, x_{i_1}, \dots, x_{i_{n_{zp}}}, n_1, \dots, n_{n_{zr}} \right) \quad (2.4)$$

qui représente l'ensemble des choix possibles pour la gestion du combustible à l'équilibre du rechargement. Certains choix peuvent être fixés d'autres optimisés. L'ensemble des choix variables est regroupé dans le vecteur de décision. La longueur totale du vecteur de décision sera notée n_{var} par la suite. Pour les études subséquentes du CANDU-6 que nous présenterons seuls les burnups moyen de sortie seront optimisés. Le vecteur de décision se résumera donc à \vec{X} , donné par:

$$\vec{X} = \left(B_1^e, \dots, B_{n_{zb}}^e \right) \quad (2.5)$$

2.7 But de l'optimisation

Le but de l'optimisation est de minimiser la "fonction objectif". Dans la mesure où les critères de sécurité sont respectés, le critère économique de rentabilité peut servir de comparaison entre différents choix de gestion du combustible. Dans ce cas, l'objet de l'optimisation est de réduire le prix de revient de la production d'énergie. La fonction objectif se traduit donc par une fonction de coût qui doit également être définie.

Le coût en combustible est la composante la plus évidente aux frais de fonctionnement. Son évaluation n'est cependant pas facile et peut se faire en deux étapes. Premièrement, le coût du combustible (i.e. d'une grappe neuve) peut soit être fixé si l'enrichissement n'est pas une variable de décision, soit être calculé dans le cas contraire. Le prix de revient d'une grappe de combustible (C_F) est donné par la formule suivante [15]:

$$C_F(\varepsilon) = \sum C_{UN}(\varepsilon) \times (1 + int)^{t_1} + \sum C_{SWU}(\varepsilon) \times (1 + int)^{t_2} + C_{FAB} \quad (2.6)$$

où C_{UN} représente le coût en uranium naturel, C_{SWU} le coût de l'enrichissement de l'uranium naturel, C_{FAB} le coût de fabrication d'une grappe (usinage, montage, ...), ε l'enrichissement dans la grappe, int le taux d'actualisation, t_1 et t_2 les temps d'obtention et d'enrichissement de l'uranium naturel.

NB: Le signe de somme \sum dans cette formule représente le fait que la composition du combustible n'est pas forcément uniforme dans tous les crayons d'une grappe.

Les coefficients C_{UN} et C_{SWU} sont donnés par les formules suivantes :

$$C_{SWU}(\varepsilon) = m_G \cdot c_S \cdot \frac{S}{P}(\varepsilon) \quad (2.7)$$

$$C_{UN}(\varepsilon) = m_G \cdot c_U \cdot \frac{F}{P}(\varepsilon) \quad (2.8)$$

avec

$$\begin{aligned}
\frac{S}{P}(\varepsilon) &= V(\varepsilon) + \frac{W}{P}(\varepsilon) \cdot V(\varepsilon_W) - \frac{F}{P}(\varepsilon) \cdot V(\varepsilon_{UN}) \\
\frac{F}{P}(\varepsilon) &= \frac{\varepsilon - \varepsilon_W}{\varepsilon_{UN} - \varepsilon_W} \\
\frac{W}{P}(\varepsilon) &= \frac{\varepsilon - \varepsilon_{UN}}{\varepsilon_{UN} - \varepsilon_W} \\
V(\varepsilon) &= (2\varepsilon - 1) \ln \left(\frac{\varepsilon}{1 - \varepsilon} \right)
\end{aligned}$$

où m_G représente la masse d'uranium dans la grappe, c_U le prix de l'uranium naturel au kg, c_S le prix de l'unité de séparation au kg, ε_{UN} la teneur en ^{235}U de l'uranium naturel, ε_W la teneur en ^{235}U de l'uranium appauvri, $\frac{F}{P}$ le nombre de kg d'uranium naturel requis par kg d'uranium enrichi, $\frac{S}{P}$ le nombre d'unités de séparation requis par kg d'uranium enrichi. L'origine détaillée de ces formules est donnée par Rozon [18].

La deuxième étape consiste à calculer le nombre de grappes de combustible nécessaire au fonctionnement du réacteur [9]. Pour le rechargement du canal j , le nombre de grappes utilisées par unité de temps est donné par la fréquence de rechargement du canal (équation 2.3). La puissance canal est simplement la somme des puissances de grappes définie par l'équation (1.4).

En prenant en compte le fait que la composition (ε) des grappes neuves peut varier selon le canal j , le coût total en combustible est donné par la somme sur tous les canaux du produit des équations (2.3) et (2.6), ce qui nous donne en utilisant (1.4):

$$\frac{\text{coût}}{\text{unité de temps}} = \sum_j \frac{n_j}{T_j} \cdot C_F(\varepsilon_j) = \sum_j \frac{C_F(\varepsilon_j)}{B_j^e} \cdot P_j = \left\langle \frac{C_F(\varepsilon)}{B_j^e} \cdot H, \phi \right\rangle_{\text{réacteur}} \quad (2.9)$$

En normalisant par la puissance totale du réacteur, nous obtenons pour la fonction

coût F_C associé au combustible :

$$F_C = \frac{\langle \frac{C_F(\varepsilon)}{B_j^e} \cdot H, \phi \rangle_{\text{réacteur}}}{\langle H, \phi \rangle_{\text{réacteur}}} \quad (2.10)$$

La fonction coût F_C représente donc le coût en combustible de l'énergie produite par unité de temps et de puissance. Elle est généralement exprimée en $\$/MW - an$. Le burnup B étant habituellement donné en MWj/t et le prix du combustible en $\$/grappe$, la relation précédente sera multipliée par un coefficient pour obtenir l'unité désirée.

D'autres paramètres peuvent être pris en compte dans les frais de fonctionnement. Par exemple dans un réacteur CANDU "traditionnel", des barres de compensation sont positionnées dans le réacteur pour améliorer la distribution du flux neutronique. Ces barres peuvent être en cobalt. Or cet élément absorbe les neutrons et forme du ^{60}Co qui est radioactif. Ce dernier est utilisable comme source de rayons-X en médecine et peut donc être vendu à cet effet. Dans son étude, Beaudet [17] a tenu compte de ce gain pour la fonction coût. Une deuxième composante est alors retranchée à la fonction de coût puisque le cobalt est vendu et vu donc comme un crédit.

2.8 Contraintes

Si le choix des paramètres de la gestion du combustible semble relativement large étant donné leur nombre, les contraintes physiques pour maintenir le réacteur dans un état de fonctionnement sécuritaire réduisent beaucoup l'éventail possible des variations relatives des paramètres contrôle. Nous présenterons seulement les contraintes tels que définies par Rozon [15]. Les spécificités pour le ACR-700 seront présentés ultérieurement.

2.8.1 Criticité du réacteur

La première et la plus importante contrainte est de maintenir la puissance du réacteur constante. Il doit donc être dans un état stationnaire, d'où :

$$k_{eff} = \frac{1}{\lambda} = \frac{\langle \phi^*, \mathbf{F}\phi \rangle}{\langle \phi^*, \mathbf{A}\phi \rangle} = k_{ref} = \frac{1}{\lambda_{ref}} \quad (2.11)$$

avec

$$(\mathbf{A} - \lambda \mathbf{F})^T \phi^* = 0$$

où $k_{ref} = 1.000$ si le réacteur est critique et si toutes les barres de contrôle sont présentes, sinon $k_{ref} > 1.000$ pour tenir compte de l'anti-réactivité des mécanismes de contrôle non présents dans la simulation ou des erreurs de simulations, ϕ^* représente le flux adjoint.

2.8.2 Puissance canal maximum

La deuxième contrainte est de maintenir l'intégrité du réacteur. Pour cela, nous devons assurer un refroidissement adéquat de tous les canaux. Le débit dans un canal ne pouvant être infini, la puissance générée ne doit donc pas dépasser un maximum P_{CHF} pour que le flux de chaleur soit entièrement évacué, et n'atteigne pas le flux de chaleur critique. Pour tenir compte des erreurs (U) faites lors de l'évaluation de la puissance critique P_{CHF} , et du fait que nous devons avoir une certaine marge de manœuvre (M) pour le bon fonctionnement du réacteur, la puissance limite P_{lim} pour un canal est donnée par :

$$P_{lim} = \frac{P_{CHF}}{(1 + M).(1 + U)} \quad (2.12)$$

avec 7% pour M et 10% pour U comme valeurs types, d'après Rozon [15].

Comme nous l'avons vu au début de ce chapitre, l'optimisation de la gestion du combustible à l'équilibre du rechargement utilise la méthode en moyenne dans le temps (TA). Or le réacteur doit être dans son état sécuritaire en tout temps. Plusieurs calculs instantanés sont alors faits, et le plus grand rapport entre la puissance canal instantanée $P_{j_{ins}}$ et la puissance canal moyennée dans le temps P_j définit le facteur de pic de puissance canal $CPPF_j$ (*channel power peaking factor*) pour chaque canal j [18]. Pour respecter l'intégrité de tout le réacteur en tout temps, la contrainte sur la puissance doit être validée ce qui se traduit par:

$$\max_j \{P_{j_{ins}}\} \leq \max_j \{CPPF_j \cdot P_j\} \leq P_{lim} \quad (2.13)$$

En pondérant l'équation précédente par la puissance totale P_{tot} , en remplaçant les termes de puissance par des densités de puissance volumique, et en utilisant la définition intégrale de la puissance (équation 1.4), nous obtenons:

$$q = \max_j \left\{ CPPF_j \cdot \frac{V \langle H, \phi \rangle_{V_j}}{V_j \langle H, \phi \rangle_V} \right\} \leq \frac{\bar{P}_{lim}}{\bar{P}} = f_{lim} \quad (2.14)$$

où V_j et V représentent respectivement le volume du canal j et du cœur.

D'un point de vue pratique pour calculer le gradient de cette fonction, une séparation de la contrainte du maximum sur tous les canaux en plusieurs contraintes portant chacune sur un seul canal est effectuée. En effet, le canal dans lequel le maximum est atteint peut changer avec les variables de décision, \vec{X} , ce qui rend difficile le calcul du gradient. f_{lim} reste néanmoins le même pour chaque canal j . Dans la pratique, seul un nombre restreint de canaux de référence sont surveillés (n_{cs} , nombre de canaux de surveillance). En effet, les canaux ayant une puissance moyenne petite ne peuvent atteindre la puissance critique. L'équation précédente est alors

écrite comme un ensemble d'équations:

$$q_j = CPPF_j \cdot \frac{V \langle H, \phi \rangle_{V_j}}{V_j \langle H, \phi \rangle_V} \leq \frac{\bar{P}_{lim}}{\bar{P}} = f_{lim} \quad j \in [1, n_{cs}] \quad (2.15)$$

Une approche similaire peut être faite pour la puissance de grappe maximum ou pour une zone en général. Le facteur de pic de puissance pour la grappe *BPPF* (*bundle power peaking factor*) ou la zone de surveillance *ZPPF* (*zone power peaking factor*) est alors défini.

2.8.3 Réactivité des barres de compensation

Une efficacité minimum doit être atteinte par l'ensemble des barres de compensation pour pouvoir faire face à l'effet Xénon lors de manœuvre de puissance. La réactivité des barres de compensation présente donc un minimum.

Pour calculer la réactivité des barres de compensation, nous devons connaître la réactivité du réacteur avec et sans ces mécanismes de contrôles. Les variations du flux neutronique associées aux différentes configurations des barres de compensation sont tellement fortes que la théorie des perturbations n'est pas valide pour calculer directement la perturbation sur la valeur propre du système associée à la suppression des barres de compensation. Deux calculs de réacteur doivent être faits pour évaluer cette contrainte:

$$(\mathbf{A} - \lambda \mathbf{F})\phi = 0 \quad \text{avec les barres de compensation}$$

$$(\mathbf{A}_0 - \lambda_0 \mathbf{F}_0)\phi_0 = 0 \quad \text{sans les barres de compensation}$$

La réactivité des barres de compensation est alors donnée par l'équation suivante:

$$\Delta k_{eff}(\vec{X}) = \frac{1}{\lambda_0} - \frac{1}{\lambda} \geq \Delta K_{\text{barres}} \quad (2.16)$$

Dans ces travaux, Tajmouati [9] a rajouté une contrainte sur la réactivité de chaque banc de barres de compensation. Celle-ci doit être inférieure à 50% du déplacement des barres liquides pour éviter des oscillations entre ces deux systèmes de contrôle. Pour les évaluer, un calcul du flux neutronique supplémentaire est nécessaire par contrainte.

2.9 Méthodes d'optimisation utilisées

2.9.1 Introduction

Les différentes fonctions objectifs et contraintes qui ont été utilisées dans les travaux de Rozon [6, 14, 15, 19], Beaudet [16, 17] et Tajmouati [9] dépendent généralement du flux neutronique et des sections efficaces comme nous l'avons vu aux sections précédentes. Le problème d'optimisation est donc non linéaire. La formulation générale d'un problème d'optimisation de la gestion du combustible dans les différents réacteurs CANDU peut se résumer sous la forme suivante:

$$\min_{\vec{X}} F_C(\vec{X}) \text{ avec } \begin{cases} k_{eff}(\vec{X}) & = & k_{ref} \\ q_j(\vec{X}) & \leq & f_j & j \in (1, n_{zc}) \\ \frac{1}{\lambda_0} - \frac{1}{\lambda} & \geq & \Delta K_{\text{barres}} & \text{(si applicable)} \\ X_n^{INF} \leq X_n & \leq & X_n^{SUP} \end{cases} \quad (2.17)$$

où X_n^{INF} et X_n^{SUP} représentent respectivement les bornes inférieures et supérieures des variables de décision.

Nous n'avons pas retenu la contrainte sur la réactivité des barres de compensation lors de l'implantation. Ce type de contrôleur n'existe pas dans un ACR-700, d'une part, et le design pour les CANDU-6 est dorénavant final, d'autre part. Aucune optimisation n'est donc à envisager pour ce type de paramètres.

Le nombre total de contraintes sera noté n_{cst} dorénavant. Afin de simplifier les équations qui suivent, nous utiliserons une notation plus générale pour le problème d'optimisation pour la fonction objectif et les différentes contraintes, c'est-à-dire:

$$\min_{\vec{X}} f(\vec{X}) \text{ avec } \begin{cases} h_i(\vec{X}) = b_i & i \in (1, NCE) \\ g_j(\vec{X}) \leq c_j & j \in (1, NCI) \\ X_n^{INF} \leq X_n \leq X_n^{SUP} \end{cases} \quad (2.18)$$

avec f qui représente la fonction coût F_C , h_i qui se résume à une seule contrainte d'égalité soit la criticité du réacteur k_{eff} , g_j pour les contraintes d'inégalité sur la puissance zonale q_j .

L'ensemble des vecteurs d'état (i.e. les configurations du réacteur) qui correspondent à un rechargement en combustible respectant les critères de sécurité est appelé le domaine réalisable. Jusqu'à présent pour résoudre le problème d'optimisation de la gestion du combustible définie par le système (2.18), la plupart des auteurs [6, 9, 14–17, 19] ont utilisé des méthodes de programmation linéaire et quasi-linéaire. Or le problème étant non-linéaire, une procédure itérative de linéarisation de la fonction objectif et des contraintes est appliquée. À partir d'un vecteur d'état du domaine réalisable, un nouveau vecteur d'état est calculé en résolvant le problème linéarisé. Pour cela, un développement de Taylor au premier ordre des fonctionnelles est nécessaire. Ainsi pour n'importe quelle fonctionnelle u , nous avons:

$$u(\vec{X}^{k+1}) = u(\vec{X}^k) + \nabla u^k \cdot \Delta \vec{X}^k \quad (2.19)$$

où

$$\nabla u^k = \left\{ \frac{du}{dX_n} \right\}^k \quad (2.20)$$

Ainsi, à chaque itération k pour un vecteur d'état \vec{X}^k , le problème se résume à

calculer le pas optimal $\Delta\vec{X}^k$ tel que la fonction objectif soit optimisée. Le nouveau vecteur d'état sera alors simplement donné par la formule suivante:

$$\vec{X}^{k+1} = \vec{X}^k + \Delta\vec{X}^k \quad (2.21)$$

Le problème d'optimisation linéarisé se présente donc sous la forme suivante:

$$\min_{\Delta\vec{X}} \nabla f^k \cdot \Delta\vec{X} \text{ avec } \begin{cases} \nabla h_i^k \cdot \Delta\vec{X} = b_i - h_i(\vec{X}^k) \\ \nabla g_j^k \cdot \Delta\vec{X} \leq c_j - g_j(\vec{X}^k) \\ X_n^{INF} - X_n^k \leq \Delta X_n \leq X_n^{SUP} - X_n^k \end{cases} \quad (2.22)$$

Cependant, le problème linéarisé n'est qu'une approximation locale du problème originel. Afin de préserver l'exactitude de cette approche, le pas calculé par la méthode de programmation quasi-linéaire quelqu'elle soit, doit être "suffisamment petit". $\Delta\vec{X}^k$ devra donc être borné pour que le développement de Taylor reste valide. Cette limite sur $\Delta\vec{X}^k$ se traduit par une contrainte supplémentaire quadratique donnée par la formule suivante:

$$\sum w_n \cdot \Delta X_n^2 \leq S^2 \quad (2.23)$$

où w_n représente le poids de la variable X_n , $n \in (1, n_{var})$, S la limite quadratique. Celle-ci est initialisée puis ajustée au cours des itérations.

Ainsi, afin de pouvoir résoudre ce problème, les auteurs [6,9,14–17,19] ont dû dans un premier temps calculer les gradients des fonctionnelles. Or, les fonctionnelles qui peuvent être exprimées comme une fonction explicite des variables de décision ($u(\vec{X})$) sont extrêmement rares puisque la plupart dépendent du flux neutronique ($u(\vec{X}, \phi(\vec{X}))$). Ils ont donc eu recours à la théorie des perturbations de premier ordre ainsi qu'à la théorie des perturbations généralisées pour calculer les gradients.

2.9.2 Théorie des perturbations du premier ordre

L'objectif de la théorie des perturbations du premier ordre est de pouvoir estimer les changements directs d'une quantité physique (représentée par une fonctionnelle u) lors d'un changement des propriétés qui définissent le réacteur [20]. Les propriétés physiques perturbées seront notées de la manière suivante:

$$\begin{aligned}\vec{X} &\rightarrow \vec{X}_p = \vec{X} + \Delta\vec{X} \\ \Sigma &\rightarrow \Sigma_p = \Sigma + \Delta\Sigma \\ A &\rightarrow A_p = A + \Delta A \\ B &\rightarrow B_p = B + \Delta B \\ \lambda &\rightarrow \lambda_p = \lambda + \Delta\lambda\end{aligned}$$

La variation directe de la fonctionnelle u associée à une perturbation $\Delta\vec{X}$ sera donc donnée par la formule suivante:

$$\Delta u = u(\vec{X}_p) - u(\vec{X}) \quad (2.24)$$

et pour le gradient :

$$\begin{aligned}\nabla u &= \left\{ \frac{du}{dX_n} \right\} \\ &= \lim_{\Delta X_n \rightarrow 0} \left\{ \frac{u(\vec{X}_p) - u(\vec{X})}{\Delta X_n} \right\} \quad n \in (1, n_{var})\end{aligned} \quad (2.25)$$

avec $\vec{X}_p = \vec{X} + (0, \dots, 0, \Delta X_n, 0, \dots, 0)$ dans le cas du gradient. Généralement les paramètres X_n apparaissent explicitement dans la fonctionnelle. Nous pourrions donc dériver analytiquement cette composante directe.

Pour le cas particulier de k_{eff} , l'application de la formulation précédente à la fonctionnelle définie par $\lambda = 1/k_{eff}$, nous donne la dérivée analytique de la formule de Rayleigh [20]:

$$\frac{\partial \lambda}{\partial X_n} = \lambda \left(\frac{\langle \phi^*, \frac{\partial \mathbf{A}}{\partial X_n} \phi \rangle}{\langle \phi^*, \mathbf{A} \phi \rangle} - \frac{\langle \phi^*, \frac{\partial \mathbf{F}}{\partial X_n} \phi \rangle}{\langle \phi^*, \mathbf{F} \phi \rangle} \right) \quad (2.26)$$

ce qui donne avec la notation précédente et après simplification:

$$\begin{aligned} \frac{\partial \lambda}{\partial X_n} &= \lim_{\Delta X_n \rightarrow 0} \frac{\Delta \lambda}{\Delta X_n} \\ &= \lim_{\Delta X_n \rightarrow 0} \lambda \left(\frac{\langle \phi^*, \frac{\mathbf{A}_p}{\Delta X_n} \phi \rangle}{\langle \phi^*, \mathbf{A} \phi \rangle} - \frac{\langle \phi^*, \frac{\mathbf{F}_p}{\Delta X_n} \phi \rangle}{\langle \phi^*, \mathbf{F} \phi \rangle} \right) \quad n \in (1, n_{var}) \end{aligned} \quad (2.27)$$

démonstration:

La démonstration de l'équation (2.26) est détaillée ci dessous car la méthode utilisée sera reprise pour des équations futures.

$$\begin{aligned} \frac{\partial \lambda}{\partial X_i} &= \frac{\langle \frac{\partial \phi^*}{\partial X_i}, \mathbf{A} \phi \rangle + \langle \phi^*, \frac{\partial \mathbf{A}}{\partial X_i} \phi \rangle + \langle \phi^*, \mathbf{A} \frac{\partial \phi}{\partial X_i} \rangle}{\langle \phi^*, \mathbf{F} \phi \rangle} \\ &\quad - \langle \phi^*, \mathbf{A} \phi \rangle \frac{\langle \frac{\partial \phi^*}{\partial X_i}, \mathbf{F} \phi \rangle + \langle \phi^*, \frac{\partial \mathbf{F}}{\partial X_i} \phi \rangle + \langle \phi^*, \mathbf{F} \frac{\partial \phi}{\partial X_i} \rangle}{\langle \phi^*, \mathbf{F} \phi \rangle^2} \\ &= \frac{\langle \frac{\partial \phi^*}{\partial X_i}, \overbrace{(\mathbf{A} - \lambda \mathbf{F})}^{(a)} \phi \rangle + \langle \phi^*, \left(\frac{\partial \mathbf{A}}{\partial X_i} - \lambda \frac{\partial \mathbf{F}}{\partial X_i} \right) \phi \rangle + \langle \overbrace{\phi^*, (\mathbf{A} - \lambda \mathbf{F})}^{(b)} \frac{\partial \phi}{\partial X_i} \rangle}{\langle \phi^*, \mathbf{F} \phi \rangle} \end{aligned} \quad (2.26a)$$

Les facteurs (a) et (b) sont nuls, d'où l'équation (2.26).

Le gradient de la fonctionnelle k_{eff} est alors donné par:

$$\frac{\partial k_{eff}}{\partial X_i} = \frac{\partial(1/\lambda)}{\partial X_i} = \frac{-1}{\lambda^2} \frac{\partial \lambda}{\partial X_i} = -k_{eff}^2 \frac{\partial \lambda}{\partial X_i} \quad (2.28)$$

2.9.3 Théorie des perturbations généralisées

L'utilisation des travaux de Stacey [21, 22] sur l'application de la théorie des méthodes variationnelles en physique du réacteurs ont permis de calculer les gradients des différentes fonctions objectif et contraintes avec plus d'exactitude. L'objectif de la TPG est aussi de pouvoir estimer les changements d'une quantité physique lors d'un changement des propriétés qui définissent le réacteur. Cependant, en plus des effets du premier ordre, cette méthode permet de prendre en compte les effets indirects associés à la variation du flux neutronique sur la variation de la fonctionnelle.

Par définition le gradient d'une fonctionnelle u est donné par:

$$\nabla u(\vec{X}, \phi) = \nabla u = \left\{ \underbrace{\frac{\partial u}{\partial X_n}}_{Directe} + \underbrace{\left(\frac{\partial u}{\partial \phi} \cdot \frac{\partial \phi}{\partial X_n} \right)}_{Indirecte} \right\} \quad n \in (1, n_{var}) \quad (2.29)$$

où:

$\frac{\partial u}{\partial X_n}$ représente la variation directe de u selon les composantes du vecteur d'état \vec{X} . Cela correspond à la théorie des perturbations du premier ordre. Pour les fonctionnelles données en exemple comme la fonction objectif (2.10) et les contraintes sur la puissance (2.15), les composantes directes du gradient sont données par:

pour la fonction coût, F_C :

$$\frac{\partial F_C}{\partial X_n} = \frac{\langle \frac{\partial C u}{\partial X_n}, \frac{H}{B^e} \phi \rangle_{V_n}}{\langle H, \phi \rangle_V} + \frac{\langle C u \left(\frac{1}{B^e} \frac{\partial H}{\partial X_n} - \frac{H}{B^{e^2}} \frac{\partial B^e}{\partial X_n} \right), \phi \rangle_{V_n}}{\langle H, \phi \rangle_V}$$

$$-F_C \cdot \frac{\langle \frac{\partial H}{\partial X_n}, \phi \rangle_{V_n}}{\langle H, \phi \rangle_V} \quad n \in (1, n_{var}) \quad (2.30)$$

pour la limite zonale de puissance, q_j :

$$\begin{aligned} \frac{\partial q_j}{\partial X_n} = & \frac{q_j}{ZPPF_j} \cdot \frac{\partial ZPPF_j}{\partial X_n} \cdot \delta_{nj} + \frac{q_j}{\langle H, \phi \rangle_{V_j}} \langle \frac{\partial H}{\partial X_n}, \phi \rangle_{V_j} \cdot \delta_{nj} \\ & - \frac{q_j}{\langle H, \phi \rangle_V} \langle \frac{\partial H}{\partial X_n}, \phi \rangle_V \quad n \in (1, n_{var}) \text{ et } j \in (1, n_{zc}) \end{aligned} \quad (2.31)$$

où $ZPPF$ peut représenter le $CPPF$ ou le $BPPF$.

$\frac{\partial u}{\partial \phi} \cdot \frac{\partial \phi}{\partial X_n}$ correspond à la composante indirecte. Elle provient du fait que de changer \vec{X} fait aussi changer le flux ϕ et donc u . Nous ne pouvons pas calculer analytiquement cette composante étant donné que le calcul du flux en fonction de \vec{X} ne peut être que numérique dans la quasi-totalité des cas. Stacey a proposé une approche utilisant la théorie des perturbations généralisées du premier ordre pour évaluer cette composante. Dans l'annexe A de [21], Stacey démontre que cette composante est donnée par la formule suivante:

$$\frac{\partial u}{\partial \phi} \frac{\partial \phi}{\partial X_i} = \langle S^*, \Gamma_n \rangle = \langle S_n, \Gamma^* \rangle \quad n \in (1, n_{var}) \quad (2.32)$$

où S_n représente la source directe, S^* la source adjointe, Γ_n l'adjoint généralisé explicite, Γ^* l'adjoint généralisé implicite. La définition des termes de sources et d'adjoints généralisés est donnée dans la suite.

Ainsi grâce à la théorie des perturbations généralisées, les fonctions (flux, adjoint, adjoints généralisés) qui correspondent à l'état de référence seulement sont nécessaires pour le calcul du gradient. Ceci permet de minimiser l'effort de calcul. Trois étapes sont à suivre pour obtenir le gradient.

Calcul des sources

La première étape consiste à calculer les deux types de termes de source.

Les sources directes S_n sont données par la formule suivante en notation continue:

$$S_n = \frac{\partial}{\partial X_n} (\mathbf{A} - \lambda \mathbf{F}) \phi \quad n \in (1, n_{var}) \quad (2.33)$$

Nous pouvons noter que le terme de source explicite S_n est orthogonal au flux adjoint ϕ^* [20].

Les sources indirectes S^* sont données par la formule suivante en notation continue:

$$S_j^* = \frac{\partial u}{\partial \phi} \quad j \in (1, n_{cst} + 1) \quad (2.34)$$

Le terme de source implicite S_j^* est orthogonal au flux ϕ [20]. Il représente la variation directe de la fonctionnelle u avec le flux ϕ .

Pour les fonctionnelles données en exemple comme la fonction objectif (2.10) et les contraintes sur la puissance (2.15), les termes de source indirecte sont donnés par: pour la fonction coût, F_C :

$$\frac{\partial F_C}{\partial \phi} = S_{F_C}^* = \frac{\frac{C_u}{B^e} \cdot H(\vec{r}) - F_C \cdot H(\vec{r})}{\langle H, \phi \rangle_V} \quad (2.35)$$

pour la limite zonale de puissance, q_j :

$$\frac{\partial q_j}{\partial \phi} = S_{q_j}^* = \frac{ZPPF_j \frac{V}{V_j} \cdot H(\vec{r}_j) - q_j \cdot H(\vec{r})}{\langle H, \phi \rangle_V} \quad j \in (1, n_{zc}) \quad (2.36)$$

$$\text{où } H(\vec{r}_j) = \begin{cases} H_j & \vec{r} \in V_j \\ 0 & \text{sinon} \end{cases}$$

Calcul des adjoints généralisés

Lors de la deuxième étape, un des deux types d'adjoints généralisés est calculé.

L'adjoint généralisé explicite Γ_i est donné par la formule :

$$(\mathbf{A} - \lambda\mathbf{F})\Gamma_n = S_n \quad n \in (1, n_{var}) \quad (2.37)$$

Γ_n représente la variation du flux avec la variable de décision n .

Quant à l'adjoint généralisé implicite Γ^* , il est donné par la formule :

$$(\mathbf{A} - \lambda\mathbf{F})^T \Gamma_j^* = S_j^* \quad j \in (1, n_{cst} + 1) \quad (2.38)$$

La résolution de ces deux équations représente numériquement le même effort de calcul. Le choix se fait donc sur le nombre de fois qu'il faut les résoudre. En général, une comparaison entre le nombre de paramètres n_{var} et le nombre de contraintes n_{cst} permet de décider la méthode à employer. Si l'équation (2.37) est résolue, alors la méthode sera dite explicite. Dans le cas de l'équation (2.38), la méthode implicite sera utilisée. Le choix de la méthode est fait pour minimiser le nombre d'adjoints généralisés à résoudre.

Calcul du gradient

La dernière étape consiste simplement à évaluer numériquement l'équation 2.29 en utilisant l'équation 2.32. Les termes $\langle S^*, \Gamma_n \rangle$ et $\langle S_n, \Gamma^* \rangle$ étant égaux, deux moyens de calculer la contribution indirecte au gradient de la variation des paramètres sont possibles.

Pour la méthode explicite, le gradient est donné par:

$$\nabla u(\vec{X}, \phi) = \frac{\partial u}{\partial X_n} + \langle S^*, \Gamma_n \rangle \quad n \in (1, n_{var}) \quad (2.39)$$

et pour la méthode implicite:

$$\nabla u(\vec{X}, \phi) = \frac{\partial u}{\partial X_n} + \langle S_n, \Gamma^* \rangle \quad n \in (1, n_{var}) \quad (2.40)$$

Les résultats obtenus par Rozon [15] démontrent également l'équivalence numérique de ces deux approches.

2.9.4 Méthodes de programmation quasi-linéaire simples

Avec les gradients calculés et donc les coefficients de l'équation du système linéarisé (2.22) obtenus, différentes approches ont été proposées [9,14,15,17] pour la résolution du problème d'optimisation de la gestion du combustible.

L'approche la plus simple consiste à ne pas tenir compte de la contrainte quadratique (éq. (2.23)). Le problème devient donc entièrement linéaire et peut être résolu par la méthode du SIMPLEX. Cette méthode mathématique a l'avantage d'être simple et très répandue dans le domaine de l'optimisation en général. Cependant, le fait de ne pas tenir compte de la contrainte quadratique peut faire en sorte que le problème diverge.

La méthode MAP, une approche directement dérivée du SIMPLEX a été implantée par Rozon et al. [14] dans le code de calcul OPTTEX. Elle consiste simplement à utiliser de manière itérative la méthode du SIMPLEX avec la contrainte quadratique linéarisée. Le désavantage de cette méthode est de devoir résoudre plusieurs fois un SIMPLEX pour chaque itération externe. Cependant, à chaque itération interne aucun calcul de flux ou d'adjoint généralisé n'est nécessaire. Étant donné la simplicité du SIMPLEX, ce désavantage est négligeable en comparaison du fait que la contrainte quadratique est respectée. De plus, l'implantation de la méthode

reste très simple un fois la méthode du SIMPLEX programmée. Cette méthode a aussi été utilisé par Beaudet [17].

Une autre approche du code de calcul OPTeX a été utilisée par Beaudet [17] et Tajmouati [9]. Cette méthode basée sur la programmation quadratique et les conditions de Kuhn-Tucker, permet de prendre en compte directement la contrainte quadratique. À partir des travaux de Lemke en 1965 [23] sur la méthode du pivot complémentaire, une approche a été développée par Ferland [24] pour le cas spécifique d'un problème linéaire avec une contrainte quadratique. Cette méthode est référée par méthode de LEMKE par Rozon [15]. Son avantage principal est de pouvoir tenir compte en un seul calcul d'optimisation de la contrainte quadratique. Les résultats de Beaudet [17] et de Rozon (annexe B [15]) démontrent une bonne robustesse et efficacité de cette méthode, d'où son utilisation par Tajmouati. Dans le programme OPTeX, cette méthode est couplée avec la méthode du SIMPLEX ou de LEMKE-linéaire pour minimiser le temps de calcul dans le cas où la contrainte quadratique n'est pas active.

Alaoui [25] a également implanté la méthode du gradient réduit généralisé (GRG) dans OPTeX pour des simulations du réacteur en 2D. Cette méthode mathématique donne les mêmes résultats que la méthode MAP développée précédemment, avec pour avantage un nombre de calcul d'optimisation réduit pour arriver au même résultat. Beaudet [17] a également utilisé cette méthode pour des simulations 3D du réacteur. Par contre dans ses travaux, Beaudet affirme que la méthode du GRG n'est pas assez robuste pour de tels calculs.

CHAPITRE 3

IMPLANTATION DES MÉTHODES DE GRADIENTS DANS DONJON

La première approche choisie pour la résolution du problème d'optimisation de la gestion du combustible à l'équilibre du rechargement est basée sur la méthode des gradients. L'approche présentée dans le chapitre précédent et originellement utilisée dans le code OPTEX a été implantée dans le code DONJON. Le principe des méthodes de gradients est de 'suivre la pente' vers l'optimum petit à petit. Ainsi à chaque itération le gradient est calculé et est utilisé pour calculer le pas d'avance. Ce dernier est généralement limité pour faire en sorte que l'approximation linéaire des fonctionnelles faite à chaque itération reste valide.

Nous présenterons dans ce chapitre l'approche suivie pour la programmation des différentes étapes de la résolution du problème (2.18) dans DONJON. La validation du code sera également décrite, avec une attention plus particulière pour l'étape du calcul des gradients. Par la suite, nous nous attarderons sur deux premiers exemples de calcul d'optimisation pour valider l'algorithme général proposé. Nous soulignerons également le problème singulier du point de départ réalisable.

3.1 Algorithme général d'optimisation avec la théorie des perturbations généralisées

3.1.1 Description de l'algorithme

DONJON est un code de diffusion modulaire. [7] Chaque module a pour but soit de définir une partie du réacteur ou les options de rechargement, soit de discrétiser les équations ou enfin de résoudre l'équation de diffusion. Ainsi les modules créent ou modifient un ensemble de données qui peuvent par exemple représenter la géométrie, les matrices systèmes ou le flux neutronique. Ces données sont regroupées dans différents objets, qui peuvent être à leur tour utilisés par d'autres modules.

Cette approche modulaire a donc été utilisée pour l'implantation des calculs d'optimisation. L'algorithme général de résolution est présenté à la figure 3.1. Après avoir défini la géométrie et les composantes du réacteur, nous pouvons rassembler tous les choix pris pour la gestion du combustible dans un vecteur d'état. La première étape du problème d'optimisation de la gestion du combustible consiste alors à définir les paramètres de décision ainsi que la fonction objectif et les contraintes. Cette étape est effectuée avec le module `FOBJCT:`. Les options pour la résolution du problème d'optimisation, telles que la méthode ou les critères de convergence, sont fixées avec le module `QLPUTL:`. L'initialisation du problème est alors complétée.

Pour un ensemble de paramètres de décision, l'étape suivante consiste à évaluer les fonctionnelles décrites au chapitre précédent, pour cela le flux et son adjoint doivent être calculés. Le flux et son adjoint pour un réacteur vide de caloporteur peuvent être également requis. Les fonctionnelles sont évaluées avec le module `FOBJCT:` ce qui permet de savoir si la fonction objectif est améliorée (`QLPUTL:`). Si tel est le cas, l'algorithme continue avec la même contrainte quadratique.

La prochaine grande étape est d'évaluer les gradients au point courant pour linéariser le problème d'optimisation (2.18). Chaque variable de décision est perturbée à tour de rôle, ce qui permet de calculer les sections efficaces perturbées et donc les matrices systèmes perturbées correspondantes. Une attention plus particulière sera apportée sur la méthode de calcul des propriétés perturbées dans une section suivante. Celles-ci sont utilisées pour obtenir la dérivée de la valeur propre λ grâce au module `PERTUR:`, éq. (2.26). Les sources directes peuvent alors être obtenues avec le module `GPTSRC:`, éq. (2.33). La partie directe du gradient est aussi calculée (éq. (2.29)) ce qui permet de ne pas avoir à stocker les propriétés perturbées pour chaque variable de décision, seules les sources directes sont sauvegardées dans un objet. La prochaine sous étape du calcul des gradients est l'évaluation des sources indirectes avec le module `GPTSRC:`, éq. (2.34). Enfin, la plus petite série d'adjoints généralisés utile au calcul des gradients (méthode explicite (2.37) ou implicite (2.38)) est obtenue avec le module `GPTFLU:`, ce qui permet d'évaluer la partie indirecte des gradients avec `GPTGRD:`, éq. (2.29).

Une fois les gradients des fonctionnelles obtenus, nous pouvons résoudre le problème d'optimisation linéarisé (2.22) avec le module `QLP:`. Celui-ci utilise les options par défaut données au tableau 3.1 si aucune n'a été spécifiée lors de l'initialisation du problème. Nous reviendrons sur le choix des valeurs de défaut à la fin de cette section. L'algorithme tel que suggéré à la figure 3.1 prévoit également un resserrement de la contrainte quadratique si la fonction objectif ne s'est pas améliorée avec le vecteur de décision calculé à l'itération précédente. Dans ce cas, les gradients calculés précédemment avec le vecteur de décision correspondant à une valeur valide de la fonction objectif sont utilisés avec la nouvelle valeur de la contrainte quadratique pour résoudre l'équation (2.22). Nous reviendrons ultérieurement plus en détail sur l'évaluation de la validité du nouveau vecteur de décision calculé à chaque itération.

Une fois le pas d'avance calculé, le nouveau vecteur de décision est obtenu avec le module `QLP`: et un test sur la convergence est effectué (`QLPUTL`:). L'algorithme s'arrête lorsque la convergence est atteinte. Une description plus précise des tests de convergence est faite à la section suivante. Si la résolution du problème linéarisé s'avère impossible avec le module `QLP`:, cela signifie que le vecteur de décision courant est trop éloigné du domaine réalisable (2.9.1) pour la contrainte quadratique actuelle. Cela peut advenir proche de la frontière du domaine réalisable quand le vecteur de décision courant, validé à l'itération précédente, est en fait trop imprécis une fois les gradients recalculés. Pour que la résolution par itération puisse continuer, nous faisons un retour en arrière. Le nouveau vecteur de décision est alors donné par:

$$\vec{X}^{k+1} = \frac{\vec{X}^k + \vec{X}_v^{k-1}}{2}$$

où \vec{X}_v^{k-1} représente l'avant dernier point valide (qui a donc une résolution du problème linéarisé réussie). La validité de ce nouveau point sera alors imposée (sans aucun test sur l'amélioration de la fonction objectif ou le respect des contraintes).

À la dernière étape, les variables de décision sont alors mises à jour dans le vecteur d'état et les objets qui le contiennent. Le calcul du flux neutronique à la prochaine itération prendra alors en compte la nouvelle valeur du vecteur de décision.

En ce qui concerne les valeurs de défaut des différentes options d'optimisation, nous avons premièrement choisi la minimisation, car c'est le cas qui s'est toujours présenté. La méthode de programmation mathématique de Lemke a été retenue pour sa rapidité par rapport à la méthode MAP. La contrainte quadratique a essentiellement été initialisée pour être sûr qu'elle ne soit pas nulle. Sa valeur est cependant très discutable, étant donné que les variables de décision envisagées sont du burnup (de l'ordre de $10^4 MWj/t$) et de l'enrichissement (de l'ordre de 1%).

Tableau 3.1 Option par défaut pour l'optimisation dans le module QLPUTL :

grandeur	valeur par défaut
type d'optimisation	minimisation
méthode de résolution de (2.22)	Lemke
$S_{initial}$	1.
perturbation δ	0.01=1%
critère de convergence externe ϵ	0.001

Ainsi, à moins de spécifier des poids différents de 1 dans les fichiers d'entrée, la contrainte quadratique $S_{initial}$ est soit trop faible ou trop grande. Les gradients ne devrait pas dépendre de la valeur de la perturbation des variables de décision, si la bonne approche est prise pour les calculer, comme nous le verrons ultérieurement. La valeur de défaut est donc une valeur initialisée. Enfin, le critère de convergence est par défaut égal à la valeur du critère de convergence de la forme de flux axiale généralement utilisé.

3.1.2 Validation du nouveau point et convergence

La validation du nouveau point et le choix des tests de convergence sont très importants pour obtenir de bons résultats dans un temps de calcul raisonnable. Rozon ([15]) propose plusieurs choix pour ces deux étapes. Nous décrivons dans cette section les différentes options implantées dans le code, ainsi que les recommandations que nous ont suggérées les premiers résultats.

La validation du nouveau vecteur de décision à chaque itération consiste simplement à comparer la nouvelle valeur de la fonction objectif avec celle de l'itération précédente. En cas d'amélioration, l'algorithme continue sans changer la contrainte quadratique. Dans le cas contraire, cela signifie que l'approximation faite en linéarisant le problème (2.18) n'est plus assez précise avec la contrainte quadra-

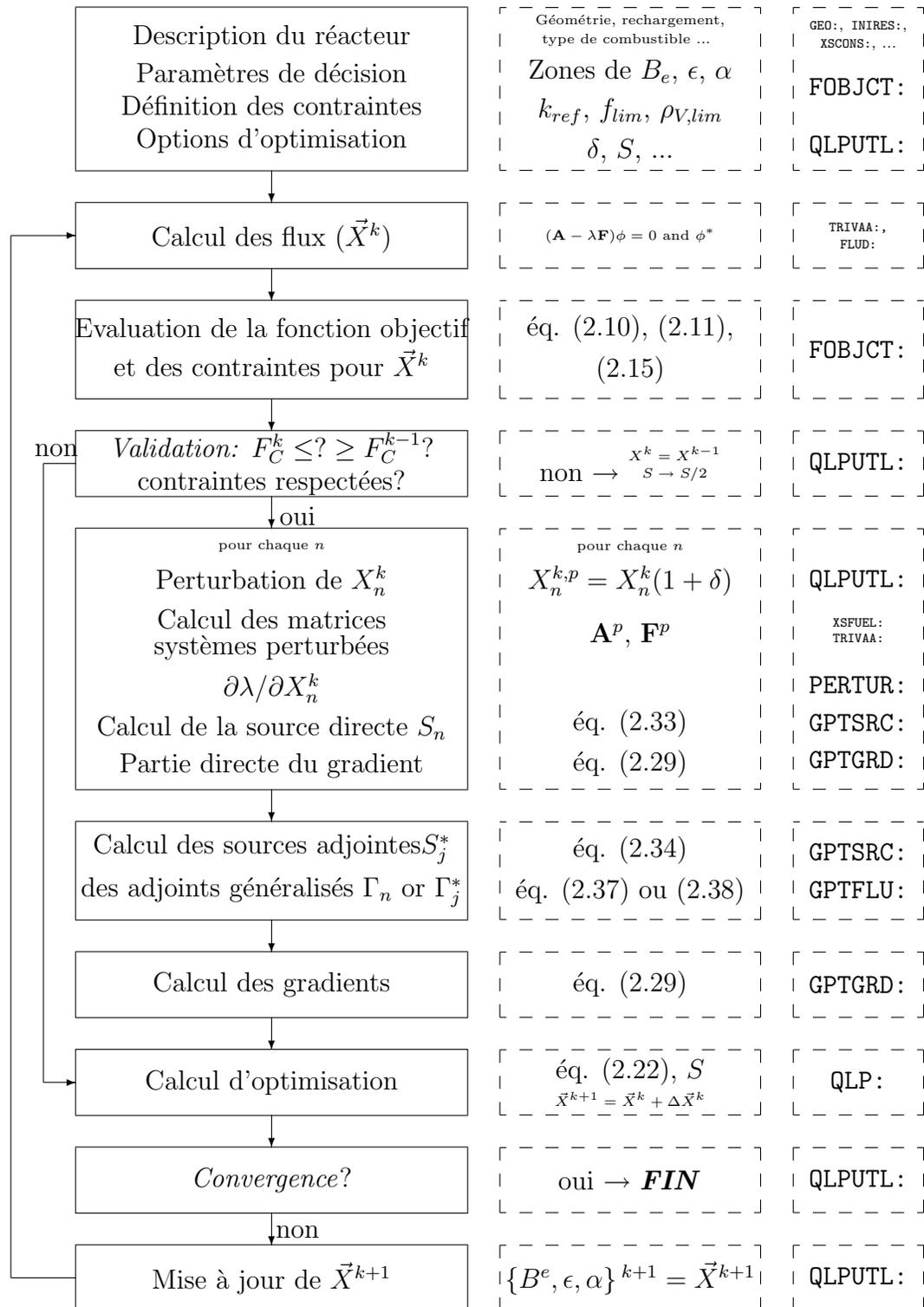


Figure 3.1 Algorithme général pour l'optimisation avec la méthode des gradients.

tique actuelle. L'optimum a été dépassé. Le vecteur de décision valide de l'itération précédente est alors repris et la contrainte quadratique doit être resserrée pour que le problème linéarisé (2.22) reste valable. Deux moyens sont implantés dans le code pour réduire la contrainte S . Le premier choix consiste simplement à diviser S par 2. La deuxième option consiste à faire une interpolation d'ordre 2 entre la valeur courante (non valide) de la fonction objectif f^k et celle de l'itération précédente f^{k-1} . La nouvelle valeur S^{k+1} de la contrainte quadratique correspond alors à l'extremum de la parabole qui passe par f^k et f^{k+1} . Cette valeur est donnée analytiquement par :

$$S^{k+1} = \frac{df^{k-1} \|\Delta \vec{X}^{k-1}\|^2 / 2}{df^{k-1} \|\Delta \vec{X}^{k-1}\| - (f^k - f^{k-1})} \quad (3.1)$$

où df^{k-1} est la dérivée de la fonction objectif à l'itération précédente donnée par :

$$df^{k-1} = (\vec{\nabla} f^{k-1} \cdot \Delta \vec{X}^{k-1}) / \|\Delta X^{k-1}\|$$

Afin de réduire le nombre d'itérations, une contrainte quadratique grande au début de la résolution peut être utilisée. Cependant, cela peut entraîner une divergence de l'algorithme. En effet, proche de la frontière du domaine réalisable une linéarisation précise est requise. Une trop grande approximation peut entraîner, selon la méthode décrite précédemment, la validation d'un point qui ne respecte pas les contraintes mais qui améliore la fonction objectif de la meilleure solution trouvée jusqu'à présent (selon l'équation (2.10)). Afin d'éviter ce phénomène, nous avons rajouté en option une vérification de la faisabilité du vecteur de décision actuel. En cas de non respect des contraintes (à une tolérance près), la contrainte quadratique est réduite.

Plusieurs critères d'arrêt pour la convergence de l'algorithme peuvent être choisis: la variation du vecteur de décision $\|\Delta X^k\| / \|X^k\| < \epsilon$, la fonction objectif

$|\Delta f^k|/|f^k| < \epsilon$, le nombre d'itérations maximum atteint ou enfin la contrainte quadratique devenue trop petite $S^k < S_{min} = \|X^k\|\epsilon$. Nous supposons par la suite que le nombre d'itérations maximum est toujours pris en compte comme critère d'arrêt. Prendre juste la variation de X ou de f peut entraîner un arrêt de l'algorithme sans avoir atteint la convergence. En effet, deux cas extrêmes peuvent se présenter. Soit la fonction varie très peu pour une itération donnée, cela correspond géométriquement à un plateau local. Soit la fonction varie beaucoup, mais une contrainte fait en sorte que la variation du vecteur de décision est faible, auquel cas l'erreur d'approximation du problème linéarisé peut être importante. Même si la fonction objectif ne s'améliore que très peu et que le vecteur de décision ne change presque pas, la contrainte quadratique reste inchangée. Nous ne recommandons donc pas de prendre seulement S comme seul autre critère d'arrêt, car le nombre d'itérations maximum peut être atteint sans avoir une amélioration notable de la fonction objectif. Cela représenterait alors juste une perte de temps de calcul. Par contre le critère sur S peut éviter de faire des calculs inutiles. En effet s'il est actif, alors quelque soit le résultat pour ΔX à la prochaine itération, nous aurons $\|\Delta X\| \leq S_{min} = \|X^k\|\epsilon$, et le critère sur ΔX sera obligatoirement actif (variation relative de ΔX trop faible).

3.1.3 Vérification successive des étapes

De nombreuses étapes sont nécessaires à la résolution du problème d'optimisation de la gestion du combustible comme le montre l'algorithme général à la figure 3.1. Aussi afin de valider chacune d'entre elles, nous avons créé plusieurs fichiers d'entrée pour vérifier les résultats obtenus. Certaines vérifications étaient triviales et consistaient juste à regarder une valeur dans un objet, d'autres au contraire nécessitaient de recalculer indépendamment les résultats obtenus. Pour cela, nous avons utilisé MATLAB afin de vérifier sur des cas test la programmation des di-

verses équations. Il serait fastidieux et inutile de reprendre ici tous ces tests. Cependant nous pouvons noter que l'évaluation des fonctionnelles a été testée pour le cas où elles sont prédéfinies et implantées comme pour le problème (2.22) ou définies par l'utilisateur (cf. [7] ou annexe). Nous avons également utilisé MATLAB pour vérifier la dérivation des sections efficaces, ainsi que la résolution de plusieurs problèmes de programmation quasi-linéaire.

3.2 Précision du calcul des gradients

3.2.1 Introduction

La principale approximation utilisée pour la résolution du problème d'optimisation de la gestion du combustible est la linéarisation des fonctionnelles. Aussi, pour l'utilisation de la méthode des gradients, la précision dans l'évaluation de ces derniers est primordiale pour la stabilité et l'efficacité de cette méthode. Nous décrirons donc dans cette section plus en détail comment les gradients sont calculés et les vérifications faites sur la précision des résultats.

Afin de tester la précision du calcul des gradients, nous avons simulé un réacteur type CANDU-6 avec les contrôleurs liquides (tous à 50%) et sans barres de compensation. Tous les calculs présentés dans cette section ont été réalisés en utilisant une représentation du cœur à l'équilibre du rechargement. Ce modèle est utilisé pour l'optimisation du burnup moyen de sortie dans un CANDU-6 avec 2 zones de combustion radiale. Une discrétisation spatiale d'un point par grappe a été utilisée pour les calculs de réacteur.

Les fonctionnelles testées dans cette section sont le coût en combustible F_C comme fonction objectif, la puissance canal maximum q_j et le coefficient de multiplication

des neutrons k_{eff} comme contrainte tels que décrits par les équations (2.10), (2.15) et (2.11) respectivement.

3.2.2 Théorie

Comme nous l'avons vu dans le chapitre précédent, la dérivée des matrices systèmes ($\partial\mathbf{A}/\partial B^e$ et $\partial\mathbf{F}/\partial B^e$) sont nécessaires pour calculer le terme de source directe (2.33) et la dérivée de la formule de Rayleigh (2.26) ou (2.27). Comme le suggère l'équation (2.25), deux approches sont envisageables pour obtenir les dérivées des sections efficaces utiles au calcul des dérivées des matrices systèmes.

La première méthode, la plus intuitive, consiste simplement à perturber directement les variables de décisions et à recalculer les sections efficaces correspondantes. Cela revient donc à faire une approximation de la dérivée des sections efficaces par la pente numérique d'ordre 1 (à droite si la perturbation $\delta > 0$ sinon à gauche). Cette méthode sera par la suite nommée le calcul semi-analytique des gradients. La dérivée des matrices systèmes est alors donnée par:

$$\begin{aligned}\partial\mathbf{A}/\partial B^e &= \frac{\mathbf{A}(\bar{\Sigma}(B_p^e)) - \mathbf{A}(\bar{\Sigma}(B^e))}{\delta B^e} \\ \partial\mathbf{F}/\partial B^e &= \frac{\mathbf{F}(\bar{\Sigma}(B_p^e)) - \mathbf{F}(\bar{\Sigma}(B^e))}{\delta B^e}\end{aligned}$$

d'où

$$\begin{aligned}\mathbf{A}_p &= \mathbf{A}(\bar{\Sigma}(B_p^e)) = \mathbf{A}(\bar{\Sigma}((1 + \delta)B^e)) \\ \mathbf{F}_p &= \mathbf{F}(\bar{\Sigma}(B_p^e)) = \mathbf{F}(\bar{\Sigma}((1 + \delta)B^e))\end{aligned}$$

La deuxième approche consiste à appliquer la formule de Taylor aux propriétés des

matériaux. Cette méthode sera nommée le calcul analytique des gradients. Dans ce cas, les matrices systèmes perturbées sont données par :

$$\begin{aligned}\partial\mathbf{A}/\partial B^e &= \mathbf{A} \left(\frac{\partial\bar{\Sigma}(B^e)}{\partial B^e} \right) \\ \partial\mathbf{F}/\partial B^e &= \mathbf{F} \left(\frac{\partial\bar{\Sigma}(B^e)}{\partial B^e} \right)\end{aligned}$$

d'où

$$\begin{aligned}\mathbf{A}_p &= \mathbf{A} \left(\bar{\Sigma}(B^e) \right) + \delta B^e \frac{\partial\mathbf{A}}{\partial B^e} = \mathbf{A} \left(\bar{\Sigma} + \delta B^e \frac{\partial\bar{\Sigma}(B^e)}{\partial B^e} \right) \\ \mathbf{F}_p &= \mathbf{F} \left(\bar{\Sigma}(B^e) \right) + \delta B^e \frac{\partial\mathbf{F}}{\partial B^e} = \mathbf{F} \left(\bar{\Sigma} + \delta B^e \frac{\partial\bar{\Sigma}(B^e)}{\partial B^e} \right)\end{aligned}$$

$d\bar{\Sigma}(B_e)/dB_e$ est la dérivée des sections efficaces moyennées dans le temps par rapport au burnup moyen de sortie. Or, selon l'équation (2.1), les propriétés moyennées dans le temps sont données par :

$$\bar{\Sigma}_{jk} = \frac{1}{B_{jk}^{eoc} - B_{jk}^{boc}} \int_{B_{jk}^{eoc}}^{B_{jk}^{boc}} \Sigma(B) \cdot dB$$

En définissant les facteurs de formes f_1 et f_2 constants pour une distribution de flux donnée, par:

$$\begin{aligned}B_{jk}^{boc} &= f_1 B_j^e \\ B_{jk}^{eoc} &= f_2 B_j^e\end{aligned}$$

l'équation (2.1) devient :

$$\bar{\Sigma} = \frac{1}{(f_2 - f_1)B_j^e} \int_{f_1 B_j^e}^{f_2 B_j^e} \Sigma(B) \, dB \quad (3.2)$$

D'où :

$$\begin{aligned}
\frac{\partial \bar{\Sigma}}{\partial B_j^e} &= -\frac{1}{(f_2 - f_1)B_j^{e^2}} \int_{f_1 \cdot B_j^e}^{f_2 B_j^e} \Sigma(B) dB \\
&\quad + \frac{1}{(f_2 - f_1)B_j^e} \left(f_2 \Sigma(f_2 B_j^e) - f_1 \Sigma(f_1 B_j^e) \right) \\
&= -\frac{\bar{\Sigma}}{B_j^e} + \frac{1}{B_{jk}^{eoc} - B_{jk}^{boc}} \left(\frac{B_{jk}^{eoc}}{B_j^e} \Sigma(B_{jk}^{eoc}) - \frac{B_{jk}^{boc}}{B_j^e} \Sigma(B_{jk}^{boc}) \right) \quad (3.3)
\end{aligned}$$

L'implantation de cette méthode nous a conduit à développer un nouveau module **XSFUEL** : dans **DONJON**. Pour plus de détails pratiques, le lecteur pourra se référer au guide de l'utilisateur [26].

Le reste de la procédure pour l'obtention des gradients à partir des propriétés et matrices systèmes perturbées suit l'algorithme décrit à la figure 3.1.

Afin de vérifier les résultats obtenus avec les approches analytique et semi-analytique, nous avons également calculé les gradients numériquement. Le calcul numérique des gradients est fait en utilisant la méthode de Ceschino [27] qui est implantée dans le module de vérification **GPTVRF** : de **DONJON**. Ce type de calcul ne fait bien sûr pas partie de l'algorithme d'optimisation en général, puisqu'il requière un nombre important de calculs de flux.

3.2.3 Comparaison et résultats

Parmi les 114 canaux surveillés, trois contraintes sur la puissance canal ont été retenues en plus de la contrainte sur la réactivité et de la fonction objectif F_C pour l'analyse des résultats . Pour ces trois canaux numérotés 94, 60 et 19, la puissance canal varie respectivement beaucoup, moyennement et la moins entre $\pm 20\%$ de perturbation.

La méthode semi-analytique ne donne pas des résultats précis, car ils dépendent grandement de la valeur de la perturbation choisie. Aussi nous n'avons pas rapporté les gradients obtenus par cette approche.

Pour le calcul numérique des gradients, différents ensembles de points ont été utilisés pour comparer l'influence des perturbations choisies. Celles-ci sont données par le tableau 3.2. Le tableau 3.3 donne la valeur des gradients des fonctionnelles par rapport au burnup moyen de sortie dans la zone centrale, évalués analytiquement et numériquement pour ces différents nombres de points. Nous comparons les gradients analytiques aux valeurs numériques. Premièrement, nous pouvons remarquer que la valeur des gradients calculés numériquement varie beaucoup avec le choix et le nombre de points utilisés par la méthode de Ceschino sauf pour la contrainte sur la réactivité. La dernière ligne du tableau 3.3 donne la variation des fonctionnelles lorsque la valeur de décision est perturbée entre $\pm 20\%$. Quand la fonctionnelle varie beaucoup, comme pour F_C et q_{94} , l'amplitude des perturbations utilisées est plus influente que le nombre de points pour la précision des gradients numériques obtenus. L'utilisation de faibles perturbations des variables de décision ($\sim 1\%$, cas 'num4p') est alors recommandée pour le calcul numérique des gradients dans ce cas. Quand la fonctionnelle varie peu ($\sim 1\%$), comme pour q_{60} , la même constatation peut être faite. Cependant dans ce cas, si la perturbation est trop faible, les erreurs numériques sont non négligeables. Si la variation de la fonctionnelle est très faible ($\sim 0.1\%$) alors les erreurs numériques deviennent prépondérantes et les gradients évalués numériquement n'ont plus de signification. Enfin pour la contrainte sur la réactivité, la variation est du même ordre que pour q_{60} . Cependant les résultats numériques sont plutôt constants. Ceci peut s'expliquer par le caractère global de cette contrainte qui porte sur tout le réacteur.

Les résultats présentés au tableau 3.3 ne permettent cependant pas de valider le calcul analytique des gradients étant donné les fortes variations des gradients évalués

Tableau 3.2 Perturbations utilisés pour le calcul numérique des gradients

méthode	% de perturbation de la variable de décision								
	0	± 0.25	± 0.5	± 1	± 2	± 5	± 10	± 15	± 20
num 17	X	X	X	X	X	X	X	X	X
num 13	X	X	X	X	X	X	X		
num 4p	X		X	X					
num 4m	X					X	X		
num 4l	X						X		X

numériquement. Nous avons donc tracé la valeur des fonctionnelles F_C , k_{eff} et q_{19} pour vérifier si le gradient analytique calculé est bien tangent à la fonctionnelle, ce qui correspond aux figures 3.2 à 3.4 respectivement. Pour la fonction objectif (fig. 3.2), le gradient analytique est très précis. Il peut également être utilisé pour obtenir une bonne extrapolation de F_C pour une faible perturbation ($\pm 2\%$) même si la fonctionnelle est non linéaire pour de plus grandes perturbations. Pour la contrainte sur la réactivité (fig. 3.3), le gradient analytique est encore très proche de la pente en 0% de perturbation. La fonction est presque linéaire dans l'intervalle de perturbation de $\pm 20\%$. Ceci explique les résultats constants pour l'évaluation numérique des gradients pour cette fonctionnelle. Enfin, pour la contrainte sur la puissance canal qui correspond au pire résultat du calcul numérique des gradients q_{19} , nous pouvons voir que même quand les erreurs numériques sont prépondérantes, la théorie des perturbations généralisées donne une bonne approximation du gradient. En effet à la figure 3.4, le gradient analytique (segment de droite) suit la tendance de la courbe faite par les points à grande perturbation ($\geq 5\%$ en valeur absolue). Le comportement des fonctionnelles q_{60} et q_{94} est le même que pour la fonction objectif.

Tableau 3.3 Gradient des fonctionnelles suivant les deux méthodes de calcul et l'amplitude de la variation entre $\pm 20\%$ de perturbation

méthode	$dF_C \cdot 10^4$	$dk_{eff} \cdot 10^5$	$dq_{94} \cdot 10^4$	$dq_{60} \cdot 10^5$	$dq_{19} \cdot 10^6$
analytique	-.095554	-.544985	-.398026	-.667170	-.075373
num 17	-.166459	-.548926	-.334857	-.459642	0.854479
$\Delta\%$	74.204	0.7231	-15.87	-31.10	-1233.
num 13	-.125357	-.551918	-.364329	-.529252	-.517865
$\Delta\%$	31.190	1.2721	-8.466	-20.67	587.07
num 4p	-.098221	-.563608	-.417016	0.928672	5.28247
$\Delta\%$	2.7914	3.4170	4.7709	44.6433	-7108.
num 4m	-.125274	-.551111	-.374562	-.517047	0.900792
$\Delta\%$	31.103	1.1240	-5.895	-22.50	-1295.
num 4l	-.164207	-.551854	-.335938	-.461314	0.112798
$\Delta\%$	71.848	1.2603	-15.59	-30.85	-1596.

u	F_C	k_{eff}	q_{94}	q_{60}	q_{19}
$ \Delta u /u$ (%)	~ 22	~ 1.7	~ 9.6	~ 1.7	~ 0.1

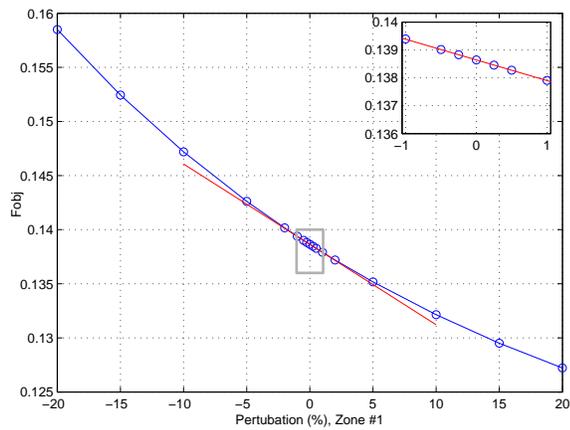


Figure 3.2 Variation de F_C avec le burnup moyen de sortie de la zone centrale.

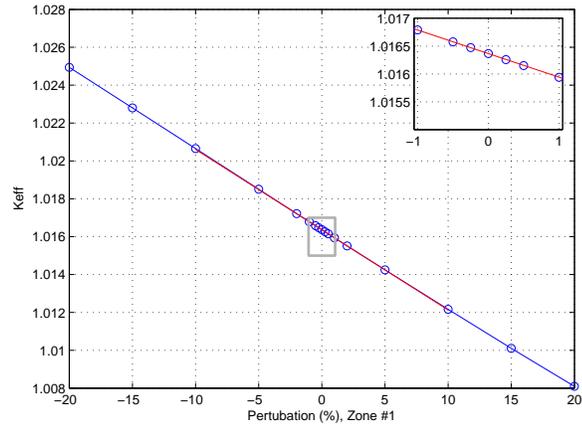


Figure 3.3 Variation de k_{eff} avec le burnup moyen de sortie de la zone centrale.

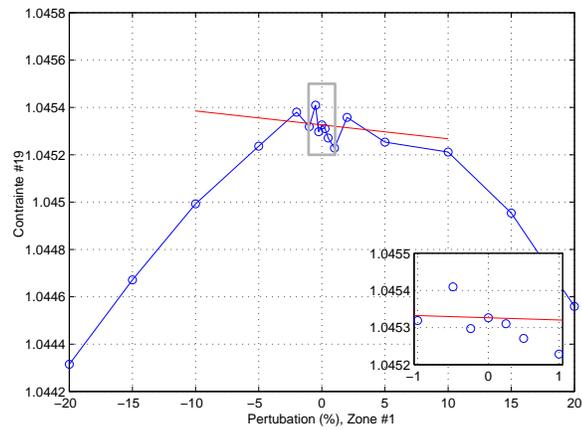


Figure 3.4 Variation de q_{19} avec le burnup moyen de sortie de la zone centrale.

3.3 Vérification de la convergence des résultats

Toutes les étapes de la résolution du problème d'optimisation de la gestion du combustible ont été validées séparément. Afin de tester la convergence de l'algorithme, nous avons optimisé la distribution de burnup moyen de sortie dans un réacteur CANDU-6. Les résultats ont été comparés au comportement attendu analytiquement. Le modèle de réacteur choisi pour tester la convergence ne comprend que les barres liquides remplies à 50%, aucune barre de compensation n'est présente. Deux études seront présentées, premièrement l'optimisation avec 2 variables de décision correspondant au burnup moyen de sortie des 2 zones de combustion. La deuxième étude traitera de l'influence d'un plus grand nombre de zones de combustion (donc du nombre de variables de décision) sur les résultats de l'optimisation.

3.3.1 Première optimisation d'un cas simple

La répartition des 2 zones de combustion est donnée par la figure 3.5 en quart de cœur. La figure 3.6a représente la valeur des burnups moyens de sortie (au centre $B_{e,1}$ et en périphérie B_2^e) en fonction de la contrainte sur la puissance canal. Sur la figure 3.6b nous avons tracé la valeur de la fonction objectif ainsi que la puissance canal maximum obtenue en fonction de la puissance canal maximum autorisée. Afin d'analyser les résultats obtenus, nous avons réécrit la fonction objectif (équ. (2.10)) pour le cas particulier à 2 zones de combustion, comme suit:

$$F_c = \left(n_1 \underbrace{\frac{C_u \bar{P}_1}{B_1^e}}_{f_1} + n_2 \underbrace{\frac{C_u \bar{P}_2}{B_2^e}}_{f_2} \right) / P_{tot} \quad (3.4)$$

où n_1 et n_2 représentent le nombre de canaux dans la zone 1 et 2 respectivement. Supposons que B_1^e soit plus grand que B_2^e , ce qui est généralement le cas pour aplatir la distribution de puissance. Si B_1^e diminue alors la puissance moyenne \bar{P}_1 augmente car le combustible est plus neuf, alors B_2^e doit également augmenter pour maintenir la criticité et donc \bar{P}_2 diminue (car la puissance totale est constante). Par conséquent le facteur f_1 augmente et f_2 diminue. Étant donné que dans notre cas, $n_1 = 176$ et $n_2 = 204$, la deuxième partie de F_c a plus de poids que la première. Ainsi B_2^e va avoir tendance à augmenter pour que F_c diminue. Une approche plus physique consiste à dire que quand B_2^e augmente, le flux neutronique en périphérie diminue, et donc les pertes aussi d'où la diminution de la fonction coût. Or plus B_2^e augmente plus la distribution de puissance présente un pic au centre et donc la puissance canal maximum augmente. La contrainte sur la puissance canal maximum autorisée limite B_2^e et donc le burnup moyen de sortie B_{moy}^e du cœur. Cependant, même en relaxant la contrainte de puissance canal maximum autorisée, nous atteignons un point où B_1^e ne peut plus être réduit. En effet, petit à petit quand B_1^e diminue, f_1 augmente plus rapidement que f_2 ne diminue. Alors un équilibre est atteint et le facteur de multiplication des neutrons k_{eff} devient la seule contrainte active. À partir de ce moment, même si la puissance canal autorisée (P_{lim}) est augmentée, la puissance canal maximum ($P_{C,max}$) reste constante et les burnups moyens de sortie restent inchangés.

Les résultats présentés à la figure 3.6 sont en accord avec le comportement analytique décrit précédemment: diminution de la fonction objectif et de B_1^e quand la puissance canal autorisée (P_{lim}) augmente. À partir d'environ 11.1MW, un plateau est atteint pour toutes les valeurs. Cela correspond au moment où la contrainte sur la puissance canal n'est plus active ($P_{C,max} < P_{lim}$). La contrainte sur k_{eff} est alors la seule active.

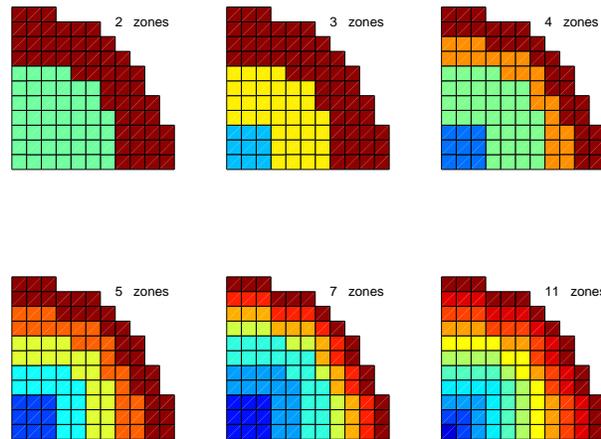


Figure 3.5 Répartition radiale des zones de combustion pour différents nombres de zones.

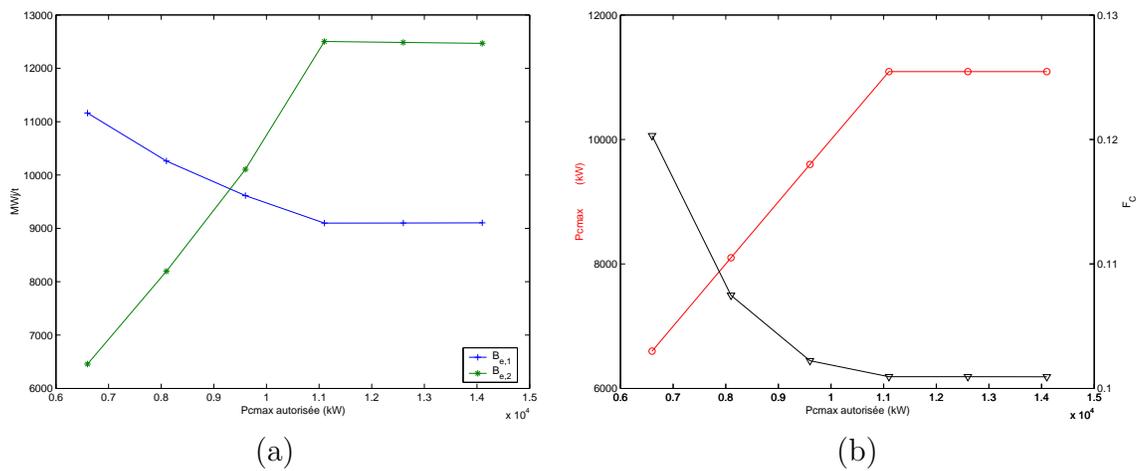


Figure 3.6 Impact de la contrainte de puissance canal maximum sur l'optimisation du burnup moyen de sortie à 2 zones dans un réacteur CANDU-6 sans barres de compensation.

3.3.2 Influence du nombre de zones de combustion optimisées

La répartition radiale pour différents nombres de zones de combustion ($N = 2, 3, 4, 5, 7$ et 11) est donnée à la figure 3.5. Les figures 3.7a, b, c et d représentent la distribution du burnup moyen de sortie le long de la rangée L pour des puissances canal maximum autorisées de 6600, 8100, 9600 et 11100kW respectivement. Le même raisonnement mathématique qu'avec les calculs à 2 zones peut être appliqué à l'équation F_C (2.10). Quand la contrainte sur la puissance canal maximum (P_{lim}) est faible (figure 3.7a), les burnup moyens de sortie au centre doivent être élevés pour aplatir la forme du flux et donc la distribution de puissance ($B_i^e \sim 10000 MWj/t$). Par contre plus cette contrainte est relâchée, plus le burnup moyen de sortie diminue au centre pour augmenter en périphérie. Ceci est d'autant plus marqué quand le nombre de zones est grand (figures 3.7b, c et d), ce qui s'explique par le fait qu'à N zones radiales, nous avons $n_N > n_{N-1} > \dots > n_1$. L'exemple le plus frappant de cette différence de répartition du burnup moyen de sortie se trouve pour une contrainte de 8100kW de puissance canal maximum. En périphérie (canal L22), le burnup moyen de sortie pour un calcul à 2 zones est de $8195 MWj/t$, et de $11191 MWj/t$ pour 11 zones, soit une différence de plus de 36%. Ce phénomène peut également s'expliquer physiquement par l'importance des fuites. En effet, plus le nombre de zones est grand, plus le profil de burnup peut être ajusté radialement pour minimiser les fuites. Analytiquement, le flux doit être diminué en périphérie pour réduire les fuites de neutrons, ce qui revient à augmenter le burnup moyen de sortie dans les zones correspondantes.

Comme l'enrichissement du combustible est fixe, la fonction objectif est inversement proportionnelle au burnup moyen de sortie du cœur. Ainsi plus ce dernier est grand, plus la fonction objectif est minimisée. La figure 3.8 donne le burnup moyen de sortie du cœur en fonction du nombre de zones de combustion pour

différentes contraintes de puissance canal. Un grand nombre de zones de combustion permet d'améliorer l'optimum de la fonction objectif quelque soit la contrainte sur la puissance canal. Cependant, le gain en burnup (augmentation du burnup moyen de sortie entre 2 et 11 zones, représenté par les doubles flèches verticales) avec le nombre de zones est plus grand quand la contrainte sur la puissance canal est petite (courbe bleue 6600kW). En effet, quand la contrainte sur la puissance canal maximum (P_{lim}) est petite, le profil du burnup à deux zones est très limité et reste très 'grossier' par rapport la minimisation des fuites. Ainsi avec le passage à 11 zones, même si elle demeure très contrainte par P_{lim} , la distribution de burnup diminue notablement les fuites. Dans le cas opposé, avec P_{lim} grand, le burnup à 2 zones est déjà dans une configuration qui permet de beaucoup minimiser les fuites. Le passage à 11 zones n'apporte alors qu'une faible amélioration.

Afin d'étudier plus en détails l'influence de la forme du flux initiale sur le burnup moyen de sortie, nous avons également optimisé la gestion du combustible d'un réacteur CANDU-6 avec les barres de compensation présentes en plus des contrôleurs liquides pour 2 et 11 zones de combustion. Les résultats sont présentés au tableau 3.4. Lorsque les barres de compensation sont présentes, elles entraînent une perte de neutrons. Le burnup moyen de sortie est donc plus faible. De plus, la distribution du flux est également naturellement aplatie. Les fuites de neutrons sont donc plus importantes (quelque soit le nombre de zones de combustion choisi). Le gain en burnup (entre 2 à 11 zones) est alors plus grand quand les barres sont absentes. Dans ce cas, la distribution de burnup moyen de sortie est l'effet le plus important pour réduire les fuites.

Le comportement de notre solution optimale avec un nombre de plus en plus élevé de zones de combustion est analogue à celui obtenu analytiquement avec un modèle de rechargement continu en 1D. La figure 3.9 montre les résultats obtenu par Wight [28] dans ce cas. Pour le réacteur cylindrique présenté, nous pouvons voir que le

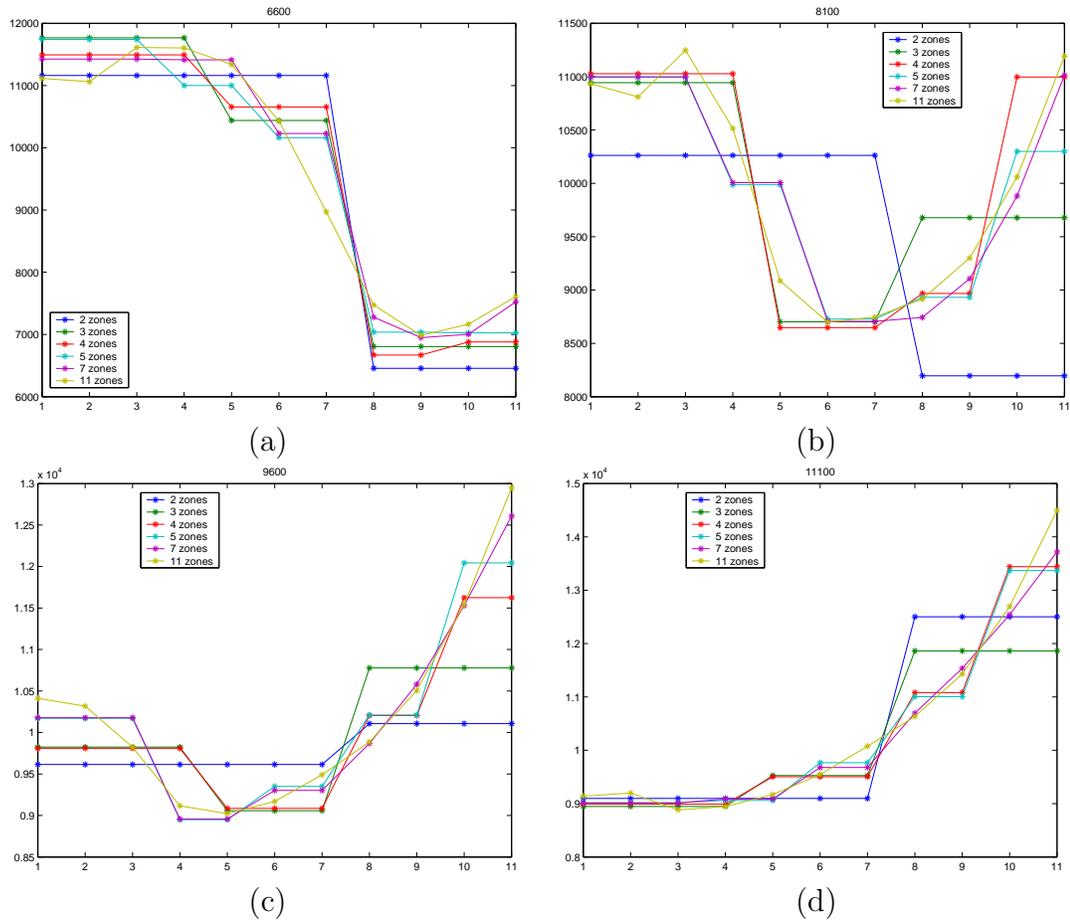


Figure 3.7 Variation de la distribution du burnup moyen de sortie à n zones pour différentes contraintes de puissance canal maximum (L11 centre - L22 périphérie).

burnup augmente en périphérie. Le haut burnup uniforme au centre du réacteur est une contrainte imposée par l'auteur sur la distribution du flux (flux au centre plat).

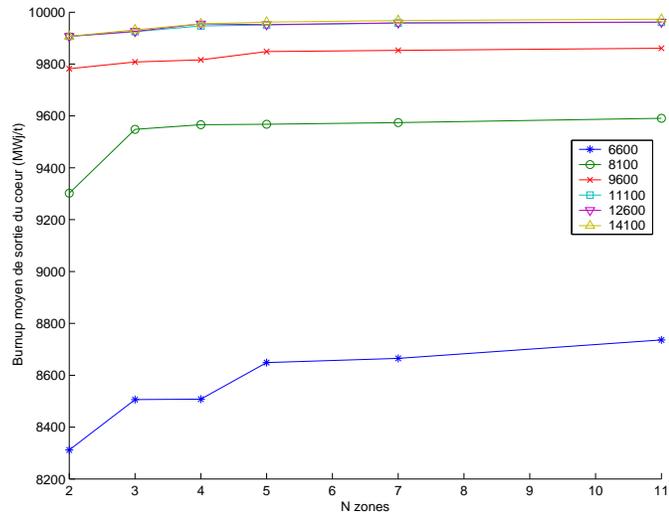


Figure 3.8 Burnup moyen de sortie du cœur en fonction du nombre de zones de combustion pour différentes contraintes de puissance canal.

	2 zones	11 zones	gain (%)
B_{moy}^e sans	8312.	8736.	5.1
B_{moy}^e avec	7552.	7610.	0.8

Tableau 3.4 Variation du burnup moyen de sortie du cœur (MWj/t) optimisé pour un réacteur CANDU-6 avec et sans barres de compensation.

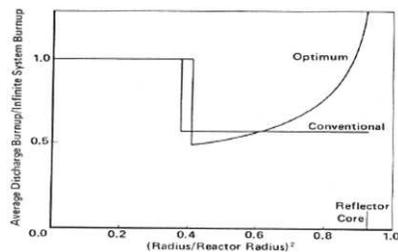


Figure 3.9 Distribution radiale du burnup moyen de sortie obtenue par Wight.

CHAPITRE 4

IMPLANTATION DE MÉTHODES DE GRADIENTS ALTERNATIVES

4.1 Introduction

Dans le chapitre précédent, nous avons présenté et vérifié l'algorithme standard d'optimisation de la gestion du combustion à l'équilibre du rechargement. Nous avons cependant implicitement omis un point important du processus d'optimisation à savoir l'obtention du point de départ de l'algorithme du simplexe. En effet, pour la méthode des gradients classique basée sur la résolution par le simplexe, nous avons noté à plusieurs reprises la nécessité de toujours avoir un point réalisable. Certains ajustements ont d'ailleurs été faits pour revenir à un tel point, au cas où la linéarisation devienne trop approximative. Cependant, les techniques utilisées supposent qu'au moins un point réalisable a déjà été trouvé. Donc la configuration initiale du réacteur, qui sert de point de départ, doit respecter toutes les contraintes, c'est-à-dire correspondre à un point réalisable. Pour un type de réacteur en service depuis longtemps, plusieurs données sont disponibles, mais cela s'avère problématique pour un nouveau design de réacteur. De plus, différentes contraintes peuvent s'ajouter pour lesquelles peu d'expérience est disponible.

La méthode de programmation quasi-linéaire classique, telle que décrite au chapitre précédent, présente donc la difficulté d'exiger un point de départ réalisable. Deux méthodes alternatives sont proposées pour résoudre ce problème et ainsi trouver un point réalisable au problème d'optimisation (2.18). La première utilisera l'algorithme général présenté au chapitre précédent, mais appliqué à différentes

fonctions objectif. La deuxième procédure utilise les méthodes duales. Nous présenterons dans ce chapitre la théorie et l'implantation de ces deux méthodes. Ensuite, nous comparerons l'efficacité des deux approches sur un exemple d'optimisation de gestion du combustible d'un CANDU-6. Enfin, une nouvelle approche tirant les avantages de ces deux méthodes sera présentée.

4.2 Méthode multi-étapes

Pour décrire et tester l'implantation de la recherche d'un point réalisable, nous avons repris l'exemple d'un réacteur CANDU-6 avec les barres liquides mais sans barres de compensation. Les seules variables de décision sont les burnups moyens de sortie de 2 zones de combustion. Les contraintes à satisfaire pour avoir un point réalisable sont un réacteur critique et des puissances canal qui ne dépassent pas la limite requise, éq.(2.11) et 2.15) respectivement. Pour atteindre cet objectif, nous procéderons en deux étapes: premièrement trouver une distribution de burnup de sortie pour que le réacteur soit critique et ensuite minimiser la puissance canal en imposant simultanément une contrainte sur la réactivité. Cette approche sera appelée par la suite l'optimisation multi-étapes (MS).

4.2.1 Contrainte sur la réactivité

Afin de satisfaire la contrainte sur la réactivité, un nouveau problème d'optimisation est défini. Le but est de minimiser l'écart entre la valeur recherchée du coefficient de multiplication des neutrons (k_{ref}) et sa valeur actuelle (k_{eff}) sans aucune contrainte. Le nouveau problème d'optimisation est alors défini par:

$$\min_{\{B_{e,i}\}} (k_{eff} - k_{ref})^2 \quad (4.1)$$

où $\{B_{e,i}\}$ représente une distribution de burnup moyen de sortie.

Cette fonction objectif a été implantée dans les modules d'optimisation (F**OBJECT**: et G**PTGRD**: pour sa dérivée) du code DONJON.

Si le nombre de zones de combustion choisie est plus grand que 1, alors le nombre de solutions possibles est infini. En effet, les solutions correspondent au sous-ensemble défini par $k_{eff} = k_{ref}$. Pour assurer l'unicité de la solution, des contraintes sur le ratio entre les burnups moyen de sortie doivent être définis. Afin de vérifier la convergence de notre implantation du problème (4.1), nous lui avons rajouté des contraintes analytiques qui définissent le ratio entre les burnup de sortie. Cette technique était utilisée précédemment pour ramener le problème de criticité du réacteur de plusieurs variables à une seule. La convergence se faisait alors en utilisant une méthode de Brent pour trouver le zéro de la fonction $k_{eff} - k_{ref}$ entre deux bornes (un point tel que $k_{eff} > k_{ref}$ et un autre tel que $k_{eff} < k_{ref}$).

Les résultats obtenus avec la méthode de Brent et notre méthode de programmation quasi-linéaire sont présentés au tableau 4.1. Nous avons également donné les points de départ utilisés pour les deux algorithmes ainsi que la valeur de k_{eff} correspondante à ces configurations. Le rapport choisi pour les burnups moyens de sortie est de 1. Les résultats présentés au tableau 4.1 démontrent bien l'équivalence entre les méthodes de Brent et la méthode d'optimisation développée lorsqu'utilisée avec des contraintes. La différence négligeable obtenue sur le burnup moyen de sortie peut être attribuée essentiellement à la tolérance sur la précision du flux neutronique. De plus, les deux algorithmes d'optimisation présentent également des critères de convergence différents: la variation du burnup moyen de sortie en valeur absolue inférieure à une faible tolérance pour la méthode de Brent, ou la variation relative pour notre approche.

Tableau 4.1 Burnup moyen de sortie pour que $k_{eff} = k_{ref}$

Méthode	B_1^e	B_2^e	k_{eff}
Brent	28480.14	28480.14	0.9999997
<i>Borne supérieure</i>	19000.	19000.	1.070030
<i>Borne inférieure</i>	35000.	35000.	0.9549444
<i>Point initial</i>	19000.	19000.	1.070030
Optimisation	28474.08	28474.08	1.000012
Optimisation sans contrainte	27051.19	29522.72	0.9999810

Nous avons également effectué le calcul pour le cas où aucune contrainte de ratio n'est fixée (dernière ligne du tableau 4.1). Nous pouvons alors observer que B_1^e et B_2^e sont différents, ce qui illustre le fait que le nombre de solutions n'est unique qu'à la condition de contraindre les valeurs respectives des burnups de sortie par un ratio.

4.2.2 Contraintes de puissance maximum

Une fois que la contrainte sur la réactivité est respectée, la dernière étape est de respecter les contraintes sur la puissance maximum pour obtenir un point réalisable au problème d'optimisation (2.18) défini au chapitre précédent.

Le moyen envisagé pour garantir que les contraintes sur la puissance canal soient respectées, est de minimiser la puissance canal maximum P_{Cmax} et de la comparer à la valeur de la contrainte P_{lim} . Nous aurions pu aussi minimiser l'écart entre P_{Cmax} et P_{lim} . Un nouveau problème d'optimisation est alors défini comme suit:

$$\min_{\{B_{e,i}\}} P_{Cmax} = \min_{\{B_{e,i}\}} \|P_j, j \in [1, n_{cha}]\|_{\infty} \quad (4.2)$$

$$\text{avec } k_{eff} = k_{ref} \quad \text{avec } k_{eff} = k_{ref}$$

Si l'optimum sur P_{Cmax} est inférieur à la contrainte sur la puissance maximum de canal, alors le problème d'optimisation original (2.18) a une solution et cet optimum peut lui servir de point initial.

Cependant la norme infinie, utilisé dans (4.2), est non dérivable. Mathématiquement toutes les normes sont équivalentes dans des espaces de dimension finie [29]. De plus, par définition la norme infinie est:

$$\|P_j, j \in [1, n_{cha}]\|_{\infty} = \lim_{N \rightarrow \infty} \sqrt[2N]{\sum_j P_j^{2N}} \quad (4.3)$$

La norme $2N$ est quant à elle dérivable par rapport à B_e et pourrait être utilisée comme fonction objectif dans (4.2) à la place de la norme infinie. Cependant avant d'implanter cette fonction objectif, nous avons étudié le minimum obtenu en utilisant différentes normes de types $2N$, c'est-à-dire différentes valeurs de N . Le cas choisi correspond à un réacteur CANDU-6 critique sans barres de compensation avec 2 zones de combustion radiales. La figure 4.1 représente la variation de la puissance canal maximum en fonction du ratio des deux burnups moyen de sortie, pour lesquels nous avons réalisé un grand nombre de calculs de réacteur. Le minimum de la puissance canal maximum est atteint pour un ratio B_1^e/B_2^e d'environ 1.77, ce qui signifie que, quelque soit la contrainte P_{lim} dans (2.13), la valeur minimale possible est de $6527kW$. Nos cherchons donc une approximation de la norme infinie, donc de N qui nous permette de trouver la puissance minimale. Pour chaque distribution de puissance obtenue pour tracer la figure 4.1, nous avons calculé la norme $2N$ selon N , donnée par F_1^P .

$$F_1^P = \|P_j\|_{2N} = \sqrt[2N]{\sum_j P_j^{2N}} \quad (4.4)$$

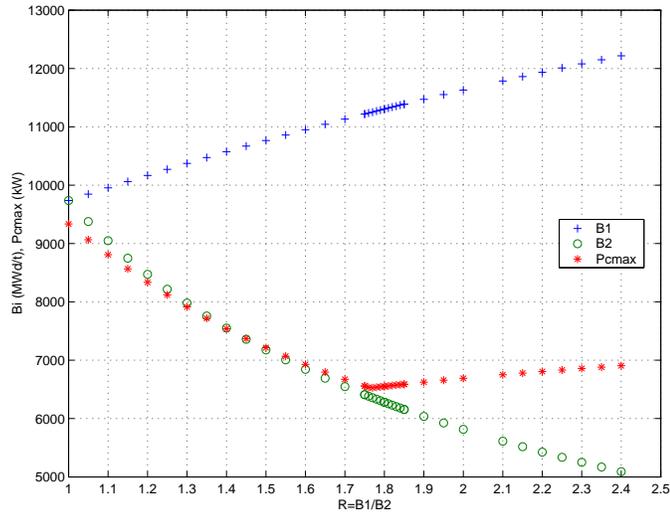


Figure 4.1 Puissance canal maximum en fonction du rapport des burnups moyens de sortie

Les résultats de cette première méthode sont présentés à la figure 4.2. Quand N augmente, le minimum mathématique se rapproche du minimum physique. Cependant, même avec une norme 30 ($N = 15$) le minimum mathématique est loin de l'optimum. Nous ne pouvons cependant pas augmenter N indéfiniment car la valeur maximale d'un réel en représentation 4 octets ($3.402823E+38$) peut être atteinte même si les puissances canal sont normalisées.

Nous avons alors testé trois autres types de fonctionnelles similaires. La première modification part du fait que le minimum pour la puissance canal maximum est obtenu quand le flux est le plus plat possible. Ainsi nous avons calculé la norme $2N$ de la différence entre la distribution de puissance et une puissance uniforme égale à P_{moy} . La deuxième méthode consiste donc à minimiser:

$$F_2^P = \|P_j - P_{moy}\|_{2N} = \sqrt[2N]{\sum_j (P_j - P_{moy})^{2N}} \quad (4.5)$$

Les mêmes calculs de norme sont présentés à la figure 4.3. Même si les résultats

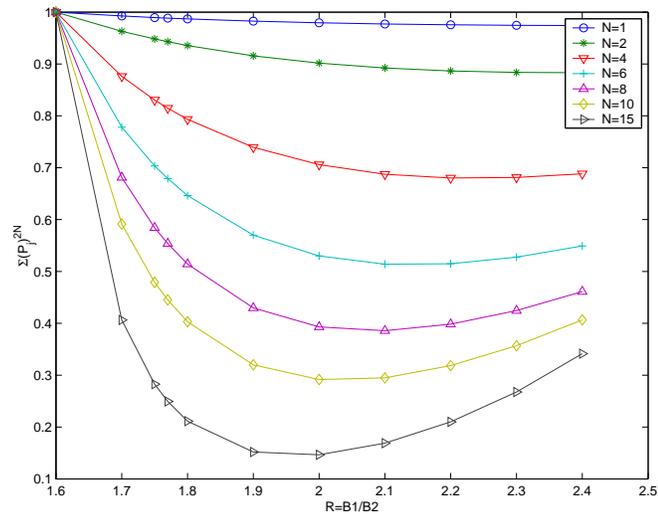


Figure 4.2 Norme $2N$ de la distribution de puissance canal en fonction du ratio des burnups moyen de sortie, méthode 1

s'améliorent quand N augmente, le minimum mathématique n'est jamais atteint même pour un ratio de 2.4. Physiquement, ce comportement s'explique par le fait que de nombreux canaux en périphérie ont une puissance canal largement inférieure à la moyenne. Le minimum sera donc atteint pour une distribution de burnup moyen de sortie faible en périphérie (ratio grand) afin d'augmenter la puissance canal dans les canaux correspondants. Cette méthode soulève donc le problème des canaux à faible puissance. Elle serait peut être appropriée dans le cas d'un réacteur où le flux est plat radialement dans le cœur, mais ce n'est pas le cas pour les CANDU-6.

Afin d'avoir une méthode plus générale, nous avons envisagé deux moyens de ne pas tenir compte des canaux à faible puissance. Une troisième alternative à la norme infinie du problème (4.2) consiste à ne faire que la somme sur les canaux où

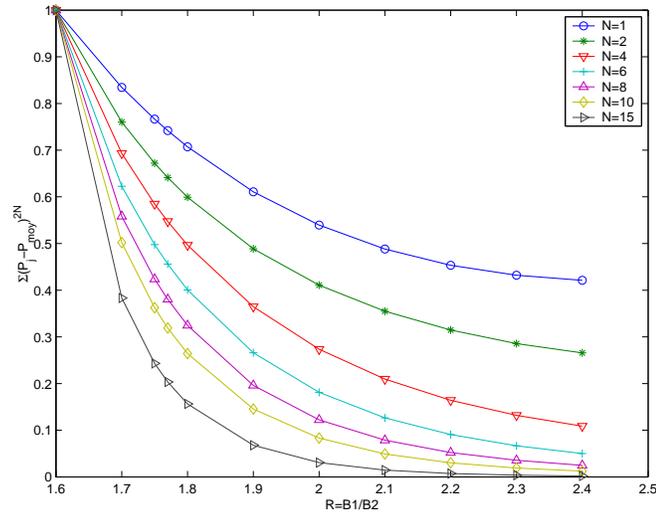


Figure 4.3 Norme $2N$ de la distribution de puissance canal en fonction du ratio des burnups moyen de sortie, méthode 2

la puissance canal est supérieure à la moyenne. La norme est alors donnée par:

$$F_3^P = \|\max\{0, P_j - P_{moy}\}\|_{2N} = \sqrt[2N]{\sum_j (\max\{0, P_j - P_{moy}\})^{2N}} \quad (4.6)$$

Comme le montre la figure 4.4, le minimum mathématique trouvé s'approche rapidement de l'optimum physique. En effet, pour $N = 8$ le minimum est atteint pour un ratio d'environ 1.8 ce qui correspond physiquement à une puissance canal de 6551. au lieu de 6527. pour un ratio optimum de 1.77.

Enfin, la dernière variante proposée pour la fonction objectif de (4.2) consiste à ne faire la somme que sur la zone où le $CPPF$ est calculé (voir chapitre 2). Cela revient donc également à éliminer dans le calcul de la norme les canaux de faible puissance, comme le montre l'équation suivante:

$$F_4^P = \|P_j - P_{moy}, j \in Z_{CPPF}\|_{2N} = \sqrt[2N]{\sum_{j \in Z_{CPPF}} (P_j - P_{moy})^{2N}} \quad (4.7)$$

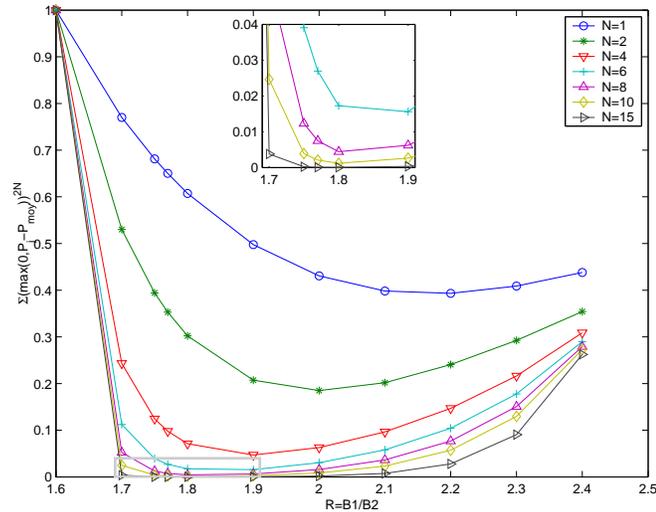


Figure 4.4 Norme $2N$ de la distribution de puissance canal en fonction du ratio des burnups moyen de sortie, méthode 3

où Z_{CPPF} représente la zone où le $CPPF$ est calculé.

Pour notre exemple, tous les canaux sauf une rangée de deux de largeur en périphérie font partie de Z_{CPPF} . Les résultats présentés à la figure 4.5 démontrent un bon comportement de cette dernière méthode. Elle présente cependant intrinsèquement un désavantage par rapport à la méthode précédente du fait qu'il faut avoir une connaissance à priori de la forme du flux pour estimer Z_{CPPF} .

Nous avons testé la troisième méthode pour trouver un point initial au problème général d'optimisation (2.18). Une modification a été apportée à la fonction (4.6). Étant donné qu'un calcul automatique des contraintes q_j était déjà fait, nous les avons utilisées à la place de la puissance canal en prenant $CPPF_j = 1$. Le problème (4.2) devient donc:

$$\min_{\{B_{e,i}\}} \sqrt[2N]{\sum_j (q_j - q_{moy})^{2N}} \quad (4.8)$$

avec $k_{eff} = k_{ref}$ avec $k_{eff} = k_{ref}$

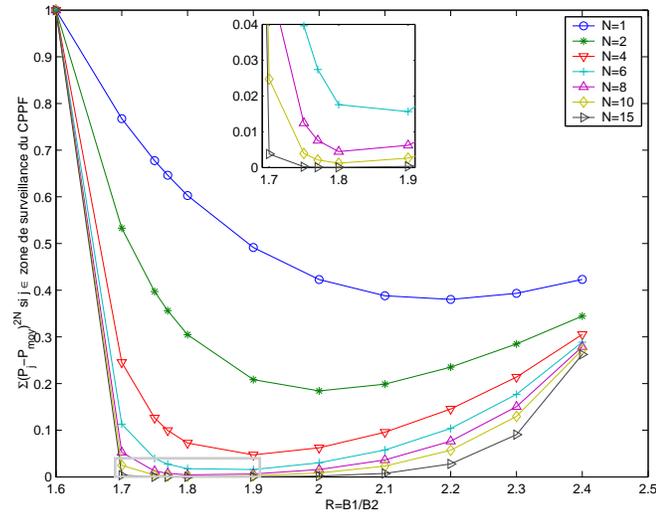


Figure 4.5 Norme $2N$ de la distribution de puissance canal en fonction du ratio des burnups moyen de sortie, méthode 4

Cet échange a aussi pour avantage de traiter avec des nombres de l'ordre de l'unité plutôt que du millier. L'exposant N de la norme peut alors être pris plus grand si nécessaire.

Nous avons donc fait des calculs d'optimisation en utilisant pour différentes valeurs de N la fonction objectif (4.8) et la contrainte $k_{eff} = k_{ref}$. Les résultats sont donnés au tableau 4.2. Les valeurs proches de celles recherchées (dernière ligne du tableau) sont atteintes pour une valeur de $N > 8$, idéalement 20.

4.2.3 Conclusions

L'approche en deux étapes donne une méthode générale robuste pour trouver un point initial réalisable. Les résultats ont aussi permis de valider l'algorithme d'optimisation défini à la figure 3.1 pour des fonctions objectifs différentes de la fonction coût du combustible.

Tableau 4.2 Minimum de la puissance canal maximum obtenue par la méthode 3

N	B_1^e	B_2^e	ratio	P_{Cmax}
1	11863.65	5489.39	2.1612	6781.
2	11608.74	5812.69	1.9971	6684.
4	11442.06	6043.89	1.8932	6615.
6	11371.09	6146.21	1.8501	6585.
8	11336.55	6198.44	1.8289	6561.
10	11308.00	6238.70	1.8126	6557.
15	11277.73	6293.91	1.7918	6541.
20	11266.12	6304.36	1.7870	6530.
30	11247.73	6330.13	1.7769	6530.
∞	11252.05	6357.09	~ 1.77	6527.

Dans le cas du ACR-700, une contrainte supplémentaire sur la réactivité du vide est présente. La même méthode peut alors être envisagée. Une troisième sous-étape peut être programmée pour déplacer le point trouvé après la minimisation de la puissance canal dans le domaine réalisable où la réactivité du vide est en dessous de la limite prescrite.

Enfin, une recommandation doit être faite quant à la deuxième sous étape. Trouver le minimum de la puissance canal maximum ne va pas forcément dans le même sens que l'optimisation du coût en combustible. Aussi, il n'est pas forcément judicieux de pousser l'optimisation jusqu'à la convergence. En effet le but final de résoudre (4.8) est de trouver un point réalisable au problème (2.18). Donc dès que la valeur de la puissance canal maximum est en dessous de la limite définie par l'équation (2.15), la deuxième sous étape peut être arrêtée. Nous avons cependant tenu à pousser la minimisation de la puissance canal jusqu'au bout dans l'exemple étudié précédemment pour valider la méthode ainsi que le choix de la fonction objectif (éq. (4.8)).

4.3 Méthodes duales

Le désavantage de l'approche présentée dans la section précédente est de devoir résoudre plusieurs problèmes d'optimisation. Ceci nécessite plusieurs manipulations de la part de l'utilisateur. Les méthodes duales nous permettent de résoudre le problème d'optimisation de la gestion du combustible (1.6) directement. Nous décrirons dans un premier temps le principe de ces méthodes ainsi que la définition mathématique du problème d'optimisation correspondant. L'algorithme commun utilisé pour les deux méthodes proposées sera ensuite décrit plus en détail. Une comparaison de l'efficacité des différentes méthodes (duales et multi-étapes) sera donnée à la section suivante.

4.3.1 Théorie

Le principe général des méthodes duales [30,31] est de se départir des contraintes du problème d'optimisation original pour ainsi obtenir un problème sans contrainte. La méthode utilisée consiste à tenir compte dans la nouvelle fonction objectif non seulement de la fonction objectif d'origine mais aussi des contraintes. Plus précisément, l'écart entre la valeur des contraintes et leur limite respective est retenue dans le nouveau problème d'optimisation. À chaque méthode correspond une manière de tenir compte de ces écarts. Nous avons implanté deux de ces méthodes, à savoir le lagrangien augmenté et une pénalisation extérieure. La formulation analytique de ces fonctions objectifs dépendant non seulement des fonctionnelles du problème d'optimisation original mais aussi du choix de l'influence des contraintes.

Le lagrangien augmenté (L_a) du problème d'optimisation (2.18) est alors donné

par:

$$\begin{aligned}
L_a(\vec{X}, \nu, \mu, \eta, \gamma) &= f(\vec{X}) + \sum_i \nu_i (h_i(\vec{X}) - b_i) \\
&+ \sum_j \mu_j (g_j(\vec{X}) - c_j + \max\{0, -g_j(\vec{X}) + c_j - \frac{\mu_j}{\gamma_j}\}) \\
&+ \sum_i \frac{\eta_i}{2} (h_i(\vec{X}) - b_i)^2 \\
&+ \sum_j \frac{1}{2\gamma_j} (\max^2\{0, \gamma_j(g_j(\vec{X}) - c_j) + \mu_j\} - \mu_j^2) \quad (4.9)
\end{aligned}$$

où ν et μ sont les coefficients de Lagrange; η et γ sont les poids des contraintes.

La fonction de pénalité (**P**) du problème d'optimisation (2.18) est quant à elle donnée par:

$$\begin{aligned}
P(\vec{X}, \eta, \gamma) &= f(\vec{X}) + \sum_i \frac{\eta_i}{2} (h_i(\vec{X}) - b_i)^2 \\
&+ \sum_j \frac{\gamma_j}{2} (\max\{0, g_j(\vec{X}) - c_j\})^2 \quad (4.10)
\end{aligned}$$

Avec ces méthodes, n'importe quelle valeur du vecteur de décision qui satisfait les bornes sur les variables de décision peut être un point initial puisque le nouveau problème est sans contrainte. Ceci présente un très grand avantage quand le domaine réalisable peut difficilement être atteint.

L'algorithme général de ces deux méthodes est identique. La résolution se fait par itération. Premièrement, le lagrangien augmenté ou la fonction pénalité sont évalués avec un estimé des coefficients qui pondèrent l'influence des contraintes dans la nouvelle fonction objectif. Le problème d'optimisation sans contrainte est alors résolu. Pour cela, plusieurs itérations internes sont requises car la nouvelle fonction objectif est aussi non linéaire. Une linéarisation avec la théorie des perturbations généralisées est alors effectuée. Après convergence interne sur le minimum

du lagrangien augmenté ou de la fonction pénalité, les coefficients qui pondèrent les contraintes sont réévalués pour l'itération suivante. Le processus est répété jusqu'à convergence externe du minimum du lagrangien augmenté ou de la fonction pénalité.

L'algorithme détaillé de la méthode implantée est donné pour le lagrangien augmenté dans la section suivante. La dérivation des équations linéarisées nécessaires est également présentée dans cette section. Pour la méthode de pénalisation, seules les formules nécessaires à la résolution seront présentées puisque l'algorithme est le même que pour le lagrangien augmenté.

4.3.2 Algorithme implanté

La méthode du lagrangien augmenté (LA) est utilisée pour la description de l'algorithme. Celle-ci est représenté à la figure 4.6. La première colonne correspond à l'algorithme général, soit les itérations externes. La séquence de calcul peut être résumée par les points suivants:

- Les facteurs de Lagrange sont initialisés à 0 ($\nu^0 = \mu^0 = 0$). Ce choix permet de ne pas tenir compte dans le lagrangien augmenté des contraintes tant qu'elles sont respectées. Le poids initial η^0 et γ^0 de chaque contrainte doit être fixé judicieusement pour que l'influence relative de chaque contrainte soit égale.
- 1) À chaque itération k , le problème d'optimisation sans contrainte suivant est résolu: $\vec{X}^{k+1} = \min L_a(\vec{X}, \nu^k, \mu^k, \eta^k, \gamma^k)$

- 2) Ensuite s'il y a convergence de la solution \vec{X}^{k+1} , l'algorithme s'arrête, sinon il continue à l'étape suivante.
- 3) Avec la nouvelle valeur du vecteur de décision les contraintes sont réévaluées, ce qui nous permet de mettre à jour

(a) les coefficients de lagrange :

$$\nu_i^{k+1} = \nu_i^k + \eta_i^k (h_i(\vec{X}^k) - b_i) \text{ et } \mu_j^{k+1} = \max\{0, \mu_j^k + \gamma_j^k (g_j(\vec{X}^k) - c_j)\}$$

(b) et les poids des contraintes :

$$\begin{cases} \eta_i^{k+1} = \alpha \eta_i^k \text{ si } h_i(\vec{X}^{k+1}) - c_i \neq 0, \text{ sinon } \eta_i^{k+1} = \max\{\eta_i^k / \alpha, \eta_i^{min}\} \\ \gamma_j^{k+1} = \alpha \gamma_j^k \text{ si } g_j(\vec{X}^{k+1}) - c_j > 0, \text{ sinon } \gamma_j^{k+1} = \max\{\gamma_j^k / \alpha, \gamma_j^{min}\} \end{cases}$$

α est le facteur de multiplication pour le poids des contraintes.

Retour à l'étape 1 ($k = k + 1$).

En théorie, les poids des contraintes peuvent être limités à une certaine valeur maximum, ce qui permet de garder une certaine stabilité numérique de la méthode du LA [30]. Cependant, aucune méthode analytique ne permet de calculer cette valeur sans devoir calculer des dérivées secondes des fonctionnelles. La valeur du poids des contraintes n'est donc pas bornée avec l'algorithme tel qu'implanté.

Delbos [31] insiste sur le fait que le poids des contraintes utilisé (η_i et γ_j) aux étapes 1 et 2a doit être le même, si les propriétés de convergence du LA veulent être conservées. Dans l'approche présentée par Gauvin [30], le poids des contraintes est systématiquement augmenté à chaque itération externe par un même facteur. L'approche que nous avons suivi permet d'adapter le poids des contraintes individuellement comme le permet l'algorithme de Delbois. Le choix de diminuer le poids des contraintes a pour but de relaxer les contraintes qui sont satisfaites. Nous avons cependant préféré que le facteur de diminution du poids des contraintes ($(\alpha + 1)/2$) soit plus petit que celui d'augmentation (α). Le but de cette manœuvre est qu'en moyenne un contrainte qui oscillerait entre un état violé et respecté verrait son

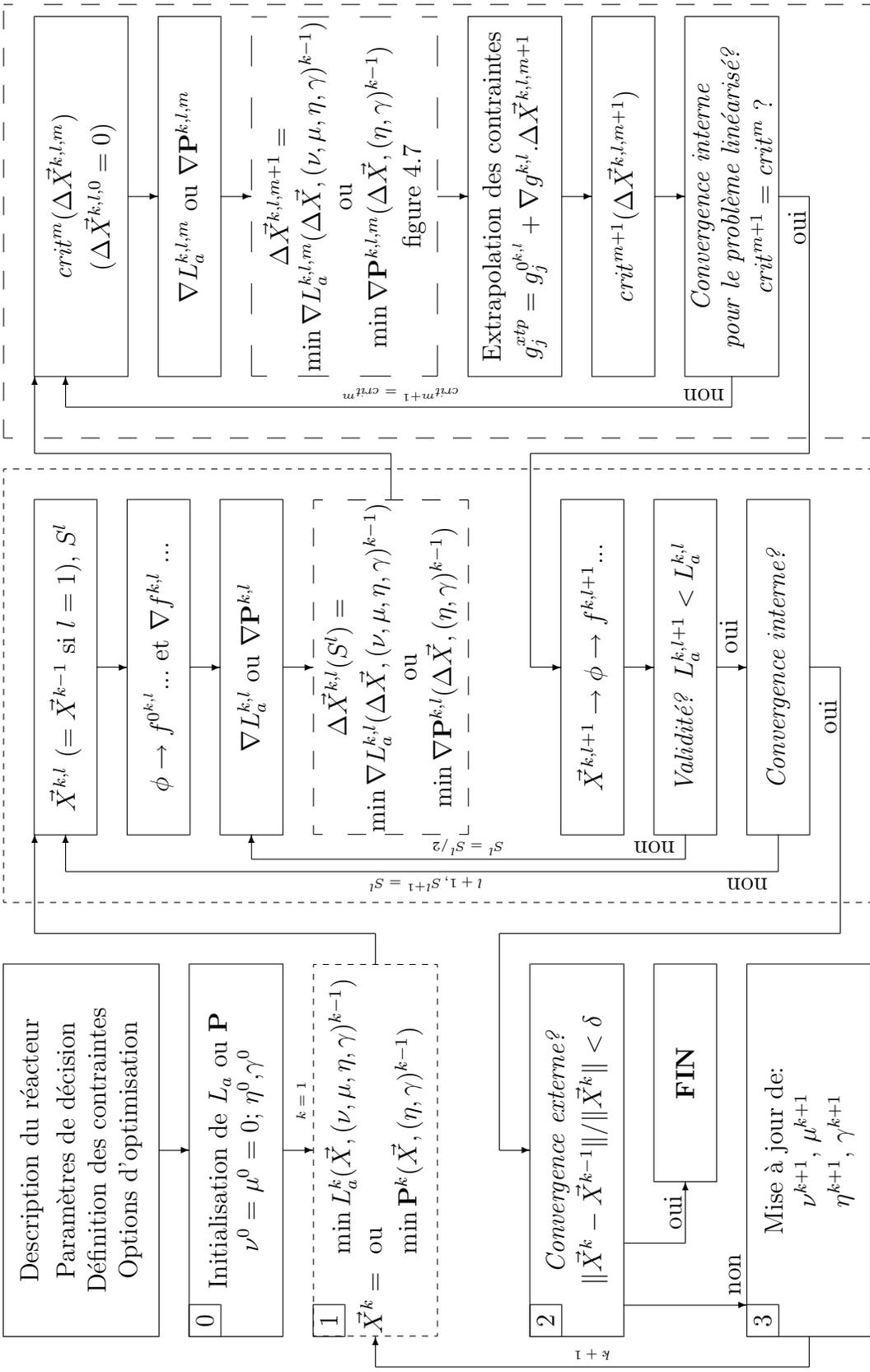


Figure 4.6 Algorithme général pour l'optimisation avec la méthode du lagrangien augmenté ou de pénalisation.

poids augmenter.

Pour résoudre le problème sans contrainte donné à l'étape 1, une linéarisation du LA en \vec{X}^k doit être faite. Nous emploierons les notations suivantes:

$$\begin{aligned} f(\vec{X}) &= f(\vec{X}^k) + \nabla \vec{f} \cdot \Delta \vec{X} = f^0 + \nabla \vec{f} \cdot \Delta \vec{X} \\ h_i(\vec{X}) &= h_i(\vec{X}^k) + \nabla \vec{h}_i \cdot \Delta \vec{X} = h_i^0 + \nabla \vec{h}_i \cdot \Delta \vec{X} \\ g_j(\vec{X}) &= g_j(\vec{X}^k) + \nabla \vec{g}_j \cdot \Delta \vec{X} = g_j^0 + \nabla \vec{g}_j \cdot \Delta \vec{X} \end{aligned}$$

Les indices d'itération externe des multiplicateurs de Lagrange et des poids des contraintes ont été enlevés pour alléger la notation. De plus ces coefficients sont constants lors de l'étape 1.

Avec ces notations, l'équation (4.9) devient:

$$\begin{aligned} L_a(\vec{X}, \nu, \mu, \eta, \gamma) &= f^0 + \nabla \vec{f} \cdot \Delta \vec{X} + \sum_i \nu_i (h_i^0 + \nabla \vec{h}_i \cdot \Delta \vec{X} - b_i) \\ &\quad + \sum_j \mu_j (g_j^0 + \nabla \vec{g}_j \cdot \Delta \vec{X} - c_j \\ &\quad \quad \quad + \max\{0, -g_j^0 - \nabla \vec{g}_j \cdot \Delta \vec{X} + c_j - \frac{\mu_j}{\gamma_j}\}) \\ &\quad + \sum_i \frac{\eta_i}{2} (h_i^0 + \nabla \vec{h}_i \cdot \Delta \vec{X} - b_i)^2 \\ &\quad + \sum_j \frac{1}{2\gamma_j} (\max^2\{0, \gamma_j (g_j^0 + \nabla \vec{g}_j \cdot \Delta \vec{X} - c_j) \\ &\quad \quad \quad + \mu_j\} - \mu_j^2) \end{aligned} \tag{4.11}$$

En ne gardant que les termes de premier ordre et en enlevant les constantes, nous obtenons le problème linéarisé ∇L_a suivant:

$$\begin{aligned} \nabla L_a(\Delta \vec{X}, \nu, \mu, \eta, \gamma) &= \nabla \vec{f} \cdot \Delta \vec{X} + \sum_i \nu_i \nabla \vec{h}_i \cdot \Delta \vec{X} \\ &\quad + \sum_i \eta_i (h_i^0 - b_i) (\nabla \vec{h}_i \cdot \Delta \vec{X}) \\ &\quad + \sum_j \frac{1}{\gamma_j} (\max\{0, \gamma_j (g_j^0 + \nabla \vec{g}_j \cdot \Delta \vec{X} - c_j) \\ &\quad \quad \quad + \mu_j\} - \mu_j^2) \nabla \vec{g}_j \cdot \Delta \vec{X} \end{aligned}$$

$$+ \sum_j \left\{ \begin{array}{l} ((\gamma_j(g_j^0 - c_j) + 2\mu_j)(\nabla \vec{g}_j \cdot \Delta \vec{X})) \\ \quad \text{si } \gamma_j(g_j^0 + \nabla \vec{g}_j \cdot \Delta \vec{X} - c_j) + \mu_j > 0 \\ 0 \quad \text{si } \gamma_j(g_j^0 + \nabla \vec{g}_j \cdot \Delta \vec{X} - c_j) + \mu_j \leq 0 \end{array} \right. \quad (4.12)$$

La somme des constantes correspond à la valeur du LA pour $\Delta \vec{X} = 0$, et est donnée par:

$$L_a^0(\vec{X}^k, \nu, \mu, \eta, \gamma) = f^0 + \sum_i \nu_i (h_i^0 - b_i) + \sum_i \frac{\eta_i}{2} (h_i^0 - b_i)^2 + \sum_j \left\{ \begin{array}{l} 2\mu_j (g_j^0 - c_j) + \frac{\gamma_j}{2} (g_j^0 - c_j)^2 \\ \quad \text{si } \gamma_j (g_j^0 + \nabla \vec{g}_j \cdot \Delta \vec{X} - c_j) + \mu_j > 0 \\ \frac{-3\mu_j^2}{2\gamma_j} \\ \quad \text{si } \gamma_j (g_j^0 + \nabla \vec{g}_j \cdot \Delta \vec{X} - c_j) + \mu_j \leq 0 \end{array} \right. \quad (4.13)$$

Plusieurs itérations internes l sont nécessaires pour la minimisation de ∇L_a . Le schéma de calcul correspondant est présenté à la figure 4.6 en colonne 2. Afin que la linéarisation reste valide à chacune des itérations internes, une contrainte quadratique est ajoutée. Ainsi, pour chaque l , la première étape consiste à calculer le flux ϕ correspondant au vecteur de décision actuel $\vec{X}^{k,l}$. Cela nous permet de calculer la valeur des fonctionnelles f^0 du problème d'origine (2.18) ainsi que leur gradient ∇f (2.22). Le problème linéarisé du LA est évalué puis résolu. Nous obtenons ainsi une nouvelle valeur du vecteur de décision pour laquelle nous réévaluons le flux et donc la valeur du LA. Si ce nouveau point n'est pas valide, l'itération l est reprise en diminuant la contrainte quadratique. Dans le cas contraire, une nouvelle itération est commencée si la convergence interne n'a pas été obtenue.

La résolution du problème du LA linéarisé mérite une attention particulière. En effet, pour évaluer le problème d'optimisation (4.12), nous devons connaître le signe de $\gamma_j(g_j^0 + \nabla \vec{g}_j \cdot \Delta \vec{X} - c_j) + \mu_j$, que nous appellerons critères *crit*. Or $\Delta \vec{X}$ est notre inconnue. Le problème est donc linéaire par morceaux seulement. Un algorithme tel que la méthode de Lemke de programmation quasi-linéaire ne peut être directement utilisé puisque les coefficients du problème d'optimisation peuvent varier dans le domaine réalisable (à l'intérieur de la contrainte quadratique). Nous proposons la méthode itérative suivante pour résoudre le problème (4.12). Pour une itération m , nous calculons le signe de *crit* ^{m} pour un estimé de $\Delta \vec{X}^{\vec{k},l,m}$, et supposons que celui-ci ne change pas autour de $\vec{X}^{\vec{k},l}$ dans l'hypersphère de rayon S^l . Ceci nous permet de calculer le pas d'avance du vecteur de décision $\Delta \vec{X}^{\vec{k},l,m+1}$. Les contraintes d'inégalité sont alors extrapolées en utilisant les gradients et le nouveau pas d'avance. Le signe des critères est alors réévalué *crit* ^{$m+1$} . Si les critères restent inchangés, la convergence sur les itérations m est atteinte, sinon le LA linéarisé est réévalué avec les critères *crit* ^{$m+1$} . Cette approche est basée sur le fait que la valeur du problème linéarisé (4.12) ne devrait pas changer très proche de $\Delta \vec{X}^{\vec{k},l}$. Pour contrecarrer des instabilités possibles, un nombre d'itération m maximum est imposé.

Pour la minimisation du LA à une itération m donnée, deux cas de figures sont possibles. La figure 4.7 représente ces deux cas pour une optimisation à deux variables. Le cercle représente la contrainte quadratique dans les deux cas. La solution $\Delta \vec{X}^{\vec{k},l,m+1}$ sera donc sur le cercle ou à l'intérieur, elle correspond au point noir sur la figure. L'opposé du gradient de la fonction objectif est donné par le vecteur $-u$. Le trait pointillé est orthogonal au vecteur $-u$, il représente une courbe de niveau de u étant donné que la fonction u est linéaire. Dans le premier cas de figure (4.7 cas a), la contrainte quadratique est assez grande pour qu'une variable de décision au moins atteigne son maximum ou minimum permis. La méthode de

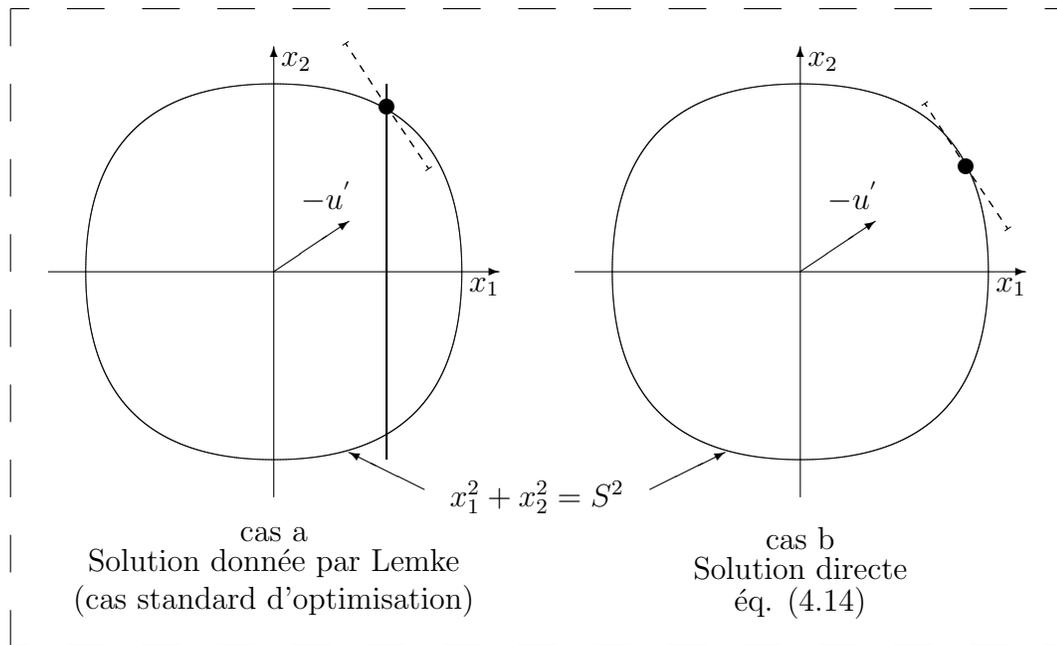


Figure 4.7 Calcul du pas d'avance avec la méthode du lagrangien augmenté ou de pénalisation.

Lemke de programmation mathématique est alors utilisée pour résoudre le problème quadratique. Dans le deuxième cas (4.7 cas b), aucune limite de X_i ne peut être atteinte. La solution est directement donnée par l'intersection de l'hypersphère avec le gradient de ∇L_a , ce nous donne analytiquement l'équation suivante:

$$X_i = -S \frac{\nabla_i u / \omega_i}{\sqrt{\sum \nabla_i u^2 / \omega_i}} \quad (4.14)$$

démonstration:

Le problème à résoudre est défini par:

$$\min u = \min \nabla u \cdot \Delta \vec{X} \begin{cases} \sum \omega_i \Delta X_i^2 \leq S^2 \\ X_{i,inf} - X_i^k \leq X_i \leq X_{i,sup} - X_i^k \end{cases} \quad (4.15)$$

où u correspond à L_a ou \mathbf{P} .

Dans le cas a de la figure 4.7, la solution est à l'intersection de l'hypersphère et de la droite donnée par le gradient de u . Or, étant donné que toutes les variables de décision n'ont pas le même poids, la contrainte quadratique s'écrit:

$$(\|\vec{X}\|_{2^P})^2 = \sum \omega_i \Delta X_i^2 = S^2 = \sum x_i^2 = (\|\vec{x}\|_2)^2 \quad \text{avec } x_i = \sqrt{\omega_i} X_i \quad (4.16)$$

Dans le repère orthonormé (x_i) , la fonction objectif est alors donnée par:

$$u = \sum \nabla u \cdot \Delta \vec{X} = \sum \frac{\nabla_i u}{\sqrt{\omega_i}} x_i = \vec{u}' \cdot \vec{x} \quad \text{avec } u'_i = \frac{\nabla_i u}{\sqrt{\omega_i}} \quad (4.17)$$

La solution du problème est donc donnée par:

$$x_i = -S \frac{u'_i}{\|\vec{u}'\|_2} \quad (4.18)$$

En remplaçant x_i et u'_i par leur expression (4.16) et (4.17) respectivement dans (4.18), nous obtenons l'équation (4.14).

Pénalisation extérieure

L'algorithme général de la méthode de pénalisation externe (MPE) est en fait le même que celui du LA, seule la fonction objectif change. Cette dernière est donnée

par la fonction objectif plus l'écart des contraintes au carré fois un poids associé à chacune. Dans le cas de la MPE, l'algorithme de résolution peut se résumer comme suit:

- Initialisation du poids des contraintes η^0 et γ^0
- 1) À chaque itération k , le problème d'optimisation sans contrainte suivant est résolu: $\vec{X}^{k+1} = \min_{\vec{X}} \mathbf{P}(\vec{X}, \eta^k, \gamma^k)$
- 2) Ensuite s'il y a convergence de la solution \vec{X}^{k+1} , l'algorithme s'arrête, sinon il continue à l'étape suivante.
- 3) Avec la nouvelle valeur du vecteur de décision les contraintes sont réévaluées, ce qui nous permet de mettre à jour leur poids :

$$\begin{cases} \eta_i^{k+1} = \alpha \eta_i^k \text{ si } h_i(\vec{X}^{k+1}) - c_i \neq 0, \text{ sinon } \eta_i^{k+1} = \max\{\eta_i^k / \alpha, \eta_i^{min}\} \\ p_i^{k+1} = \alpha p_i^k \text{ si } g_i(\vec{X}^{k+1}) - c_i > 0, \text{ sinon } p_i^{k+1} = \max\{p_i^k / \alpha, p_i^{min}\} \end{cases}$$
 Retour à l'étape 1 ($k = k + 1$).

Pour la linéarisation de \mathbf{P} , les mêmes notations que pour le LA sont utilisées, ce qui nous donne:

$$\begin{aligned} \mathbf{P}(\vec{X}, \eta, \gamma) &= f^0 + \nabla f \cdot \Delta \vec{X} + \sum_i \frac{\eta_i}{2} (h_i^0 + \nabla h_i \cdot \Delta \vec{X} - b_i)^2 \\ &\quad + \sum_j \frac{\gamma_j}{2} (\max\{0, (g_j^0 + \nabla g_j \cdot \Delta \vec{X} - c_j)\})^2 \end{aligned} \quad (4.19)$$

En ne gardant que les termes de premier ordre et en enlevant les constantes, le problème linéarisé de la méthode de pénalité $\nabla \mathbf{P}$ devient:

$$\nabla \mathbf{P}(\Delta \vec{X}, \eta, \gamma) = \nabla f \cdot \Delta \vec{X} + \sum_i \eta_i (h_i^0 - b_i) (\nabla h_i \cdot \Delta \vec{X})$$

$$+ \sum_j \begin{cases} \gamma_j (g_j^0 - c_j) (\nabla \vec{g}_j \cdot \Delta \vec{X}) \\ \quad \text{si } g_j^0 + \nabla \vec{g}_j \cdot \Delta \vec{X} - c_j > 0 \\ 0 \quad \text{si } g_j^0 + \nabla \vec{g}_j \cdot \Delta \vec{X} - c_j \leq 0 \end{cases} \quad (4.20)$$

La somme des constantes correspond à la valeur de la fonction de pénalité pour $\Delta \vec{X} = 0$, et est donnée par:

$$\mathbf{P}^0(\vec{X}^k, \eta, \gamma) = f^0 + \sum_i \frac{\eta_i}{2} (h_i^0 - b_i)^2 + \sum_j \begin{cases} \frac{\gamma_j}{2} (g_j^0 - c_j)^2 \\ \quad \text{si } g_j^0 + \nabla \vec{g}_j \cdot \Delta \vec{X} - c_j > 0 \\ 0 \quad \text{si } g_j^0 + \nabla \vec{g}_j \cdot \Delta \vec{X} - c_j \leq 0 \end{cases} \quad (4.21)$$

Le détail de la MPE est en fait le même que celui du LA, si ce n'est qu'aucun coefficient de Lagrange n'est calculé. La MPE est réputée pour être moins stable que le LA, à cause de la manière de mettre à jour le poids des contraintes. En effet, pour le LA, un maximum sur le poids des contraintes peut normalement être choisi, ce qui n'est pas le cas pour la MPE. La matrice contenant les gradients de la fonction objectif peut alors devenir mal conditionnée pour la MPE entraînant ainsi une divergence. Or dans l'implantation du LA, nous n'avons pas retenu un poids maximum des contraintes comme paramètre, car celui-ci dépend du problème, et aucune formule analytique ne permet de le calculer. Nous avons plutôt adapté le poids des contraintes individuellement, et restreint le nombre d'itérations externes pour que le poids des contraintes ne grossissent pas trop. Nous avons alors également implantée la MPE, pour d'une part la tester et comparer son efficacité par rapport au LA, et d'autre part parce que sa définition de la fonction objectif est utilisée par une autre méthode non basée sur les gradients.

4.4 Comparaison des méthodes

Afin de comparer les méthodes de pénalisation externe (MPE) et du lagrangien augmenté (LA) duales avec l'optimisation multi-étapes (MS), nous avons simulé un réacteur CANDU-6 avec toutes les barres de contrôle (liquides et de compensation). Le facteur de multiplication des neutrons visé (k_{ref}) est de 1; la puissance canal maximum autorisée est de $6700kW$. Dans un premier temps nous résoudrons un premier problème simple à 2 variables de décision seulement pour valider l'implantation des méthodes. Ensuite, nous testerons la robustesse et la stabilité des algorithmes MS et LA en augmentant la complexité du problème, en particulier le nombre de zones de combustion.

4.4.1 Cas à 2 zones

Pour tester les méthodes alternatives de gradient, nous avons utilisé une configuration du réacteur où les barres liquides ne sont pas toutes au même niveau. Ce choix permet d'avoir plus facilement une distribution plane de la puissance. Pour ce cas simple, les deux variables de décision correspondent à deux burnups moyens de sortie pour 2 zones radiales dans le réacteur comme illustré à la figure 3.5 et rappelé à la figure 4.9. Le même point de départ est utilisé pour toutes les méthodes. Ce point ne fait volontairement pas parti du domaine réalisable pour tester l'efficacité des méthodes. La même précision a été choisie pour toutes les approches. Le minimum de la contrainte quadratique a été fixé à $S_{min} = 1$. Pour le LA et la MPE, celui-ci est plus grand pour les deux premières itérations externes (10 puis 5) pour réduire le nombre total de calculs de flux nécessaires. Par définition (normalisation), le poids de la fonction objectif dans le LA et la MPE est de 1. Celui de la contrainte sur k_{eff} est de 1000 pour les deux méthodes duales.

En ce qui concerne les contraintes sur la puissance maximum, nous avons utilisé un poids initial supérieur pour la MPE ($\gamma_j^0 = 20$) que pour le LA ($\gamma_j^0 = 2$). Nous avons remarqué que la réduction parabolique du pas d'avance converge plus rapidement que lorsqu'une diminution de moitié du pas d'avance est utilisée. Nous avons remarqué que la réduction parabolique du pas d'avance converge plus rapidement que lorsqu'une diminution de moitié du pas d'avance est utilisée. Nous avons donc utilisé la réduction parabolique du pas d'avance.

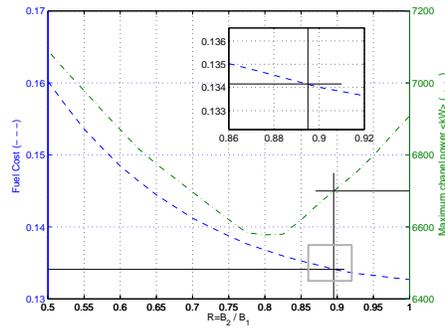
Les résultats d'optimisation de la gestion du combustible sont présentés au tableau 4.3. B_1 et B_2 représentent les burnups obtenus au centre et en périphérie respectivement à la fin de chaque optimisation. R est égal au ratio B_2/B_1 . F_C est la valeur adimensionnelle de la fonction de coût en combustible. Les trois dernières colonnes représentent le nombre d'itérations externes (MPE et LA), de calcul de flux et d'évaluation de l'ensemble des gradients, c'est à dire l'effort de calcul correspondant aux diverses optimisations.

Les méthodes MS et LA donnent des résultats similaires ($\sim 0.1\%$ sur B_1 , B_2 et F_C) qui sont dans le domaine réalisable. Le LA requiert cependant un effort de calcul plus important. Les résultats obtenus avec la MPE sont légèrement meilleur que pour les deux autres méthodes (B_i supérieur de $\sim 0.8\%$ et F_C réduit de $\sim 0.8\%$). Cependant, les contraintes ne sont pas respectées de manière aussi précise pour la MPE (k_{eff} sous-critique de $0.5mk$) que pour le LA et la MS (k_{eff} dans une fourchette de $0.05mk$). Nous avons également testé un poids plus grand pour la contrainte sur k_{eff} pour plus de précision, mais une convergence prématurée a été obtenue avec la méthode MPE. Nous avons jugé que l'écart sur k_{eff} obtenu avec la MPE est inacceptable, alors que celui avec les deux autres méthodes est raisonnable voir négligeable. De plus, l'effort de calcul avec la MPE est beaucoup plus grand. cette méthode ne semble donc pas prometteuse.

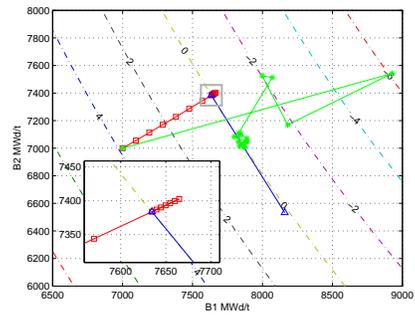
Pour vérifier la précision des résultats de la MS et du LA, nous avons tracé la variation des fonctionnelles du problème d'optimisation (2.18) à la figure 4.8. Sur le graphe a), nous avons représenté la valeur de la puissance canal maximum et de la fonction coût en fonction du rapport R , et ce, dans le domaine où $k_{eff} = 1$. La fonction objectif ne cesse de diminuer avec R , cependant la puissance canal maximum dépasse la limite prescrite $P_{lim} = 6700kW$. L'optimum de la gestion du combustible est alors obtenu pour un rapport $R = B_2^e/B_1^e \approx 0.895$ (ligne verticale noire), limité en fait par P_{lim} (ligne horizontale noire de droite). Le même rapport a été obtenu pour la MS et du LA, ce qui confirme l'exactitude de leurs résultats. Les figures b), c) et d) présentent une carte de k_{eff} , $P_{C,max}$ et F_C respectivement dans le domaine de recherche. Les lignes avec des petits chiffres sont en fait des courbes de niveau. Sur ces figures, nous avons également représenté la valeur du vecteur de décision pour toutes les itérations valides. Les deux méthodes donnent les mêmes résultats finaux, mais elles ne suivent pas du tout le même chemin. En effet, le LA est plus rapide pour trouver un point presque réalisable mais a besoin de nombreuses itération pour converger. Sur la partie agrandie des figures, nous pouvons voir que le LA oscille autour de la contrainte $k_{eff} = 1$.

4.4.2 Cas à N zones de combustion

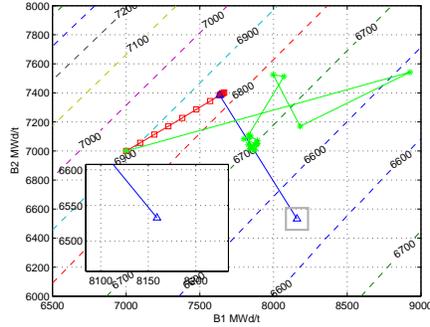
Nous avons appliqué l'optimisation de la gestion du combustible à des cas plus complexes pour tester la stabilité des méthodes alternatives. Seul le niveau des barres liquides est changé par rapport au cas simple. En effet, un niveau uniforme de 50% est utilisé pour garder l'aspect plus général de l'optimisation. La méthode MPE n'a pas été retenue étant donné ses résultats peu précis quant au respect des contraintes et aux grands efforts de calculs nécessaires pour le cas le plus simple. Des problèmes avec 3, 4, 5, 7 et 11 variables de décision ont été envisagés. La répartition des zones de combustion correspondante est donnée à la figure 4.9, seul



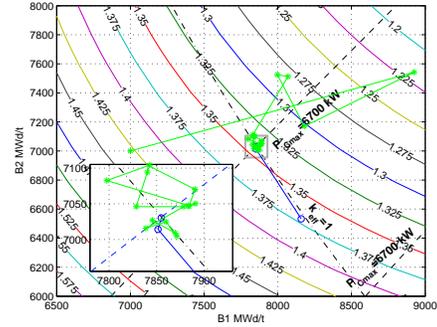
$P_{C,max}$ (kW) pour $k_{eff} = 1$, (a)



k_{eff} (mk), (b)



$P_{C,max}$ (kW), (c)



F_C , (d)

Figure 4.8 Fonction objectif et puissance canal maximum vs. rapport $R = B_2^e / B_1^e$ pour $k_{eff} = 1$

Tableau 4.3 Résultats pour 2 zones de combustion pour un CANDU-6 avec toutes les barres de contrôle ($\delta = 0.0001$, $S_{min} = 10/5/1$)

Méthode	B_1	B_2	R	F_C	k_{eff}	$P_{C,max}$	nb ext it.	tot nb it.int.	nb GPT cal
Départ	7000	7000	1.0						
$k_{eff} = k_{ref}$	7634	7383						20	14
min $P_{C,max}$	8159	6532						2	1
LEMKE	7854	7030	0.8950	1.3415	0.999994	6700.6		7	3
total								29	18
LA	7845	7026	0.8956	1.3426	1.000049	6701.3	7	55	26
MPE	7905	7096	0.8976	1.3310	0.999588	6702.6	30	150	68
<i>parabolique</i>									

un quart de réacteur est représenté étant donné la symétrie du cœur. Les zones sont réparties radialement pour cette étude.

Le point de départ pour les deux méthodes retenues (LA et MS) est une distribution uniforme du burnup moyen de sortie ($7000 MW_j/t$). Pour l'approche MS, nous avons tout d'abord trouvé un point réalisable à deux zones après les étapes $k_{eff} = 1$ et min $P_{C,max}$. Ensuite, nous avons conservé cette même distribution comme point de départ de la dernière étape des calculs quelque soit le nombre de zones de combustion. Pour la méthode LA, le poids associé à la contrainte sur k_{eff} est de 1000, et celui pour la limite de puissance canal q_j est de 2. Les résultats obtenus sont donnés dans le tableau 4.4.

Plus le nombre de zones est grand, plus la distribution de burnup peut être ajustée pour augmenter l'efficacité du combustible. Les résultats de la deuxième colonne montrent en effet une diminution du coût en combustible (F_C) avec le nombre de variable de décision, et ce pour les deux méthodes. Les distributions de burnup

obtenues sont aussi très similaires avec les deux algorithmes (pour le même nombre de zones). La principale différence se retrouve généralement seulement dans les trois anneaux centraux (R_1 à R_3). Le fait que les zones correspondantes contiennent peu de canaux (figure 4.9) jouent beaucoup sur les résultats, nous reviendrons ultérieurement sur ce point dans la section 4.4.5.

Comme nous l'avons vu à la section 3.3.2 et par expérience, le burnup est grand au centre pour réduire la puissance, et il reste faible sur les bords pour y augmenter la puissance. Ceci permet d'aplatir la distribution de puissance lorsqu'un nombre faible de zones de combustion est utilisé (typiquement 2 ou 3). Cela a pour conséquence par contre d'augmenter le flux en périphérie, et donc les fuites vers l'extérieur du réacteur, réduisant ainsi l'efficacité énergétique totale du combustible. Ainsi, dès que le nombre de zones de combustion augmente, le profil du flux s'ajuste pour minimiser les fuites, tout en respectant la contrainte sur la puissance canal maximum. Aussi, nous pouvons observer dans les résultats du tableau 4.4 pour les deux méthodes (MS et LA), que le comportement burnup élevé au centre et faible en périphérie ne reste valide que pour deux zones. En effet, dès que les zones deviennent plus petites au centre, la barre liquide du centre du réacteur (figure 2.1) a un impact majeur sur le burnup dans les zones correspondantes. Pour compenser la forte absorption de la barre, le burnup diminue dans les trois anneaux centraux (R_1 à R_3 du cas à 11 zones). De plus, nous pouvons remarquer une augmentation du burnup dans les zones en périphérie (R_{10} et R_{11}), qui va à l'encontre de l'expérience à deux zones. En fait, plus le burnup est grand en "extrême" périphérie plus le flux est petit, et donc plus les fuites neutroniques sont faibles. L'efficacité du combustible est alors augmentée, et la fonction de coût F_C diminue. Une analyse plus détaillée de la distribution de burnup pour les différents nombres de zones est donnée dans [32].

En ce qui concerne les résultats de la distribution de burnup, les méthodes LA et

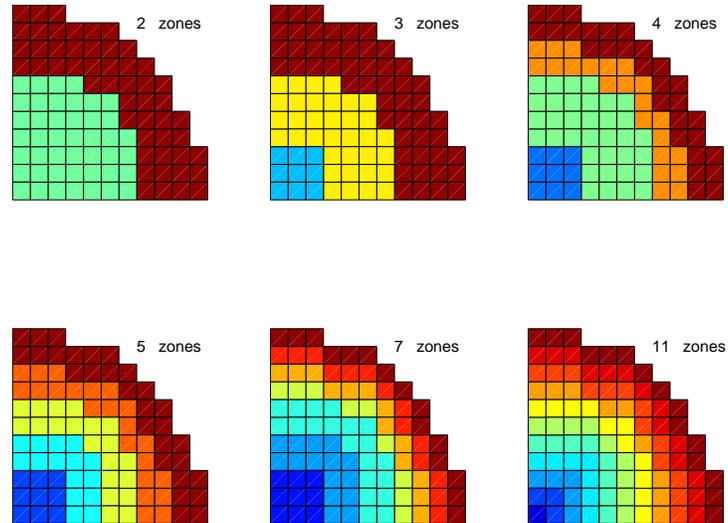


Figure 4.9 Distribution des zones de burnup pour différents nombres de zones

MS sont équivalentes. La comparaison des efforts de calcul sera traité à la section 4.4.4.

4.4.3 Influence de la répartition des zones

La répartition des zones à la figure 4.9 est arbitraire. Le choix de leur aspect radial vient en partie d'une habitude historique et surtout d'une supposition à propos des mécanismes de contrôle. Leur impact sur la gestion du combustible est loin d'être négligeable quand les distributions de flux avec ou sans contrôleurs sont comparées. Par contre leur position non symétrique haut/bas et radialement n'est pas pris en compte avec une distribution radiale du burnup. Pour illustrer ce point, nous avons tracé la répartition de burnup et la puissance canal correspondant au calcul à 11 zones avec la méthode MS à la figure 4.10 respectivement à gauche et à droite. Les rectangles blancs indiquent la partie remplie d'eau des barres liquides, et les

Tableau 4.4 Résultats pour de plus grands nombres de zones de combustion

zones	F_C	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}						
$P.I.*$		7000																
Méthode MS																		
$P.R.**$	138246	8108						6464										
2	136085	7852						6837										
3	135700	7103			7939						6958							
4	135635	7180			7964						6709		7194					
5	135346	7328			8126			7649			6859		7189					
7	135097	7013			8256			7582			6915		6893		7111		7659	
11	135016	7507	6929	7094	8039	8374	7516	7593	6958	6864	7114	7734						
Méthode LA																		
2	1.36065	7851						6840										
3	1.35780	7343			7907						6921							
4	1.35571	7302			7943						6683		7217					
5	1.35198	7008			8264			7568			6862		7384					
7	1.35096	7014			8256			7564			6943		6919		7116		7615	
11	1.34995	6877	6892	7240	8036	8342	7557	7533	7006	6924	7119	7640						

(*) $P.I.$: Point initial

(**) $P.R.$: Point réalisable

signes 'X' correspondent aux canaux où la puissance canal maximum est atteinte. Seul un canal par zone voit sa puissance égaler la limite, ce qui veut dire que ce canal a en fait un caractère limitatif sur l'appâtissement potentiel de la répartition de puissance. De plus, tous les canaux ayant atteint la puissance canal limite sont dans la moitié supérieure. Un débalancement net haut/bas de la puissance est visible également sur la moitié droite de la figure.

Pour palier à ce phénomène, nous avons reconduit les optimisations avec les mêmes zones (2 à 11 de la figure 4.9) mais chacune séparée en deux moitiés haut/bas. Ces cas sont notés $N \times 2$. Les résultats pour 2×2 et 11×2 sont donnés au tableau 4.5 pour la méthode MS. Premièrement, la fonction de coût est nettement améliorée par rapport aux cas à N zones. Dans le cas des zones radiales, la fonction de coût est améliorée de 0.33% et 0.79% en passant de 2 à 4 ou 11 zones respectivement. Ici, l'optimisation avec 2×2 zones donne quant à elle un gain de 0.94%, soit une amélioration supérieure au cas à 11 zones radiales.

Nous pouvons également noter que le burnup est généralement plus grand dans la partie supérieure du cœur pour un même anneau R_i . Ceci permet de réduire la différence de puissance canal entre les moitiés inférieures et supérieures du réacteur. En effet, pour le cas à 11 zones radiales, cette différence est d'en moyenne $257kW$, alors qu'elle n'est que de $100kW$ pour le cas à 11x2 zones. Pour illustrer ce point, nous avons représentée à la figure 4.11 la distribution de burnup obtenue pour 11x2 zones de manière similaire à la figure 4.10. Nous remarquons à nouveau qu'un seul canal dicte le burnup dans les zones avec les canaux les plus chauds. Par contre, les canaux à la limite de puissance ne sont plus tous situés dans la moitié supérieure du réacteur.

Une analyse plus poussée notamment sur les différences 2/11 versus 2x2/11x2 est faite dans [32]. En résumé, nous observons que:

- un plus grand nombre de zones ne garantie pas une distribution de burnup avec un coût en combustible plus faible.
- la répartition des zones doit tenir compte des asymétries du réacteur le plus possible.
- le burnup dans une zone est limité par un seul canal.

Tableau 4.5 Burnups pour des zones séparées haut/bas.

zones	F_C	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}	
<i>P.I.*</i>		7000											
<i>P.R.**</i>	138246	8108						6464					
2x2	134805	7683						7189					
		7605						7129					
11x2	133969	7536	7316	7464	7984	8042	7354	7130	7030	7187	7624	8451	
		7505	7287	7390	8106	7487	7608	6966	6911	7137	7519	8305	

(*) *P.I.* : Point initial

(**) *P.R.* : Point réalisable

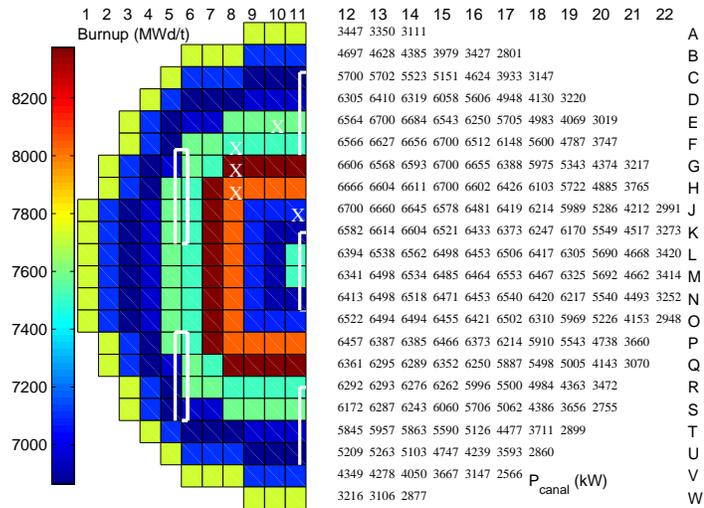


Figure 4.10 Distribution du burnup et de la puissance canal pour 11 de zones

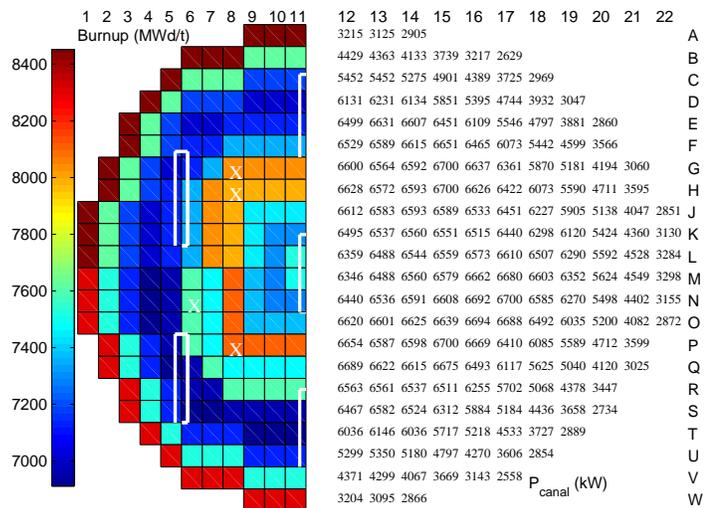


Figure 4.11 Distribution du burnup et de la puissance canal pour 11x2 de zones

4.4.4 Influence du poids des contraintes et effort de calcul

Dans les calculs précédents, pour la méthode LA, le poids initial associé à la contrainte sur k_{eff} est de 1000, et celui pour la limite de puissance canal q_j est de 2. Ces poids ont été choisis en fonction de la précision requise et de l'ordre de grandeur de l'écart entre les contraintes et leur limite correspondante. Nous référerons à ce cas par la suite comme le cas 'LA-lw', les poids correspondent à de petites valeurs.

Dans la section 4.3.2, nous avons remarqué que le poids des contraintes s'ajuste à chaque itération externe. Il serait cependant important de savoir comment celui-ci varie, et surtout quelle influence il a sur la fonction objectif et le respect des contraintes. Pour répondre à cette question, nous avons repris les calculs d'optimisation avec la méthode LA pour différents poids initiaux: soit 1000 pour k_{eff} et 20 pour P_{lim} qui correspond au cas 'LA-mw', ou 5000 et 100 respectivement pour le cas 'LA-hw'. Nous avons reconduit les calculs sur le réacteur CANDU-6 pour différents nombres de zones radiales.

Les résultats pour le burnup moyen de sortie du cœur ainsi que pour la fonction objectif sont présentés respectivement à la figure 4.12 et au tableau 4.6. Le burnup moyen de sortie du cœur est la moyenne des variables de décision pondérée par la puissance délivrée par les zones. Nous avons reproduit les résultats avec la méthode MS et avec le cas 'LA-lw'. Pour le cas le plus simple (2 zones), le burnup moyen de sortie est similaire pour toutes les méthodes testées. Lorsque le nombre de zones de burnup augmente, les résultats pour des faibles poids ('LA-lw') et des moyens poids ('LA-mw') de contraintes initiales restent semblables. Par contre, lorsque les poids initiaux des contraintes sont grands ('LA-hw'), le burnup moyen de sortie et la fonction objectif se détériorent. Les contraintes sont quant à elles toujours respectées. En effet, l'erreur obtenue sur k_{eff} est généralement comprise dans une fourchette de $\pm 0.01mk$ et celle sur la puissance canal maximum ne dépasse pas

1kW.

Le tableau 4.7 donne l'effort de calcul des optimisations présentées au tableau 4.6. Les nombres It_e et It_i représentent respectivement le nombre d'itérations externes et internes. Il y a 1 calcul du flux neutronique par itération interne plus 1 pour le point de départ. L'ensemble des gradients est calculé $Grad$ fois, c'est à dire à chaque pas d'avance valide et pour chaque initialisation. Les résultats pour la méthode MS incluent les itérations pour trouver un point initial réalisable à deux zones: soient $17+6=23$ calculs de flux et $12+3=15$ évaluation des gradients (cf. figure 4.8). La méthode du LA demande généralement beaucoup plus d'itérations que la méthode MS. Cependant, avec un grand poids initial pour les contraintes ('LA-hw') le nombre de calculs de flux est plus petit que pour les cas avec un poids faible ou moyen. La convergence est donc plus rapidement atteinte dans le cas 'LA-hw', mais étant donné des résultats moins optimaux observés au tableau 4.6, cela suggère une convergence prématurée. Nous avons suivi plus en détail le poids des contraintes pour le cas à 11 zones, et avons remarqué que, pour certaines contraintes sur la puissance canal, leur poids atteint une valeur 3 fois plus grande dans le cas 'LA-hw' que pour les deux cas 'LA-mw' et 'LA-lw'. Ce large poids (environ 300) entraîne une instabilité pour le problème linéarisé (4.12) qui engendre une convergence prématurée de l'algorithme.

En résumé, le LA demande plus de calculs que la MS. En général, les résultats pour la fonction objectif avec le LA sont proches de ceux obtenus avec la MS. Cependant, un trop grand poids des contraintes entraîne une instabilité de la méthode et une convergence prématurée.

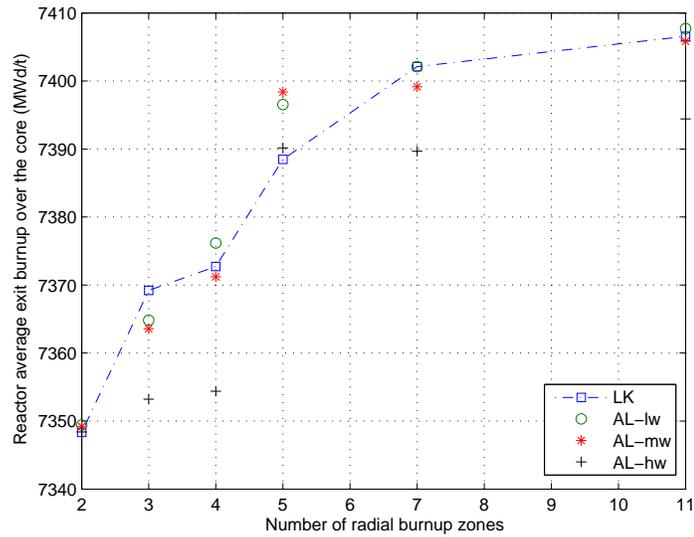


Figure 4.12 Burnup moyen de sortie \overline{B}^e suivant le nombre de zones pour différentes méthodes d'optimisation

Tableau 4.6 Fonction objectif F_C suivant le nombre de zones pour différentes méthodes d'optimisation (sans dimension).

# zones	MS	LA-lw	LA-mw	LA-hw
2	13609	13607	13607	13609
3	13570	13578	13580	13599
4	13563	13557	13566	13597
5	13535	13520	13516	13531
7	13510	13510	13515	13532
11	13501	13499	13503	13524

Tableau 4.7 Efforts de calculs pour les résultats du tableau 4.6.

# zones	MS *		LA-lw			LA-mw			LA-hw		
	It_i	$Grad$	It_e	It_i	$Grad$	It_e	It_i	$Grad$	It_e	It_i	$Grad$
2	27	18	12	73	34	20	138	64	6	42	19
3	34	22	10	89	49	15	144	80	9	69	34
4	38	27	4	66	48	19	139	69	6	53	30
5	48	29	6	58	32	20	170	88	8	83	47
7	44	30	14	112	58	14	115	59	3	41	28
11	45	27	16	116	62	7	92	58	6	86	57

* les résultats incluent les itérations pour trouver un point initial réalisable (17+6=23 et 12+3=15).

4.4.5 Influence du point de départ

Dans cette section, nous avons étudié l'influence du point initial sur la distribution de burnup optimisée pour les différentes méthodes. Cependant, avant de présenter des résultats, nous tenons à faire quelques remarques, et à apporter quelques précisions quant aux calculs obtenus jusqu'à présent. Premièrement, la méthode MS telle qu'utilisée jusqu'à présent peut être qualifiée *d'hybride* par rapport à la méthode PQL. En effet, seulement 2 zones ont été utilisées pour trouver un point réalisable, ensuite la méthode PQL à N zones est utilisée pour finir de résoudre le problème d'optimisation. Deuxièmement, pour savoir si cela a une importance, nous avons reconduit la méthode MS pour les calculs du cas à 11 zones de burnup, en utilisant la discrétisation fine des zones dès la recherche du point réalisable. Le terme de méthode MS se référera par la suite à cette approche et le qualificatif de *complète* sera utilisé en cas de doute. Le résultat est donné au tableau 4.8 avec un rappel du résultat obtenu avec la version hybride de la méthode MS. Les distributions de burnup sont similaires dans les deux cas sauf pour deux anneaux du centre (R_1 et R_3). Le résultat de la méthode MS complète est en fait plus proche du résultat de la méthode LA avec des faibles poids initiaux des contraintes que le résultat de la MS hybride.

Le burnup dans la zone R_1 qui correspond au point réalisable avec la méthode MS complète est de $7091MWj/t$, alors qu'il est de $8108MWj/t$ pour la méthode MS hybride. La méthode PQL est donc très influencée par le point de départ pour des cas à grand nombre de zones. Ce comportement s'explique en fait par un biais mathématique de la fonction objectif. Afin de mieux illustrer ce point, nous pouvons renormaliser et récrire la fonction objectif (éq. (2.10)) comme suit:

$$F_C \frac{P_{tot}}{C_U} = \sum_j \frac{1}{B_j} P_j = \sum_z \frac{1}{B_z} \bar{P}_z n_z \quad (4.22)$$

où C_U est le prix constant du nouveau combustible, les indices j et z correspondent aux canaux et aux zones de burnup respectivement, \bar{P}_z est la puissance moyenne de la zone z et n_z le nombre de canaux dans la zone z .

Premièrement, quand le burnup moyen de sortie B_z augmente, la physique des réacteurs entraîne que la puissance canal moyenne \bar{P}_z diminue, et donc le rapport $R_z = \frac{\bar{P}_z}{B_z}$ devient moins important dans la sommation de l'équation (4.22). Or pour maintenir k_{eff} égal à 1, le burnup moyen de sortie doit aussi diminuer dans au moins une autre zone. Ainsi, afin de minimiser *plus rapidement* le coût en combustible, les zones avec un grand nombre de canaux n_z ont tendance à voir leur ratio diminuer plus fortement que les zones avec un nombre de canaux plus restreint. La figure 3.5 montre que la zone numéro 1 ne contient que 4 canaux, et donc sa contribution dans la somme de la fonction objectif est faible. Il apparait alors logique que le burnup final obtenu dans ces petites zones soit proche du point initial. Ce phénomène apparait cependant moins à 2 zones, puisque dans ce cas le nombre de canaux dans chaque zone est plus constant.

Afin de savoir si les méthodes MS et LA sont également influencées par le biais mathématique de la fonction objectif, nous avons reconduit l'optimisation pour ces deux méthodes en partant toujours d'une distribution de burnup uniforme mais plus grande soit $8000MWj/t$. Les résultats sont donnés au tableau 4.9, la différence

Méthode	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}
PQL *	7507	6929	7094	8039	8374	7516	7593	6958	6864	7114	7734
MS +	6945	6918	7210	8035	8374	7509	7599	6926	6946	7048	7751

* Point initial: $B_1 = \dots = B_7 = 8108MWd/t$ et $B_8 = \dots = B_{11} = 6464MWd/t$.

+ Point initial: $B_1 = \dots = B_{11} = 7000MWd/t$.

Tableau 4.8 Distribution du burnup pour 11 zones pour la méthode MS 'hybride' et 'complète'.

Methode	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}
MS +	7022	6911	7200	8037	8374	7512	7611	6898	6847	7196	7725
$ \Delta \%$	1.11	0.10	0.14	0.02	0.00	0.04	0.16	0.40	1.42	2.10	0.34
LA-lw+	7563	7041	7048	8025	8336	7538	7598	6934	6873	7126	7722
$ \Delta \%$	9.97	2.16	2.65	0.14	0.07	0.26	0.86	1.03	0.74	0.10	1.07
LA-hw+	7788	7431	7312	7802	8263	7692	7556	7086	6844	6986	7392
$ \Delta \%$	10.67	4.19	0.00	1.28	2.55	1.89	1.25	0.88	2.16	0.77	3.02

+ Point initial: $B_1 = \dots = B_{11} = 8000MWd/t$.

Tableau 4.9 Distribution du burnup pour 11 zones pour un autre point initial.

relative avec les cas précédents $|\Delta|\%$ est aussi rapportée. La forme générale des distributions du burnup obtenues est très similaire aux résultats précédents. La méthode MS n'est quasiment pas influencée par le point de départ. En effet, la différence maximum entre les résultats à partir d'un burnup faible et grand n'est que de 2%. Au contraire, pour la méthode LA, les résultats dans les petites zones semblent très dépendants du point de départ. Une différence d'environ 10% est obtenue pour B_1 , que ce soit en utilisant un faible ou un grand poids initial pour les contraintes.

4.4.6 Conclusions

Les trois méthodes PQL, MS et LA donnent des résultats similaires en ce qui concerne la distribution optimale de burnup. Contrairement aux méthode PQL et MS, pour le LA, la méthode ne garantie pas que les contraintes soient respectées, mais nous avons toujours trouvé une solution qui respecte $k_{eff} = 1$ et $P_{C,max} <$

P_{lim} . Elles ont cependant toutes des avantages et des inconvénients. Premièrement la méthode PQL :

- elle est rapide.
- mais, son application est limitée aux problèmes pour lesquels un point réalisable est connu. De plus, les résultats dépendent du point de départ.

Deuxièmement, la méthode MS:

- elle est générale dans le sens où elle n'est pas limitée à un point réalisable. Elle ne semble pas influencée par le choix du point initial.
- par contre, elle est relativement longue / difficile à utiliser par l'utilisateur, c'est à dire qu'il est nécessaire de créer un fichier d'entrée pour chaque étape. Il faut d'une étape à l'autre soit copier "manuellement" les résultats, ce qui peut être une source d'erreur, soit sauvegarder l'objet `L_MAP` qui contient les variables de décision (options de rechargement), ce qui implique une connaissance "avancée" des options de DONJON. De plus, les résultats des calculs à 2 zones dans la section 4.4.1 semblent indiquer que la MS est plus lente pour trouver un point réalisable que la méthode LA.

Enfin, la méthode du LA:

- cette dernière est relativement simple à utiliser et aussi très générale.
- cependant, même si elle se rapproche rapidement d'un point réalisable (cf. section 4.4.1), elle converge lentement et nécessite un bon choix du poids des contraintes. Ses résultats sont également dépendants du point initial.

4.5 Méthode mixte

4.5.1 Définition

Dans la section précédente 4.4.6, nous avons résumé les avantages et limites de chacune des méthodes employées jusqu'à présent. À partir de ces constatations, nous avons développé une méthode qui tente de ne retenir que les avantages des méthodes PQL, MS et LA. Cette méthode dite mixte (MM) conserve l'idée de la méthode MS à savoir trouver un point réalisable dans un premier temps et ensuite résoudre le problème original d'optimisation de la gestion du combustible donné par l'équation (2.18) avec la méthode PQL qui converge rapidement. La méthode LA est utilisée pour sa rapidité à trouver un point réalisable. La méthode MM est plus simple à utiliser que la méthode MS étant donné qu'un seul problème d'optimisation est à définir.

Le but de la méthode LA ayant changé, le choix du poids initial faible des contraintes est à revoir. Nous avons tracé à la figure 4.13 la variation des contraintes avec les itérations pour le cas à 11 zones pour les trois choix de poids initiaux des contraintes. Les traits verticaux en pointillés représentent les itérations externes, et le trait horizontal en tiret représente la limite des contraintes. Avec un large poids initial des contraintes, la convergence vers un point réalisable est très rapide. En effet, dès la deuxième itération externe, les deux types de contraintes sont respectées dans le cas 'LA-hw', alors que $k_{eff} \neq 1$ pour des poids faible ou moyen. De plus, une convergence grossière sur les itérations internes est rapidement atteinte pour le cas 'LA-hw', elle se traduit par un plateau à partir de 10 itérations. Ainsi dans ce cas, le nombre maximum d'itérations internes peut même être réduit de 20 à 10.

En résumé, la méthode mixte consiste à trouver un point réalisable avec le lan-

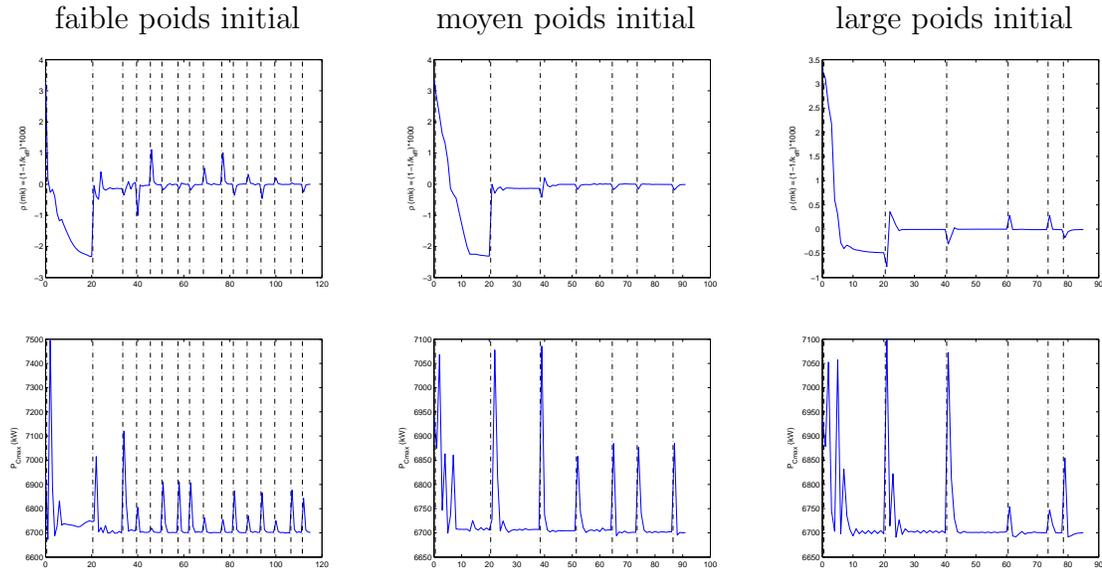


Figure 4.13 Variation de $\rho = 1 - 1/k_{eff}$ et $P_{C,mac}$ avec les itérations internes de la méthode LA.

grangien augmenté en utilisant des poids de contraintes grands: soit 5000 pour k_{eff} et 100 pour P_{lim} , puis ce vecteur de décision est amélioré avec la méthode de programmation quasi-linéaire.

4.5.2 Résultats de la fonction coût

Nous avons déjà appliqué la méthode MM pour l'optimisation de la gestion du combustible du CANDU-6 en utilisant de 2 à 11 zones radiales de burnup. les résultats sont présentés dans [33]. En résumé, cette méthode s'avère en général plus rapide que la méthode MS et n'est pas influencée par le choix du point initial pour les cas envisagés. Le biais mathématique associé à la définition de la fonction objectif ne semble pas être très important pour cette méthode. Elle semble donc efficace et prometteuse. Cependant, nous pouvons nous demander jusqu'à quel niveau de complexité du problème d'optimisation (nombre de zones de burnup) les constatations faites restent valables.

Tableau 4.10 Fonction coût pour les cas à 2, 11 et 190 zones.

	PQL	MS f7	LA-lw f7	LA-hw f7	MM f7
2	13609	13609	13607	13609	13609
11	13502	13501	13499	13524	13500
190	13399	13408	13408	13434	13422

Afin d'éprouver les limites des différentes méthodes, nous avons envisagé le cas le plus complexe pour l'optimisation de la distribution du burnup de sortie: une zone par canal. Étant donné la symétrie droite/gauche du réacteur, 190 zones sont utilisées. Chacune des zones comprenant 2 canaux, aucun biais mathématique dû à la définition de la fonction objectif n'est donc introduit.

Les résultats de la fonction de coût (sans dimension) sont présentés au tableau 4.10 pour 190 zones avec ceux à 2 et 11 zones radiales pour les différentes méthodes. Le point de départ de toutes les méthodes correspond à une distribution uniforme du burnup de $7000 MWj/t$. Cependant pour la méthode PQL, le point de départ est le point réalisable trouvé avec la méthode MS pour 2 zones quelque que soit le nombre de zones optimisées. Avec 190 zones, toutes les méthodes arrivent à trouver un point réalisable et converger vers un optimum. Elles ne trouvent cependant pas toutes le même optimum. En effet, une variation de la fonction objectif allant jusqu'à 35 est observée, alors que pour 2 zones, cette variation n'est que de 2. Le pire résultat (pour la fonction objectif) est obtenu avec le LA avec un poids élevé des contraintes. Ceci va dans le même sens que les résultats déjà présentés pour 11 zones, et rappelés au tableau 4.10. Par contre, le résultat pour 190 zones avec la MM n'est pas aussi bon que celui de la MS. La MM ne semble donc pas aussi efficace pour un problème avec un nombre de zones très grand, certainement à cause de la première étape avec le LA-hw.

Nous avons également repris l'optimisation pour 190 zones à partir d'une distri-

Tableau 4.11 Fonction coût pour le cas à 190 zones à partir de $8000MWj/t$.

	MS f8	LA-lw f8	LA-hw f8	MM f8
F_C	13387	13390	13395	13383
$ \Delta $	21	18	39	35

bution uniforme du burnup de $8000MWj/t$ pour tester l'influence du point de départ dans ce cas très complexe pour les différentes méthodes alternatives proposées dans ce chapitre. Les résultats pour la fonction objectif sont donnés au tableau 4.11. Toutes les valeurs obtenues avec les méthodes alternatives présentent une amélioration par rapport à la méthode PQL de référence, alors que les résultats à partir de $7000MWj/t$ sont moins minimisés pour les méthodes alternatives que pour PQL. L'écart entre résultats obtenus à partir des deux points initiaux est également donné au tableau 4.11. Suivant le point de vue, cet écart peut sembler être négligeable ou important. Pour l'ingénieur, la fraction représentée par la différence par rapport aux résultats est faible (au maximum 0.3%). Toutes les méthodes semblent donc équivalentes. Par contre, toujours d'un point de vue ingénieur, cet écart peut sembler grand quand il est comparé à la réduction de la fonction objectif grâce au nombre de zones plus grand. En effet, le fait d'utiliser 190 zones permet de réduire F_C approximativement de 0.75% et 1.5% seulement par rapport aux cas à 11 et 2 zones respectivement. De plus d'un point de vue mathématique, le point de départ introduit clairement un biais sur les résultats de cas aussi complexes que 190 zones. En effet, ils sont soit améliorés pour toutes les méthodes si le burnup de départ est grand, et détériorés dans le cas contraire.

4.5.3 Effort de calcul

Les différentes méthodes proposées ne requièrent pas le même effort de calcul. De plus, celui dépend également du point de départ utilisé. L'effort de calcul correspondant aux résultats présentés à la section précédente est donné au tableau 4.12. It_e , ϕ et $Grad$ correspondent respectivement aux nombres d'itérations externes, de calculs de flux et de calculs de l'ensemble des gradients. Pour la méthode MM, It_e est égal au nombre d'itération externe pour trouver un point réalisable plus un pour la méthode PQL comme dernière étape. Les nombres entre parenthèses pour ϕ et $Grad$ correspondent également à cette dernière étape. 'f7' et 'f8' correspondent aux distributions uniformes de burnup de $7000MWj/t$ et une de $8000MWj/t$ respectivement.

Le tableau 4.12 montre également que la méthode LA requiert beaucoup plus de calculs de flux et de gradients que les méthodes PQL et MS. Cependant, pour la méthode MM, même basée sur la méthode LA, la convergence est atteinte avec seulement un petit effort de calcul supplémentaire par rapport aux méthodes PQL et MS. Que ce soit avec la méthode MS ou MM, un point réalisable est trouvé deux fois plus rapidement en partant d'une distribution de faible burnup que d'une de grand burnup. En effet, pour la méthode MS, seulement 26 itérations sont nécessaires à partir de $7000MWj/t$ en comparaison de 58 à partir de $8000MWj/t$.

Pour la méthode MM, le rapport est similaire avec 42 en comparaison de 103 pour trouver un point réalisable à partir d'une distribution de faible burnup et de grand burnup respectivement. Cependant, comme nous venons de le voir, même si un point réalisable est plus rapidement trouvé à partir d'une distribution uniforme de $7000MWj/t$, la fonction objectif est moins bonne. Deux explications peuvent être avancées: soit un minimum local est détecté, soit la convergence est légèrement prématurée. Encore plus difficilement qu'à 11 zones, il n'y a pas

Tableau 4.12 Nombre d'itérations requises pour le cas à 190 zones.

	It_e	ϕ	$Grad$
PQL	-	70=18+7+45	54=12+3+39
MS f7	-	52=13+13+26	45=8+12+25
MS f8	-	72=22+36+14	63=17+33+13
LA-lw f7	14	204	149
LA-lw f8	7	129	105
LA-hw f7	11	213	169
LA-hw f8	11	199	163
MM f7	5=4+1	49(7)	40(6)
MM f8	11=10+1	121(18)	94(15)

moyen de représenter la fonction objectif analytiquement ou graphiquement pour 190 zones. L'existence d'un minimum local est donc purement hypothétique. Quant à la deuxième théorie, nous avons remarqué précédemment que la méthode PQL est influencé par le point de départ. Plus il y a de variables de décision, plus les options d'optimisation (contrainte quadratic initiale, critère de convergence, ...) sont difficiles à choisir. La deuxième explication semble donc plus probable, bien que l'une n'exclut pas l'autre.

4.5.4 Résultats de la distribution de burnup

Les méthodes MS et MM en partant d'une valeur grande ($8000MWj/t$) du burnup donnent des résultats pour la fonction objectif similaires, légèrement meilleur que la méthode PQL. Cependant, ces résultats sont globaux, et aucune différence locale sur la distribution de burnup ou de puissance n'est faite.

Nous avons tracé à la figure 4.14 le burnup obtenu pour les 190 zones pour les trois méthodes PQL, MS et MM. La puissance canal correspondante est donnée à la figure 4.15. Sur la partie gauche de ces deux figures, les résultats de la méthode PQL sont illustrés. La différence entre ces derniers et les résultats des deux autres méthodes est représentée sur la droite de ces figures. La différence est exprimée en pourcentage avec PQL comme référence, les chiffres en caractères normaux et italiques correspondent respectivement aux méthodes MS et MM. Comme à la

figure 4.10, les rectangles blancs représentent les barres liquides et les 'X' les canaux qui ont atteint la puissance canal limite.

Le côté gauche de la figure 4.14 et du tableau 4.4 montrent que les résultats à 190 et 11 zones sont semblables du point de vue de la distribution de burnup obtenue. En effet, la valeur du burnup dans la partie centrale du réacteur (B_1 à B_3 pour le cas à 11 zones) est intermédiaire dans les deux cas. Ensuite, le burnup augmente de manière significative et atteint son maximum pour B_4 et B_5 à 11 zones, et pour les canaux G8/H8 et P8 à 190 zones. Puis, il diminue pour atteindre ses valeurs minimums dans les anneaux suivants (B_6 à B_9 pour le cas à 11 zones). Finalement, le burnup réaugmente dans la périphérie (B_{10} et B_{11} pour le cas à 11 zones). La distribution plus complexe du burnup obtenue à 190 zones correspond à une distribution plus aplatie de la puissance canal. Les pics dans la distribution du burnup à 190 zones correspondent en fait aux canaux qui limitent les grandes valeurs du burnup obtenues dans le cas à 11x2 zones (G8, H8 et P8) comme le montre la figure 4.11. L'avantage de prendre 190 zones est donc de rendre plus indépendantes les puissances canal les unes des autres. Elles sont quand même liées par l'équation de diffusion, mais ne sont plus contrôlées par un autre canal de la même zone ce qui se traduit par un nombre de canaux ayant atteint la puissance limite plus grand comme le montre la figure 4.15.

Les résultats sur la droite de la figure 4.14 montrent que les distributions de burnup obtenues avec les méthodes MS et MM sont similaires à celle obtenue avec PQL. En effet, la différence maximum est de 3.4% et la moyenne de sa valeur absolue n'est que de 0.87% pour la méthode MS et de 1.12% pour la méthode MM. La différence est généralement négative où le burnup est petit et positive où le burnup est grand. Ainsi la puissance canal est augmentée au centre et diminuée en périphérie. D'un point de vue bilan neutronique, cela réduit les fuites vers l'extérieur du réacteur. L'efficacité du combustible est donc améliorée. La différence est plus importante

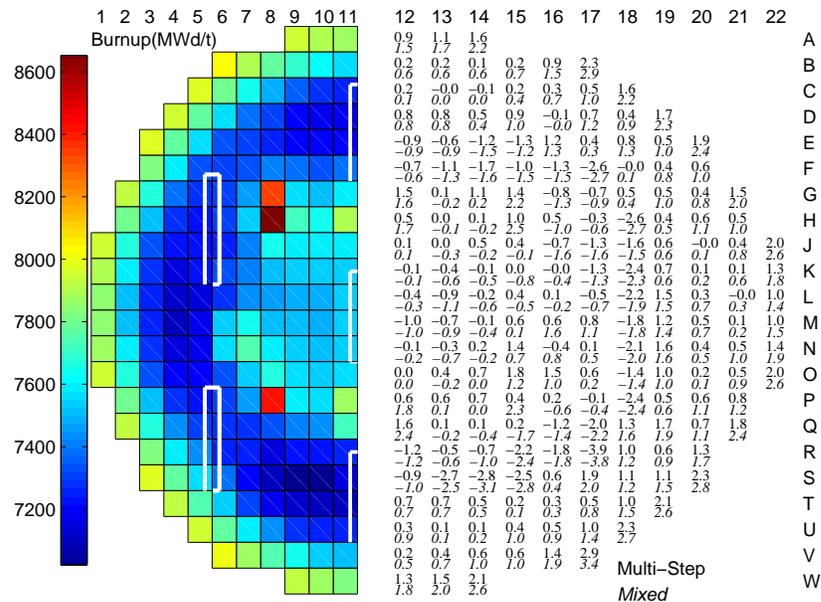


Figure 4.14 Distribution du burnup pour le cas à 190 zones.

avec la méthode MM qu'avec la méthode MS, ce qui explique la valeur plus petite du coût en combustible avec cette première.

4.6 Conclusion de l'implantation

Afin de résoudre le problème potentiel de trouver un point de départ réalisable, nous avons dans ce chapitre implanté avec succès trois méthodes alternatives: méthode multi-étapes, lagrangien augmenté et pénalisation externe. Cette dernière s'est avérée peu efficace sur un cas très simple, et n'a donc pas été testée plus en détail. La programmation qui lui est associée, n'est cependant pas vaine, car elle sera utilisée plus tard avec la méthode de recherche tabou. Les deux autres méthodes sont quant à elles assez efficaces. Les options pour la méthode LA peuvent être néanmoins difficiles à déterminer. De plus, cette dernière est relativement lente.

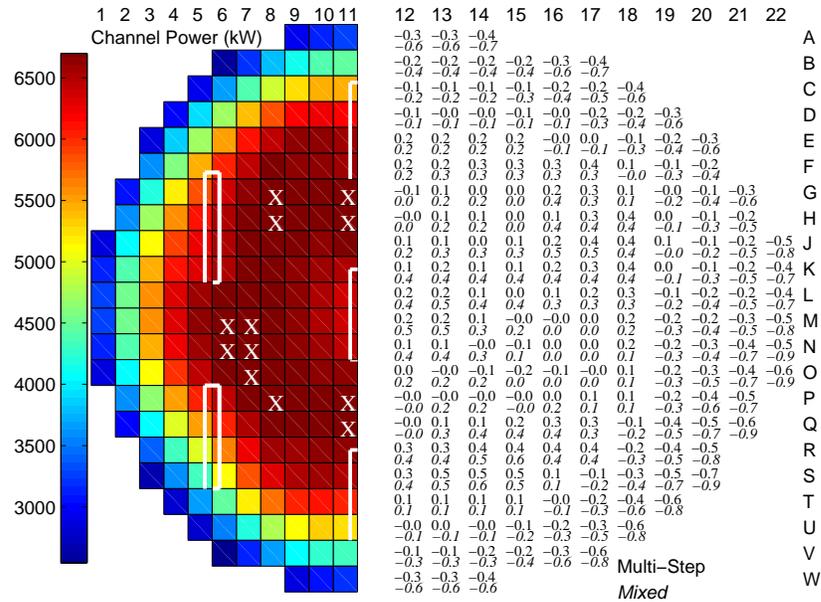


Figure 4.15 Distribution de la puissance canal pour le cas à 190 zones.

Une dépendance des résultats avec le point initial a également été notée pour toutes les méthodes sur des cas moyennement complexes. Aussi nous avons développé une dernière approche: la méthode mixte MM, qui consiste à tirer parti du meilleur des méthodes précédentes: LA, MS et PQL. Les résultats démontrent une bonne efficacité. Seul le cas le plus complexe permet de voir quelques imprécisions pour certains choix des options de départ.

Ces conclusions sont basées sur les calculs faits avec un réacteur CANDU-6. Elles ne se veulent donc pas générales. Néanmoins, certaines tendances semblent relativement généralisables. Aussi nous ne retiendrons pour la suite des simulations avec les méthodes basées sur le gradient que les approches MS et MM.

CHAPITRE 5

GESTION DU COMBUSTIBLE DU ACR-700 PAR MÉTHODES DE GRADIENTS

5.1 Description du “Advanced CANDU Reactor”

Les deux différences principales du “Advanced CANDU Reactor” (ACR-700) par rapport au CANDU-6 sont d’une part l’emploi d’eau légère comme caloporteur, et d’autre part un pas de réseau réduit de 28.5cm à 22cm ainsi qu’une augmentation de la séparation entre les tubes de force et de calandre. L’impact sur la gestion du combustible est l’utilisation de combustible légèrement enrichi pour pouvoir maintenir la réaction en chaîne. Le contrôle de la répartition de puissance se fait avec des barres en acier qui sont partiellement insérées par le haut et le bas du réacteur. Elles remplacent les barres liquides (réservoir d’eau légère à niveau variable) du CANDU-6 réparties “uniformément” à travers le réacteur.

Le design du ACR-700 est le résultat de plusieurs versions intermédiaires. Le concept original utilisant de l’eau légère a été exposé par Chan et al [34], design alors dénommé par “CANDU de nouvelle génération (CANDU-NG)”. Afin de réduire la réactivité du vide et ainsi d’augmenter la sûreté inhérente de ce réacteur, l’utilisation de poisons consommables a été envisagée dans la première version du ACR-700 [35]. Enfin, le dernier modèle du ACR-700 est similaire au précédant, si ce n’est un nombre de canaux un peu plus grand (300 au lieu de 284), et l’utilisation de combustible légèrement plus enrichi (2.5% au lieu de 2%) et une concentration de poisons consommables plus grande (15% au lieu de 4%). Le modèle de la grappe a également légèrement changé entre les deux versions.

Nous avons utilisé les deux modèles du ACR-700 dans notre étude. La version à 284 canaux, représentée à la figure 5.1, a servi pour une étude préliminaire sur la réactivité du vide. Le modèle du ACR-700 utilisé pour l'optimisation de la gestion du combustible correspond à la version à 300 canaux. Il est représenté à la figure 5.2, avec sur la gauche le modèle de la cellule et sur la droite le modèle du réacteur. Les barres de contrôle sont insérées à 50% (rectangles blancs). La limite du réflecteur tel que modélisée est donnée par la ligne extérieure sur la vue du réacteur. Nous pouvons remarquer que ce dernier design du ACR-700 n'est plus symétrique haut/bas (voir les 4 canaux de chaque côté). Sans rentrer dans le détail des résultats pour l'instant, nous pouvons dire que les deux modèles sont équivalents en ce qui concerne la réactivité du vide, à savoir une valeur bien en dessous de la limite prescrite. Lorsque nous avons réalisé l'étude préliminaire, seul le modèle à 284 canaux était disponible. Cependant, nous avons jugé plus utile par la suite de prendre le dernier modèle présenté pour l'optimisation de la gestion du combustible, afin de tirer des conclusions plus applicables.

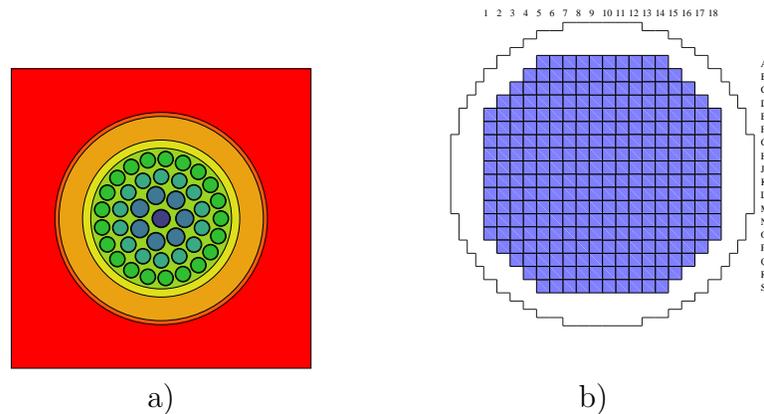


Figure 5.1 Modèle du ACR-700 à 284 canaux.

Plusieurs modifications et autres approches sont nécessaires pour tenir compte des différences entre le ACR-700 et le réacteur CANDU-6. Premièrement, l'enrichissement ainsi que la concentration en poisons initiaux sont de nouveaux paramètres par rapport à l'uranium naturel utilisé dans les CANDU-6. Afin de pouvoir prendre

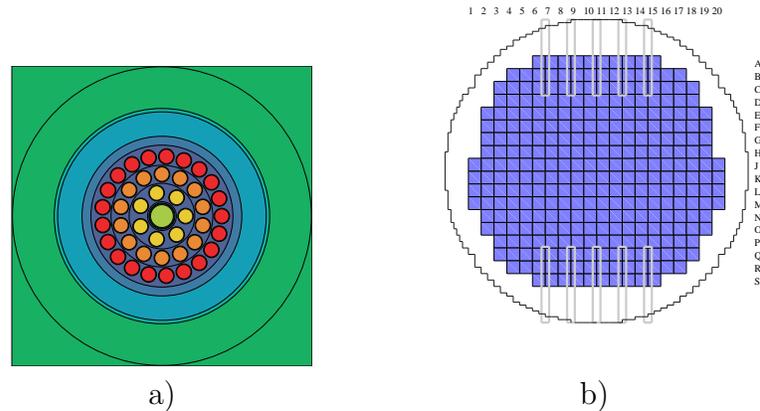


Figure 5.2 Modèle du ACR-700 à 300 canaux.

l'enrichissement et la concentration en poison comme variables de décision, une des modifications consiste à implanter dans DONJON une dépendance des sections efficaces par rapport à ces paramètres. Nous reviendrons sur ce point dans les deux dernières sections de ce chapitre.

Deuxièmement, une contrainte sur la réactivité du vide doit être modélisée. Cette contrainte ressemble beaucoup à la contrainte sur la réactivité des barres de compensation pour les réacteurs CANDU-6 [9, 15]. La définition théorique de cette contrainte ainsi que la méthode choisie pour son calcul seront présentées ultérieurement.

Enfin, la dernière modification consiste plus en un changement de mentalité. Pour les CANDU-6, historiquement la répartition des zones de burnup était radiale, le design du cœur prédisposant à une telle approche. Or comme nous l'avons vu à la section 4.4.3, la légère dissymétrie du réacteur CANDU-6 a une nette influence sur les résultats d'optimisation. Dans le cas du ACR-700, les barres de contrôles sont insérées par le haut et le bas. Ce design a pour conséquence de compliquer la répartition des zones de burnup. En effet, dans leur travaux [1, 35], Boubcher et al ont utilisé 5 zones dont la répartition est donnée à la figure 5.3 (modèle du

quart du cœur du ACR-700 à 284 canaux). Selon l'expérience que nous avons déjà acquise, cela s'avère en fait le minimum pour respecter la contrainte sur la puissance canal maximum. La conséquence sur l'optimisation de la gestion du combustible du ACR-700 est double: nous ne possédons pas de configuration simplifiée reflétant le design de ce réacteur pour vérifier les modifications faites d'une part. D'autre part, l'interprétation des résultats "optimaux" est plus difficile.

L'implantation des modifications mentionnées plus haut a été faite dans l'ordre suivant. Premièrement, nous avons représenté le nouveau type de contrainte sur la réactivité du vide. Sa définition théorique a été détaillée, et différentes approches approximatives ont été testées. Des calculs d'optimisation de la distribution du burnup ont alors été conduit pour un enrichissement fixe. Enfin, l'enrichissement et la concentration en poison ont été retenus comme variable de décision. L'implantation de la dépendance des sections efficaces en fonction de ces paramètres est présentée. Un réacteur fictif simple a alors été utilisé pour tester l'implantation. L'application au ACR-700 est ensuite présentée.

5.2 Réactivité du vide

Une étude préliminaire sur la réactivité du vide est réalisée pour le ACR-700. Nous avons utilisé le modèle à 284 canaux (figure 5.1). La réactivité du vide est définie dans son cadre général et aussi pour l'approche moyennée dans le temps. Des approches simplifiées sont également définies et testées.

5	3	5	5	5				
3	3	3	5	5	5			
2	2	2	2	3	3	5		
2	2	2	2	2	3	3	4	
1	1	1	2	2	2	3	3	4
1	1	1	1	2	2	2	3	4
1	1	1	1	2	2	2	3	4
1	1	1	1	1	1	2	3	3
1	1	1	1	1	1	2	2	3

Figure 5.3 Exemple de répartition de zones de burnup de sortie [1], symétrie quart de réacteur

5.2.1 Définition

La réactivité du vide représente l'impact d'une perte du caloporteur sur la réaction en chaîne et donc sur la réactivité du réacteur. Une des particularités du réacteur ACR-700 par rapport aux autres CANDU provient de l'utilisation d'eau légère comme caloporteur et de combustible enrichi contenant un poison consommable. Le but du poison est d'assurer une réactivité du vide négative pour respecter le concept de sûreté inhérente (baisse de puissance en cas de perte de caloporteur).

La réactivité du vide est alors donnée par la différence entre les valeurs propres du cœur vide de caloporteur et dans son état normal refroidi:

$$\rho_V = \frac{1}{k_{eff}} - \frac{1}{k_{eff,V}} = \lambda - \lambda_V \quad (5.1)$$

Les propriétés pour le réacteur vide sont obtenues avec DRAGON en perturbant la cellule unitaire de référence par l'élimination du caloporteur à chaque étape du calcul d'évolution.

Une réactivité du vide $\rho_{V,lim}$ négative est recherchée. Cela se traduit comme une nouvelle contrainte à rajouter au problème d'optimisation original (2.17). Comme nous l'avons vu précédemment pour la contrainte sur la réactivité des barres de compensation, une telle contrainte peut être prise en compte en calculant le flux neutronique dans le réacteur dans deux états: cœur refroidi et cœur vidé de son caloporteur (équation (5.1)). Le nouveau problème d'optimisation de la gestion du

combustible du ACR-700 s'écrit alors :

$$\min_{\vec{X}} F_C(\vec{X}) \text{ avec } \begin{cases} k_{eff}(\vec{X}) = k_{ref} \\ q_j(\vec{X}) \leq f_j & j \in (1, NCS) \\ \rho_V \leq \rho_{V,lim} \\ X_n^{INF} \leq X_n \leq X_n^{SUP} \end{cases} \quad (5.2)$$

La notation plus générale définie à l'équation (2.18) peut être conservée, à condition de bien noter que dans ce cas g_j représente à la fois les contraintes d'inégalités sur la puissance maximum et celle sur la réactivité du vide. Il en va de même pour la définition des fonctions du lagrangien augmenté et de la pénalisation externe.

5.2.2 Calcul moyenné dans le temps

Par définition, la réactivité du vide pour un réacteur représente la variation de la criticité due à une vidange totale de tout le caloporteur dans le cœur. Une notion d'état instantané est donc intrinsèquement reliée à cette grandeur, puisque la vidange du cœur est supposée avoir lieu instantanément pour une configuration précise, c'est à dire une distribution du burnup et une position des contrôleurs donnée dans le réacteur. Or, la plupart des calculs de design de réacteur et de gestion du combustible sont effectués avec le modèle moyenné dans le temps (TA). Il serait donc utile de pouvoir évaluer directement la réactivité du vide avec ce même modèle. En effet, passer d'un modèle TA à un modèle instantané requiert de connaître l'âge de chaque canal, c'est à dire la fraction du temps écoulée entre les rechargements à l'équilibre. La distribution des âges des canaux est appelé le patron d'âge. Le passage d'un modèle TA à un modèle instantané correspond à un choix du patron d'âge. Or ce choix est arbitraire, aucune distribution instantanée n'est plus valide qu'une autre pour représenter le réacteur. Une approche avec le modèle

TA permettrait donc de contourner ce choix arbitraire de l'approche instantanée. Nous allons donc comparer les deux modèles pour le calcul de la réactivité du vide.

Premièrement, une réactivité du vide équivalente $\rho_{V,TA}$ est calculée avec les propriétés moyennées dans le temps du modèle TA. Ainsi, la première étape est de résoudre l'équation de diffusion avec le modèle TA pour un cœur normal et d'obtenir ainsi les bornes d'intégrations pour le calcul des sections efficaces en TA. Ensuite, le calcul des sections efficaces est repris avec les propriétés pour un cœur vidé de son caloporteur pour les mêmes bornes d'intégration. La réactivité du vide en TA est alors donné par:

$$\rho_{V,TA} = \lambda_{TA} - \lambda_{V,TA} \quad (5.3)$$

Deuxièmement, plusieurs patrons d'âge ont été testés pour le calcul de la réactivité du vide avec le modèle instantané. Le premier patron d'âge utilisé (cas1) a été obtenu en utilisant la séquence de bloc définie par Rozon et Shen [19]. Cette technique permet d'avoir une répartition homogène de l'âge des canaux. Elle a pour but d'éliminer la possibilité d'obtenir un tilt de la répartition de flux, et d'avoir deux canaux adjacents rechargés à un intervalle de temps réduit. Les cas 2 et 3 sont en fait similaires au cas 1 avec un décalage dans le cycle pour l'affectation de l'âge des canaux de 95 et 190 (sur 284) respectivement. Enfin les cas 4 à 6 sont des répartitions de patron d'âge générées aléatoirement. Dans tous les cas, la réactivité du vide est donnée par une formule similaire à l'équation 5.3:

$$\rho_{V,Ins} = \lambda_{Ins} - \lambda_{V,Ins} \quad (5.4)$$

Les résultats obtenus présentés au tableau 5.1 correspondent à la réactivité du vide pour le modèle ACR-700 à 284 canaux avec 2% d'enrichissement sauf pour le crayon central composé d'uranium naturel avec 4% de Dy [35], mais sans contrôleurs. Pour

Tableau 5.1 Réactivité du vide pour différents modes de calculs

Méthode	ρ_V (mk)
TA	-20.451
Ins cas 1	-20.530
Ins cas 2	-20.522
Ins cas 3	-20.522
Ins cas 4	-20.479
Ins cas 5	-20.721
Ins cas 6	-20.741

le modèle TA de référence, 2 zones radiales concentriques de burnup moyen de sortie ont été utilisées ($28429.4MWj/t$ au centre et $25017.8MWj/t$ en périphérie), les contrôleurs n'étant pas présents, la puissance canal maximum n'est pas vérifiée. La réactivité du vide calculée avec le modèle TA est proche de celle calculée avec le modèle instantané, l'écart est inférieur à 0.6% pour tous les cas sauf 5 et 6 avec 1.5%. Les résultats instantanés avec un patron d'âge basé sur la séquence de blocs sont plus constants (variation de 0.5%) que ceux basés sur un patron d'âge aléatoire (variation de 2%). Ceci s'explique en fait par la définition même de la séquence de blocs, soit une répartition "homogène" de l'âge des canaux.

Même si le nombre de cas testés est faible, la constance dans les résultats permet d'affirmer que le modèle TA peut être utilisé pour l'évaluation de la réactivité du vide. De plus, la génération aléatoire du patron d'âge est peu recommandée pour le calcul de la réactivité du vide par rapport à la génération par blocs.

5.2.3 Approximation du calcul

5.2.3.1 Théorie

La définition de la réactivité du vide implique un calcul du flux neutronique après vidange du cœur pour obtenir λ_V , en plus du calcul standard pour le cas refroidi. Or un tel calcul peut être coûteux en temps de calcul. Afin d'accélérer l'optimisation, nous avons étudié des approches de perturbation pour le calcul de cette nouvelle contrainte et de son gradient.

La théorie des perturbations n'est théoriquement pas valide pour des perturbations trop fortes (comme l'extraction d'un banc de barres de compensation). [21] À priori, la vidange du cœur est également une forte perturbation. En effet dans le cas du ACR-700, la vidange du caloporteur eau légère a un impact significatif sur la distribution du flux, en particulier sur le spectre des neutrons. Mais nous ne pouvons pas cependant savoir analytiquement la limite des formules de perturbation. Aussi, nous proposons plusieurs méthodes à tester pour le calcul de la réactivité du vide et de son gradient par une méthode TPG.

On peut estimer la réactivité du vide de trois façons différentes :

1. Effectuer un nouveau calcul de flux (perturbé) et faire la différence des valeurs propres. C'est l'approche directe décrite précédemment. Elle est la plus précise si la perturbation est significative.
2. Utiliser une formule de perturbation de premier ordre, ce qui est moins précis mais ne nécessite aucun nouveau calcul de diffusion.
3. Utiliser une formule de perturbation d'ordre élevé, tel que proposée par Favorite et Stacey [36]. Cette approche doit être plus précise, puisqu'elle suggère

une correction pour tenir compte des variations de la forme (spectre) du flux dans le calcul de la variation de la valeur propre. L'intérêt d'utiliser la formule de perturbation généralisée est d'avoir à calculer deux adjoints généralisés plutôt que deux flux. Ceci devrait être moins dispendieux puisqu'il n'y a pas d'itérations sur la valeur propre pour le calcul des adjoints généralisés.

a) Approche directe

En pratique, la réactivité du vide est donnée par l'équation (5.1), les valeurs propres nécessaires étant obtenues lors du calcul de flux pour le cas standard et perturbé. Les composantes du gradient de la réactivité du vide s'obtiennent en dérivant (5.1):

$$\frac{\partial \rho_V^{(exact)}}{\partial X_n} = \frac{\partial \lambda}{\partial X_n} - \frac{\partial \lambda_V}{\partial X_n} \quad (5.5)$$

et avec la formule de perturbation (2.26) pour évaluer chaque terme.

En utilisant le quotient de Rayleigh pour exprimer λ_V , la réactivité du vide peut également être écrite sous la forme suivante:

$$\begin{aligned} \rho_V^{(exact)} &= \lambda - \frac{\langle \phi_V^*, \mathbf{A}_V \phi_V \rangle}{\langle \phi_V^*, \mathbf{F}_V \phi_V \rangle} \\ &= \frac{\langle \phi_V^*, (\lambda \mathbf{F}_V - \mathbf{A}_V) \phi_V \rangle}{\langle \phi_V^*, \mathbf{F}_V \phi_V \rangle} \end{aligned} \quad (5.6)$$

Cette formulation n'est pas utilisée en pratique. Cependant, elle est exacte, et elle nous sert de référence pour les approches simplifiées qui suivent et qui ont une formulation similaire.

b) Perturbation de premier ordre

Nous pouvons évaluer la réactivité du vide à l'aide d'une formule de perturbation de premier ordre. Ainsi en appliquant l'équation (2.27) à une perturbation absolue ($-\Delta\lambda_V =$ la vidange du caloporteur) plutôt qu'une perturbation relative ($\Delta\lambda/\Delta X_n$), nous obtenons une première estimation de la réactivité du vide:

$$\rho_V^{(1)} = -\Delta\lambda_V^{(1)} \approx \frac{\langle \phi^*, (\lambda \mathbf{F}_V - \mathbf{A}_V) \phi \rangle}{\langle \phi^*, \mathbf{F} \phi \rangle} \quad (5.7)$$

En comparant avec la formule exacte (5.6), on constate que la formule de perturbation de premier ordre utilise le flux ϕ et son adjoint ϕ^* non perturbés au lieu de ϕ_V et ϕ_V^* , ainsi que l'opérateur de production non perturbé \mathbf{F} au lieu de \mathbf{F}_V au dénominateur.

Les composantes du gradient s'obtiennent alors en dérivant (5.7):

$$\frac{\partial \rho_V^{(1)}}{\partial X_n} \approx \frac{\langle \phi^*, \left(\frac{\partial \lambda}{\partial X_n} \mathbf{F}_V + \lambda \frac{\partial \mathbf{F}_V}{\partial X_n} - \frac{\partial \mathbf{A}_V}{\partial X_n} - \rho_V^{(1)} \frac{\partial \mathbf{F}}{\partial X_n} \right) \phi \rangle}{\langle \phi^*, \mathbf{F} \phi \rangle} \quad (5.8)$$

démonstration:

$$\begin{aligned} \frac{\partial \rho_V^{(1)}}{\partial X_n} &= \frac{\left\langle \frac{\partial \phi^*}{\partial X_n}, \overbrace{(\lambda \mathbf{F}_V - \mathbf{A}_V - \rho_V^{(1)} \mathbf{F})}^{(a)} \phi \right\rangle + \left\langle \phi^*, \left(\frac{\partial (\lambda \mathbf{F}_V - \mathbf{A}_V)}{\partial X_n} - \rho_V^{(1)} \frac{\partial \mathbf{F}}{\partial X_n} \right) \phi \right\rangle}{\langle \phi^*, \mathbf{F} \phi \rangle} \\ &+ \frac{\left\langle \phi^*, \overbrace{(\lambda \mathbf{F}_V - \mathbf{A}_V - \rho_V^{(1)} \mathbf{F})}^{(b)} \frac{\partial \phi}{\partial X_n} \right\rangle}{\langle \phi^*, \mathbf{F} \phi \rangle} \quad (5.8a) \end{aligned}$$

Or en substituant ϕ_V par $\phi + \Delta\phi_V$ dans l'équation de diffusion pour le réacteur

vide et en négligeant les termes d'ordre élevé, nous trouvons :

$$\begin{aligned}
\mathbf{A}_V \phi_V &= \lambda_V \mathbf{F}_V \phi_V \\
\Leftrightarrow (\mathbf{A} + \Delta \mathbf{A}_V)(\phi + \Delta \phi_V) &= (\lambda + \Delta \lambda_V^{exact})(\mathbf{F} + \Delta \mathbf{F}_V)(\phi + \Delta \phi_V) \\
\Leftrightarrow (\mathbf{A} - \lambda \mathbf{F}) \Delta \phi_V &= -(\mathbf{A} + \Delta \mathbf{A}_V - \lambda(\mathbf{F} + \Delta \mathbf{F}_V) - \Delta \lambda_V \mathbf{F}) \phi \\
\Leftrightarrow (\mathbf{A} - \lambda \mathbf{F}) \Delta \phi_V &= -(\mathbf{A}_V - \lambda \mathbf{F}_V + \rho_V^{exact} \mathbf{F}) \phi \tag{5.9}
\end{aligned}$$

Avec l'approximation des perturbations du premier ordre, la variation du flux $\Delta \phi_V$ est négligée, et ρ_V^{exact} est approximée par $\rho_V^{(1)}$. Nous obtenons donc que le facteur (a) de l'équation (5.8a) est nul. La même approche peut être suivie avec l'équation adjointe pour le réacteur vidé, ce qui donne également zéro pour le facteur (b).

c) Perturbation généralisée

La formule de premier ordre ne tient pas compte de l'effet des changements dans la forme (et dans le spectre) du flux lorsque la perturbation du vide est appliquée. Nous pouvons corriger cette formule en calculant explicitement $\Delta \phi_V$ par une formule de perturbation généralisée. En utilisant l'équation perturbée (5.9), $\Delta \phi_V$ peut être approximé par:

$$(\mathbf{A} - \lambda \mathbf{F}) \Delta \phi_V = -(\mathbf{A}_V - \lambda \mathbf{F}_V + \rho_V^{(1)} \mathbf{F}) \phi \tag{5.10}$$

Nous observons que (5.10) est une équation singulière avec source (équation adjointe généralisée de type explicite). Une solution particulière n'existera que si le terme

de source est orthogonal au flux adjoint, i.e. si :

$$\langle \phi^*, (\mathbf{A}_V - \lambda \mathbf{F}_V + \rho_V^{(1)} \mathbf{F}) \phi \rangle = 0$$

Ceci est facilement vérifié avec la formule de premier ordre (5.7) utilisée pour évaluer $\Delta \lambda_V^{(1)}$ dans (5.9).

En substituant $\phi_V = \phi + \Delta \phi_V$ et $\phi_V^* = \phi^* + \Delta \phi_V^*$ dans la formule exacte, nous obtenons enfin un estimé de la réactivité qui tient compte de la variation du flux :

$$\rho_V^{(2)} = \frac{\langle (\phi^* + \Delta \phi_V^*), (\lambda \mathbf{F}_V - \mathbf{A}_V) (\phi + \Delta \phi_V) \rangle}{\langle (\phi^* + \Delta \phi_V^*), \mathbf{F}_V (\phi + \Delta \phi_V) \rangle} \quad (5.11)$$

où $\Delta \phi_V^*$ est donné par l'équation adjointe correspondante à l'équation (5.10).

Les composantes du gradient pourront donc s'écrire :

$$\frac{\partial \rho_V^{(2)}}{\partial X_n} \approx \frac{\langle (\phi^* + \Delta \phi_V^*), \left(\frac{\partial \lambda}{\partial X_n} \mathbf{F}_V + \lambda \frac{\partial \mathbf{F}_V}{\partial X_n} - \frac{\partial \mathbf{A}_V}{\partial X_n} - \rho_V^{(2)} \frac{\partial \mathbf{F}_V}{\partial X_n} \right) (\phi + \Delta \phi_V) \rangle}{\langle (\phi^* + \Delta \phi_V^*), \mathbf{F}_V (\phi + \Delta \phi_V) \rangle} \quad (5.12)$$

démonstration:

$$\begin{aligned} \frac{\partial \rho_V^{(2)}}{\partial X_n} &= \frac{\overbrace{\left\langle \frac{\partial \phi_V^*}{\partial X_n}, (\lambda \mathbf{F}_V - \mathbf{A}_V - \rho_V^{(2)} \mathbf{F}_V) \phi_V \right\rangle}^{(a)} + \left\langle \phi_V^*, \left(\frac{\partial (\lambda \mathbf{F}_V - \mathbf{A}_V)}{\partial X_n} - \rho_V^{(2)} \frac{\partial \mathbf{F}_V}{\partial X_n} \right) \phi_V \right\rangle}{\langle \phi_V^*, \mathbf{F}_V \phi_V \rangle} \\ &+ \frac{\overbrace{\left\langle \phi_V^*, (\lambda \mathbf{F}_V - \mathbf{A}_V - \rho_V^{(2)} \mathbf{F}_V) \frac{\partial \phi_V}{\partial X_n} \right\rangle}^{(b)}}{\langle \phi_V^*, \mathbf{F}_V \phi_V \rangle} \end{aligned} \quad (5.12a)$$

Or,

$$\begin{aligned}
(\lambda \mathbf{F}_V - \mathbf{A}_V - \rho_V^{(2)} \mathbf{F}_V) \phi_V &= (\lambda_V \mathbf{F}_V - \Delta \lambda_V^{exact} \mathbf{F}_V - \mathbf{A}_V - \rho_V^{(2)} \mathbf{F}_V) \phi_V \\
&= (\rho_V^{exact} - \rho_V^{(2)}) \mathbf{F}_V \phi_V \\
&= 0
\end{aligned}$$

Car l'approximation de la théorie des perturbations généralisées que nous avons pris donne $\rho_V^{exact} \approx \rho_V^{(2)}$. Les facteurs (a) et (b) de l'équation (5.12a) sont donc nuls.

d) Implantation

Dans la section (3.2), nous avons vu que les opérateurs \mathbf{A} , \mathbf{F} , \mathbf{A}_V et \mathbf{F}_V (notés \mathbf{M} en général) sont formés par des combinaisons linéaires des sections efficaces et des coefficients de diffusions. Le calcul de leur dérivée ($\partial \mathbf{M} / \partial X_n$) peut donc être généralisé par la formule suivante:

$$\begin{aligned}
\partial \mathbf{M}(\Sigma) / \partial X_n &= \mathbf{M} \left(\frac{\partial \Sigma}{\partial X_n} \right) = \left(\mathbf{M} \left(\Sigma + \Delta X_n \frac{\partial \Sigma}{\partial X_n} \right) - \mathbf{M}(\Sigma) \right) / \Delta X_n \\
&= (\mathbf{M}^P - \mathbf{M}) / \Delta X_n
\end{aligned} \tag{5.13}$$

Dans l'équation (2.27) les dérivées des matrices systèmes $\partial \mathbf{M} / \partial X_n$ sont alors remplacées par $\mathbf{M}^P / \Delta X_n$, car avec l'équation de diffusion (1.2) les termes $\mathbf{M} / \Delta X_n$ s'annulent. La formulation exacte de la dérivée de la réactivité du vide peut donc directement utiliser ce résultat pour simplifier son évaluation. La même transformation peut être appliquée aux équations (5.8) et (5.12) en utilisant les

Tableau 5.2 Calcul de la réactivité du vide par quatre méthodes

Méthode	exacte	1 ^{er} ordre	TPG (ϕ_V^*)	TPG (ϕ_o^*)
ρ_V (mk)	-20.451	-1.652	-18.148	-14.333
$\partial\rho_V/\partial B_{e,1}$	-1.435E-07	4.537E-08	-1.631E-07	1.342E-04
$\partial\rho_V/\partial B_{e,2}$	-3.868E-07	10.234E-08	-3.136E-07	1.524E-04

équations (5.7) et (5.11) respectivement. Nous avons en effet vérifié que les termes du type $\mathbf{M}/\Delta X_n$ s'annulent. Ces transformations permettent de simplifier le problème en utilisant des matrices perturbées plutôt que des dérivées de matrices systèmes. Nous tenons cependant à insister sur le fait que ces matrices perturbées sont obtenues avec la dérivée des propriétés et non par une perturbation directe de X_n .

5.2.3.2 Résultats

Pour calculer la réactivité du vide, nous avons simulé un réacteur ACR-700 (284 canaux) sans contrôleur, avec deux zones radiales de combustion ($B_{\text{centre}} = 28429 \text{ MWj/t}$ et $B_{\text{périphérie}} = 25017 \text{ MWj/t}$). Le burnup moyen de sortie de chaque zone est choisie pour que $k_{eff} = 1$ et que le maximum de la puissance canal soit minimum. Le tableau 5.2 regroupe les résultats des calculs de la réactivité du vide et de ses gradients par rapport au deux zones de burnup, et ce pour quatre méthodes. La méthode exacte sert de référence. Que ce soit pour ρ_V ou ses dérivées partielles la méthode de perturbation du premier ordre donne des résultats trop grossiers. L'utilisation de la théorie de perturbation généralisée (TPG) donne de meilleurs résultats pour ρ_V . L'utilisation de ϕ_V^* comme pondération donne cependant de meilleurs résultats et est en fait indispensable pour le calcul des dérivées. Ceci peut s'expliquer par le fait qu'avec ϕ_o^* , les facteurs (a) et (b) de l'équation (5.12a) ne s'annulent pas. Cette quatrième méthode ne sera donc pas retenue par la suite.

L'effort de calcul pour la réactivité du vide sont donnés au tableau 5.3. Puisque toutes les méthodes doivent calculer les matrices \mathbf{M}_V et \mathbf{F}_V , les temps du tableau 5.3 n'incluent pas cette partie du calcul. Le module de DONJON qui calcule le flux adjoint (FLUD:) recalcule la valeur propre, et donc le coût de calcul est le même que celui d'un calcul de flux. La méthode du premier ordre est en effet beaucoup plus rapide que la méthode exacte puisque aucun flux ou adjoint généralisé n'est nécessaire. La méthode de perturbation généralisée nécessite deux calculs de sources, deux calculs d'adjoints généralisés et l'addition de ces derniers avec le flux et son adjoint. Le temps de calcul est plus grand et la perte de précision pour ρ_V est de l'ordre de 10%.

Afin de vérifier l'influence du signe de la réactivité de la perturbation, nous avons également calculé la réactivité inverse du vide, c'est à dire perturber un réacteur vide. Les résultats sont présentés au tableau 5.4. Les conclusions quant à la précision des méthodes de perturbation de premier ordre et généralisée restent les mêmes. Les écarts sont cependant plus grands (environ 30%) entre l'estimation de la TPG et la méthode exacte pour une perturbation positive que pour une perturbation négative (écart d'environ 10%). Cette différence entre perturbation positive ou négative a également été remarquée par Favorite et Stacey [36]. Par contre dans l'exemple qu'ils ont utilisé, la différence est plus grande pour une perturbation négative. Il est cependant difficile d'expliquer la divergence des résultats. En effet, d'une part les géométries sont très différentes: modèle ACR-700 en 3D vs. petit réacteur à eau légère en 1D ('slab'). D'autre part, le type perturbation est très différent. Dans le cas du réacteur 1D, la perturbation correspond à une variation de la section efficace d'absorption dans le groupe thermique pour le quart gauche du réacteur. Une augmentation de cette section efficace entraîne directement une baisse de la réactivité. Dans le cas du ACR-700, la vidange du caloporteur pour tous les canaux fait varier toutes les propriétés. Aussi, même si la section efficace

Tableau 5.3 Effort de calcul pour la réactivité du vide par trois méthodes

Méthode	exacte	1 ^{er} ordre	TPG (ϕ_V^*)
Temps de calcul de ρ_V (s)	0.54	0.02	0.74
Calculs de flux	2+2	2	2
Calculs d'adjoints généralisés	$n_{cst,PC_{max}} + 1$	$n_{cst,PC_{max}} + 1$	$n_{cst,PC_{max}} + 1 + 2$

Tableau 5.4 Réactivité inverse du vide par trois méthodes

Méthode	exacte	1 ^{er} ordre	TPG (ϕ_V^*)
ρ_{V+} (mk)	20.485	46.983	26.975

d'absorption diminue, la réactivité du réacteur suite à la vidange diminue. Dans le cas du ACR-700, non seulement la forme de flux change, mais le rapport flux thermique / flux rapide varie aussi beaucoup ce qui a une importance plus grande sur les fuites. L'effet observé est donc inverse à celui auquel nous pourrions nous attendre juste en regardant la section efficace d'absorption.

5.2.3.3 Conclusions

L'approche la plus précise pour calculer le gradient de la réactivité du vide exige de calculer le flux perturbé. L'approche de perturbation classique, qui évite ce calcul additionnel, est de loin trop imprécise, et est donc rejetée. Par ailleurs, la méthode suggérée par Favorite et Stacey, même si elle tient compte des variations de la forme (spectre) du flux dans le calcul de la variation de la valeur propre, se rapproche plus de la valeur exacte, mais elle n'est pas assez précise. De plus, l'intérêt d'utiliser la formule de perturbation généralisée est d'avoir à calculer deux adjoints généralisés $\Delta\phi_V$ et $\Delta\phi_V^*$ plutôt que deux flux, ce qui devrait être plus rapide puisque la valeur propre est connue. Cela ne s'est cependant pas traduit par un gain de temps. Aucune des approximations ne peut être retenue pour le calcul

de la réactivité du vide dans le ACR-700.

5.3 Optimisation en burnup

Pour l'optimisation de la gestion du combustible en burnup seulement, les propriétés du combustible ont été générées avec DRAGON, en utilisant les compositions initiales du combustible fournies par AECL. Elles correspondent à du combustible enrichi à 2.5% environ, sauf dans le crayon du centre qui contient un mélange d'uranium naturel et de dysprosium à 15%. Les calculs d'optimisation de la gestion du combustible ont été fait avec le modèle de cœur à 300 canaux.

5.3.1 Influence des paramètres d'optimisation

Parmi toutes les approches envisagées pour l'optimisation de la gestion du combustible dans les réacteurs CANDU-6, deux ont été retenues pour le ACR-700: la méthode multi-étapes (MS) et la méthode mixte (MM). Comme nous l'avons remarqué précédemment, les résultats peuvent dépendre du choix du point initial. Aussi, nous avons utilisé deux distributions initiales du burnup: l'une non uniforme basée sur l'expérience déjà acquise par AECL et l'autre avec une estimation grossière du burnup (identique dans tous les canaux), appelées "AECL" et "flat" respectivement. L'optimisation a également été conduite pour 7 et 150 zones de burnup. En prenant en compte la symétrie droite/gauche du réacteur, le cas à 150 zones correspond à deux canaux par zone (1 pour chaque moitié du cœur). Pour le cas simplifié à 7 zones, le choix des zones a été fait à partir de la distribution détaillée du burnup fournie par AECL, pour laquelle les canaux avec un burnup de sortie similaire ont été regroupés. La distribution des zones est illustrée à la figure 5.4.

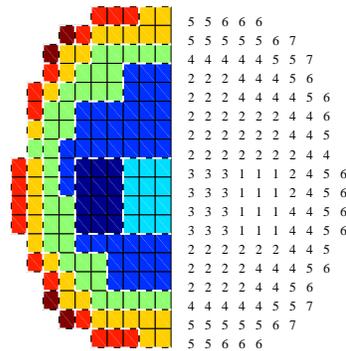


Figure 5.4 Distribution des 7 zones de burnup pour le ACR-700.

Les résultats de l'optimisation sont rassemblés dans le tableau 5.5. Les paramètres d'optimisation sont donnés par n_z le nombre de zones, P.I. le point initial et *meth* la méthode. Pour la méthode MM, les poids des contraintes w_k pour la criticité du réacteur et w_q pour la puissance canal maximum sont respectivement de 50000 et 1000. En ce qui concerne les résultats, F_C représente la valeur adimensionnelle du coût en combustible, B_{min} et B_{max} correspondent aux minimum et maximum du burnup moyen de sortie sur tout le cœur. L'effet des différentes options a été étudié.

Tableau 5.5 Résultats pour différentes distributions initiales du burnup et différentes méthodes d'optimisation .

n_z	P.I.	<i>meth</i>	F_C	B_{min}	B_{max}
7	AECL	MS	4.4756	15938	27258
7	flat	MS	4.4751	15805	27336
7	AECL	MM	4.4756	15939	27258
7	flat	MM	4.4754	15779	27330
150	AECL	MS	4.3357	16926	27565
150	flat	MS	NA	NA	NA
150	AECL	MM	4.3400	16708	27564
150	flat	MM	4.3058	19466	27769

Dans le cas le plus simple (7 zones), ni la distribution initiale du burnup ni la

méthode choisie ne semblent avoir d'influence sur l'optimum trouvé, en effet les valeurs de la fonction coût obtenues sont toutes comprises dans une fourchette de $6.10^{-4} \approx 0.015\%$, ce qui est proche du critère sur la convergence axiale du flux (10^{-4}). Les distributions de burnup et de puissance canal obtenues avec la méthode MM sont donnés à la figure 5.5 pour les deux distributions initiales de burnup (cas AECL (a) et cas flat (b)). Pour chaque cas, les résultats se présentent sous la forme du cœur du réacteur séparé en deux, la partie gauche représente la distribution de puissance canal et la droite celle du burnup de sortie. Les traits larges en pointillés blancs représentent la limite des zones de burnup, les rectangles gris les barres de contrôles et enfin la fine ligne extérieure la limite externe du modérateur. Les résultats de la colonne de gauche montrent qu'il n'y a pas de différence significative obtenue pour le cas simple. Le seul écart notable est obtenu pour la zone #7. Cette zone est très petite (8 canaux seulement) par rapport aux autres. Or comme nous l'avons remarqué au chapitre précédent, les grandes zones voient leur burnup optimisé préférentiellement à celui des petites zones. Autrement dit, le burnup dans les petites zones a tendance à rester proche de sa valeur initiale. La différence observée de $3GWj/t$ provient donc du fait que le burnup initial dans la zone #7 est de $10GWj/t$ dans le cas AECL et de $23GWj/t$ dans le cas flat.

Afin d'éviter le bias mathématique inhérent à la définition de la fonction objectif, toutes les zones doivent avoir le même nombre de canaux. C'est un des buts du cas à 150 zones. Pour ce cas plus complexe, cependant, certaines limites de la méthode MS ont été atteintes pour le cas 'flat'. En effet, pour la minimisation de la puissance canal, la contrainte prescrite n'a pas été atteinte même après 200 itérations, et l'optimisation a donc été arrêtée avant convergence, d'où la marque NA dans le tableau 5.5.

Pour la MM, nous observons que les résultats dépendent du point de départ. En effet, la fonction objectif est approximativement 0.8% plus petite pour le cas 'flat'

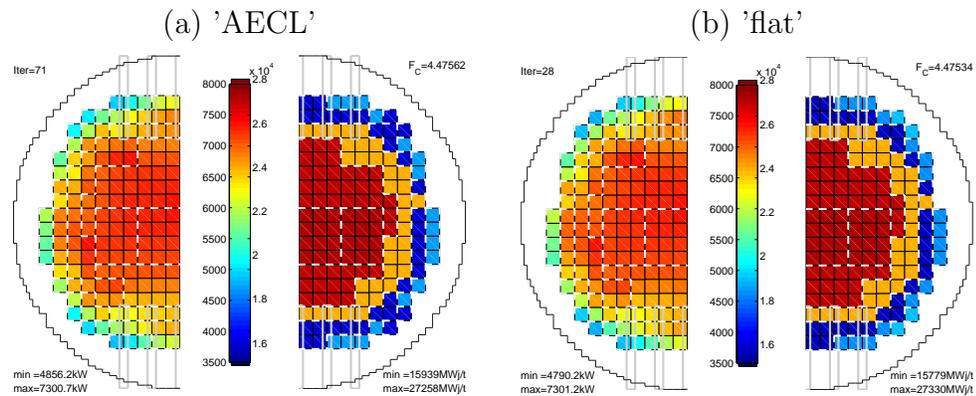


Figure 5.5 Distributions de la puissance canal et du burnup obtenues avec la MM pour 7 zones.

que pour le cas 'AECL'. Les distributions de la puissance canal et du burnup obtenues avec la MM pour 150 zones sont présentées à la figure 5.6, avec le même formalisme que pour 7 zones à la figure 5.5. Nous pouvons observer une nette différence du burnup obtenu dans toute la périphérie. Nous avons déjà noté ce phénomène pour l'optimisation de la gestion du combustible du réacteur CANDU-6 dans le chapitre précédent. [33] En fait, de manière similaire avec le cas à 7 zones, le burnup en périphérie est initialement d'environ $10 MWj/t$ pour le cas AECL, alors qu'il est de $23 GWj/t$ pour le cas flat. Ainsi, le burnup augmente pour atteindre sa valeur optimale d'environ $17 GWj/t$ (figure 5.6a, pour le cas AECL) et diminue jusqu'à environ $19.5 GWj/t$ (figure 5.6b, pour le cas flat). Ces résultats démontrent clairement que la méthode MM est très sensible au point de départ pour le cas complexe.

Avec la fonction coût F_C comme critère, nous pouvons conclure que l'optimum n'est pas aussi bon dans le cas "AECL" que dans le cas "flat" pour le cas complexe. Ainsi, soit un minimum local est obtenu dans le cas où la distribution 'AECL' est utilisée, ou alors le critère de convergence est trop large. Il est quasi impossible de prouver

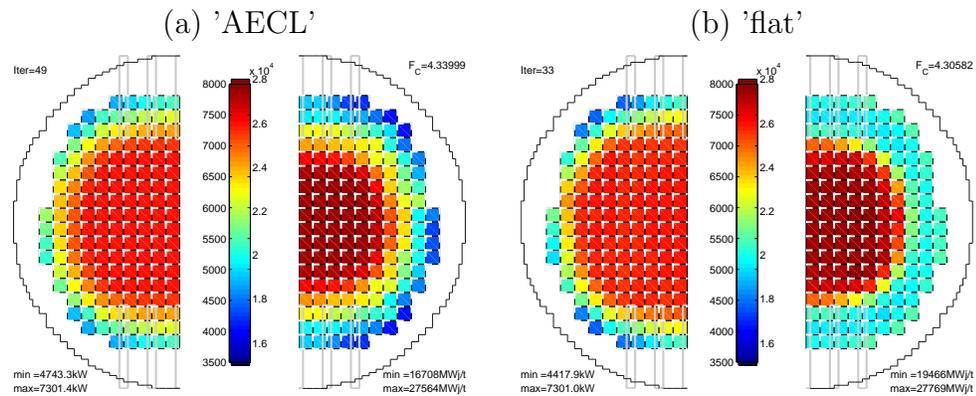


Figure 5.6 Distributions de la puissance canal et du burnup obtenues avec la MM pour 150 zones.

la première hypothèse. En effet, le nombre de variables (150) est trop grand pour soit résoudre analytiquement le problème, tracer les fonctionnelles ou même juger qualitativement du couplage entre toutes les zones de burnup simultanément en prenant en compte toutes les contraintes. Quant aux critères de convergence, ils sont limités par la précision à laquelle la distribution de flux est calculée, et qui dicte la précision des fonctionnelles.

Les résultats au tableau 5.5 montrent cependant qu'une réduction significative de la fonction objectif (le coût en combustible) peut être obtenue en prenant 150 zones. Suivant le point initial, cette réduction est d'environ 3 ou 4%. Pour le cas du réacteur CANDU-6, le gain du grand nombre de zones est plus petit, un maximum de 1.5% est obtenu entre les optimisations avec 2 et 190 zones. Ceci s'explique en fait par la grande variation du burnup moyen de sortie pour les deux types de réacteur. Pour le réacteur CANDU-6, le burnup varie entre 6.8 and 8.2 GWj/t , i.e. une variation de $\sim 15\%$, alors que pour le ACR-700, nous pouvons observer une fourchette beaucoup plus grande ($\sim 45\%$). Une comparaison des figures 5.5 et 5.6 montre également que la distribution de puissance canal peut être très aplatie en

prenant 150 zones.

Une comparaison des résultats obtenus avec MS et MM démontre que les deux méthodes sont équivalentes avec la fonction objectif et les burnups minimum et maximum comme critère (tableau 5.5). Enfin, différents poids pour les contraintes (w_k pour la criticité du réacteur et w_q pour la puissance canal maximum) ont été testés pour le cas à 7 zones, et ce pour les deux points initiaux envisagés. 4 ordres de grandeurs pour le poids des contraintes avec un rapport w_k/w_q constant, ainsi que 3 ordres de grandeur pour le rapport w_k/w_q ont été testés. Aucune différence significative sur les résultats n'a été observée, ce qui "valide" le choix initial de $w_k = 5000$ et $w_q = 1000$ en particulier.

5.3.2 Effort de calcul

Les efforts de calcul correspondant aux résultats présentés au tableau 5.5 sont donnés au tableau 5.6. It_e , ϕ et $Grad$ représentent le nombre d'itérations externes, de calculs de flux et d'évaluation de l'ensemble des gradients respectivement. Généralement, la méthode MM requiert moins de calculs de flux pour converger vers la distribution optimum de burnup. La seule exception est le cas à 7 zones à partir de la distribution initiale "AECL", où le nombre de calcul de flux est quasiment identique entre les deux approches. Le nombre d'évaluation des gradients est quant à lui toujours plus petit avec la méthode MM qu'avec la méthode MS.

Comme nous l'avons mentionné précédemment, dans le cas à 150 zones à partir d'un burnup uniforme (flat), aucun point réalisable n'a été trouvé. Une analyse détaillée des résultats a montré que dans ce cas la contrainte quadratique demeure faible ($S = 50$ par rapport à 2000 au départ) pour maintenir la linéarisation de la contrainte sur k_{eff} valide. Le pas d'avance à chaque itération est donc si petit,

Tableau 5.6 Efforts de calculs pour la résolution du problème d'optimisation.

n_z	I.P.	meth	It_e	ϕ	Grad
7	AECL	MS		103=7+6+90	82=3+4+75
7	flat	MS		101=11+65+25	82=7+56+19
150	AECL	MS		83=13+10+60	65=7+8+50
150	flat	MS		NA=18+201 ^(α) +NA	NA=13+176+NA
7	AECL	MM	5=4+1	105=33+72	60=21+39
7	flat	MM	6=5+1	79=50+29	64=40+24
150	AECL	MM	2=1+1	61=11+50	52=8+44
150	flat	MM	10=9+1	118=84+34	91=62+29

que quand la méthode MS est utilisée à partir d'un point initial trop éloigné du domaine réalisable, la convergence peut prendre de nombreuses heures. Cependant, il est important de noter que la limitation vient purement d'un aspect *temps de calcul* et non de la capacité de la méthode à résoudre le problème; en d'autres mots, le problème n'est pas divergent mais simplement long à résoudre.

Comparons maintenant l'effort de calcul pour trouver un point réalisable. Tout d'abord, celui-ci est donné par la somme des deux premiers chiffres à droite du signe "=" de la colonne ϕ pour la méthode MS et directement par le premier chiffre pour la méthode MM. Le nombre d'itérations est toujours plus petit avec la distribution "AECL" comme point de départ. L'effort de calcul pour la minimisation de la fonction coût est donné par le dernier chiffre de la somme pour les deux méthodes MS et MM. La convergence pour cette dernière étape est plus rapide à partir d'une distribution uniforme du burnup (en moyenne 30 itérations) qu'à partir d'une distribution initiale détaillée (en moyenne 75 itérations). Ceci signifie que pour les deux méthodes (MS et MM), le point réalisable trouvé à partir de la distribution "flat" est plus proche de l'optimum que celui obtenu à partir de la distribution "AECL". Dans le cas à 7 zones, la méthode MM est au total plus rapide en partant du cas "flat". Pour le cas à 150 zones, l'optimum est plus rapidement obtenu à

partir du point “AECL”.

À la vue des résultats obtenus précédemment, nous retiendrons la méthode MM pour les prochains calculs à cause de sa vitesse de convergence.

5.3.3 Analyse de sensibilité de la réactivité du vide

Un des critères du design du ACR-700 est d’avoir une réactivité du vide négative en tout temps. Nous avons donc calculé sa valeur lors des calculs d’optimisation présentés à la section précédente afin de connaître sa dépendance par rapport à la distribution du burnup moyen de sortie. Les résultats avec la méthode d’optimisation MS sont donnés à la figure 5.7 pour le cas à 7 zones en partant des deux points initiaux utilisés. La réactivité du vide cœur complet (équation 5.1) n’est représentée que pour l’étape de minimisation du coût en combustible. En effet, pour les deux premières étapes de la méthode MS, le vecteur de décision ne correspond pas à une configuration réaliste du réacteur, la réactivité du vide n’a donc aucune signification dans ce cas.

Les résultats montrent que la réactivité du vide est très petite, et ceci pour les deux points de départ. Quand le cœur est vidé, le flux thermique augmente dans le réflecteur, donc les fuites deviennent plus grandes, ce qui conduit donc à une réactivité du vide négative. Pour le point initial “AECL”, le burnup moyen de sortie en périphérie est d’environ $13GWj/t$ au début de l’étape de minimisation du coût en combustible, et finit à environ $17GWj/t$. Ainsi, le flux diminue en périphérie avec les itérations d’optimisation, ce qui a pour conséquence de moins augmenter les fuites lorsque le vide du caloporteur advient. Pour le cas “flat”, c’est exactement le même raisonnement, mais en partant de la situation opposée, c’est à dire un burnup moyen de sortie qui diminue d’environ $20GWj/t$ à $17GWj/t$ en

périphérie, et donc une réactivité du vide qui augmente légèrement. La valeur finale est bien entendu très similaire pour les deux points initiaux puisque la configuration finale de burnup l'est aussi comme nous l'avons déjà mentionné.

Lors d'un scénario réaliste d'une perte de caloporteur, le vide apparaît d'abord dans un canal sur 2 (une boucle sur deux), ce qui donne une configuration en damier du vide. Ensuite, la perturbation se propage en quelques minutes aux deux boucles et donc à tous les canaux, ce qui donne la vidange complète du cœur, i.e. la réactivité du vide telle que définie précédemment. La réactivité du vide en damier (un canal sur deux) a également été suivie pour savoir comment la distribution de burnup peut l'influencer. Les tendances observées sont exactement les mêmes. La valeur de réactivité du vide en damier est par contre moins négative: environ $-9.2mk$ pour la distribution de burnup optimale (vs. $-28.2mk$ pour le cœur complet). Une variation maximum de seulement $0.1mk$ est observée au cours des itérations (vs $0.5mk$ pour le cœur complet).

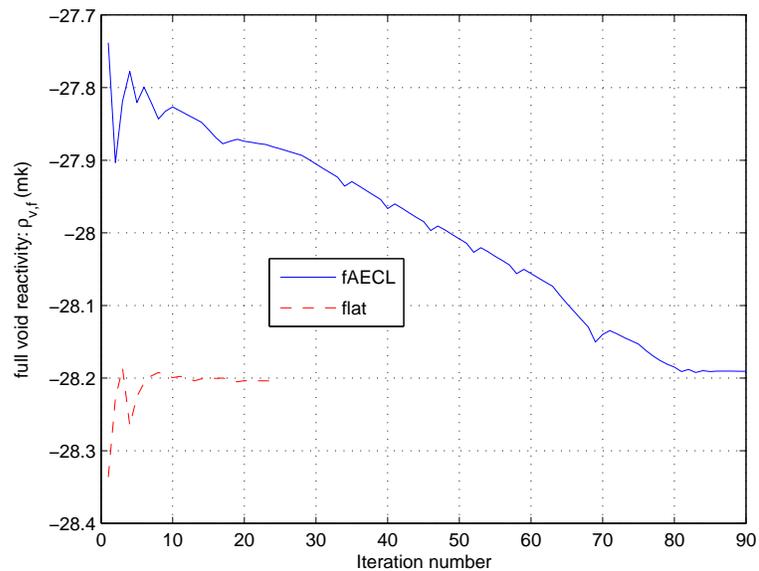


Figure 5.7 Réactivité du vide cœur complet en fonction des itérations pour la méthode MS.

5.3.4 Influence de la puissance canal maximum permise

Le maximum permis pour la puissance canal dans les calculs précédents était de $7300kW$. Cette limite est cependant arbitraire, et ne sera pas forcément celle retenue par le concepteur, puisque celui-ci doit également tenir compte de différents paramètres lors du design de la boucle du caloporteur. Nous avons pris une valeur relativement petite pour nos calculs afin de tester la validité et la robustesse des méthodes.

Les résultats de la fonction objectif démontrent une forte dépendance de l'optimum avec le nombre de zones de combustion. Plus leur nombre est grand, plus la distribution de burnup peut être ajustée finement afin de minimiser le coût en combustible. Les résultats au tableau 5.5 montrent qu'une réduction d'environ 3% de la fonction objectif est obtenue en passant de 7 à 150 zones. De plus, les distributions de puissance présentées à la figure 5.5 montrent que le burnup moyen de sortie d'une zone est limité par la puissance d'un seul canal dans le cas à 7 zones, alors qu'il est beaucoup plus "continu" dans le cas à 150 zones. Des résultats similaires ont été observés pour le CANDU-6. [32].

En plus du nombre de zones, la distribution optimale du burnup dépend aussi de la puissance maximale autorisée, et donc par conséquent le coût en combustible aussi. Pour étudier cette dépendance, plusieurs optimisations pour le cas à 7 zones ont été faites avec la méthode MM à partir du point initial "AECL". La variation de la fonction objectif optimum selon la puissance canal maximum autorisée est donnée à la figure 5.8. Plus la limite est grande, plus le coût en combustible peut être réduit. Nous avons également représenté la distribution de la puissance canal obtenue pour différentes limites de la contrainte à la figure 5.9. Les profils sont pris sur la diagonale entre les canaux R4 et C18 ce qui permet de voir les canaux avec la puissance minimum. Le nom des canaux est donné en abscisse, et peuvent être

localisé sur la figure 5.2. Plus la limite de puissance canal est élevée, plus le profil de puissance ressemble à une cloche, en d'autres mots, plus la puissance devient faible en périphérie. Les fuites sont alors diminuées, et par conséquent l'efficacité énergétique du combustible augmente.

L'étude a été réalisée avec le cas à 7 zones pour des raisons de temps de calcul. En effet, même si le nombre d'itérations est similaire pour les cas 7 et 150 zones, le temps de calcul pour tous les gradients est beaucoup plus grand pour le cas complexe. Ainsi, cela prend environ 1 heure pour 7 zones et 1 jour pour 150 zones pour résoudre l'équation (2.18).

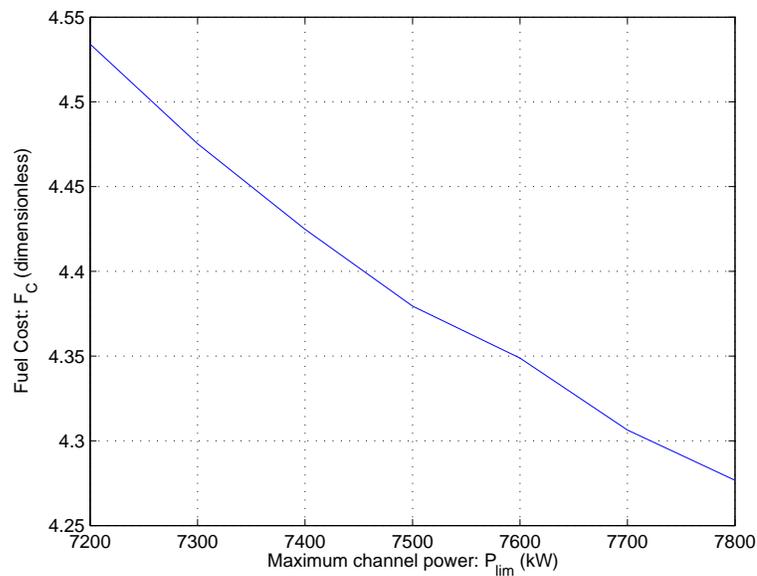


Figure 5.8 Fonction coût vs. Puissance canal maximum

5.3.5 Conclusions

Deux méthodes alternatives de gradients multi-étapes et mixte ont été utilisées avec succès pour résoudre le problème d'optimisation de la distribution du burnup du ACR-700. La méthode MS s'est cependant avérée trop lente pour converger

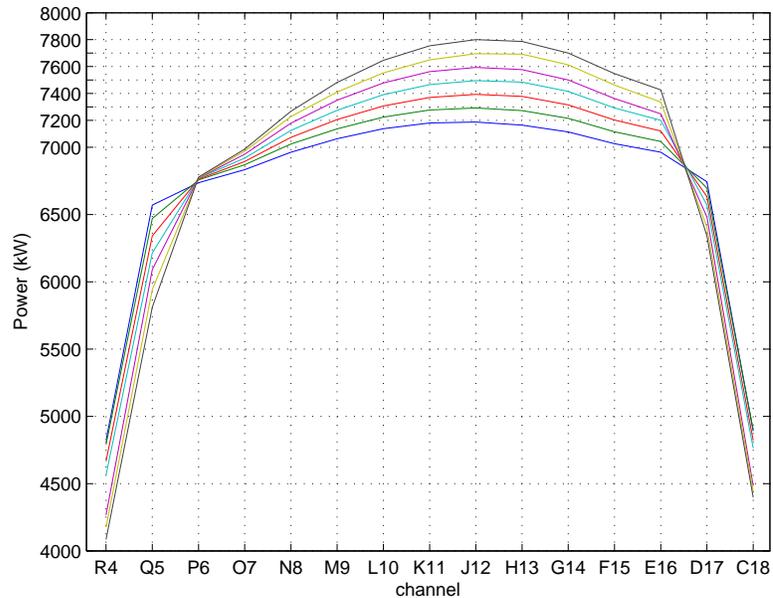


Figure 5.9 Profil de la puissance canal vs. Puissance canal maximum

dans des cas complexes. Aussi, la méthode MM est recommandée. Même si cela implique une augmentation non négligeable du temps de calcul, un plus grand nombre de zones est préférable pour obtenir une distribution optimale du burnup moyen de sortie plus fine. Une combinaison d'études préliminaires avec un nombre réduit de zones puis quelques calculs finaux avec un nombre élevé de zones est bien sûr l'idéal. Nous avons également remarqué que la variation de la réactivité du vide (complet ou en damier) est faible dans le domaine réalisable pour ces optimisations où le type de combustible est fixé. Enfin, la puissance canal maximum autorisée représente la principale limite pour le coût en combustible.

5.4 Dépendance en enrichissement des sections efficaces

5.4.1 Implantation

Afin de pouvoir optimiser l'enrichissement du combustible neuf, il est impératif de pouvoir interpoler les sections efficaces selon ce paramètre. Or, les modules qui utilisent les objets COMPO (L_COMPO) qui proviennent du calcul de transport dans DRAGON ne sont pas adaptés à une interpolation selon un autre paramètre que le burnup. La première étape d'un calcul classique dans DONJON, en ce qui concerne le traitement des propriétés des matériaux, est de créer des tables de sections efficaces (L_TABLE) avec le module XSCONS. Celles-ci sont plus pratiques à utiliser que les fichiers L_COMPO. En effet, elles contiennent normalement non seulement les sections efficaces $\Sigma(B)$, stockées dans les fichiers L_COMPO, mais aussi la dérivée de ces propriétés par rapport au burnup $\frac{d\Sigma(B)}{dB}$. Cette dernière est nécessaire aux calculs d'interpolation dans le module XSFUEL qui remplace le module CRE utilisé jusqu'à présent. Lorsque l'enrichissement est aussi pris comme paramètre pour les propriétés, alors la dérivée des sections efficaces selon celui-ci ainsi que la dérivée croisée avec le burnup sont également requises pour les calculs d'interpolations. Dans ce cas, l'objet L_TABLE contiendra donc les éléments suivant: $\Sigma(B, \varepsilon)$, $\frac{\partial \Sigma(B, \varepsilon)}{\partial B}$, $\frac{\partial \Sigma(B, \varepsilon)}{\partial \varepsilon}$ et $\frac{\partial^2 \Sigma(B, \varepsilon)}{\partial B \partial \varepsilon}$.

Le détail des formules d'interpolations avec la méthode de Ceschino, qui sont utilisées dans le module XSFUEL, est donné à l'annexe III. Les calculs d'optimisation se font avec le modèle TA dans notre étude. Il est donc nécessaire de pouvoir interpoler non seulement les propriétés, mais aussi leur intégrale ainsi que la dérivée partielle par rapport aux variables d'optimisation pour le calcul des gradients. La première version du module XSFUEL consista donc à extraire du module CRE la partie qui interpole à partir des objets L_TABLE et de rajouter la possibilité de

dériver les propriétés moyennées dans le temps par rapport au burnup de sortie. La dernière version interpole également en fonction de l'enrichissement et peut calculer la dérivée partielle des sections efficaces instantanées ou moyennées dans le temps par rapport à ce nouveau paramètre.

La dernière modification apportée pour l'interpolation à deux paramètres a consisté à adapter le module REFUEL. En effet, dans le cas des calculs en TA, la forme axiale du flux est réévaluée à chaque itération pour recalculer les burnups au début et à la fin du cycle de rechargement pour chaque canal (B_{jk}^{boc} et B_{jk}^{eoc} respectivement, éq. 2.1). Pour cela, la distribution de puissance est nécessaire, et donc indirectement les facteurs $H(B, \varepsilon)$ voir l'équation (1.3). Succinctement, d'un point de vue programmation fortran, l'enrichissement rajoute une dimension aux tableaux de données, et donc elle nécessite la réécriture de toutes les sous-routines. La sous-routine qui fait l'interpolation en tant que telle est bien entendue complètement changée pour pouvoir utiliser deux paramètres. Par contre, toute la partie interpolation est transparente en ce qui concerne le nombre de paramètres pour l'utilisateur. Seule la définition de la table est légèrement différente dans le cas à deux paramètres, sinon elle reste la même qu'auparavant. Le guide de l'utilisateur et les modifications des structures sont données dans l'annexe II.

Pour les modules d'optimisation, la dépendance en enrichissement n'a que très peu impliqué de changement. En effet, seuls le flux et les propriétés des matériaux, déjà interpolées avec d'autres modules, sont nécessaires au calcul des fonctionnelles de l'équation (2.18) et de leurs gradients. La principale différence vient de la définition de la fonction coût qui contient explicitement l'enrichissement comme paramètre, une modification du premier code implanté a donc dû être apportée pour le calcul de cette fonctionnelle et surtout de sa dérivée.

5.4.2 Vérification de l'implantation

5.4.2.1 Description du cas envisagé

L'implantation de la dépendance des sections efficaces a été vérifiée sur un exemple simple. La géométrie du réacteur fictif utilisé est donnée à la figure 5.10. Les marques C, R et 0 correspondent respectivement aux canaux du cœur, au réflecteur et à l'extérieur du réacteur. Les calculs de flux ont été fait en TA afin de tester tous les modules potentiellement impliqués dans une interpolation à 2 variables des sections efficaces. Le burnup moyen de sortie doit donc être spécifié. Deux zones de combustions ont été utilisées. Elles sont représentées sur la droite de la figure 5.10. Les COMPO utilisées contiennent des facteurs de discontinuités [37], cette nouvelle option a donc été vérifiée en même temps.

Réacteur						Zones de burnup			
0	R	R	R	R	0				
R	C	C	C	C	R	2	2	2	2
R	C	C	C	C	R	2	1	1	2
R	C	C	C	C	R	2	1	1	2
R	C	C	C	C	R	2	2	2	2
0	R	R	R	R	0				
<i>4 plans en Z</i>									

Figure 5.10 Schéma du réacteur fictif

5.4.2.2 Résultats

Pour vérifier l'interpolation à deux paramètres des propriétés, nous avons procédé par plusieurs étapes. Premièrement, nous avons obtenu des tables avec la version normale et celle modifiée de DONJON pour une seule variable pour vérifier l'équivalence du nouveau code. Une comparaison de celle-ci a démontré la simili-

tude des deux approches (aux erreurs numériques près). Le facteur de multiplication des neutrons k_{eff} a été retenu comme critère de comparaison des méthodes, plutôt que les sections efficaces, car ces dernières sont trop nombreuses.

Les résultats sont présentés au tableau 5.4.2.2. La première colonne correspond aux différents enrichissements envisagés. Les nombres en gras représentent l'enrichissement utilisé dans les calculs de transport. Les colonnes 'Trad' et 'Nouv' donnent les résultats pour la version normale et celle modifiée de DONJON. Les noms "1D" et "2D" correspondent à une interpolation à une ou deux variables. Pour les cas "1D", les résultats correspondent à des sections efficaces interpolées en burnup seulement à partir des COMPO relatives à chaque enrichissement. Les cas "2D" représentent vraiment la double interpolation.

Tous les résultats pour k_{eff} sont les mêmes pour les enrichissements notés en gras, quelque soit la méthode. Ceci valide donc deux choses. Premièrement, l'implantation de l'interpolation avec la nouvelle approche ne change rien par rapport à l'ancienne méthode pour un seul paramètre. Ensuite, nous pouvons noter que les formules d'interpolation à deux paramètres se ramènent à celles à un seul, car les enrichissements en gras sont utilisés comme point de référence pour créer les tables (cf. annexe III). Il était donc théoriquement attendu d'avoir exactement les mêmes résultats pour une ou deux variables. Enfin, pour valider l'interpolation à deux paramètres dans des cas plus généraux, nous avons utilisé des enrichissements pour lesquels aucun calcul de transport n'a été fait. Les résultats ainsi obtenus pour 2.1 à 2.4% ne sont accessibles qu'avec la méthode "Nouv 2D". k_{eff} augmente avec l'enrichissement comme prévu, mais surtout il suit la tendance obtenue entre 2.0 et 2.5% d'enrichissement.

Tableau 5.7 k_{eff} pour différents modes de calculs et enrichissements

ε	Trad	Nouv 1D	Nouv 2D
1.5	1.064802	1.064803	1.064803
2.0	1.147715	1.147715	1.147715
2.1	-	-	1.161977
2.2	-	-	1.175560
2.3	-	-	1.188505
2.4	-	-	1.200836
2.5	1.212593	1.212593	1.212593
3.0	1.263650	1.263649	1.263649

5.4.3 Influence du poids des variables

Lorsque l'enrichissement est pris comme variable de décision, un nouveau problème se pose à nous: l'influence du poids relatif de chaque type de variables. En effet, les ordres de grandeur des burnups de sortie et de l'enrichissement sont de 10^4 et 10^{-2} respectivement. Donc, si nous considérons, l'équation donnant la contrainte quadratique, le poids relatif pour les variables d'enrichissement devrait être de l'ordre de 10^{12} pour avoir un pas d'avance similaire en terme de pourcentage. Trois notions importantes sont à noter. Premièrement, le terme de "poids" des variable est en fait une terminologie simplifiée pour facteur de mise à l'échelle. Le poids des variables n'a rien à voir avec la notion de poids des contraintes. Deuxièmement, lors de la résolution du problème linéarisé, la première étape est de mettre à l'échelle toutes les variables en utilisant leur poids correspondant, pour que celui-ci soit égal à 1. La valeur absolue du poids n'est donc pas importante, seules les valeurs relatives des poids entre les variables comptent. Le pas d'avance maximum dépend par contre des valeurs absolues de la contrainte quadratique S et des poids des variables. Enfin, les gradients des fonctionnelles ne sont pas forcément tous du même ordre de grandeur même après la remise à l'échelle du problème linéarisé. Avoir un poids similaire pour tous les types de variables pour la contrainte quadratique n'est donc

pas une garantie d'avoir une influence égale des variables sur la fonction objectif.

Pour mieux comprendre et apprécier l'influence du poids des variables sur les résultats, nous avons repris le réacteur fictif présenté précédemment, et nous avons optimisé sa gestion du combustible. Plusieurs enrichissements pour le point initial ont été testés: 1.6, 2.3 et 2.9% (burnup initiaux fixés: $B_1 = 25010MWj/t$ et $B_2 = 12050MWj/t$). Le poids des variables de burnup est toujours fixé à $w_B = 1$, et celui pour l'enrichissement varie parmi les valeurs 10^{10} , 10^{11} , 10^{12} , 10^{13} , 10^{14} et 10^{16} . Enfin, les méthodes multi-étapes (MS) et du lagrangien augmenté (LA) ont été envisagées. La méthode LA a été retenue exceptionnellement à la place de la méthode MM car l'optimisation de la gestion du combustible pour ce réacteur fictif est suffisamment simple et rapide.

Avant de montrer les résultats obtenus par l'optimisation, nous avons trouvé l'optimum global de manière semi-analytique et graphique. La méthode et les résultats sont illustrés aux figures 5.11 et 5.12. La première étape consiste à trouver tous les points pour lesquels $k_{eff} = 1$. Les deux burnups sont fixés et l'enrichissement est optimisé pour avoir un réacteur critique. Les résultats à la figure 5.11a montrent que si B_1 et B_2 sont tous les deux trop petits, alors l'enrichissement atteint le minimum imposé sans avoir un réacteur critique. Ces points ne font donc pas parti du domaine réalisable et sont supprimés sur les autres graphiques de la figure. La puissance canal maximum, la fonction objectif et l'enrichissement obtenus sont illustrés respectivement sur les graphiques b, c et d. Une puissance canal de $P_{lim} = 1050kW$ est retenue pour la contrainte thermique (16MW pour la puissance totale du réacteur). En prenant en compte toutes les contraintes, le domaine réalisable ainsi que la valeur de la fonction objectif sur celui-ci sont présentés à la figure 5.12. Le minimum est obtenu pour $B_1 \approx 25MWj/t$, $B_2 \approx 26MWj/t$ et $\varepsilon = 1.55\%$.

Les résultats de l'optimisation obtenus pour les différents points initiaux avec les

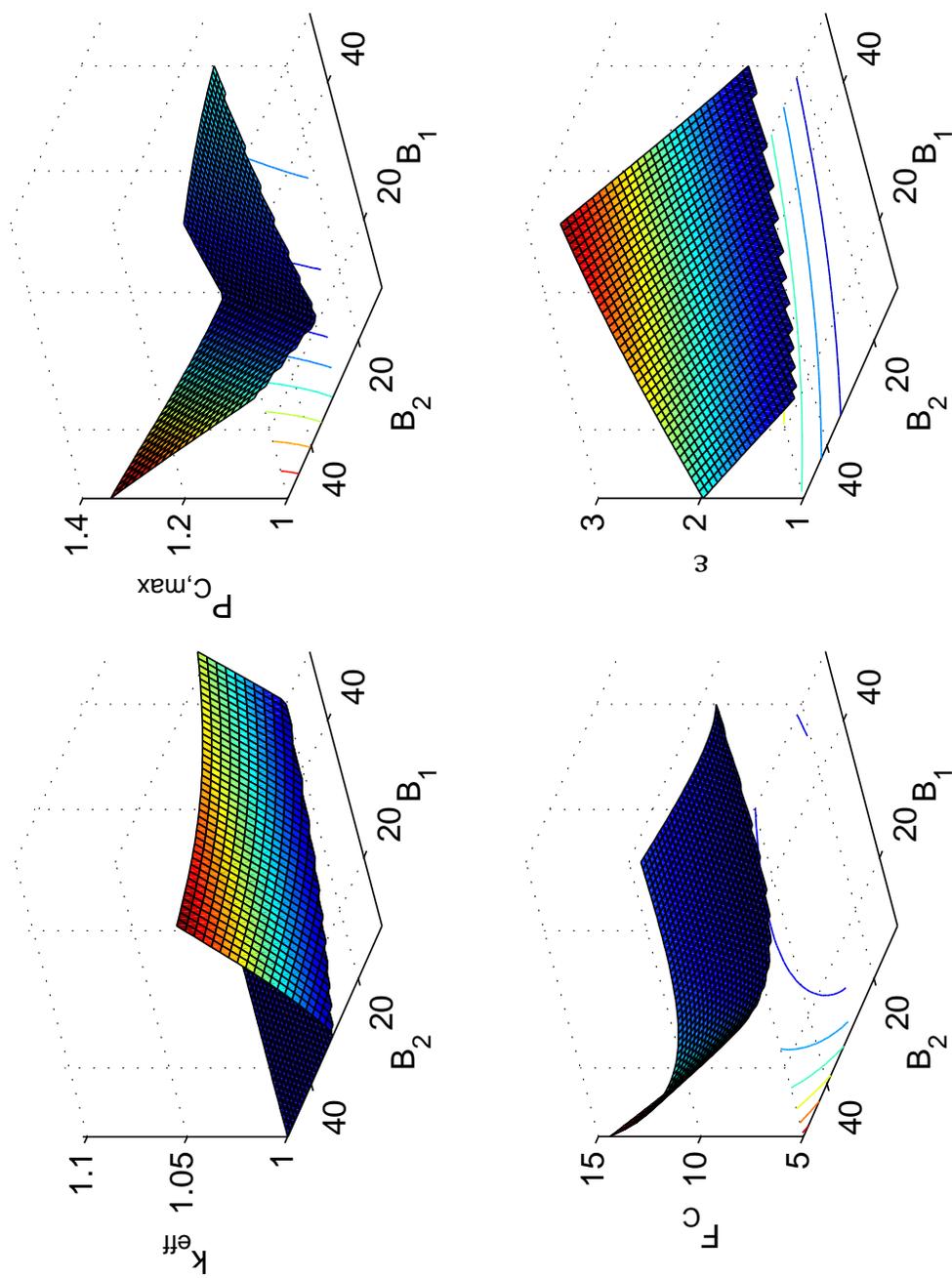


Figure 5.11 Valeur des fonctionnelles pour le réacteur fictif.

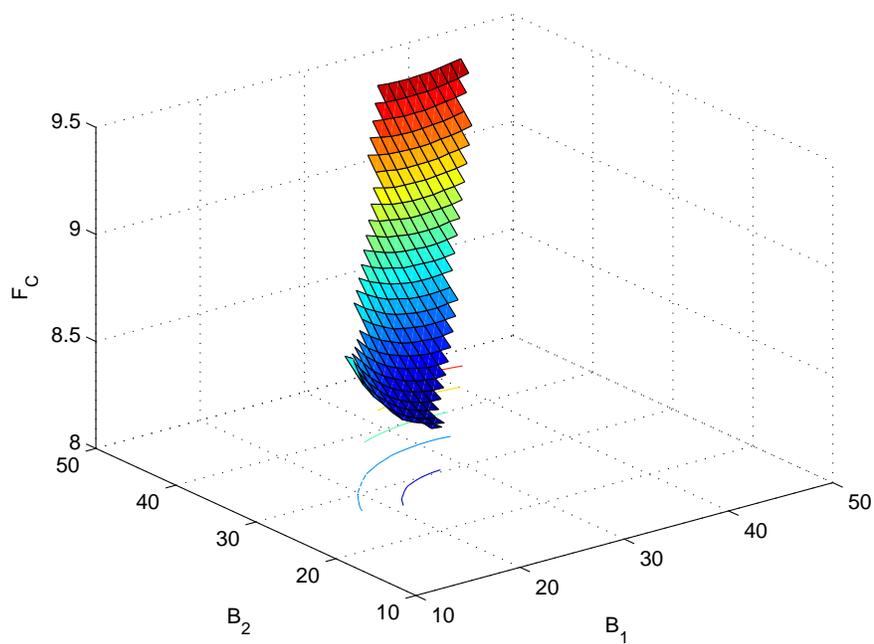


Figure 5.12 Valeur de la fonction objectif pour le réacteur fictif dans le domaine réalisable seulement.

méthodes MS et LA sont présentés aux tableaux 5.8 et 5.9 respectivement. Le signe moins “-” au tableau 5.8 correspond aux calculs pour lesquels la recherche d’un point réalisable n’a pu être complétée. Les chiffres avec une astérix signifient qu’un point réalisable a été obtenu, mais que la fonction objectif n’a pas été améliorée par la dernière étape de la méthode MS, les résultats sont néanmoins tout aussi valables que ceux sans astérix.

Pour des poids intermédiaires 10^{11} et 10^{12} , on obtient des résultats en général très similaires: B_1 entre 24.3 et 25.3 GWj/t , B_2 entre 25.7 et 26.7 GWj/t et ε entre 1.55 et 1.57% d’enrichissement. La plupart des résultats pour les poids intermédiaires peuvent en fait être considérés comme égaux, étant donné qu’une différence de 0.1 GWj/t ou de 0.01% d’enrichissement est négligeable. De plus, ces résultats coïncident avec le minimum trouvé *analytiquement*. Au contraire, si le poids w_ε pour l’enrichissement est trop petit (10^{10}) ou trop grand ($\geq 10^{13}$) alors les résultats sont biaisés. En effet, pour un poids de 10^{10} , la méthode MS ne trouve pas de configuration correspondant à un réacteur critique et donc s’arrête. La méthode LA converge sur un point non réalisable qui est en fait très proche du point initial pour le burnup: $B_1 = 25GQWj/t$ et $B_2 = 12GQWj/t$. Pour un poids grand ($\geq 10^{13}$), un point réalisable est trouvé, mais l’enrichissement final trouvé ne rapproche du minimum analytique que lorsque l’enrichissement initial est proche (i.e. 1.6%).

Pour expliquer ceci, nous devons revenir à la définition de la contrainte quadratique. La variation maximum d’une variable i à chaque itération est donnée par $S/\sqrt{w_i}$. En général, la contrainte initiale utilisée est de $S = 2000$. Étant donné que le burnup varie entre 10^4 et 5.10^4 , le poids de 1 pour w_B fait en sorte que les variables de burnup peuvent varier de 5 à 10% de leur valeur totale, et la fonction de coût peut diminuer de l’ordre de 2.10^{-4} . Pour l’enrichissement qui est compris entre 0.015 et 0.03 (valeur absolue), des poids w_ε de 10^{10} , 10^{12} et 10^{14} représentent une

Tableau 5.8 Résultats d'optimisation pour le réacteur fictif avec la méthode MS

$\frac{\varepsilon_{ini}}{w_\varepsilon}$	1.6%			2.3%			2.9%		
	B_1	B_2	ε	B_1	B_2	ε	B_1	B_2	ε
10^{10}	-	-	-	-	-	-	-	-	-
10^{11}	25.3	26.3	1.570 *	25.5	26.5	1.577 *	25.5	26.49	1.588 *
10^{12}	24.4	26.7	1.573 *	24.5	26.7	1.573 *	24.6	26.7	1.575 *
10^{13}	25.1	26.4	1.571	30.0	31.5	1.778	33.1	31.9	1.826
10^{14}	24.7	26.2	1.563 *	39.3	39.8	2.165	-	-	-

Tableau 5.9 Résultats d'optimisation pour le réacteur fictif avec la méthode LA

$\frac{\varepsilon_{ini}}{w_\varepsilon}$	1.6%			2.3%			2.9%		
	B_1	B_2	ε	B_1	B_2	ε	B_1	B_2	ε
$10^{10} \alpha$	25.0	12.0	1.540	25.2	12.9	1.542	25.2	12.7	1.540
10^{11}	24.7	25.9	1.553	25.0	25.9	1.556	25.2	26.1	1.563
10^{12}	24.4	25.9	1.550	25.1	26.3	1.568	24.7	25.9	1.552
10^{13}	23.8	26.0	1.547	29.8	31.8	1.787	33.9	42.5	2.188
10^{14}	24.3	26.4	1.564	34.2	43.3	2.218	46.1	49.5	2.598

α : point non réalisable $k_{eff} \neq 1$ et $P_{C,max} > P_{lim}$.

variation maximum autorisée d'environ 100, 10 et 1% respectivement pour ce type de variable, ce qui correspond à des variations de la fonction objectif de l'ordre de $2 \cdot 10^{-2}$, $2 \cdot 10^{-3}$ et $2 \cdot 10^{-4}$ respectivement. Ainsi, si le critère pour choisir le poids relatif des variables est d'avoir le même pourcentage de variations des variables, le meilleur poids est théoriquement environ 10^{12} . Pour avoir une variation de la fonction objectif similaire selon toutes les variables (au moins au début puisque les gradients ne changent pas forcément tous de manière proportionnelle), le meilleur poids est théoriquement environ 10^{14} .

En pratique, les résultats des tableaux 5.8 et 5.9 montrent qu'un poids w_ε entre 10^{11} et 10^{12} est recommandé (pour un poids w_B de 1). Pour un poids trop petit, l'enrichissement est le seul à changer, dans le cas opposé, il ne varie presque pas.

5.5 Optimisation du burnup moyen de sortie et de l'enrichissement

Pour les calculs en burnup seulement, l'enrichissement initial moyen de la grappe (sans tenir compte du crayon au centre) est de 2.5% pour une concentration de Dy de 15%, donc un rapport de 1 pour 6. Ainsi pour les calculs en burnup et en enrichissement, nous avons conservé ce rapport pour un enrichissement variant entre 1.5 et 3.5%. La variation de concentration en Dy est donc implicite. Nous reviendrons ultérieurement sur le fait d'avoir pris un rapport R "enrichissement initial/concentration de Dy" constant. Enfin, 5 enrichissements différents ont été utilisés pour l'interpolation de sections efficaces à deux paramètres. Grâce à l'application développée dans MATLAB, nous avons tracé la variation de la section efficace totale du combustible dans le groupe thermique selon le burnup et l'enrichissement à la figure 5.13. Aucune variation brusque de la section efficace n'est observée selon l'enrichissement. L'interpolation de Ceschino d'ordre 4 est donc suffisamment précise. La traduction du format L_COMPO en format MATLAB, est assurée par une des options du module MATLAB :

Lorsque l'enrichissement du combustible peut varier, le prix du combustible neuf dans la fonction de coût F_C n'est plus constant. Il est donné par la formule (2.6). Les coûts de l'uranium naturel et de séparation que nous avons utilisés proviennent des données prises en janvier 2006 sur le site web de la "World Nuclear Association" [38]. Le coût de fabrication est estimé à partir de celui des grappes pour les réacteurs CANDU-6 [39]. L'ensemble des paramètres utilisé pour le calcul du coût des grappes est résumé dans le tableau 5.10.

L'optimisation de la gestion du combustible pour le ACR-700, avec deux types de variables de décision, a été faite en utilisant 7 zones de burnup et 1 zone d'enrichissement (tout le cœur). La répartition des zones de burnup est la même que pour les calculs avec juste le burnup comme variable de décision (figure 5.4).

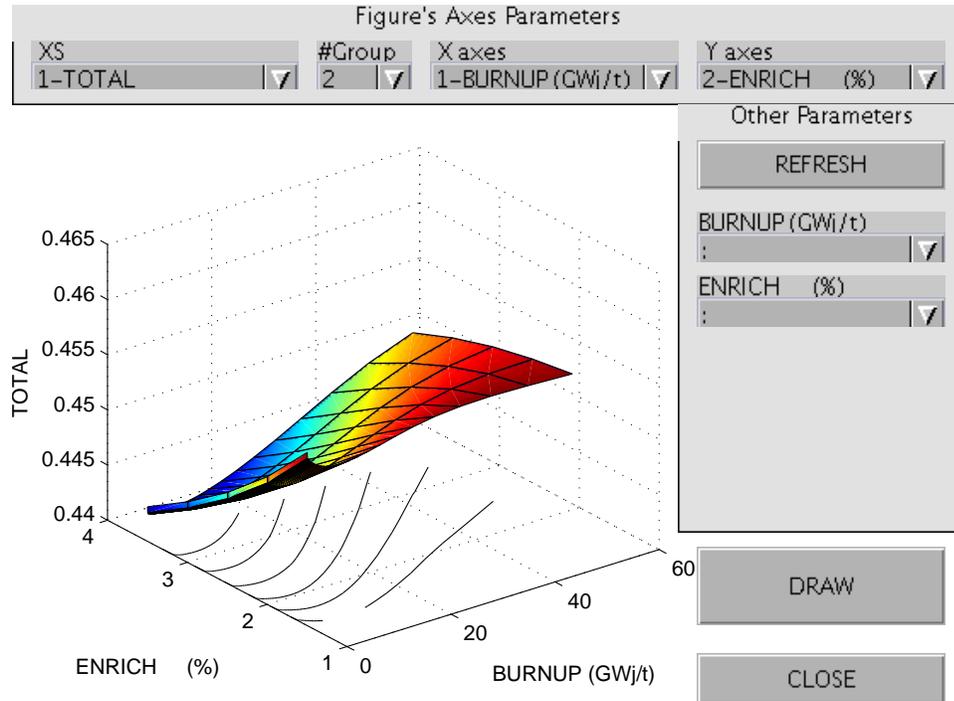


Figure 5.13 Section efficace totale dans le groupe thermique.

Tableau 5.10 Paramètres pour le calcul du coût des grappes.

paramètre	valeur	unité	description
C_{UN}	100	\$/kg	coût de l'uranium naturel
C_{SWU}	140	\$/SWU	coût de l'enrichissement
C_{FAB}	80	\$/kg	coût de fabrication de la grappe
int	10	%	taux d'intérêt annuel
t_1	0.5	an	délai d'obtention de l'uranium naturel
t_2	0.5	an	délai pour l'enrichissement
ε_w	0.2	%	teneur en U^{235} de l'uranium appauvri

5.5.1 Influence du poids des variables

Comme nous venons de le voir à la section 5.4.3, un nouveau paramètre peut influencer les résultats: le poids relatif entre les variables X_i , i.e. leur facteur w_{X_i} de mise à l'échelle. Nous avons donc reconduit les calculs d'optimisation de la gestion du combustible avec les méthodes multi-étapes (MS) et mixte (MM) pour différentes valeurs du facteur w_ε pour l'enrichissement ($w_B = 1$). Les résultats sont donnés aux tableaux 5.11 et 5.12. L'enrichissement peut varier entre 1.5 et 3.5%. Nous avons testé trois valeurs initiales différentes pour l'enrichissement, qui sont données à la colonne ε_{ini} . F_C et ε_f correspondent aux valeurs optimales de la fonction objectif (unité arbitraire) et de l'enrichissement du combustible neuf. La distribution initiale du burnup moyen de sortie est soit celle dérivée de la distribution détaillée fournie par AECL (section 5.3, cas "AECL") pour les résultats du tableau 5.11, soit une distribution uniforme de $23000 MWj/t$ (cas "flat") au tableau 5.12.

Avant d'analyser les résultats obtenus, nous pouvons remarquer que plusieurs calculs ont divergé (mention NA dans les tableaux 5.11 et 5.12). Nous reviendrons sur ces cas spécifiques ultérieurement. Les résultats des tableaux 5.11 et 5.12 montrent que plus l'enrichissement est grand plus la fonction objectif est faible. De plus, les deux méthodes MS et MM donnent des valeurs différentes de la fonction objectif et de l'enrichissement final pour un même poids w_ε et un même point initial (différentes distributions de burnup de type "AECL" ou "flat" et différents enrichissements ε_{ini}). En fait, comme nous le verrons plus tard, quasiment tous les calculs se sont arrêtés après avoir atteint le nombre maximum d'itérations. Or, ce nombre est différent pour les deux méthodes: 100 et 150 pour les méthodes MS et MM respectivement. Cela explique donc le fait que les résultats obtenus avec la méthode MM soient systématiquement meilleurs (F_C plus petit).

Même si les algorithmes d'optimisation n'ont pas été poussés jusqu'à la convergence, les résultats "intermédiaires" obtenus nous permettent de tirer plusieurs observations. Premièrement, un poids extrême (très grand 10^{14} ou très petit 10^{10}) peut entraîner une défaillance des méthodes quelque soit le point initial. De plus, pour un poids grand (10^{13}), l'enrichissement obtenu est très dépendant de sa valeur initiale. En fait, comme nous l'avons vu à la section précédente, pour un tel poids w_ε , dès que la variable change un peu cela entraîne une grande variation de la fonction objectif. À l'opposé les variables de type burnup ont alors une plus faible influence sur la fonction objectif, et ont donc tendance à changer moins vite. Or ces dernières doivent varier avec l'enrichissement pour toujours respecter les contraintes. Autrement dit c'est l'arrosee arrosé. Un poids élevé de l'enrichissement entraîne une faible variation du burnup qui limite à son tour les changements d'enrichissement. Pour un poids plus faible (10^{11} et 10^{12}), les résultats sont les mêmes quelque soit l'enrichissement initial. En fait, l'enrichissement peut varier plus, et dans la recherche d'un point réalisable, c'est cette variable qui va s'ajuster aux variables de burnup pour respecter les contraintes, plutôt que le contraire.

Ainsi, comme nous l'avons déjà remarqué à la section 5.4.3, lorsque le poids w_ε est trop grand, la valeur de la variable d'enrichissement tend à rester proche de sa valeur initiale. À l'opposé, quand le poids w_ε est trop petit, les variables de burnup varient plus difficilement, seul l'enrichissement varie ce qui entraîne parfois une défaillance de l'algorithme pour trouver un point réalisable. Un poids entre 10^{11} et 10^{12} semble recommandé, préférentiellement 10^{12} car un problème de convergence est survenu avec la méthode MM pour un poids de 10^{11} dans le cas "flat".

L'effort de calcul associé aux résultats présentés aux tableaux 5.11 et 5.12 est présenté au tableau 5.13. Les nombres correspondent aux nombres de calculs de flux de chaque étape des méthodes MS et MM. "NA" correspond aux résultats non disponibles. Le nombre total de calcul de flux n'est pas donné, car, comme

Tableau 5.11 Résultats à partir de la distribution “AECL”.

ε_{ini}	w_ε	MS		MM	
		F_C	$\varepsilon_f\%$	F_C	$\varepsilon_f\%$
1.6%	10^{10}	NA	NA	NA	NA
	10^{11}	4.4640	2.826	4.3540	3.014
	10^{12}	4.5080	2.762	4.3692	2.978
	10^{13}	4.6468	2.593	4.4499	2.822
	10^{14}	NA	NA	4.5702	2.659
2.5%	10^{10}	NA	NA	NA	NA
	10^{11}	4.4593	2.832	4.3518	3.019
	10^{12}	4.4703	2.815	4.3420	3.038
	10^{13}	4.5916	2.664	4.3727	2.958
	10^{14}	4.6020	2.628	4.4758	2.774
3.4%	10^{10}	4.4611	2.830	NA	NA
	10^{11}	4.4515	2.844	4.3812	2.960
	10^{12}	4.4562	2.842	4.4007	2.926
	10^{13}	4.3210	3.095	4.2268	3.377
	10^{14}	NA	NA	4.2176	3.401

Tableau 5.12 Résultats à partir de la distribution “flat”.

ε_{ini}	w_ε	MS		MM	
		F_C	$\varepsilon_f\%$	F_C	$\varepsilon_f\%$
1.6%	10^{10}	NA	NA	NA	NA
	10^{11}	4.3095	3.087	4.2588	3.245
	10^{12}	4.3458	3.000	4.2804	3.173
	10^{13}	NA	NA	4.4194	2.859
	10^{14}	NA	NA	4.5672	2.662
2.5%	10^{10}	4.3102	3.085	NA	NA
	10^{11}	4.3256	3.046	NA	NA
	10^{12}	4.3190	3.063	4.2759	3.187
	10^{13}	4.3508	2.987	4.2907	3.139
	10^{14}	4.3961	2.898	4.3219	3.052
3.4%	10^{10}	4.3134	3.077	NA	NA
	10^{11}	4.3055	3.098	4.2717	3.201
	10^{12}	4.3314	3.031	4.2465	3.290
	10^{13}	4.2564	3.247	4.2161	3.410
	10^{14}	4.2599	3.301	4.3422	3.305

nous l'avons signalé précédemment, le nombre maximum d'itérations est quasiment toujours atteint à la dernière étape, i.e. 100 et 150 pour les méthodes MS et MM respectivement. Le nombre de calcul de flux est le même +1 pour le point initial. Deux raisons nous ont amenés à utiliser un nombre d'itérations maximum assez faible. Premièrement, à priori cela semblait initialement suffisant compte tenu des résultats obtenus pour les simulations avec juste les 7 zones de burnup sans la variable d'enrichissement (tableau 5.6, maximum 118 itérations au total). Deuxièmement, le but principal était de tester les différents poids pour la variable d'enrichissement pour savoir quelle fourchette est acceptable, i.e. savoir si un point réalisable est trouvé et si une convergence prématurée est obtenue, et ceci dans un temps raisonnable.

De ces calculs, même s'ils sont limités, plusieurs indications importantes peuvent être tirées. Premièrement, le nombre de calculs de flux nécessaire pour trouver un point réalisable est donné par la somme des premiers nombres pour la méthode MS, et directement par le premier pour la méthode MM. Comme nous l'avons déjà remarqué précédemment pour les calculs sans enrichissement, un point réalisable est la plupart du temps plus facilement trouvé à partir de la distribution de burnup "AECL" que de celle du type "flat", quelque soit l'algorithme d'optimisation choisi. Pour la méthode MM, le poids de la variable d'enrichissement de 10^{12} correspond presque systématiquement aux calculs les plus rapides pour trouver un point réalisable. ceci ajoute un autre critère pour le sélectionner comme poids idéal. Pour la méthode MS, les différences proviennent essentiellement de l'étape pour respecter la contrainte sur la puissance canal maximum. Pour un enrichissement initial de 2.5%, le nombre d'itérations de chaque étape est équivalent quelque soit le poids de la variable d'enrichissement w_e , car cet enrichissement est proche de celui nécessaire pour avoir un réacteur critique pour les deux distributions de burnup initiales. Dans ce cas, le burnup est donc la principale variable ajustée, ce qui

Tableau 5.13 Effort de calcul

ε_{ini}	w_ε	AECL		flat	
		MS	MM	MS	MM
1.6%	10^{10}	NA+NA+NA	117+NA	NA+NA+NA	94+NA
	10^{11}	17+4+101	14+151	15+71+101	88+151
	10^{12}	17+5+101	29+151	17+84+101	65+151
	10^{13}	20+40+101	54+151	21+151+NA	80+151
	10^{14}	31+151+NA	158+151	27+151+NA	113+151
2.5%	10^{10}	10+4+NA	21+NA	9+69+101	109+NA
	10^{11}	10+4+101	18+151	12+69+101	64+NA
	10^{12}	13+4+101	5+151	14+70+101	41+151
	10^{13}	14+7+101	26+151	11+49+101	60+151
	10^{14}	12+7+101	35+151	12+56+101	74+151
3.4%	10^{10}	11+4+101	10+NA	16+70+101	144+NA
	10^{11}	12+4+101	34+151	15+67+101	74+151
	10^{12}	13+4+101	21+151	16+55+101	36+151
	10^{13}	20+13+101	48+151	19+11+101	49+151
	10^{14}	23+72+NA	88+101	22+9+40	63+11

implique une influence minimale de l'enrichissement, et donc de son poids. Pour les deux autres enrichissements initiaux, par contre, le poids w_ε choisi fait varier beaucoup le nombre de calculs de flux nécessaire à l'obtention d'un point réalisable. Quand le point initial est très loin du domaine réalisable, le nombre maximum d'itérations pour la minimisation de $P_{C,max}$ (150) est atteint, ce qui entraîne une défaillance de la dernière étape de la méthode, i.e. la minimisation de F_C par la programmation linéaire. Ainsi, le poids idéal pour la variable d'enrichissement a été déterminé à 10^{12} .

5.5.2 Fonction objectif en fonction de l'enrichissement

5.5.2.1 Résultats intermédiaires

Si nous tenons compte de deux types de variables de décision (burnup et enrichissement) simultanément, les résultats des tableaux 5.11 et 5.12 démontrent une forte dépendance de la vitesse de convergence des méthodes MS et MM selon le choix du point de départ et le choix du facteur de mise à l'échelle w_ε . Une question peut alors être posée par rapport aux résultats trouvés: y a-t-il une variable qui ralentit

tit la convergence, voir même empêche la convergence elle-même étant donné que les calculs ont été arrêtés par un nombre maximum d'itérations? Pour tenter de répondre à cette interrogation, nous avons tracé à la figure 5.14 la valeur de la fonction objectif optimale par rapport à l'enrichissement optimum selon les résultats des tableaux 5.11 et 5.12. Sur cette figure, sont également reportés les résultats d'optimisation effectués en burnup seulement pour différents types de combustibles, i.e. pour différentes valeurs fixées d'enrichissement (points correspondant aux cercles). L'optimisation a dans ce cas été faite en utilisant la méthode MS à partir de la distribution "AECL". Le choix de la méthode et du point initial n'a cependant aucune influence sur les résultats de la fonction objectif pour le cas à 7 zones comme nous l'avons remarqué à la section 5.3. Dans ces derniers cas, la fonction objectif obtenue est un optimal.

À l'exception d'un point, l'ensemble des résultats des tableaux 5.11 et 5.12 se trouve sur une courbe du type " $1/x$ " ou "fond de parabole", qui suit les résultats des optimisations à enrichissement fixe. Ceci démontre que la distribution de burnup est toujours optimisée en premier, et qu'elle atteint une valeur optimale correspondante à la valeur de l'enrichissement au début des algorithmes d'optimisation. Ensuite, cette dernière variable est minimisée et la distribution de burnup s'ajuste ce qui fait en sorte que le couple (F_C, ε) suit pour le reste des itérations la courbe formée par les points de la figure 5.14. Un *nuage* de points aurait plutôt été attendu dans le cas où les deux types de variables auraient été optimisés simultanément et directement jusqu'au minimum global. De plus, l'enrichissement n'a pas toujours atteint la même valeur après un nombre fixe d'itérations pour la dernière étape de chacune des deux méthodes. La disparité des résultats provient donc d'une influence conjointe du choix du point de départ et surtout du poids de la variable d'enrichissement sur la convergence des résultats. Nous pouvons également remarquer que les résultats obtenus à partir de la distribution uniforme de burnup

sont en général plus optimisés que dans le cas “AECL” quelque soit la méthode d’optimisation. En effet, les points correspondant à la distribution “flat” (les “x”) ont en moyenne une fonction objectif plus petite que les points correspondant à la distribution “AECL” (les “+”)

Le cas particulier du point tout seul à la figure 5.14 correspond au calcul à partir d’un enrichissement de 3.4% et une distribution de burnup type “flat” pour un poids w_ε de 10^{14} avec la méthode MM (voir tableau 5.12). Nous pouvons remarquer au tableau 5.13 que le calcul d’optimisation a convergé très rapidement dans ce cas, en fait prématurément.

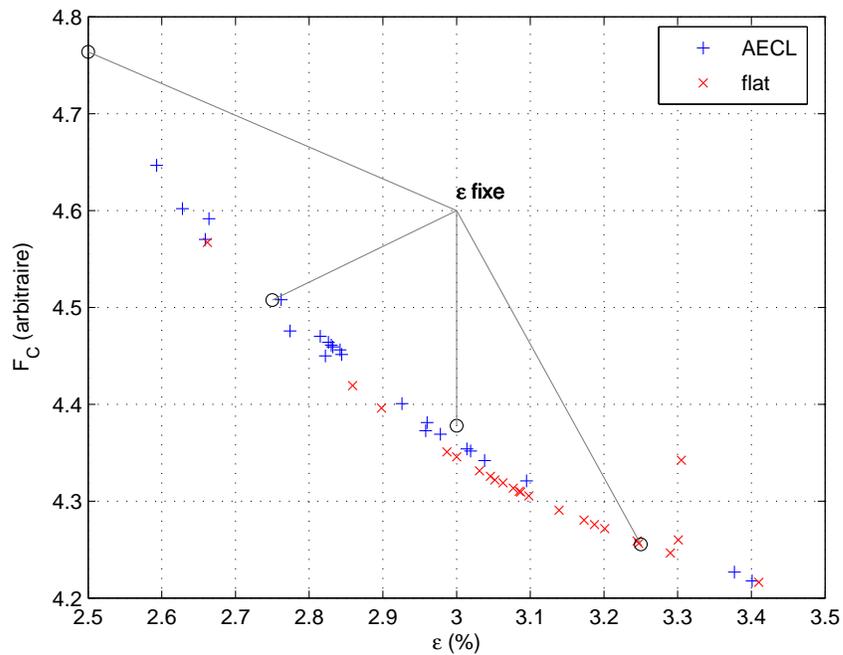


Figure 5.14 Fonction objectif optimale vs. Enrichissement optimal.

5.5.2.2 Théorie

À la section précédente, nous avons remarqué sur la figure 5.14 que la fonction objectif F_C (i.e. le coût en combustible par kW) diminue avec l'enrichissement. Cela peut paraître à priori paradoxal puisque le coût du combustible augmente avec l'enrichissement. Pour expliquer ce comportement, nous devons revenir à la définition de la fonction objectif donnée par l'équation (2.10). Dans le cas où une seule zone d'enrichissement est utilisée, nous pouvons écrire:

$$F_C = C_F(\varepsilon) \frac{\langle \frac{1}{\overline{B}_j^e} \cdot H, \phi \rangle_{réacteur}}{\langle H, \phi \rangle_{réacteur}} = C_F(\varepsilon) \frac{\sum_j \frac{P_j}{\overline{B}_j^e}}{\sum_j P_j} \quad (5.14)$$

Dans cette équation, le ratio correspond en fait à la définition du burnup moyen de sortie du cœur, \overline{B}^e . En utilisant l'équation (2.6) qui définit le coût du combustible, nous pouvons réécrire l'équation (2.10) sous la forme très simple suivante:

$$F_C = \frac{C_F(\varepsilon)}{\overline{B}^e} = \frac{C(\varepsilon) + C_{FAB}}{\overline{B}^e} \quad (5.15)$$

où $C(\varepsilon)$ regroupe tous les termes dépendants de l'enrichissement de l'équation (2.6), et donc $C(\varepsilon)$ dépendant également de C_{UN} et C_{SWU} .

Si le coût de fabrication d'une grappe est très grand par rapport au coût de l'uranium naturel et de l'enrichissement (i.e. coût des matériaux), $C_{FAB} \gg C(\varepsilon)$, alors la fonction objectif peut être approximée par $F_C = \frac{C_{FAB}}{\overline{B}^e}$. Le burnup moyen de sortie a donc tendance à augmenter au maximum. Pour que la contrainte sur la réactivité soit satisfaite, cela veut dire que l'enrichissement augmente aussi. D'un point de vue ingénierie, cela revient à dire que si les grappes coûtent cher à fabriquer, il est préférable d'utiliser de l'uranium le plus enrichi possible pour recharger le moins souvent possible.

Dans le cas contraire, si le coût de fabrication d'une grappe est très petit par rapport au coût des matériaux, $C_{FAB} \ll C(\varepsilon)$, alors la fonction objectif peut être approximée par $F_C = \frac{C(\varepsilon)}{B^e}$. Dans ce cas, minimiser F_C revient à augmenter le burnup moyen de sortie du cœur et à réduire le coût en matériaux, i.e. réduire l'enrichissement. Or, cela est contradictoire en ce qui concerne la contrainte sur la réactivité du réacteur ($k_{eff} = 1$), un compromis entre augmenter $\overline{B^e}$ et réduire $C(\varepsilon)$ doit être fait. Pour savoir à quel moment se situe la balance, regardons maintenant plus en détail la variation de F_C par rapport à l'enrichissement. En supposant que le burnup moyen de sortie du cœur $\overline{B^e}$ correspond à la distribution optimale $\{B_i^e\}$ pour un enrichissement donné, nous pouvons écrire $\overline{B^e}(\{B_i^e\}, \varepsilon) = \overline{B^e}(\varepsilon)$. Ainsi en différentiant selon l'enrichissement l'équation (5.15), nous obtenons:

$$\begin{aligned} dF_C/d\varepsilon &= \frac{dC_F/d\varepsilon}{\overline{B^e}} - F_C \frac{d\overline{B^e}/d\varepsilon}{\overline{B^e}} \\ \Leftrightarrow \frac{dF_C(\varepsilon)}{F_C(\varepsilon)} &= \frac{dC_F(\varepsilon)}{C_F(\varepsilon)} - \frac{d\overline{B^e}(\varepsilon)}{\overline{B^e}(\varepsilon)} \end{aligned} \quad (5.16)$$

Tout d'abord, nous définissons le taux de variation d'une quantité u par rapport à l'enrichissement, comme étant du/u . Sachant que la fonction objectif F_C est toujours positive, elle augmente ($dF_C > 0$) si le taux de variation burnup moyen de sortie du cœur $\overline{B^e}$ est plus petit que celui du coût en combustible, et F_C diminue dans le cas contraire.

Revenons maintenant aux résultats présentés à la figure 5.14. Nous avons tracé à la figure 5.15 la variation du coût du combustible en fonction de l'enrichissement pour les valeurs données au tableau 5.10. Le coût de fabrication est relativement faible par rapport au coût en matières premières. Nous sommes donc dans le deuxième cas exposé ci-dessus, à savoir un compromis entre le burnup moyen de sortie du cœur et l'enrichissement doit être fait. La figure 5.14 suggère qu'un minimum ou un plateau de la fonction objectif est atteint proche ou à partir de 3.5% d'enrichissement (i.e.

pour $\varepsilon = \varepsilon_{max}$), ce qui équivaut à dire que $R_C = \frac{dC_F}{C_F}$ et $R_B = \frac{d\overline{B^e}}{\overline{B^e}}$ sont égaux pour ε_m . Essayons d'évaluer ces deux quantités. Pour la première R_C , $C_F(\varepsilon)$ est connu analytiquement, cela ne pose donc aucun problème. Pour évaluer R_B , nous devons connaître $\overline{B^e}$ en fonction de l'enrichissement. Or, aucune formule analytique n'est disponible. Deux approximations sont possibles. La première, suite à un calcul de diffusion, consiste simplement à récupérer (ou à calculer) le burnup moyen de sortie du cœur à la suite des calculs d'optimisation. L'inconvénient de cette méthode est de ne pas être indépendante des calculs d'optimisation, mais elle tient compte de la distribution du burnup moyen de sortie. L'autre approche consiste à utiliser les résultats de calcul de transport. Alors, $\overline{B^e}$ du cœur est approximé par le burnup pour lequel l'ensemble des grappes permet en moyenne d'obtenir $k_{eff} = 1$. Pour trouver ce point, il nous suffit de calculer $\overline{k_{eff}}$ défini par:

$$\overline{k_{eff}(B)} = \frac{1}{B} \int_0^B k_{eff}(B') dB' \quad (5.17)$$

Le point qui correspond à $\overline{k_{eff}} = 1$ est le point recherché. Nous avons suivi cette deuxième approche car si elle s'avère suffisamment précise pour vérifier nos résultats, elle pourrait être alors utilisée comme approche pour optimiser grossièrement l'enrichissement sans calcul d'optimisation.

Quelque soit l'approche suivie, le burnup moyen de sortie du cœur $\overline{B^e}(\varepsilon)$ est connu pour plusieurs enrichissements (ceux utilisés pour créer les tables de propriétés, ou ceux utilisés pour une optimisation à enrichissement fixé), ce qui permet d'avoir un estimé de la quantité R_B . Nous avons tracé à la figure 5.16 ces deux rapports. L'approche avec les calculs de transport à été retenue pour le calcul de R_B . Les résultats semblent montrer qu'en effet le minimum de F_C est obtenu pour un enrichissement d'environ 3.5%, car les deux courbes se rejoignent à cette endroit. Il nous manque malheureusement le rapport R_B pour 3.5% d'enrichissement car $\overline{k_{eff}} = 1$ n'a pas été atteint pour le burnup final d'évolution en transport.

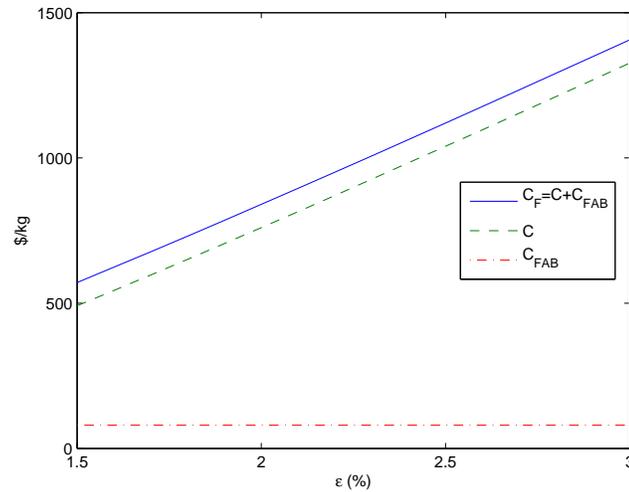


Figure 5.15 Coût du combustible C_F , des matériaux C et fabrication C_{FAB} en fonction de l'enrichissement.

Néanmoins, cette figure permet de vérifier indépendamment de toute optimisation que F_C diminue pour des enrichissements faibles ($< 3\%$) puisque le taux de variation du burnup R_B est bien supérieur au taux de variation de l'enrichissement R_C . De plus, les valeurs de C_F données montrent que plus l'enrichissement augmente, plus le coût du combustible augmente rapidement. En effet, l'écart est de $269\$/kg$ entre 1.5 et 2.0% d'enrichissement, et il devient $289\$/kg$ entre 2.5 et 3.0%. Pour le burnup, l'effet inverse est observé. En effet, l'écart passe de $11.17GWj/t$ à $9.17GWj/t$ pour les mêmes valeurs de variations d'enrichissement. Ceci explique donc que la fonction objectif diminue de moins en moins vite en fonction de l'enrichissement.

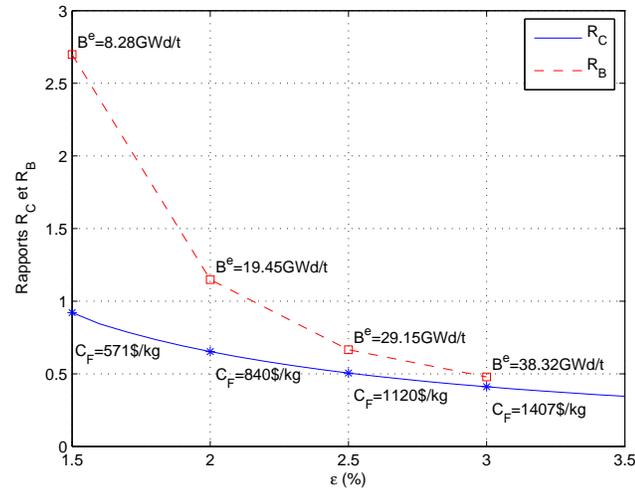


Figure 5.16 Rapport R_B et R_C en fonction de l'enrichissement.

5.5.3 Résultats optimaux

Afin de savoir si le choix du point de départ (distribution de burnup et d'enrichissement) influence les résultats, nous avons repris les points obtenus avec un poids $w_\epsilon = 10^{12}$ aux tableaux 5.11 et 5.12, et continué l'optimisation sans limiter le nombre d'itérations. Les résultats pour les deux méthodes à partir d'un enrichissement faible et grand pour les deux distributions de burnup initiales sont donnés au tableau 5.14. Les colonnes B_i correspondent aux burnups de chaque zone. En règle générale, les deux méthodes ne semblent que très peu influencées par le choix du point de départ. En effet, les valeurs optimales non seulement de la fonction objectif et de l'enrichissement, mais aussi de la distribution de burnup sont similaires. Quelque soit le point initial, nous pouvons cependant noter que la méthode MS donne des résultats légèrement plus optimisés que la méthode MM: une différence de ~ 0.01 sur F_C est observée ($\approx 0.25\%$). La méthode MM semble aussi moins fiable étant donné l'optimum (F_C) obtenu à partir de 1.6% d'enrichissement et une

Tableau 5.14 Résultats des optimisations complètes ($w_\varepsilon = 10^{12}$).

MS										
ε_{ini}	B_{ini}	F_C	B_1	B_2	B_3	B_4	B_5	B_6	B_7	ε_f
1.6	AECL	4.2145	39.64	39.77	39.90	36.75	29.92	32.07	24.48	3.446
1.6	flat	4.2163	39.36	39.45	39.57	36.44	29.53	31.55	25.64	3.424
3.4	AECL	4.2348	38.45	38.58	38.70	35.57	28.82	30.77	23.15	3.363
3.4	flat	4.2149	39.41	39.49	39.60	36.47	29.55	31.58	25.77	3.426
MM										
ε_{ini}	B_{ini}	F_C	B_1	B_2	B_3	B_4	B_5	B_6	B_7	ε_f
1.6	AECL	4.2237	39.10	39.23	39.36	36.21	29.42	31.48	23.89	3.409
1.6	flat	4.2810	35.87	35.85	35.95	32.89	26.02	27.59	23.48	3.173
3.4	AECL	4.2260	38.97	39.11	39.23	36.09	29.31	31.34	23.68	3.400
3.4	flat	4.2225	38.94	39.02	39.13	36.01	29.11	31.07	25.46	3.393

distribution uniforme initiale du burnup.

Le nombre de calculs de flux nécessaire à la minimisation du coût en combustible par la méthode PQL (dernière étape de MS et MM) peut être comparé pour les optimisations complètes. Le nombre total d'itérations requises correspondant aux calculs du tableau 5.14 est donné au tableau 5.15. Le nombre d'itérations requises pour trouver un point réalisable est rappelé entre parenthèses. Comme nous venons de le mentionner, il est plus long de trouver un point réalisable avec la distribution initiale uniforme du burnup (cas "flat"). Par contre, ce point est plus proche du minimum global quelque soit l'enrichissement initial et la méthode choisie, car le nombre d'itérations pour converger sur l'optimum est plus petit pour les cas "flat" que pour les cas "AECL". Ce nombre est obtenu par la soustraction du nombre total moins le nombre entre parenthèses.

Un des buts du design du ACR-700 est d'assurer une réactivité du vide négative. Ceci est assuré par l'utilisation de dysprosium (Dy) dans le combustible neuf. Nous avons remarqué à la section 5.3.3 qu'avec ce design de cellule, la distribution du burnup dans le réacteur (dans le domaine réalisable) n'a que peu d'influence sur la réactivité du vide. Au contraire, la concentration en Dy et la valeur de l'enrichissement devrait avoir une plus grande influence sur la réactivité du vide dans le cœur ρ_V . Pour vérifier ceci, nous avons regroupé dans le tableau 5.16 les

Tableau 5.15 Effort de calcul pour les optimisations complètes.

MS		
ε_{ini}	B_{ini}	ϕ
1.6	AECL	279(22)
1.6	flat	304(101)
3.4	AECL	240(17)
3.4	flat	257(71)
MM		
ε_{ini}	B_{ini}	ϕ
1.6	AECL	300(29)
1.6	flat	221(65)
3.4	AECL	293(21)
3.4	flat	218(36)

valeurs de la réactivité du vide cœur complet correspondant aux configurations optimales $\rho_{V,fin}^f$ et à celles du point réalisable $\rho_{V,deb}^f$ trouvées pour les différents calculs présentés au tableau 5.14.

Une variation de 3 mk est observée entre les différents calculs de ρ_V^f au début et à la fin de la minimisation de F_C . Même si la réactivité du vide varie plus avec l'enrichissement et la concentration de Dy qu'avec la distribution du burnup seule, la variation reste faible par rapport à sa valeur absolue. De plus, cette dernière reste bien en dessous de la limite négative d'une part, et elle diminue de la même manière que la fonction coût d'autre part. Ce second point est valable pour le rapport enrichissement initial/concentration de Dy de $R = 6$ que nous avons imposé. Si la concentration en Dy est diminuée, la réactivité du vide va augmenter (devenir moins négative). Nous n'avons pas testé ce paramètre pour deux raisons: premièrement, une limitation du logiciel à savoir l'interpolation des propriétés nucléaires à plus que 2 variables était impossible au moment où les tests ont été conduits, deuxièmement une question de temps, nous avons reçu le design final du ACR-700 que tardivement dans la conduite du projet.

Tableau 5.16 Réactivité du vide pour les configurations optimales obtenues avec MS (mk).

méthode	$\varepsilon_{ini}(\%)$	dist. B_{ini}	$\rho_{V,deb}^f$	$\rho_{V,fin}^f$
MS	1.6	AECL	-35.00	-38.43
MS	1.6	flat	-35.50	-38.38
MS	3.4	AECL	-35.40	-38.23
MS	3.4	flat	-35.91	-38.39
MM	1.6	AECL	-34.97	-38.34
MM	1.6	flat	-35.48	-37.74
MM	3.4	AECL	-35.32	-38.32
MM	3.4	flat	-35.90	-38.31

Nous avons également calculé la réactivité du vide en damier ρ_V^{ck} pour les mêmes configurations qu'au tableau 5.16. Elle varie entre -12.7 et -13.1 mk pour les configurations correspondant aux points réalisables et est égale à environ -14.1mk pour les configurations optimales (qui sont sensiblement les mêmes, voir tableau 5.14). Les mêmes commentaires s'appliquent à ρ_V^{ck} qu'à ρ_V^f .

5.5.4 Contrainte sur la réactivité du vide

La réactivité du vide, étant très négative, il ne paraît pas nécessaire de la considérer comme contrainte dans le problème d'optimisation (2.18) pour ce design de réacteur. Pour d'autres designs par contre, la réactivité du vide pourrait représenter une contrainte active. Nous avons prévu dans la conception des méthodes MS et MM de pouvoir tenir compte d'une telle contrainte. Il nous a donc paru utile de tester nos méthodes dans un tel cas. Pour cela nous avons envisagé le même problème de l'optimisation de la gestion du combustible du ACR-700, en modifiant la contrainte sur la réactivité du vide plein cœur de simplement négative à inférieure à -36mk, cette valeur étant fixée arbitrairement (valeur intermédiaire dans les différents résultats du tableau 5.16). L'enrichissement du combustible

neuf est initialement de 1.6% pour garantir que la contrainte ne soit pas respectée initialement, et donc pour mieux tester les méthodes.

5.5.4.1 Méthode MS

Les résultats pour la méthode MS sont présentés à la figure 5.17. La figure a) correspond aux itérations pour trouver un point réalisable, et la figure b) représente la dernière étape de la méthode MS, i.e. la minimisation de F_C . Sur la figure a), les traits verticaux représentent la limite entre les étapes de la méthode. Chaque ligne en trait continu entre les verticales correspond donc à une étape. La première courbe (en partant de la gauche) donne la valeur de k_{eff} lors de la minimisation de $(k_{eff} - 1)^2$. L'échelle est donnée par les valeurs minimum et maximum (0.9902 et 1.002 respectivement), et le trait horizontal en tirets/points correspond à la contrainte du problème d'optimisation d'origine, i.e. $k_{eff} = 1$. La courbe suivante représente la puissance canal maximum lors de la minimisation de cette dernière. La troisième courbe correspond à la réactivité du vide. L'échelle pour ces deux dernières courbes est donnée par la valeur maximale (8643kW et -35.2mk) et la valeur de la contrainte correspondante du problème d'optimisation (2.18) représentée par le trait horizontal en tirets/points (7300kW et -36mk). La courbe en trait plein sur la figure 5.17b représente le coût en combustible. L'échelle est donnée par l'axe vertical de gauche de la figure. Enfin, les lignes en tirets sur les deux graphiques représentent la valeur de la contrainte quadratique S pour chacune des étapes. L'échelle est commune à toutes les courbes et est donnée par l'axe vertical de droite des figures 5.17a et 5.17b. Seul les points valides sont retenus. Un petit plateau sur les courbes donnant les fonctionnelles correspond donc à des itérations non valides.

La valeur finale de la fonction objectif est supérieure lorsque la réactivité du vide

ρ_V^f est prise en compte ($F_C \approx 4.3$, figure 5.17b) que lorsqu'elle ne l'ai pas ($F_C \approx 4.2$, tableau 5.14). Cela ne devrait normalement pas être le cas, car les résultats du tableau 5.16 montrent que la réactivité du vide ρ_V^f diminue avec la fonction de coût F_C . Cependant, le but de cette simulation n'est pas tant de comparer l'optimum trouvé avec les résultats précédents que de démontrer que cette méthode fonctionne jusqu'au bout.

Au début de chaque étape de la méthode, la contrainte quadratique S est grande (2000) pour converger le plus rapidement possible. Cependant, quand un point est jugé non-valide, généralement à cause de l'approximation due à la linéarisation, cela se traduit par une diminution de la contrainte quadratique S . Par ailleurs, nous pouvons également remarquer que S remonte parfois (contrainte quadratique multipliée par 2). Cela a pour but d'accélérer la convergence. L'idée part du principe que si le pas d'avance est valide plusieurs fois de suite, l'optimisation pourrait aller potentiellement plus vite si le pas d'avance, i.e. la contrainte quadratique, est plus grande. Nous avons choisi 4 comme le nombre d'itérations successives $N_{it,val}$ pour augmenter S . Pour l'étape de la minimisation de la réactivité du vide, cela semble un bon choix comme paramètre d'auto-adaptation de S . Par contre pour la minimisation de F_C , cela est trop rapide car la plupart du temps cela s'est traduit par une itération non-valide. Nous devons cependant faire remarquer que lorsqu'une itération est non-valide les gradients ne sont pas recalculés à l'itération suivante mais directement repris. Une itération non-valide ne coûte donc qu'un seul calcul de flux. Il est donc préférable d'avoir une valeur $N_{it,val}$ trop petite plutôt que trop grande, car pour la plupart des optimisations, il est plus rapide de calculer un flux que un ensemble complet de gradients. Enfin, à la toute fin de l'optimisation, nous pouvons observer que la contrainte quadratique réaugmente beaucoup, mais que la fonction F_C ne diminue presque plus. En fait, le minimum étant presque atteint le pas d'avance calculer est petit et n'a plus besoin d'être limité par S pour rester

valide.

5.5.4.2 Méthode MM

La méthode MM s'est malheureusement avérée incapable de résoudre le problème d'optimisation (2.18) avec la réactivité du vide comme contrainte (active). En fait, aucun point réalisable n'a été trouvé, le lagrangien augmenté ayant convergé sur un autre point. Les résultats sont donnés au tableau 5.17. γ_ρ , γ_q et η_k représentent respectivement les poids des contraintes sur k_{eff} , $P_{C,max}$ et ρ_V . Les poids de 50000 et 1000 pour les contraintes sur la réactivité du réacteur et de sa puissance canal maximum respectivement correspondent aux valeurs utilisées pour les optimisations sans tenir compte de la réactivité du vide. Les valeurs des contraintes à la fin du lagrangien augmenté sont également données. Enfin, les nombres d'itérations externes It_e et de calculs de flux ϕ sont illustrés.

Les résultats montrent que la contrainte sur la criticité du cœur est toujours respectée. Si le poids initial sur la contrainte de réactivité du vide γ_ρ est petit, alors celle-ci ne respecte pas la limite prescrite, mais la puissance canal maximum est proche de sa limite. Une analyse plus détaillée montre qu'à la dernière itération externe, l'optimisation tendait à réduire la réactivité du vide au détriment de $(k_{eff} - 1)$. Le poids de la contrainte de réactivité du vide étant petit, tous les pas d'avance calculés étaient invalides, même avec une contrainte quadratique petite. Les variables de décision n'ayant pas changées à cette itération externe, la convergence a été obtenue. Dans le cas d'un grand poids initial sur la contrainte de réactivité du vide γ_ρ , la contrainte de réactivité est quasiment respectée, mais celle sur la puissance canal maximum ne l'est plus. De plus, le nombre de calcul de flux nécessaire est plus important dans ce cas, surtout lorsque nous le comparons à la méthode MS: 52 pour avoir un point réalisable, plutôt que 471. Ceci

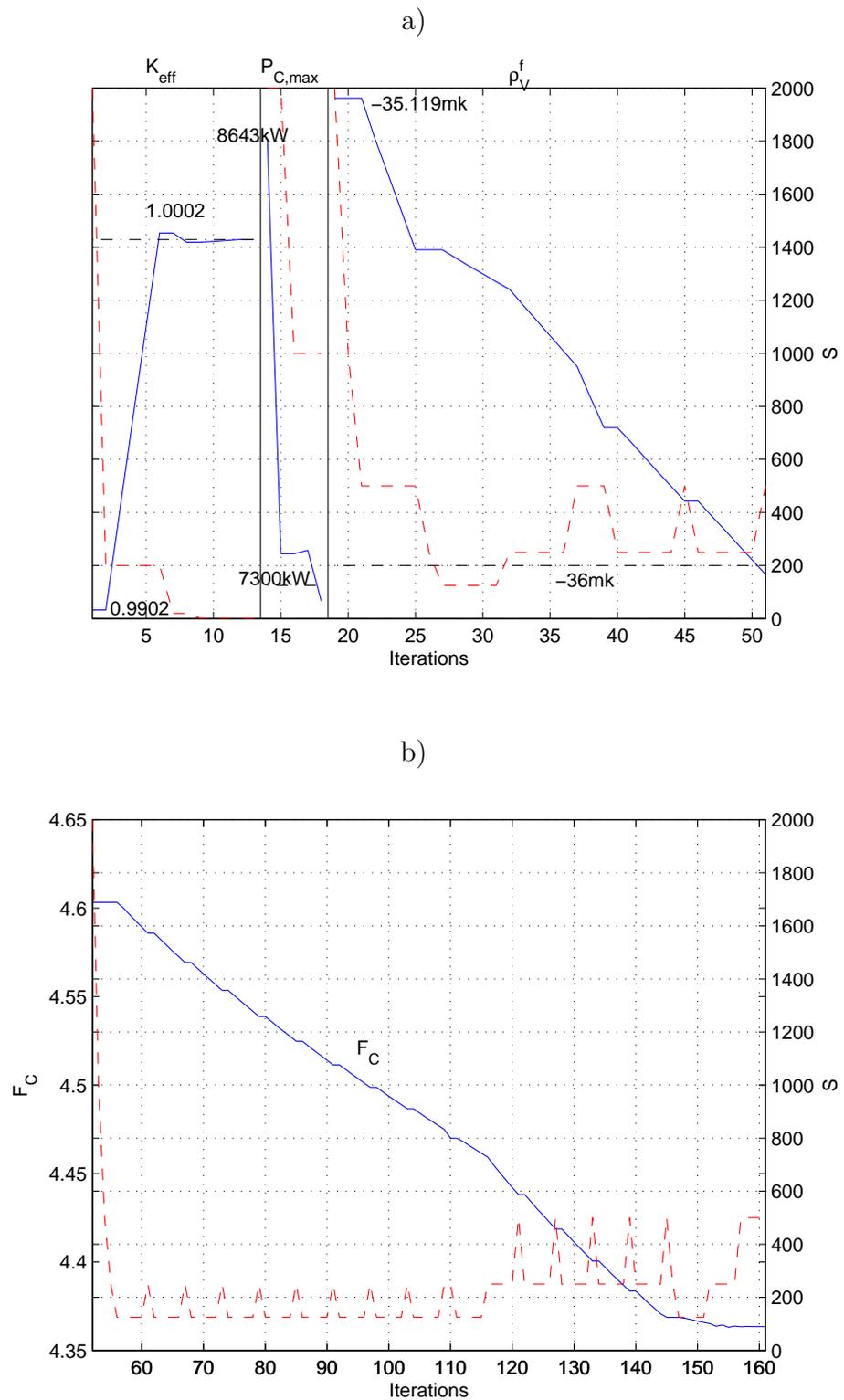


Figure 5.17 Méthode MS avec la contrainte de réactivité du vide.

Tableau 5.17 Résultats avec le lagrangien augmenté.

γ_ρ	γ_q	η_k	k_{eff}	$P_{C,max} kW$ *	$\rho_V^f (mk)$ *	It_e	ϕ
20000	1000	50000	0.9999989	7271	-35.176	4	32
200000	1000	50000	0.9999641	7557	-35.995	49	471
200000	2500	50000	0.9998692	7442	-35.912	33	317

* : rappel $P_{lim} = 7300 kW$, $\rho_{V,lim}^f = -36 mk$

a pour conséquence d'avoir des poids de contraintes qui deviennent extrêmement élevés ($\sim 10^{18}$ pour k_{eff} , $\sim 10^{14}$ pour $P_{C,max}$ et $\sim 10^{19}$ pour ρ_V), et donc entraîne une instabilité potentielle. Ainsi, pour essayer de respecter la contrainte sur la puissance canal en même temps que celle sur la réactivité du vide, nous avons augmenté le poids de la première tout en gardant une grande valeur pour le poids de la deuxième. Les résultats montrent qu'une contrainte s'est améliorée au dépend de l'autre. Le compromis entre les différents poids de contraintes est donc très difficile à faire. La méthode MM ne semble donc pas indiquée pour les optimisations avec la réactivité du vide comme contrainte et l'enrichissement comme variable de décision.

5.6 Conclusion

Les méthodes multi-étapes et mixtes s'appliquent bien à l'optimisation de la gestion du combustible à l'équilibre du rechargement du ACR-700. Pour une optimisation de la distribution de burnup seulement, elles ne sont influencés par le choix du point de départ que pour des cas à très nombreuses variables de décision. L'introduction de l'enrichissement comme variable de décision rend ces deux méthodes plus lentes. Le profil radial de la distribution optimale de burnup ressemble beaucoup à celui obtenu pour le réacteur CANDU-6, à savoir un grand burnup au centre qui diminue en s'éloignant sauf en extrême périphérie où le burnup augmente. ceci est conforme

à la physique des réacteurs.

Pour les calculs avec enrichissement, le poids de cette variable w_ε dans la contrainte quadratique est long à traiter car son choix est “difficile”. Il requiert en effet pas mal d’expériences. Cependant, ce paramètre n’est introduit que pour remettre à l’échelle les différents gradients et le pas d’avance. Un tel concept n’existe pas pour les méthodes méta-heuristiques. De plus, lorsque l’enrichissement est pris comme variable de décision, les calculs sont beaucoup plus long, juste pour une variable de plus. Une méthode méta-heuristique serait alors peut être concurrentielle. Ces deux avantages potentiels pour les méthodes méta-heuristiques nous ont donc poussé à tester une méthode en particulier, la recherche tabou.

CHAPITRE 6

MÉTHODES MÉTAHEURISTIQUES

Depuis 15 ans environ, de nouvelles méthodes dites métaheuristiques ont été développées dans le domaine de l'optimisation en général [40]. L'idée principale de ces approches est de laisser une part au hasard dans le processus de recherche de l'optimum tout en utilisant de manière automatique les tendances qui se dessinent au fur et à mesure des essais, d'où leur aspect heuristique. Ces méthodes métaheuristiques sont donc complètement différentes des approches déterministes telles que les méthodes de gradients proposées jusqu'ici. Seule de l'information sur la fonction objectif est retenue d'un essai à l'autre, aucune indication sur sa pente qui permettrait d'accélérer le processus d'optimisation n'est calculé. Il y a principalement deux raisons pour justifier cette approche. Premièrement, historiquement, ces méthodes ont été développées pour des fonctions à variables combinatoires ou discrètes. Aucun gradient ne peut donc être calculé pour ces cas. Ensuite, de toujours suivre la plus forte pente ne garantit pas d'arriver au minimum global. Par contre, l'aspect lié au hasard des méthodes permet de sortir d'un optimum local. L'apprentissage au fur et à mesure des essais permet cependant de biaiser le hasard pour chercher dans des zones du domaine de recherche qui sont prometteuses. Nous reviendrons plus en détail sur le fonctionnement des différents algorithmes qui ont été développés en général. Par contre, nous pouvons déjà dire que le nombre d'essais pour "apprendre" peut être grand, ce qui se traduit en général par de long temps de calculs.

Grâce à l'énorme amélioration des capacités de calculs des ordinateurs, ces méthodes métaheuristiques ont été appliquées à la gestion du combustible des réacteurs

nucléaires. Trois méthodes ont été utilisées spécifiquement pour l'optimisation de la gestion du combustible dans les différentes filières nucléaires, les algorithmes génétiques (GA), le recuit simulé (SA) et la recherche tabou (TS). Nous décrirons rapidement le principe de chacune de ces méthodes dans la section suivante, avant de voir comment elles ont été appliquées à la gestion du combustible. La recherche tabou a été retenue dans notre cas de l'optimisation de la gestion du combustible des réacteurs CANDU. En effet, elle semble à priori plus facile d'utilisation, et ses différentes étapes de fonctionnement sont plus intuitives. Nous décrirons donc plus en détail la théorie nécessaire à son adaptation, avant d'expliquer plus en détail son implantation. Des tests préliminaires de la méthode tabou appliquée à des fonctions analytiques seront ensuite décrits. Ils nous permettront d'acquérir de l'expérience sur cette méthode et surtout sur le réglage de ces différents paramètres.

6.1 Principe général des méthodes métaheuristiques

Comme pour les méthodes duales, les contraintes sont incluses dans la fonction objectif pour toutes les méthodes méta-heuristiques. Un poids est donc attribué à chaque contrainte en fonction de leur ordre de grandeur et de leur importance relative.

6.1.1 Algorithmes génétiques

L'approche des algorithmes génétiques (GA) [41] est basée sur la théorie de l'évolution de Darwin selon laquelle seuls les individus les mieux adaptés parviennent à survivre et à se reproduire. Ainsi, leur ADN est partiellement transmis d'une génération à l'autre. L'ensemble des individus (la population) s'améliore face au défi lancé par le milieu dans lequel il doit survivre. L'adaptation de cette théorie

aux problèmes d'optimisation a été largement illustrée par Goldberg [41]. Un individu est alors représenté par un choix précis des paramètres de décision, la valeur de la fonction objectif et des contraintes est une mesure de sa cohérence avec le milieu. L'idée de génération est quant à elle représentée par les itérations successives pour améliorer la population. Les GA sont les plus utilisés parmi les méthodes métaheuristiques.

Les GA utilisent un processus itératif qui tend à améliorer un ensemble de solutions, et non une seule solution comme dans le cas de la plupart des algorithmes déterministes, dont la programmation mathématique. Pour passer d'une génération à l'autre, trois étapes sont requises. L'algorithme général est donné par le schéma suivant:

- Un ensemble de vecteurs de décision $\{\vec{X}\}$ est choisi au hasard. Il correspond à la population initiale.
- 1) Chaque individu est évalué par rapport à son environnement, c'est-à-dire que la valeur de la fonction objectif et des contraintes est calculée. Elle est appelée l'étape de *sélection*. Ceux qui ont les meilleurs résultats ont une probabilité plus grande de se reproduire pour la prochaine génération. La taille de la population est généralement gardée constante (même si ce n'est pas un pré requis). Ainsi, les meilleurs individus se reproduiront en général plusieurs fois et les pires ne se reproduiront pas. La reproduction est cependant faite selon un choix aléatoire biaisé (le biais correspondant aux résultats de chaque individu), ce qui laisse une chance à de mauvais individus de se reproduire. À la fin de cette étape, une population intermédiaire est obtenue avec tous les individus de la génération précédente qui vont se reproduire (éventuellement plusieurs fois).
 - 2) Une partie de la population intermédiaire est directement sauvée pour la

prochaine étape. Les autres individus sont mis au hasard par paire. Cette étape est nommée *croisement*. Le choix pour un individu d'être croisé ou directement reproduit est fait au hasard selon une probabilité de croisement. Pour une paire d'individus dits "parents", deux nouveaux individus ("enfants") vont être créés lors du croisement. Trois types de représentations d'individus sont possibles: une représentation binaire, une représentation réelle ou une combinaison d'entiers. Suivant la représentation des individus (le type des paramètres de décision), la méthode de croisement varie [41].

- 3) Chaque individu peut voir son ADN (les paramètres de décision) changer aléatoirement. Cela correspond à la *mutation* spontanée. Là encore, cette mutation se manifeste différemment suivant la représentation des individus [41, 42].
- 4) La génération suivante est obtenue. Tout le processus peut être répété plusieurs fois pour améliorer la population (retour à l'étape 2). Le critère d'arrêt peut être essentiellement de deux types: soit un nombre fixé de générations, soit une fonction objectif (en moyenne et/ou la meilleure de la population) qui ne s'améliore plus.

6.1.2 Recuit simulé

Une autre méthode couramment utilisée est le recuit simulé (*simulated annealing*) développée indépendamment par Kirkpatrick et al [43] et par Cerny [44]. L'origine de cette technique vient d'un parallèle avec la cristallographie. Contrairement aux GA, cette méthode utilise un seul individu. Le but de la méthode est donc d'améliorer cet individu au fur et à mesure des itérations. Pour cela, l'algorithme suivant est suivi:

- Un vecteur de décision \vec{X} est choisi au hasard. Il correspond à l'individu initial.

- 1) Cet individu est légèrement modifié aléatoirement.
- 2) Si ce changement améliore la solution, le nouvel individu est automatiquement conservé. Dans le cas contraire, la fonction objectif a donc empiré. Cependant, l'individu a une certaine probabilité d'être conservé, probabilité qui est donnée par la loi de Boltzman:

$$P = \exp\left(-\frac{\Delta f}{CT}\right) \quad (6.1)$$

où Δf représente la valeur absolue de la détérioration de la fonction objectif, C est une constante pour mettre à l'échelle, T est une température qui peut être ajustée.

La probabilité de conserver un individu est donc dictée par la température T . En cristallographie, l'équation (6.1) donne la probabilité qu'un défaut se propage. Plus la température du corps est grande plus cela est facile, car les barrières de potentiels deviennent moins difficiles à franchir. Ce phénomène est obtenu simplement en chauffant le matériau, ce qui correspond dans le jargon des métallurgistes à un recuit (d'où le nom de la méthode d'optimisation). Passer d'un minimum local à un autre, en optimisation, revient à franchir une barrière de potentiel, en cristallographie.

- 3) La température est mise à jour au besoin. En effet, plus la probabilité de conserver un point dont la fonction objectif s'est détériorée est grande, plus il est facile de sortir d'un minimum local, par contre, moins la convergence est rapide. La température est donc réduite au bout d'un certain nombre d'itérations sans amélioration de la fonction objectif. Le critère d'arrêt peut être une température devenue trop petite, un vecteur d'état ou une fonction

objectif qui ne changent presque plus, ou un nombre maximum d'itérations.

6.1.3 Recherche tabou

La dernière méthode méta-heuristique utilisée est la recherche tabou (*tabu search*, TS) développée par Glover [45]. La recherche tabou utilise un seul vecteur de décision \vec{X} qu'elle modifie (idéalement améliore) à chaque itération. Le principe général de cette méthode peut être résumé comme suit:

- Un vecteur de décision \vec{X} est choisi au hasard.
- 1) Le voisinage V d'un vecteur de décision est défini comme l'ensemble des vecteurs qui sont obtenus par une perturbation de \vec{X} . Un sous-volume ou sous-ensemble N du voisinage $V(\vec{X})$ du vecteur de décision est choisi au hasard. N peut être égal à V , si ce dernier est suffisamment petit.
 - 2) L'optimum sur l'ensemble N est alors recherché et noté \vec{X}^* . La valeur de la fonction objectif correspondante est F_C^* .
 - 3) Si la fonction objectif F_C^* n'a pas été améliorée par rapport à la meilleure solution trouvée jusqu'à présent, pendant un nombre fixé d'itérations consécutives, alors la convergence est atteinte, l'algorithme s'arrête.
 - 4) Sinon l'individu \vec{X}^* devient le nouveau point de départ de l'itération suivante même si la fonction objectif F_C^* n'est pas améliorée par rapport au point précédent. Pour éviter que l'algorithme ne tourne en rond, et forcer une exploration de l'espace des solutions, le vecteur \vec{X}^* (ou une partie de celui-ci) est alors rendu tabou pour un nombre fixé d'itérations. Retour à l'étape 1.

6.1.4 Considérations générales

Étant donné que tous les cas de figures possibles ne sont jamais testés, l'optimum trouvé n'est pas forcément global. Cependant, l'aspect probabiliste de ces méthodes peut permettre d'explorer plusieurs recoins de l'espace des solutions possibles, suivant les options et les ajustements choisis (probabilités de croisement et de mutation / température / taille du voisinage). Ceci représente un avantage par rapport aux méthodes déterministes qui utilisent les gradients et peuvent ne pas sortir d'un minimum local.

Ces algorithmes généraux sont la plupart du temps raffinés. Plusieurs petites modifications utilisées dans la littérature seront décrites par la suite.

6.2 Applications à la gestion du combustible

L'application à la gestion du combustible des méthodes métaheuristiques a essentiellement été faite pour des réacteurs à eau légère. Les réacteurs du type PWR et BWR sont rechargés à l'arrêt une fois par longue période de temps (typiquement de 1 à 2 ans). Cela diffère donc complètement des réacteurs CANDU qui sont rechargés tous les jours. La gestion du combustible pour des réacteurs à eau légère consiste alors à choisir: quelle proportion du combustible sort du réacteur, où mettre le nouveau combustible, et enfin où relocaliser la portion de combustible qui reste. Les choix du rechargement sont donc représentés par un ensemble de combinaisons qui correspondent aux déplacements des assemblages dans le cœur. Cet ensemble est appelé le *patron de rechargement*. D'un point de vue optimisation, le vecteur d'état est donc un ensemble de variables combinatoires plutôt que des variables continues, comme dans le cas de la gestion du combustible des réacteurs CANDU à l'équilibre du rechargement. Pour les BWR, la position des

barres de contrôle dans le temps est aussi à optimiser pour non seulement augmenter l'efficacité du combustible, mais surtout pour maintenir le réacteur dans les limites de sûreté entre deux rechargements.

6.2.1 Algorithmes génétiques

Si le cadre général des GA reste relativement constant parmi l'ensemble des travaux effectués jusqu'ici, certaines particularités inhérentes à chaque problème ont poussé les auteurs à des ajustements voire des améliorations de l'algorithme de base. Parmi celles-ci, plusieurs sont à souligner pour une application future à l'optimisation de la gestion du combustible des CANDU.

Premièrement, dans ses travaux Parks [42] a fait en sorte d'avoir un choix diversifié de la population initiale pour couvrir au maximum l'espace des solutions possibles dès le début, et ainsi permettre de réduire le nombre d'itérations ou améliorer l'optimum obtenu. Le degré de diversité entre deux patrons de rechargement X et Y est alors donné par une fonction qui tient compte de la différence de facteur de multiplication, $k_{\infty,X} - k_{\infty,Y}$ pour chaque position d'assemblage.

Certains patrons de rechargement sont à priori mauvais en ce qui concerne les contraintes de sécurité. Le domaine des solutions possibles peut donc être réduit si l'expertise acquise est utilisée. DeChaine et Feltus [46] ont ainsi utilisé l'expertise acquise pour modifier la population initiale. Les résultats montrent qu'avec un nombre fixe de générations, l'optimum atteint est amélioré. Kobayashi et Aiyoshi [47,48] ont transposé les règles expérimentales pour le repositionnement en matrices de manière à utiliser l'expertise de manière automatique. Un exemple de règle est celle de placer le nouveau combustible plus vers le centre sous forme de damier.

Si la fonction à optimiser comprend plusieurs objectifs, le poids relatif de chacun

n'est pas forcément facile à déterminer. De plus, le choix du poids à priori peut éliminer de meilleures solutions. Une approche avec plusieurs fonctions objectifs non-dominantes a donc été développée par Parks [42]. L'auteur garde en archive les meilleurs résultats et les remplace le cas échéant. Ceci permet de pouvoir choisir le poids seulement à la fin de l'optimisation.

La parallélisation des GA est très facile. Elle peut cependant être faite dans deux optiques différentes. La première consiste simplement à diviser l'évaluation d'une même population sur plusieurs processeurs. La deuxième optique envisagée par Yamamoto et Hashimoto [49] consiste à traiter une population distincte sur chaque processeur et d'échanger un individu de temps en temps (selon une probabilité dite de *propagation*). Cette méthode permet d'avoir plusieurs probabilités de croisement et de mutation en même temps. Les résultats obtenus par l'auteur démontrent qu'une telle méthode permet non seulement une diminution du temps de calcul qui suit quasiment le speed-up idéal, mais surtout une amélioration des résultats finaux. Ce gain de qualité du résultat est maximum quand l'individu échangé est le meilleur de la population locale ("elite migration"). Yamamoto a utilisé cette méthode pour déterminer le patron de rechargement. Pereira et Lapa [50] ont obtenu des résultats semblables pour une optimisation combinée d'une part de la géométrie et des matériaux de la cellule et d'autre part des zones d'enrichissement du cœur du réacteur.

6.2.2 Recuit simulé

Une étude réalisée par Yamamoto et Hashimoto [51] a démontré l'applicabilité de la méthode SA pour l'optimisation du patron de rechargement dans un PWR. De plus, une adaptation de cet algorithme à un traitement parallèle donne de meilleurs résultats que pour un calcul sur un seul processeur et avec un nombre total de

patrons de rechargement testés qui est réduit. L'idée des auteurs est d'avoir une optimisation indépendante avec SA sur chaque processeur (chacune ayant sa propre température T). Ainsi, avec le même principe que la migration pour les GA, les solutions de deux processeurs peuvent être échangées selon une probabilité donnée par la loi de Boltzman.

6.2.3 Recherche tabou

L'application de la recherche tabou à la gestion du combustible est récente, et n'est pas très répandue. Les principaux travaux réalisés [52–55] sont appliqués aux réacteurs à eau légère. Ben Hmida et al. ont été les premiers à utiliser la recherche tabou pour l'optimisation de la gestion du combustible [55]. L'application envisagée par les auteurs était d'optimiser le patron de rechargement dans un PWR. Dans leur travaux, Ben Hmida et al. définissent le voisinage V comme l'ensemble des patrons de rechargement que l'on peut obtenir en permutant deux assemblages du patron de rechargement de référence. Certaines règles heuristiques ont été ajoutées pour éviter d'évaluer des vecteurs de décision qui sont physiquement inadmissibles, comme par exemple, il est interdit de permuter 2 assemblages ayant des réactivités très différentes. En plus du caractère tabou appliqué aux assemblages récemment permutés dans le processus itératif d'optimisation, un compteur est appliqué à chacun d'entre eux pour mesurer la fréquence à laquelle ils sont sélectionnés. En interdisant les assemblages trop fréquemment choisis, l'algorithme proposé par Ben Hmida et al. force une exploration du domaine de recherche. Enfin le caractère tabou peut être supprimé s'il conduit à une nette amélioration de la fonction objectif, cela correspond à un *critère d'aspiration*. Selon les résultats avec le code PANTHER, les auteurs concluent que la méthode de recherche tabou est aussi prometteuse que les algorithmes génétiques.

Jagawa et al. [54] ont optimisé le patron de rechargement dans un réacteur BWR. Leur approche est légèrement différente de l'algorithme proposé par Glover. Premièrement, le voisinage se réduit à un seul élément: le choix de 2 assemblages à intervertir dans le patron de rechargement courant, ces 2 assemblages deviennent tabous si leur permutation entraîne une amélioration de la fonction objectif. Ensuite, la fonction objectif est différente suivant la valeur des contraintes. Ainsi quand celles-ci ne sont pas toutes respectées, le but de l'optimisation est de minimiser la violation des contraintes. Dans le cas contraire, le patron de rechargement est modifié pour maximiser k_{eff} à la fin du cycle.

Dans leurs travaux, Castillo et al. ont dans un premier temps optimisé le patron de rechargement pour un réacteur à eau bouillante. [53] Dans ce cas, le vecteur de décision est alors un ensemble de combinaisons. Ils ont ensuite appliqué la recherche tabou à l'optimisation de la position des barres de contrôle entre 2 rechargements. [52] Les barres de contrôles peuvent avoir seulement un nombre discret et fini de positions. Le problème consiste donc à une optimisation en nombres entiers. Castillo et al. proposent deux principales améliorations à l'algorithme général. Dans leurs travaux, ils ont sauvegardé dans la liste des tabous seulement la variation du vecteur \vec{X} et non le vecteur au complet. La première modification apportée par les auteurs est le *critère d'aspiration* qui permet, comme dans [55], de supprimer le caractère tabou d'une variation possible du vecteur si le résultat est prometteur. Le deuxième changement consiste à arrêter d'évaluer les voisins du sous-ensemble N dès qu'un meilleur point que le point actuel est trouvé.

6.3 Recherche tabou en variables continues

La méthode de la recherche tabou a originalement été développée par Glover pour une application avec des variables combinatoires. [45] Les applications faites en ges-

tion du combustible des réacteurs sont également limitées à des variables combinatoires ou en nombres entiers. Une application de la recherche tabou à l'optimisation du burnup moyen de sortie et de l'enrichissement dans les réacteurs CANDU à l'équilibre du rechargement requiert donc une adaptation de la méthode aux variables continues.

Une des premières adaptation aux fonctions à variables continues de la recherche tabou, fidèle à l'algorithme de Glover, a été faite par Siarry et Berthiau [56]. L'algorithme général reste en effet le même. Cependant les notions de voisinage et de liste de tabou sont redéfinies. Pour des variables continues, les perturbations possibles sont infinies, le voisinage est donc un ensemble infini. L'ensemble N pour lequel la fonction objectif va être évaluée pour chaque élément, est alors défini en utilisant un nombre fini de couronnes concentriques autour du vecteur de décision courant \vec{X} . Siarry et Berthiau choisissent ensuite un point au hasard dans chacun de ces anneaux pour former le sous-ensemble N . Des exemples de voisinage sont présentés à la figure 6.1. Le nombre de couronnes concentriques détermine la taille du sous-ensemble N . Les auteurs proposent de définir le rayon de chacun des anneaux soit par une suite géométrique, soit par une suite linéaire ou encore de telle sorte que les volumes de chaque couronne soient égaux. En ce qui concerne le caractère tabou des éléments déjà testés, Siarry et Berthiau proposent de rendre tabou toute une petite boule autour des points de la liste.

Chelouah et Siarry [2,57] ont proposé une amélioration de l'algorithme proposé par Siarry et Berthiau. Ils ont introduit les concepts d'exploration et d'intensification. L'exploration correspond en fait à l'algorithme général proposé par Glover appliqué à tout l'espace des solutions. L'intensification est une recherche localisée d'un optimum. Après avoir trouvé le meilleur élément \vec{X}^* de N , celui-ci est comparé au vecteur de décision courant \vec{X} . Si ce dernier donne un meilleur résultat que pour \vec{X}^* , alors \vec{X} est dit prometteur, dans le sens où il pourrait être un optimum

local. Chelouah et Siarry ont dans un premier temps [57] proposé que, pendant la phase d'exploration, tous les vecteurs de décision prometteurs soit rassemblés dans une liste. À la fin de la phase d'exploration, l'optimisation est alors relancée à partir de l'élément le plus prometteur de cette liste, en réduisant l'espace des solutions afin de raffiner le résultat ce qui correspond à une phase d'intensification. Plus récemment, [2] ils ont présenté un algorithme hybride recherche tabou et méthode de *Nelder-Mead simplex*. Une optimisation locale, qui correspond à la phase d'intensification, est alors faite avec la méthode *Nelder-Mead simplex* dès qu'un vecteur de décision prometteur est détecté. Cette méthode consiste à tracer autour de l'optimum local un polyèdre dont la taille est réduite à chaque itération interne. Cet optimum local trouvé est alors mis dans la liste des prometteurs qui fonctionne de manière similaire à celle des tabous.

Chelouah, Siarry et Berthiau dans leurs différentes publications ont utilisé des fonctions analytiques dont les minimums locaux et globaux sont connus pour tester leurs implantations de la méthode de recherche tabou.

6.4 Implantation de la méthode de recherche tabou

Toutes les méthodes métaheuristiques utilisent des nombres aléatoires. L'algorithme utilisé pour leur génération est présenté à l'annexe IV.

Avant de rentrer trop dans les détails de l'algorithme implanté, nous devons faire une précision très importante. La méthode TS, comme les autres méthodes métaheuristiques ne permettent pas de minimiser une fonction objectif tout en respectant des contraintes séparément. Ceci est un donc problème majeur qui est en fait réglé en rajoutant la violation des contraintes à la fonction objectif originale f . La manière utilisée est d'associer un poids aux contraintes exactement comme

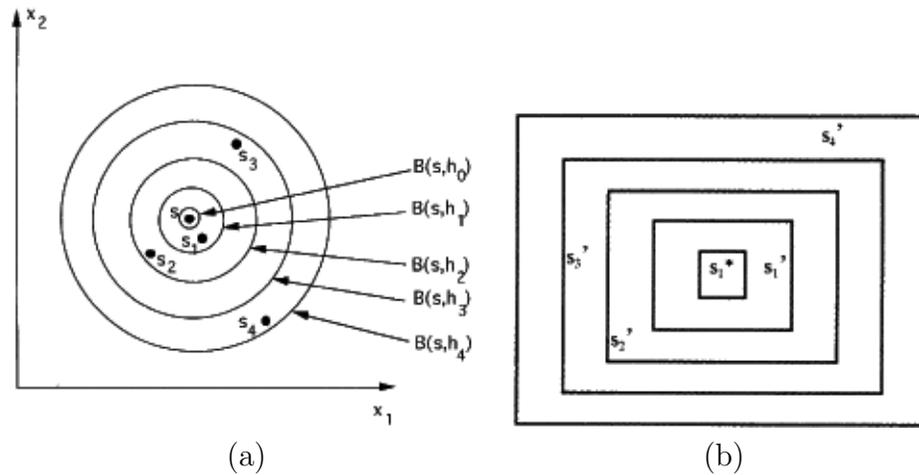


Figure 6.1 Exemples de répartition des éléments de N dans le voisinage de la solution courante \vec{X} .

(a) hypersphère (Siarry) (b) hyperrectangle (Chelouah)

pour la méthode de pénalisation externe, à la différence que ce poids reste constant tout le long de l'optimisation. La nouvelle fonction objectif, que nous avons notée P dans la section 4.3.1, est alors notée f_{TS} pour la méthode TS.

Pour l'implantation de la recherche tabou au code DONJON, plusieurs des modifications de l'algorithme de Glover présentés dans les sections précédentes ont été retenues. Différents algorithmes ont été testés sur des fonctions analytiques avant de passer à la gestion du combustible. La figure 6.2 résume les grandes étapes communes des tests faits dans DONJON à la fois sur les fonctions analytiques et sur l'optimisation de la gestion du combustible. Cette figure peut être vue à deux niveaux: les rectangles en traits doubles présentent les grandes étapes de la méthode TS, et à l'intérieur, le détail des sous-étapes et de différents compteurs pour la convergence.

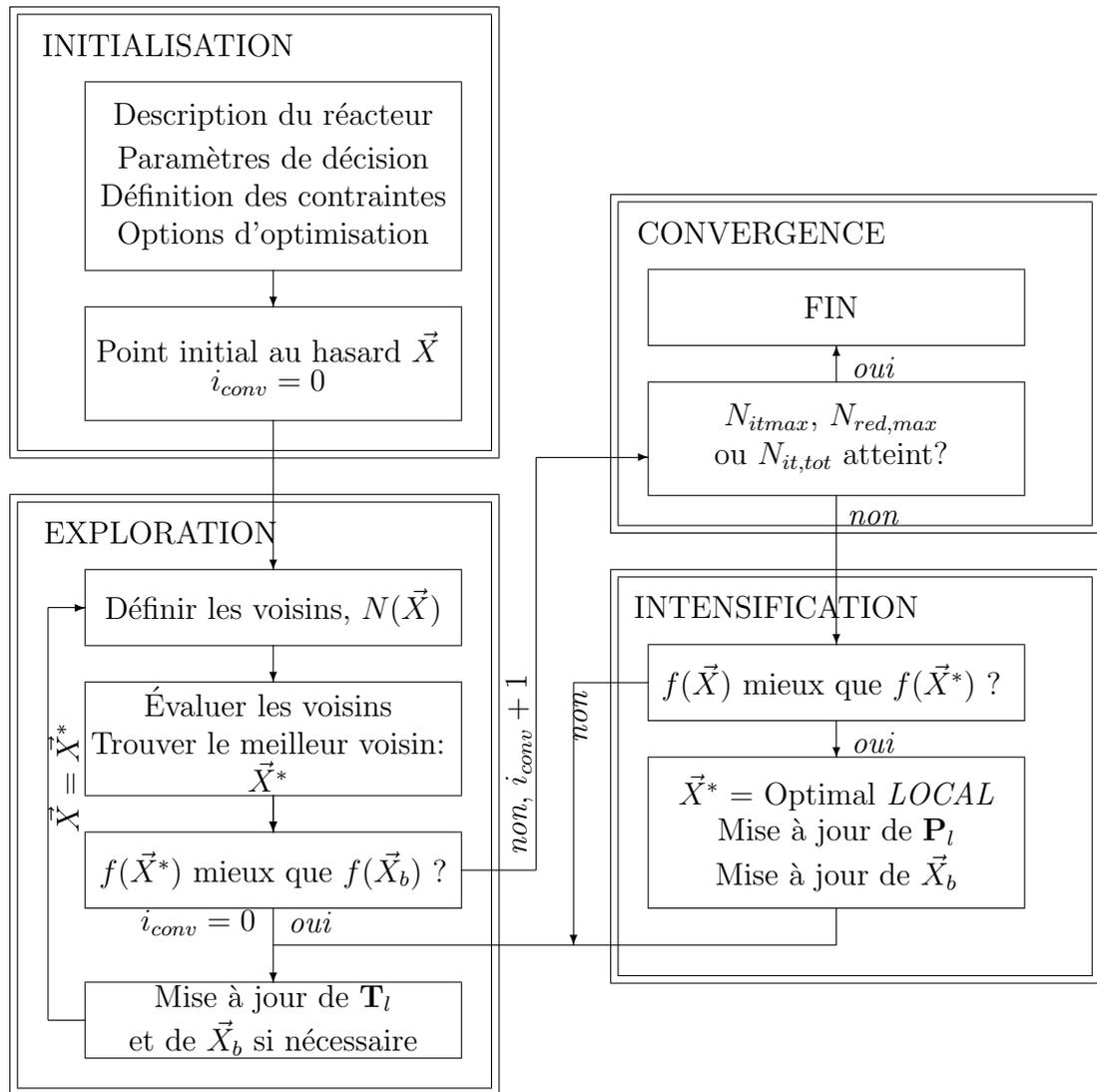


Figure 6.2 Algorithme général pour l'optimisation avec la méthode de recherche tabou.

6.4.1 Initialisation

Dans un premier temps, le problème d'optimisation est défini. Pour la gestion du combustible, cela consiste à décrire le réacteur, les options de rechargement (ex: zones de combustion, enrichissement, ...), la fonction objectif, les contraintes et leur poids. Les fonctions analytiques testées sont données à l'annexe V. Nous reviendrons dans la section 6.5 sur le choix de ces fonctions. Ensuite, les options pour la résolution du problème sont choisies. Les plus importantes sont: le nombre de voisin N_{ngb} de l'ensemble de recherche N , le rayon du voisinage R_n , la longueur maximum de la liste tabou L_t et de celle de la liste des prometteurs L_p et finalement le rayon de l'hyper-sphère ou hyper-rectangle R_t ou R_p autour des vecteurs de la liste tabou \mathbf{T}_l et des prometteurs \mathbf{P}_l respectivement. Les rayons R_n , R_t et R_p sont des valeurs normalisées entre 0 et 1. Enfin pour conclure l'initialisation du problème, un point dans l'espace de recherche est pris au hasard. Le compteur pour la convergence i_{conv} est initialement égal à 0. La liste des prometteurs est initialisée en prenant L_p points au hasard dans le domaine de recherche, ce qui permet de définir le seuil d'acceptation comme la moyenne de la fonction objectif de tous les prometteurs.

6.4.2 Exploration

Les grandes lignes de la phase d'exploration correspondent en fait à l'algorithme de base de la méthode TS qui est résumé à la section 6.1.3. La figure 6.2 donne cependant plus de détail et montre également comment le concept de zones prometteuses est introduit. La suite des explications se réfère donc à cette figure seulement.

À chaque itération, les voisins du vecteur de décision \vec{X} qui forme le sous-ensemble N sont choisis. Les trois types de discrétisation proposé par [56] pour le choix des

anneaux ont été implantés. La valeur de la fonction objectif tabou (fonction de coût plus les contraintes) est évaluée pour chacun des éléments de N pour en déterminer le meilleur \vec{X}^* . Si sa fonction objectif correspondante est améliorée par rapport au meilleur point disponible \vec{X}_b , alors ce dernier est mis à jour. Le compteur de convergence i_{conv} est réinitialisé à 0. Dans le cas contraire, i_{conv} est augmenté de 1. S'il n'y a pas eu d'amélioration de l'optimum pendant N_{itmax} itérations consécutives, alors la convergence est atteinte ($i_{conv} > N_{itmax}$), l'algorithme s'arrête. D'autres critères d'arrêt ont été retenus pour certains algorithmes. Nous reviendrons ultérieurement sur les ajouts dans la section 6.5.

Une vérification du caractère prometteur de \vec{X} est faite. Quand la fonction objective de \vec{X} est meilleure que celle de tous ces voisins, alors un minimum local probable est détecté si le seuil d'acceptation n'est pas dépassé. Une phase d'intensification est alors appliquée sur la zone prometteuse autour de \vec{X} ; elle fournit un minimum local qui remplace la valeur de \vec{X}^* (précédemment le meilleur voisin). Dans le cas où la fonction objective de \vec{X}^* est meilleure que celle de \vec{X}_b , cette dernière est mise à jour. \vec{X}^* remplace alors le pire point de la liste des prometteurs, et le seuil est mis à jour. Si le vecteur de décision \vec{X} n'est pas prometteur, alors il est mis directement dans la liste tabou selon la logique FIFO ("First In First Out", remplacement du plus vieux point de la liste).

Le vecteur de décision \vec{X}^* devient ensuite le nouveau point de départ pour l'itération suivante. Les rayons R_n , R_t et R_p peuvent être changer d'une itération à l'autre pour améliorer la recherche des zones prometteuses dans certaines versions de l'algorithme.

6.4.3 Intensification

La phase d'intensification consiste donc à trouver l'optimum local dans la zone prometteuse détectée. Le problème de gestion du combustible reste donc le même (eq. (2.18)), mais l'espace des solutions est restreint au voisinage du vecteur de décision prometteur \vec{X} . La taille du voisinage utilisé varie avec les différents algorithmes testés. La phase d'intensification est effectuée selon deux différentes méthodes d'optimisation. La première utilise directement la méthode des gradients avec en particulier la méthode de pénalisation extérieure, telle que décrite dans la section 4.3.1. La deuxième approche est la méthode de Nelder-Mead.

Après convergence de la phase d'exploration, une phase d'intensification finale à partir du meilleur point trouvé est systématiquement faite avec la méthode des gradients. L'espace de recherche n'est pas réduit pour permettre à cette dernière étape de se rendre jusqu'à l'optimum local, que nous espérons être aussi global. Le pas d'avance de la méthode de gradients est limité initialement à une petite valeur pour ne pas sortir de la vallée contenant le meilleur point.

6.4.4 La méthode de Nelder-Mead

La méthode dite de *Nelder-Mead simplex* [58] (NMS) consiste à utiliser un polytope à n dimensions pour converger vers l'optimum. Le polytope est défini par les points (x_1, \dots, x_{n+1}) , qui peuvent être initialement donnés par $x_{i+1} = x_1 + \delta e_i$, où e_i est un vecteur unitaire selon la dimension i . Nelder et Mead proposent plusieurs transformations élémentaires du polytope pour se rapprocher de l'optimum. La première étape consiste à évaluer la fonction objectif pour tous les sommets du polytope. Les notations suivantes sont prises: x_w , x_{w2} et x_b correspondent au pire point (w), au deuxième pire point ($w2$) et au meilleur (b) respectivement.

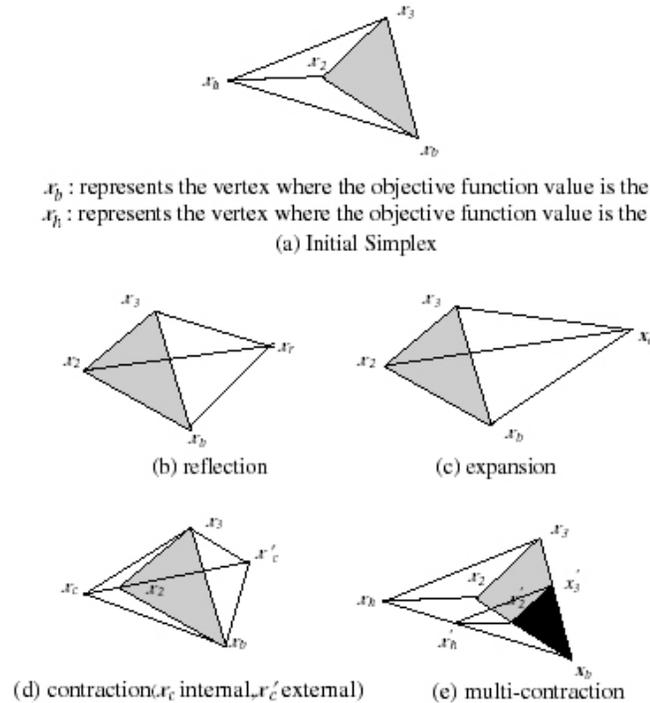


Figure 6.3 Transformations géométriques élémentaires de la méthode Nelder-Mead, pour un cas à 3 variables (Chelouah)

$\bar{x} = \frac{1}{n} \sum_{i \neq w} x_i$ est la moyenne des sommets en excluant le pire. Enfin α , β et γ sont des coefficients utilisés pour les transformations géométriques élémentaires suivantes:

- réflexion: $x_r = (1 + \alpha)\bar{x} - \alpha x_w$
- expansion: $x_e = \gamma x_r + (1 - \gamma)\bar{x}$
- contraction: $x_c = \beta x_w + (1 - \beta)\bar{x}$ (interne)
 $x'_c = \beta x_r + (1 - \beta)\bar{x}$ (externe)
- multicontraction $x'_i = \frac{x_i + x_b}{2}$

Une représentation géométrique pour un cas à trois variables est donnée à la figure 6.3.

La première transformation consiste à faire la réflexion x_r du pire sommet x_w , et d'évaluer la fonction objectif pour ce nouveau point $f(x_r)$. Si elle est meilleure que pour tous les autres sommets, alors une direction prometteuse est trouvée. Une expansion est alors faite selon x_r , et le point x_e est évalué. Si la valeur de la fonction objectif correspondante ($f(x_e)$) est meilleure que celle du point réfléchi ($f(x_r)$), alors x_e remplace le pire sommet du polytope (x_w), sinon x_r est le nouveau point du polytope à la place de x_w . Dans le deuxième cas de figure, la valeur $f(x_r)$ est moins bonne que $f(x_b)$ mais reste meilleure que $f(x_{w2})$. Le point de la réflexion x_r remplace directement le pire sommet x_w . Dans le dernier cas ($f(x_r)$ pire que $f(x_{w2})$), une contraction est effectuée. Si $f(x_r)$ est meilleure que $f(x_w)$, une contraction externe est faite, sinon une contraction interne. La valeur de la fonction au point contracté ($f(x_c)$) est alors comparée à celle du pire point ($f(x_w)$). Si le pire point est moins bon que le point contracté, alors x_c (ou x'_c) remplace x_w . Géométriquement cela correspond à une adaptation du pas d'avance dans cette direction pour mieux suivre la courbe de la fonction objectif. Dans le cas contraire, x_w est mieux que le point contracté. Alors une multicontraction est réalisée pour réduire le pas d'avance dans toutes les directions. Tous les sommets du polytopes sont remplacés par la valeur x'_i correspondante. La fonction objectif est alors évaluée pour les nouveaux sommets. L'algorithme recommence en réordonnant les sommets du nouveau polytope selon la valeur de la fonction objectif. Le critère d'arrêt consiste à estimer la taille de la transformation géométrique entre deux itérations, ce qui se traduit par l'équation suivante:

$$\|x^{k+1} - x_w\|^2 < \varepsilon$$

où x^{k+1} est égal à x_r , x_e , x_c ou x'_c suivant le cas de figure, ε est le critère de convergence.

Aucun gradient n'est donc requis pour cette méthode. Elle est applicable directe-

ment à n'importe quel problème d'optimisation sans contrainte, peu importe sa linéarité ou continuité. Cette méthode tend à suivre la courbe de la fonction objectif. Elle s'apparente donc à une méthode de gradient, et n'est utilisable que pour une recherche *local* d'un optimum.

6.5 Banc d'essais de la méthode recherche tabou

Afin de vérifier l'algorithme de recherche tabou implanté, nous avons utilisé plusieurs fonctions analytiques avec des formes très différentes (annexe V). Trois fonctions à larges vallées ont été envisagées, dont 2 avec une seule vallée (DJ et Zn) et une avec plusieurs vallées (RC). Tous les minima sont globaux pour ces trois fonctions. Ces fonctions sont donc plus utiles pour décrire l'efficacité des algorithmes pour des fonctions simples, et pour voir si les auto-ajustements ne sont pas excessifs. À l'opposé, les autres fonctions présentent des minima locaux. La fonction ES est quasiment constante presque partout sauf pour une seule vallée très profonde mais très localisée. La fonction SH est également constante presque partout, excepté pour quelques petites zones dans lesquelles des extréma très accentués et très rapprochés sont présents. Les fonctions B2 et GP ont la forme générale d'une large vallée avec respectivement de nombreuses et quelques minima locaux, dont les "sous-vallées" locales sont de tailles moyennes. Ces quatre dernières fonctions servent donc à tester l'efficacité de la phase d'exploration. La dernière fonction a pour caractéristique de présenter une seule large vallée mais donc le fond est très étroit, profond et long.

Ces fonctions nous ont également permis d'évaluer différentes variantes/approches de l'algorithme, et donc de connaître l'influence de ces différentes options et divers paramètres. De plus, les quelques approches ainsi que les formes très variées des fonctions analytiques que nous avons envisagées, ont également amené à des recom-

mandations pour le choix des options à retenir pour la gestion du combustible.

6.5.1 Description détaillée des approches

Trois approches assez différentes ont en fait été envisagées, et chacune d'entre elles a reçu une légère variante pour tester l'influence d'une des options possibles. Dans la littérature, il est parfois difficile de connaître toutes les options précises utilisées pour obtenir les résultats présentés. Aussi, nous avons juger important de détailler les choix que nous avons pris pour toutes les approches envisagées, avant de donner les résultats obtenus.

Pour toutes les méthodes, la liste des prometteurs a été initialisée en prenant L_p points au hasard dans tout l'espace de recherche. Ceci permet de définir le seuil d'acceptation. Ensuite, chaque approche peut être résumée par les options suivantes: le nombre de voisins N_{ngb} , le nombre d'itérations maximum sans amélioration du meilleur point N_{itmax} , la longueur de la liste tabou L_t , la longueur de la liste des prometteurs L_p , le rayon initial du voisinage R_n , celui des zones des points tabou R_t , celui des zones des points prometteurs R_p , le nombre d'itérations sans trouver de point prometteurs avant de réduire la taille des divers rayons $N_{npd,max}$ et enfin le nombre maximum de niveaux de réduction automatique des rayons $N_{red,max}$.

La première méthode (#1) se veut la plus *simple* possible. Les rayons R_n , R_t et R_p restent constants, et donc $N_{npd,max}$ et $N_{red,max}$ valent 0. Le critère d'arrêt est donc donné par N_{itmax} seulement. Les options choisies sont résumées dans le tableau et les points suivants:

Méthode 1: méthode générale

N_{ngb}	N_{itmax}	L_t	L_p	$R_n(*)$	$R_t(*)$	$R_p(*)$
10	100	20	10	0.2	0.01	0.02

Phase d'exploration:

Distribution des voisins: linéaire

Phase d'intensification:

$$S_{ini} = (X_{max} - X_{min}) \times R_n$$

Réduction du pas: 1/2

Méthode 1_50:

Comme la méthode 1 avec $N_{itmax} = 50$

La deuxième méthode (#2) a pour idée principale d'adapter les rayons automatiquement à la forme de la fonction. En effet, selon la taille des vallées, un voisinage trop petit ne permet pas de sortir d'un minimum local le cas échéant. Dans le cas contraire, les voisins sont trop éloignés et la recherche est faite au pur hasard sans *apprentissage* de l'algorithme. Le critère pour diminuer les rayons correspond aux cas où nous ne détectons pas de zones prometteuses. La phase d'intensification est faite avec une méthode de gradients: la méthode de pénalisation externe. Cependant, seules les itérations internes de l'algorithme présenté à la section 4.3.2 sont effectuées. Ceci permet alors de garder les poids des contraintes constants (puisque'ils ne sont pas mis à jour), et donc d'avoir toujours la même définition de la fonction objectif. Cela est indispensable pour la recherche tabou si nous voulons comparer les différents points obtenus avant et après la phase d'intensification. Les options sont encore une fois résumées dans le tableau et les points suivants:

Méthode 2: méthode autoadaptative

N_{ngb}	N_{itmax}	L_t	L_p	$R_n(*)$	$R_t(*)$	$R_p(*)$	$N_{npd,max}$	$N_{red,max}$
10	100	20	10	0.2	0.01	0.02	4	6

Phase d'exploration:

Distribution des voisins

- initialement: linéaire, pour n'avoir aucun biais
- après phase d'intensification: isovolume, pour s'éloigner du point courant
- après amélioration du meilleur point: géométrique $q = 1.25$, pour s'approcher du point courant

Réduction des rayons si aucun prometteur détecté pendant $N_{npd,max}$ itérations successives

Phase d'intensification:

$$S_{ini} = (X_{max} - X_{min}) \times R_p$$

Réduction du pas: 1/2

Augmentation des rayons ($\times 2$) s'ils sont différents de ceux du point prometteur précédent

Méthode 2-n4:

Comme la méthode 2 avec $N_{ngb} = 2n_{var}$

Enfin, la dernière approche utilise la méthode du simplex de Nelder pour la phase d'exploitation. Deux variantes ont été testées: sans et avec auto-adaptation des rayons R_n , R_t et R_p :

Méthode Nelder (N_r1):

N_{ngb}	N_{itmax}	L_t	L_p	$R_n(*)$	$R_t(*)$	$R_p(*)$	$N_{npd,max}$	$N_{red,max}$
4	100	20	10	0.2	0.01	0.02	4	1

Phase d'exploration:

Distribution des voisins:

- initialement: linéaire, pour n'avoir aucun biais
- après phase d'intensification: isovolume, pour s'éloigner du point courant
- après amélioration du meilleur point ou réduction des rayons: linéaire, pour n'avoir aucun biais

Remise à 0 du compteur de non obtention de zone prometteuse, si le meilleur est amélioré.

Phase d'intensification:

Méthode Nelder: $(\alpha, \beta, \gamma) = (1, 0.5, 2)$, $\varepsilon = 0.001$

Méthode Nelder (N_r6):

Comme la méthode Nelder (N_r1) avec $N_{red,max} = 6$

En ce qui concerne la phase d'intensification finale, une méthode de gradient est appliquée au meilleur point. Le pas d'avance s'adapte automatiquement pour minimiser le nombre d'itérations. L'idée principale est de commencer avec un pas d'avance suffisamment petit pour éviter de sortir d'une vallée et de rester proche du meilleur point trouvé, le but étant d'améliorer ce dernier avec une méthode qui converge vite. Les options choisies sont les mêmes pour toutes les approches à savoir:

Phase d'intensification finale:

$$S_{ini} = (X_{max} - X_{min}) \times R_p$$

Réduction du pas: moitié

Augmentation du pas ($\times 2$) si valide 3 fois de suite

6.5.2 Résultats et conclusions

L'optimisation de toutes les fonctions analytiques a été reconduite 100 fois et ce avec chaque méthode. Ceci permet de pouvoir juger de la fiabilité des approches. L'efficacité est quant à elle donnée par le nombre d'évaluations de la fonction nécessaire pour trouver la solution. Les résultats sont donnés au tableau 6.1. Le nombre moyen d'évaluations présenté N_{eval} est calculé sur les résultats ayant convergé vers le minimum global seulement. Le critère de réussite pour un calcul d'optimisation est donné par la formule suivante:

$$|F - F_{ana}| < \epsilon_{rel}|F_{ini}| + \epsilon_{abs} \quad (6.2)$$

où F_{ini} est la moyenne de la fonction objectif sur 100 points pris au hasard dans tout l'espace de décision. F est le résultat de l'optimisation. F_{ana} est la valeur analytique, et $\epsilon_{rel} = 10^{-4}$ et $\epsilon_{abs} = 10^{-6}$. Les valeurs de F_{ini} et F_{ana} sont résumées au tableau 6.2. Elles peuvent varier de plusieurs ordres de grandeur l'une par rapport à l'autre, ou être nulle alors que l'autre ne l'est pas.

La différence entre la méthode 1 et 1_50 est d'environ 500 évaluations en général. Cela correspond à 50 itérations avec 10 voisins chacune à évaluer. 50 itérations sans amélioration du meilleur point ($N_{itmax} = 50$) devraient donc être suffisantes. Même si les approches '2' et 'N' ont été faites avec $N_{itmax} = 100$, cela n'a pas rallongé le nombre d'évaluations. En effet, la convergence a été obtenue avec le critère de non détection de zone prometteuse, avant que celui de non détection d'un meilleur point soit actif.

En ce qui concerne l'efficacité des méthodes 2 et 2_n4, 4 voisins (6 pour DJ) semblent largement suffisants pour obtenir de bons résultats rapidement. Par contre, cela se fait au détriment de la fiabilité dans le cas où les vallées sont très localisées

par rapport à l'espace de décision (SH).

Le réglage automatique des rayons R_n , R_t et R_p pour la méthode 2 par rapport à la méthode 1 a un effet très différents suivant les fonctions. Le gain varie ainsi de très grand pour DJ et SH, en passant par faible pour B2 et GP, et quasiment nul pour ES, RC et R2 à très négatif pour Z2. Paradoxalement, ce sont les fonctions avec la plus large vallée et celle avec la plus étroite qui donne les meilleurs résultats. L'autoadaptation des rayons semble en fait plus utile lorsque le nombre de voisin est limité.

La méthode de Nelder N_r1 est beaucoup plus rapide que les deux autres, tout en restant fiable même avec seulement 4 voisins utilisés, sauf pour la fonction SH où elle perd sa précision. Lorsqu'elle est couplée avec une autoadaptation des rayons R_n , R_t et R_p (cas N_r6), cette approche permet de gagner de la fiabilité mais au dépens de l'efficacité. Les deux seules fonctions pour lesquelles cela s'avère en fait nécessaire, sont ES et SH, c'est à dire les deux dont l'optimum global est au fond d'un trou étroit. Pour les autres cela s'avère en fait non nécessaire. Il serait donc utile de pouvoir adapter le nombre de réduction des rayons automatiquement suivant la fonction, pour garantir un bon compromis entre l'efficacité et la fiabilité.

Nous avons également comparé nos résultats avec ceux obtenus par Chelouah et Siarry [2]. Leurs résultats sont donnés aux deux dernières lignes du tableaux 6.1. Nous avons choisi cette référence, car elle semble être très efficace pour un grand nombre de fonctions analytiques et que plusieurs de leurs modifications de l'algorithme TS de base ont été retenues dans notre implantation. Cependant, les chiffres présentés par les auteurs sont sujet à confusion. En effet, les notions d'itérations et d'évaluations de fonctions sont parfois mélangées. Ceci est très important, car à chaque itération, plusieurs voisins sont évalués. Le nombre d'évaluations de la fonction objectif est donc supérieur au nombre d'itérations. De

plus, les auteurs ont utilisé deux critères différents pour accepter/rejeter la convergence de l'optimisation dans la même publication. Ceci a conduit à des résultats différents pour certaines fonctions (voir très différents pour B2) sur le nombre d'évaluations sans changer le taux de réussite, ce qui n'est pas logique.

Malgré ces zones d'ombres, nous avons tenu à comparer nos résultats avec les leurs, car la méthode N_r1 est en fait très semblable à celle utilisée par Chelouah et Siarry dénommée '*CTSS_{multiple}*' dans [2]. Les résultats sont comparables en ce qui concerne la rapidité de convergence et la précision sauf pour la fonction SH, qui de part l'étroitesse des vallées des minima locaux a trouvé très utile l'autoadaptation des rayons.

Enfin, nous n'avons pas jugé nécessaire de pousser plus en profondeur la recherche sur un algorithme universel, étant donné que, le but est d'appliquer cette méthode au problème précis de la gestion du combustible dans les réacteurs de type CANDU à l'équilibre du rechargement. La fonction objectif correspondant à la gestion du combustible, à cause de la méthode de pénalisation externe, devrait ressembler à la fonction Rn, si le poids des contraintes est grand, ou GP avec un poids plus petit. En effet, comme nous l'avons vu à la figure 4.8, le domaine réalisable se limite à un bout de ligne pour deux variables. Dès que la configuration du réacteur s'éloigne de $k_{eff} = 1$ ou de $P_{C,max} \leq P_{lim}$, la nouvelle fonction objectif (qui comprend les contraintes) est alors essentiellement composée de la forme quadratique de l'écart des contraintes.

La méthode retenue pour l'optimisation de la gestion du combustible est celle qui utilise le simplex de Nelder. En effet, elle semble plus rapide tout en gardant une bonne efficacité. Par contre, nous avons remarqué que le nombre maximum de réduction des rayons R_n , R_t et R_p devrait être variable. Il doit en fait dépendre de la taille du domaine de recherche par rapport à la précision voulue et la taille des

Tableau 6.1 Résultats de la recherche tabou pour des fonctions analytiques.

méth #	ES % - N_{eval}	B2 % - N_{eval}	RC % - N_{eval}	GP % - N_{eval}
1	100-1245	100-1504	100-1280	100-1427
1.50	100-741	99-1020	100-781	100-926
2	100-1295	100-1295	100-1398	100-1290
2_n4	99-509	100-564	99-718	100-613
N_r1	91-188	97-222	100-333	99-261
N_r6	98-670	100-528	100-754	100-663
réf ^{α}	100-325	100-98	100-125	100-119
réf ^{β}	-	100-175	100-125	100-151

méth #	DJ % - N_{eval}	SH % - N_{eval}	Rn ^{π} , n=2 % - N_{eval}	Zn, n=2 % - N_{eval}
1	100-1404	100-1331	100-1950	100-1374
1.50	100-900	99-797	100-1259	100-857
2	100-1027	100-912	100-1834	100-1919
2_n4	100-644	81-364	100-1067	100-1074
N_r1	100-352	59-136	100-484	100-338
N_r6	100-1587	90-330	100-879	100-844
réf ^{α}	100-155	100-283	100-369	100-78
réf ^{β}	-	100-273	100-428	100-78

(^{π}) le critère de convergence pour la phase d'intensification finale est réduit à 0.00001 au lieu de 0.001 par défaut.

(^{α}) travaux de Chelouah [2, 59]: succès si l'équation (6.2) est valide

(^{β}) travaux de Chelouah [2]: succès si $|F - F_{ana}| < \epsilon_{rel}|F_{ana}| + \epsilon_{abs}$

vallées pour essayer de faire un nombre adéquat de calculs.

Tableau 6.2 Moyenne estimée et minimum global pour les fonctions analytiques.

	F_{ini}	F_{ana}
ES	-0.02	-1
B2	1.7	0
RC	55	0.397887
GP	55000	3
DJ	26	0
SH	2	-186.7309
Rn	140000	0
Zn	4000	0

CHAPITRE 7

APPLICATION DE LA MÉTHODE TABOU AU ACR-700

Dans ce chapitre, la méthode de recherche tabou (TS) a été utilisée pour optimiser la gestion du combustible du ACR-700. Avant de présenter les résultats obtenus, nous avons redéfini la fonction objectif ainsi que plusieurs paramètres de la méthode TS.

7.1 Choix des paramètres

La fonction objectif pour la méthode tabou f_{TS} est la même que celle de la méthode de pénalisation externe, c'est à dire la fonction de coût F_C plus l'écart des contraintes au carré fois un poids associé à chacune d'elles. En utilisant le formalisme de la gestion du combustible plutôt que celui de l'optimisation en général, la fonction objectif pour la méthode tabou f_{TS} , donnée par l'équation (4.21), peut alors s'écrire:

$$f_{TS} = F_C + \frac{w_k}{2}(k_{eff} - 1)^2 + \frac{w_q}{2} \sum_j (\max(0, q_j - q_{lim})^2) \quad (7.1)$$

7.1.1 Point initial

Le choix du point initial est fait au hasard. Il n'a idéalement pas d'influence sur les résultats obtenus. Il semble cependant évident que si deux points de départ distincts sont pris, et que la même séquence de nombres aléatoires est utilisée,

alors les résultats seront différents. Mais ceci revient à faire deux fausses suppositions. Premièrement, le nombre d'itération n'est pas forcément le même, les zones prometteuses ne sont pas obligatoirement détectées aux mêmes itérations. Le caractère aléatoire de la méthode ne garantit donc pas que la séquence de nombre est appliquée aux mêmes opérations. Le deuxième point est plus conceptuel. Prendre deux séquences de nombres identiques revient implicitement à annuler leur caractère aléatoire.

En pratique, le rayon initial du voisinage est élevé (20% de la variation totale permise pour chaque variable). Tout l'espace de recherche peut donc être parcourus en quelques itérations. De plus, le principe même de la méthode TS est de pouvoir sortir de minima locaux. Le caractère aléatoire de la méthode ne permet pas non plus de pouvoir reproduire exactement les mêmes résultats. Il ne devrait donc pas avoir de relation cause à effet entre le choix du point initial et les résultats comme pour les méthodes déterministes, tels que les méthodes de gradients.

7.1.2 Poids relatif des variables

Le poids relatif des variables n'est pas un paramètre pour la méthode TS. En effet, aucun gradient, ni aucune contrainte quadratique ne sont définis. Il n'y a aucun pas d'avance en tant que tel. Seul le rayon du voisinage *limite* l'écart entre deux points à des itérations successives. Ce rayon est défini comme une valeur relative de l'espace total de recherche pour chaque variable. Ces dernières peuvent donc varier indépendamment les unes des autres.

7.1.3 Compromis statistique

Dans le chapitre précédant, nous connaissions le minimum analytique des fonctions tests. Afin de tester les différents algorithmes TS envisagés, nous avons procédé à 100 optimisations (tirages) par fonction pour déterminer le taux de succès avec une bonne confiance. Le temps moyen pour ces 100 tirages est de l'ordre de 10 minutes. Pour la gestion du combustible, nous ne connaissons pas le minimum à l'avance. Il nous est donc impossible de savoir si le résultat trouvé correspond au minimum global. Un seul tirage n'est donc pas suffisant. Le temps de calcul pour une optimisation à 7 ou 8 variables de décision pour la gestion du combustible du ACR-700 est de l'ordre de 1 jour. Il est donc impensable d'envisager 100 tirages pour des fins de statistiques. Un compromis à 3 optimisations est utilisé. Cela peut paraître peu, mais si les résultats de tous les tirages sont proches, alors nous pouvons leur accorder une bonne confiance. En effet, même si nous pouvons parler de "statistique", c'est en fait un abus de langage. Les tirages ne sont pas indépendants et sont en fait biaisés vers l'optimum. La notion d'écart type ne s'applique pas, mais nous obtenons plutôt une tendance à avoir résultats similaires ou pas.

7.1.4 Poids des contraintes

Les poids des contraintes restent fixes tout au long de l'optimisation avec la méthode TS. Leur choix est donc très important, car la précision de l'optimum de la fonction coût et du respect des contraintes va dépendre de ce choix. En effet, deux cas extrêmes peuvent être envisagés:

- Le poids est trop petit, alors f_{TS} est égale à F_C , les contraintes n'ont que très peu d'influence à moins que l'écart avec leur limite soit très grand. Des

points en dehors du domaine réalisable seront alors retenus.

- Le poids est trop grand, alors F_C n'a que très peu d'influence sur f_{TS} à moins que toutes les contraintes soient respectées avec une très grande précision. Cela paraît idéal à première vue, cependant, l'aspect aléatoire de la méthode TS ne garantit pas que les voisins aux itérations suivantes respectent les contraintes avec autant de précision. En effet, étant donné qu'il y a une contrainte d'égalité ($k_{eff} = 1$), le domaine réalisable est en fait très difficile à atteindre. L'influence des contraintes sur f_{TS} est donc toujours très grande. L'optimisation de f_{TS} revient alors à trouver le point le plus proche du domaine réalisable possible. En fait dans ce cas, la fonction f_{TS} ressemble beaucoup à la fonction test Rn du chapitre précédent, c'est à dire une fonction à vallée étroite.

Un minimum de connaissances à propos des fonctionnelles est alors requis pour faire le bon choix des poids des contraintes. Premièrement, l'ordre de grandeur de F_C est de $5 \cdot 10^{-2}$, la précision recherchée sur le minimum est de 0.05% de F_C , c'est à dire $2.5 \cdot 10^{-5}$. Deuxièmement, nous devons décider de l'erreur acceptable pour chaque type de contraintes. Tout d'abord, pour k_{eff} , supposons une tolérance de $\delta_k = |\pm 0.05mk| = 5 \cdot 10^{-5}$. En prenant une valeur de 5000 pour le poids w_k de cette contrainte, la part maximum associée à $k_{eff} = 1$ dans f_{TS} est de $0.625 \cdot 10^{-5}$ pour un vecteur de décision qui respecte $|k_{eff} - 1| < \delta_k$. Ensuite, pour $P_{C,max}$, supposons une tolérance de $\delta_q = 5kw$. En prenant les données pour un ACR-700 appliquées dans l'équation 2.15, cela autorise à une erreur pour chaque canal sur la fonctionnelle q_j de 10^{-3} . En prenant un poids $w_q = 10$, la contribution maximum des contraintes $P_j \leq P_{lim}$ à f_{TS} est alors de $0.5 \cdot 10^{-5}$. Pour la réactivité du vide quand elle est retenue comme contrainte, la même tolérance et le même poids sont appliqués que pour k_{eff} .

Avec ces choix de poids des contraintes ($w_k = 5000$ et $w_q = 10$) et de tolérances, l'influence totale de toutes les contraintes dans f_{TS} est de l'ordre de grandeur de la précision de la fonction coût F_C , ce qui est notre but pour ne pas rencontrer un des deux cas extrêmes que nous venons de présenter.

7.2 Optimisation en burnup

La méthode TS est utilisée dans un premier temps pour optimiser la distribution de burnup du ACR-700. Nous avons repris le cas à 7 zones de burnup présenté à la figure 5.4. Afin de mieux comprendre l'influence des divers paramètres d'optimisation testés, reprenons rapidement la description de l'algorithme TS utilisé. À chaque itération, N_{ngb} voisins sont évalués. Si le meilleur d'entre eux est meilleur que le point d'avant, le processus recommence, à condition que le nombre total d'itérations soit plus petit que $N_{it,tot}$ et que le meilleur point ait été amélioré depuis moins de N_{itmax} itérations. Si le meilleur voisin est moins bon que le point de l'itération précédente, une zone prometteuse est détectée. Si aucune zone prometteuse n'est détectée pendant $N_{npd,max}$ itérations, alors la taille du voisinage est réduite. Quand $N_{red,max}$ réductions de la taille du voisinage ont été faites, l'algorithme s'arrête.

Chelouah et Siarry [2] préconisent des valeurs ajustables au nombre de variables n_{var} du problème d'optimisation. Leur formule générale dépendante du nombre de variables est donnée au tableau 7.1. Dans la dernière colonne, nous donnons les valeurs relatives à notre cas à 7 variables. Cependant nous avons ajusté les valeurs du tableau 7.1 à notre algorithme.

Pour être certain que l'algorithme a un nombre d'itérations suffisant pour adapter le rayon du voisinage à la forme de la fonction objectif, nous avons fixé la valeur de $N_{it,tot}$ à 1000 dans tous les calculs présentés par la suite. Ce paramètre n'est pas

Tableau 7.1 Valeur générale des paramètres de la méthode TS selon [2].

N_{ngb}	$\min\{2n_{var}, 10\}$	10
$N_{it,tot}$	$50n_{var}$	350
N_{itmax}	$5n_{var}$	35
$N_{npd,max}$	$2n_{var}$	14
$N_{red,max}$	$2n_{var}$	14

supprimé pour éviter des temps d'exécution extrêmement long au cas où les autres critères d'arrêt ne seraient pas rencontrés, ce qui correspondrait à une amélioration continue mais négligeable de la fonction objectif. Le nombre de voisins est le même (10). Le nombre de réduction $N_{red,max}$ a un sens différent dans l'algorithme de [2], et correspond en fait à un critère d'arrêt pour la phase d'intensification. Dans nos calculs, nous avons fixé $N_{red,max}$ à 4. Le rayon initial du voisinage est de 20% du domaine de recherche. Après 4 réductions, il est donc de $20/2^4 = 1.25\%$. Pour les variables de burnup qui sont comprises entre 10 et 40 GWj/t, cela correspond à une variation maximum de 375MWj/t soit pour le voisin le plus éloigné soit pour la taille du domaine de l'intensification. La précision de l'algorithme du simplexe de Nelder-Mead sur un domaine de cette taille est alors supposée suffisante.

Plusieurs valeurs ont été utilisées pour les deux paramètres restants. Premièrement, le nombre d'itérations sans amélioration du meilleur point N_{itmax} a été fixé à 50, c'est à dire une valeur 1.5 fois plus grande que celle suggérée par [2] pour essayer de forcer l'exploration d'une part, et d'avoir ainsi l'opportunité de trouver plus de zones prometteuses d'autre part. Deux valeurs du nombre d'itérations sans détection de zone prometteuse $N_{npd,max}$ ont été envisagées: 14 comme suggéré par [2] et 5 pour essayer de trouver plus rapidement la taille des zones de minima locaux s'il y a lieu. Les résultats sont présentés au tableau 7.2. Les colonnes B_i correspondent aux valeurs du burnup moyen de sortie GWj/t des zones 1 à 7. F_C donne la valeur de la fonction objectif coût, et ϕ correspond au nombre

d'évaluations de la fonction objectif requis (i.e. nombre de calculs de flux). Chaque ligne donne les résultats pour une optimisation différente. Ainsi, trois simulations ont été faites pour chaque choix de paramètres afin d'obtenir une certaine confiance statistique des résultats. Nous avons expliqué à la section précédente que la valeur de la fonction objectif est de 5.10^{-2} . Cependant, l'unité de cette fonctionnelle est arbitraire, et tout comme nous l'avons fait dans les chapitres précédents, nous avons mis F_C à l'échelle dans les tableaux.

Peu de différences sont observées dans les résultats du tableau 7.2 entre $N_{npd,max} = 5$ et $N_{npd,max} = 14$. Les valeurs de la fonction objectif et du nombre de calculs de flux sont similaires. Les distributions du burnup moyen de sortie sont semblables au centre du réacteur (B_1 à B_4), mais différent beaucoup en périphérie ce qui explique les petites variations de la fonction objectif. Cependant, le profil de la distribution de burnup est proche du profil idéal tel qu'obtenu à la section 5.3 et donné au tableau 7.3, i.e. le burnup est élevé au centre (zones B_1 à B_3) puis diminue en allant vers la périphérie (zones B_4 et B_5) avant de réaugmenter au bord complètement (zones B_6 et B_7). Des observations similaires pour $N_{npd,max} = 5$ et $N_{npd,max} = 14$ laissent penser qu'il n'est donc pas nécessaire d'attendre un grand nombre d'itérations sans détecter de zone prometteuse pour réduire la taille du voisinage. Cependant, une analyse plus détaillée des résultats a montré que l'optimisation s'est toujours arrêtée car le nombre maximum d'itérations sans amélioration du meilleur point N_{itmax} a été atteint, et non pas parce que le nombre de réductions de la taille du voisinage $N_{red,max}$ a été dépassé. La taille idéale du voisinage n'a donc pas forcément été atteinte.

Nous avons alors repris les calculs en augmentant N_{itmax} à 200, pour $N_{npd,max} = 14$ et $N_{npd,max} = 5$. Les résultats sont présentés au tableau 7.4 avec le même formalisme qu'au tableau 7.2. Dans ce cas, le nombre N_{itmax} n'a pas été atteint. Comme nous pouvons nous y attendre, le nombre de calculs de flux est plus grand lorsque

Tableau 7.2 Optimisation en burnup avec $N_{itmax} = 50$

$N_{npd,max} = 5$								
B_1	B_2	B_3	B_4	B_5	B_6	B_7	F_C	ϕ
26.88	26.97	27.76	24.41	15.93	17.61	18.31	4.4793	4449
26.98	27.12	27.57	24.16	14.43	19.85	26.95	4.5140	4684
26.96	27.23	27.33	23.96	14.91	21.25	15.85	4.4964	4954
$N_{npd,max} = 14$								
B_1	B_2	B_3	B_4	B_5	B_6	B_7	F_C	ϕ
27.43	26.99	27.41	24.18	17.76	14.66	14.69	4.5151	4837
27.44	27.03	27.40	24.16	16.77	15.75	19.93	4.4870	4677
27.31	27.17	27.28	24.01	15.46	18.58	21.99	4.4759	4693

Tableau 7.3 Distribution de burnup obtenue avec la méthode MM pour le cas à 7 zones à partir de la distribution “flat”.

B_1	B_2	B_3	B_4	B_5	B_6	B_7	F_C
27.32	27.16	27.24	24.02	15.77	18.29	18.98	4.4753

$N_{npd,max} = 14$ que pour $N_{npd,max} = 5$, presque le double. La distribution de burnup moyen de sortie obtenue reste encore une fois relativement constante au centre (zones B_1 à B_4) d’une optimisation à l’autre (pour des paramètres identiques), et de plus, elle varie moins en périphérie (sauf pour la très petite zone B_7). Le profil de la distribution de burnup est encore très proche du profil idéal obtenu à la section 5.3. La fonction objectif est également légèrement meilleure en moyenne sur les trois optimisations lorsque N_{itmax} n’est pas atteint.

L’amélioration de la fonction coût F_C n’est cependant que marginale si nous prenons en compte le nombre d’itérations supplémentaires nécessaires (environ 4000) pour $N_{npd,max} = 14$. Ceci laisse donc supposer que la convergence finale est difficile malgré les efforts faits pour avoir un algorithme de recherche tabou pour lequel le rayon du voisinage s’adapte. Afin d’avoir une idée plus précise de la proportion de l’effort

Tableau 7.4 Optimisation en burnup avec $N_{itmax} = 200$

$N_{npd,max} = 5$								
B_1	B_2	B_3	B_4	B_5	B_6	B_7	F_C	ϕ
27.02	27.14	27.44	24.05	15.44	19.59	17.78	4.4723	3935
26.69	26.95	28.02	24.36	15.31	19.45	16.36	4.4907	4915
27.23	27.18	27.30	24.00	15.25	19.18	21.65	4.4781	6058
$N_{npd,max} = 14$								
B_1	B_2	B_3	B_4	B_5	B_6	B_7	F_C	ϕ
27.40	27.13	27.31	24.07	15.43	17.71	25.94	4.4914	8460
27.17	27.14	27.35	24.04	15.92	18.59	16.86	4.4687	7855
27.52	27.10	27.28	24.14	16.31	16.09	23.19	4.4908	10165

de calcul passé sur la convergence finale par rapport à la recherche de la zone du minimum global, nous avons tracé à la figure 7.1 la valeur de la fonction objectif au cours des itérations pour le deuxième cas du tableau 7.4 avec $N_{npd,max} = 14$. La valeur courante et la meilleure valeur correspondent à la ligne continue et à la ligne en tirets respectivement. L'axe des ordonnées est séparé en deux parties. Les lignes verticales en pointillés indique les itérations pour lesquelles une zone prometteuse a été trouvé. Enfin, les résultats donnent la valeur de la fonction objectif pour la méthode TS, i.e. f_{TS} , et ils comprennent donc l'écart des contraintes en plus de la fonction coût F_C . Nous pouvons voir que la convergence vers le minimum de la fonction f_{TS} est très rapide. Étant donné le poids important des contraintes sur f_{TS} , celles-ci sont donc satisfaites très vite. D'un point de vue physique des réacteurs cela veut dire qu'un point réalisable est rapidement atteint.

L'effort de calcul (donné par ϕ) est comparable à celui obtenu par Chelouah [59] sur des fonctions analytiques. En effet, pour des fonctions à 5 et 10 variables, il a obtenu en terme de nombre d'évaluations de la fonction objectif: 2150 pour R_5 , 2250 pour Z_5 , 15700 pour R_{10} et 4600 et Z_{10} (voir annexe V pour les définitions des fonctions). Le temps de calcul est par contre très grand (de l'ordre d'un jour).

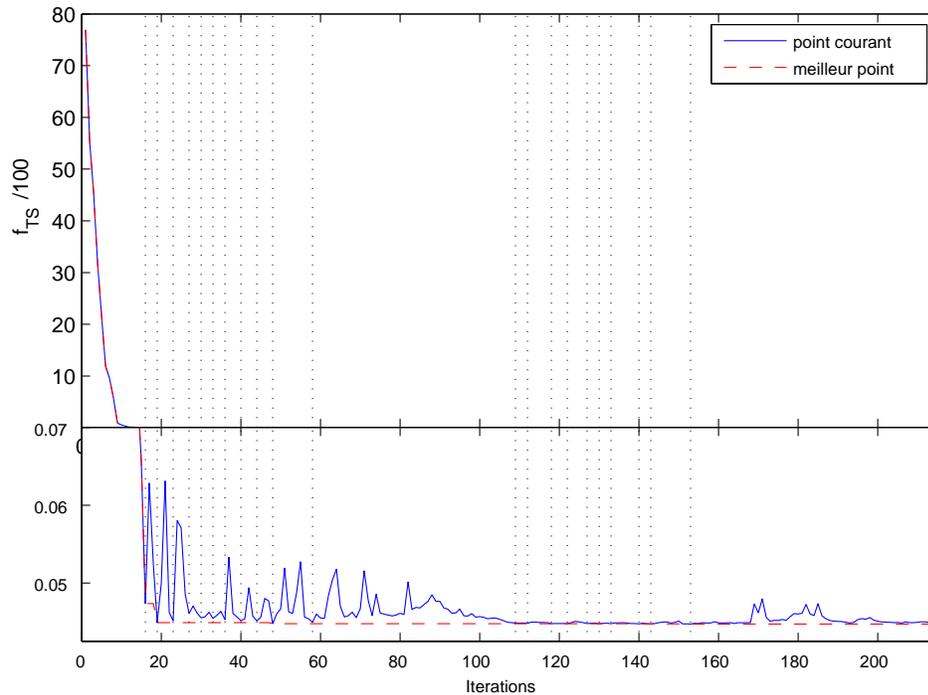


Figure 7.1 Fonction objectif f_{TS} selon les itérations

Il est donc impensable de faire 100 calculs d'optimisations à des fins statistiques comme nous l'avons précisé au début de ce chapitre.

Les résultats obtenus avec la méthode TS semblent donc confirmer deux résultats obtenus avec les méthodes de gradients. Premièrement, la distribution de burnup moyen de sortie ne présente pas de minima local pour le cas à 7 zones de burnup. En effet, les distributions de burnups sont, à toute fin pratique, identiques. Deuxièmement, il est déconseillé d'avoir des zones très petites par rapport aux autres. En effet, elles ont alors peu d'influence sur la fonction objectif, et leur burnup est mal optimisé.

7.3 Optimisation en burnup et en enrichissement

Lorsque l'enrichissement du combustible neuf est également optimisé, une variable d'un ordre de grandeur complètement différent est ajoutée. Cette nouvelle variable a également une influence sur la fonction objectif f_{TS} plus générale et donc à priori plus grande. Nous avons repris les calculs d'optimisation à 7 zones de burnup et 1 zone d'enrichissement. Avec la méthode TS, aucun facteur de mise à l'échelle (poids w_i associé aux variables) n'est utilisé.

Nous avons repris les calculs d'optimisation avec $N_{npd,max} = 2n_{var}$, c'est à dire $N_{npd,max} = 16$ dans ce cas. Les deux valeurs de N_{itmax} , 50 et 200, ont été testées. Les résultats sont présentés au tableau 7.5 avec le même formalisme qu'aux tableau 7.2 plus la colonne ε_f qui donne la valeur optimale de l'enrichissement. Les résultats avec $N_{itmax} = 50$ ont convergé car N_{itmax} a été atteint. Avec presque deux fois plus de calculs de flux, les résultats pour $N_{itmax} = 200$ ne donne pas une valeur de la fonction objectif nettement améliorée en moyenne par rapport à $N_{itmax} = 50$. De plus, la distribution de burnup est loin d'être constante. Ceci s'explique par le fait qu'elle est fortement reliée à la valeur de l'enrichissement qui n'est ni optimale ni constante.

Il est cependant intéressant de noter qu'un point à l'intérieur ou très proche du domaine réalisable a été atteint. De plus, la méthode TS semble "rapide" pour trouver un tel point, mais très lente pour la convergence finale vers les distributions de burnup et de l'enrichissement optimaux, malgré la phase d'intensification proposée par [2] à l'algorithme de base. Avec l'enrichissement comme nouvelle variable de décision, l'optimisation de la fonction f_{TS} rencontre le même problème que la fonction R_2 testée au chapitre précédent. La valeur de la fonction objectif est très proche de sa valeur optimale mais pas les variables de décision. Ceci est dû à la présence d'une "vallée" très étroite, profonde et longue. En effet, la valeur

Tableau 7.5 Optimisation en burnup et en enrichissement avec $N_{npd,max} = 16$

$N_{itmax} = 50$									
B_1	B_2	B_3	B_4	B_5	B_6	B_7	ε_f	F_C	ϕ
32.44	32.25	31.53	28.52	21.18	24.94	21.29	2.902	4.4114	6590
32.63	31.99	31.09	28.29	22.02	22.46	18.08	2.885	4.4386	6274
28.61	27.71	28.04	25.39	18.80	13.71	33.84	2.642	4.7248	5585
$N_{itmax} = 200$									
B_1	B_2	B_3	B_4	B_5	B_6	B_7	ε_f	F_C	ϕ
27.55	27.57	27.84	24.79	16.70	20.26	22.44	2.623	4.6092	7271
33.39	33.85	34.60	31.15	22.69	28.49	20.80	3.040	4.3476	9851
27.85	27.46	28.28	25.41	18.75	14.81	24.28	2.632	4.6722	10555

de la fonction f_{TS} peut être très élevée (de l'ordre de 10^4) quand un point est non réalisable. Comme le montre les résultats au tableau 7.5 et à la section 5.13, l'enrichissement et donc indirectement la distribution de burnup peuvent changer beaucoup sans que la fonction f_{TS} ne change beaucoup (valeur entre 4 et 5) par rapport à la valeur obtenu pour des points non-réalisables. La variation de la fonction coût F_C est donc loin d'être négligeable en soit, mais elle le devient lorsqu'elle est insérée dans la fonction f_{TS} . Le problème principal de l'optimisation simultanée des deux types de variables est que lorsqu'un nouvel enrichissement légèrement différent est pris, la nouvelle distribution de burnup a peu de chance de correspondre à un point réalisable, même si le point de l'itération précédente respectait les contraintes. D'un point de vue imagée, nous pouvons dire que le mouvement aléatoire d'une itération à l'autre rend très difficile, en fait très improbable, de se promener le long du fond de la vallée.

CHAPITRE 8

CONCLUSIONS ET RECOMMANDATIONS

La gestion du combustible est une des étapes importantes du design d'un réacteur nucléaire. En effet, elle permet de déterminer le mode opératoire de rechargement et l'efficacité d'un modèle de réacteur. En fonction des objectifs suivis, son optimisation permet par exemple d'améliorer la sûreté du design ou de minimiser les coûts de la production d'énergie. Le but de notre étude était double. Nous avons eu comme première tâche de développer dans le code DONJON des outils permettant d'optimiser la gestion du combustible à l'équilibre du rechargement pour différents types de réacteurs CANDU. Deuxièmement, nous avons appliqué ces outils pour les réacteurs CANDU-6 et ACR-700. Un accent particulier a été mis sur les outils développés pour le réacteur ACR-700 refroidi à l'eau légère.

8.1 Globalité de l'implantation

Afin de compléter le code de diffusion multi-usage DONJON, des capacités d'optimisation sont maintenant disponibles. Même si l'aspect le plus général possible a été suivi pour l'implantation de ces nouvelles fonctionnalités dans le cadre de cette recherche, seules les fonctions à variables continues sont considérées. Plusieurs points importants sont à retenir dans l'intégration des capacités d'optimisation dans DONJON:

- le code de diffusion DONJON a des usages différents. Il a donc été développé de manière très modulaire (langage CLE2000 pour les fichiers d'entrée). La même approche a été retenue pour les modules d'optimisation. Ceci

représente le premier avantage par rapport au code OPTEx qui ne permet pas de séparer les différentes étapes de la méthode de gradients.

- la définition des fonctionnelles se fait dans les fichiers d'entrées et non dans le code Fortran. Cela laisse une très grande liberté quant à la définition du problème d'optimisation. Pour faciliter l'utilisation du code, les fonctions les plus utilisées ont été prédéfinies: coût en combustible, k_{eff} , puissance canal, réactivité du vide, ainsi que leurs dérivées partielles.
- l'aspect modulaire permet de faire des algorithmes hybrides au niveau de CLE2000, et donc sans recompiler. La contrepartie potentielle de cette liberté d'action est d'avoir des fichiers d'entrée légèrement compliqués.
- l'utilisation de tables (L_TABLE) à deux paramètres est maintenant possible sans aucune analyse du comportement des propriétés. Ces tables sont directement obtenues à partir de plusieurs fichiers de compositions dépendant uniquement du burnup. Avant même de considérer une optimisation à deux types de variables, cela permet surtout de pouvoir calculer rapidement le flux neutronique en interpolant les propriétés en fonction du burnup et d'un autre paramètre.

8.2 Généralité des algorithmes développés

L'optimisation de la gestion du combustible a été envisagée pour deux types différents de réacteurs CANDU: refroidis à l'eau lourde (CANDU-6) et à l'eau légère (ACR-700). Les objectifs et contraintes n'étaient pas forcément tous les mêmes. Ainsi, différents problèmes se sont présentés, plus particulièrement pour le ACR-700. L'approche modulaire utilisée pour l'implantation des outils d'optimisation dans DONJON, nous a permis de développer plusieurs méthodes pour résoudre ces

différents problèmes de gestion du combustible. Les techniques développées sont rappelées par les points suivants:

- la première famille de méthodes d'optimisation est basée sur l'utilisation des gradients. Cette approche avait déjà été retenue dans le code OPTEX. La méthode de gradients utilisée dans ce code a été implantée dans DONJON: programmation quasi-linéaire (PQL). Ses composantes ont servi de base aux autres méthodes de ce type.
- la méthode de base PQL est limitée par le choix du point initial. Plusieurs méthodes alternatives de gradients ont été envisagées pour l'optimisation de la gestion du combustible. Elles ont pour principal avantage par rapport à la méthode PQL de pouvoir utiliser n'importe quel point du domaine de recherche comme estimé initial.
- Les nouvelles méthodes de gradients sont très utiles pour trouver un point réalisable. Les diverses alternatives proposées ont chacune leur avantages et leur inconvénients. Lorsqu'employées à bon escient, la méthode multi-étape (MS) et la méthode mixte (MM) s'avèrent en général très efficaces pour trouver un point réalisable et ensuite converger vers le minimum de la fonction de coût en combustible. Ces deux approches ne sont sensibles au point de départ que pour des cas relativement très détaillés de distribution de zones de burnup.
- une deuxième famille de méthode d'optimisation a été envisagée: les méthodes métaheuristiques. Ces approches ont pour caractéristique principale de pouvoir sortir d'un minimum local. Une d'entre elles a été implantée: la recherche tabou. L'application de ce type de méthode à la gestion du combustible est très récente. Elle a été faite principalement pour les réacteurs du type PWR et BWR, c'est-à-dire pour des variables combinatoires. De plus, l'application

de la méthode de recherche tabou aux variables continues en général est une des premières.

8.3 Application aux réacteurs CANDU

Les capacités d'optimisation dans le code DONJON sont entièrement nouvelles. Leur implantation a donc nécessité plusieurs tests afin de vérifier leur exactitude. Le ACR-700 est un nouveau type de réacteur, très peu d'expérience d'ingénierie, et en particulier sur la gestion du combustible, est donc disponible pour pouvoir apprécier les résultats. Nous avons donc choisi le réacteur CANDU-6 pour effectuer la plupart des tests requis pour la validation de cette nouvelle facette du code. De plus pour ce réacteur, la gestion du combustible peut être approchée de manière très simple (2 variables de décision) tout en restant réaliste.

Les méthodes de gradients représentent une partie importante des approches envisagées pour l'optimisation de la gestion du combustible. La précision du calcul des gradients est donc primordiale. Les résultats obtenus démontrent l'exactitude des gradients calculés pour toutes les fonctionnelles envisagées quelque soit leur ordre de grandeur (faible ou forte variation des fonctionnelles). Une condition sur la méthode de calcul est cependant requise. La valeur analytique de la dérivée des propriétés doit être utilisée, l'approximation numérique consistant à prendre la pente entre la valeur normale et une valeur perturbée est trop imprécise en général.

La méthode des gradients de base que nous avons implanté (PQL) n'est pas limitée par la complexité du problème en terme de nombre de variables de décision. En effet, des cas avec 190 zones de burnup pour le CANDU-6 et 150 zones pour le ACR-700 ont été optimisés avec succès.

Plusieurs constatations générales se dégagent des résultats obtenus pour le CANDU-6

et le ACR-700:

- le fait d'augmenter le nombre de zones de burnup permet d'obtenir une distribution de la puissance canal plus monotone et plate.
- le burnup tend à augmenter en extrême périphérie du cœur avec la discrétisation des zones de burnup. Cela permet en effet de minimiser les fuites en dehors du réacteur, et ainsi de diminuer le coût en combustible.
- traditionnellement, le choix de la répartition des zones de burnup correspond à une distribution radiale. Ce choix n'est cependant pas toujours optimal. En effet, tenir compte des asymétries du cœur dans le choix de la répartition des zones de burnup peut permettre de réduire le coût en combustible même avec un petit nombre de zones (2x2 zones séparation haut/bas).
- la définition de zones de burnup introduit un biais mathématique dans la fonction de coût en combustible, surtout si ces zones ont des nombres de canaux très différents. Les zones larges sont plus optimisées au dépens des plus petites zones.

Les résultats avec la méthode tabou pour le ACR-700 confirment ceux obtenus avec les méthodes alternatives de gradients. La méthode est cependant trop lente pour être utilisée comme méthode principale d'optimisation de la gestion du combustible. De plus, les résultats obtenus restent imprécis pour les zones de burnup beaucoup plus petites que les autres.

Nous avons également optimisé l'enrichissement avec succès. Pour le problème d'optimisation de la gestion du combustible tel que nous l'avons défini pour le ACR-700, les résultats montrent qu'un enrichissement élevé a tendance à être favorisé. Nous devons cependant souligner que la concentration en dysprosium était fixée par rapport à l'enrichissement. et que la limite supérieure du burnup était haute.

8.4 Limitations du problème original

Initialement, un des buts principal de cette recherche était d'optimiser la gestion du combustible des réacteurs CANDU refroidis à l'eau légère, en particulier le ACR-700. Ce but a été rempli, mais avec certainement moins de profondeur que prévu initialement. Il nous est en effet impossible de dire quel serait le meilleur design pour ce réacteur. Cette limitation n'est cependant pas due à un manque d'effort de notre part, mais à une évolution des buts de notre projet. L'emphase s'est vue déplacée de l'application pour le ACR-700 au développement d'outils d'optimisation dans DONJON. Nous avons plus utilisé la gestion du ACR-700 afin d'acquérir de l'expérience avec les nouvelles méthodes d'optimisation développées que comme une fin en soi. Cette expérience pourra ainsi être réappliquée pour le design d'autres modèles de réacteur CANDU. L'ensemble des méthodes d'optimisation, ainsi que les différentes options et modules connexes peut être utilisé comme une boîte à outils applicable à différents problèmes d'optimisation, et pas seulement à la minimisation du coût en combustible.

La raison principale de ce changement d'emphase découle surtout des nombreux changements qui ont été apportés par le concepteur AECL au modèle de référence du ACR-700 au cours de cinq dernières années [1, 32, 34, 35, 60]. Initialement, le cœur comportait 256 canaux, puis 284 et enfin 300. Le nombre de canaux n'est cependant qu'un point secondaire, le changement qui a eu le plus d'impact sur la gestion du combustible, est celui de la composition du combustible. En effet, dans le temps de notre recherche, le combustible a évolué d'une grappe type CANFLEX légèrement enrichie (1.65%) sans poisons [60], à une grappe type CANFLEX un peu plus enrichie (2%) avec peu de poisons (4% Dy dans le crayon central) [35], puis enfin à une nouvelle grappe avec plus d'enrichissement (2.5%) et plus de poisons (15% Dy dans le crayon central). Cela a eu beaucoup d'influence sur la réactivité

du vide qui est passée d'environ -9mk, à -5mk puis -28mk respectivement. Cette contrainte qui était initialement très importante, s'est retrouvée caduque. Elle n'en demeure pas moins potentiellement très influente sur la gestion du combustible pour des choix différents de la composition du combustible ou pour un autre design de réacteur. L'accent a donc plus été mis sur la méthodologie et le développement d'outils pour l'optimisation de la gestion du combustible, que sur son application particulière au ACR-700.

8.5 Recommandations

La principale limitation des outils d'optimisation n'est pas en fait reliée aux méthodes d'optimisation implantées en tant que telles, mais elle provient de l'interpolation des propriétés selon une variable au maximum en plus du burnup. Pour une meilleure utilisation du combustible, il aurait fallu permettre à la concentration de poisons de varier indépendamment, comme variable de décision. En réduisant la concentration de poison, cela aurait permis d'augmenter la réactivité du vide jusqu'à sa limite imposée (la contrainte devenant active) et de réduire les coûts en combustible. Minimale, trois variables de décision seraient donc nécessaires pour le ACR-700, soit le burnup, l'enrichissement et la concentration en poison. Si deux différents poisons sont utilisés alors le nombre de type de variables serait de quatre. Une méthode plus générale que la méthode de Ceschino est peut être requise étant donné sa complexité à deux variables seulement (voir annexe III). Nous n'avons volontairement pas retenu la variation des incréments de sections efficaces avec les variables de décision. Il serait intéressant de vérifier la précision de cette approximation sur un modèle de réacteur dont la réactivité du vide est moins négative. Nous conseillerons également que si une interpolation à plusieurs dimensions (> 2) est faite pour les propriétés du combustible, les incréments de

sections efficaces puissent également être interpolés en fonction des variables de décision.

Les optimisations avec le burnup et l'enrichissement comme variables de décision sont en général beaucoup plus longues que pour les optimisations en burnup seulement (à nombres de variables constants). De plus, le facteur de mise à l'échelle de la variable d'enrichissement est difficile à choisir. Une solution potentielle à ce problème serait d'avoir un facteur déterminé automatiquement à partir des gradients initiaux de la fonction objectif ou de k_{eff} , plutôt que fourni par l'utilisateur comme c'est le cas actuellement et dans le code OPTTEX. Si cette modification n'est néanmoins pas suffisante, il est peut-être même nécessaire d'avoir un facteur d'échelle qui soit auto-ajustable à chaque itération pour la variable d'enrichissement. Dans le cas où cela ne fonctionnerait pas, deux solutions sont encore possibles: une étude paramétrique classique, ou s'il y a plusieurs zones d'enrichissement, deux problèmes d'optimisation imbriqués: un "externe" pour l'enrichissement et un autre "interne" pour le burnup pour chaque valeur d'enrichissement.

Dans notre étude, nous avons contourné le problème du CPPF en le prenant égal à 1 et en utilisant des contraintes restrictives sur la puissance canal. Avec les problèmes très détaillés que nous avons envisagés (190 zones pour le CANDU-6 et 150 pour le ACR-700), nous avons obtenu des puissances canal très constantes au centre par rapport à ce qui est généralement obtenu avec 2 ou 3 zones de combustion dans un CANDU-6. Le CPPF correspondant peut alors être tout-à-fait différent. Il serait intéressant d'en vérifier sa valeur. De plus, à priori le CPPF augmente beaucoup avec l'enrichissement. En effet, plus ce dernier est grand, plus l'incrément de réactivité et donc la perturbation associée au rechargement sont grands. Ainsi, en pratique, la prise en compte de la variation du CPPF avec l'enrichissement et le burnup aurait eu pour effet de contraindre les variables de décision et de limiter l'enrichissement. Ainsi, pour obtenir à l'avenir des résultats d'optimisation

plus réalistes que ceux que nous avons obtenus, une évaluation du CPPF par des calculs de diffusion additionnels ou une approximation avec une corrélation seront nécessaires. Cela ne posera pas de problème au niveau de l'optimisation, car le CPPF a déjà été inclu dans la programmation des contraintes de puissance.

La méthode de recherche tabou est lente pour la convergence finale, mais rapide pour trouver un point réalisable. Elle serait donc peut-être plus indiquée pour une première partie d'un algorithme hybride, basé sur l'idée de la méthode mixte en remplaçant le lagrangien augmenté par la méthode de recherche tabou.

La méthode de recherche tabou est facilement parallélisable (l'évaluation des voisins ou du polytope pour le simplexe de Nelder-Mead). L'approche modulaire suivie pour son implantation permettrait en fait de paralléliser directement dans les fichiers d'entrées. Cela a en fait déjà été fait pour un couplage transport / diffusion [61]. Dans ce travail, le module `ADDOBJ` a permis d'ajouter (i.e. de regrouper), pour le calcul de diffusion (`DONJON`), des objets `L_MACRO` qui contenaient les propriétés de différentes parties du réacteur calculées en transport (`DRAGON`). Dans le cadre de la méthode de recherche tabou, la seule modification à apporter au code Fortran est de pouvoir regrouper les résultats de différentes évaluations sur un processeur. Cela semble relativement facile et tout indiqué pour le même module `ADDOBJ`.

La plupart de ces modifications pourrait être apportées rapidement pour améliorer le code. Elles permettraient également d'élargir encore plus les problèmes d'optimisation envisageables dans le cadre de la gestion du combustible des réacteurs de type CANDU.

8.6 Travaux futurs

Avec les outils déjà développés et validés dans cette recherche, nous pouvons proposer d'ores et déjà des études de gestion du combustible d'envergure. Plusieurs aspects peuvent être pris en compte pour améliorer l'optimisation du ACR-700 ou encore être utilisés pour le ACR-1000 actuellement développé par AECL [62].

Dans nos travaux, nous n'avons pas limité la valeur que pouvait prendre le burnup de sortie, si ce n'est par la limite à laquelle nous avons fait évoluer la grappe de combustible en transport. Des burnups proches de $40GWj/t$ ont été atteints pour des hautes valeurs d'enrichissement initial. Cependant, ces valeurs sont impensables dans un réacteur CANDU à l'heure actuelle, car aucune analyse de tenue de ce type de combustible à de si hauts burnups n'a été faite. Il est donc probable que le burnup moyen de sortie du cœur soit plafonné ou fixé à une valeur inférieure.

Un nouveau problème d'optimisation serait de minimiser l'enrichissement initial du combustible moyen du cœur pour atteindre le burnup prescrit. Ceci est un exemple d'optimisation de la gestion du combustible qui pourrait être le sujet de travaux futurs. Avec une interpolation des propriétés à 3 variables ou plus, la distribution des poisons pourrait être également optimisée avec l'enrichissement pour ce nouveau problème de gestion du combustible. La concentration en poisons serait alors à priori minimisée indirectement pour augmenter l'efficacité du combustible tout en respectant la contrainte sur la réactivité du vide.

L'étude avec la méthode tabou est relativement succincte. Il serait notamment intéressant d'essayer d'autres partages des zones de burnup. Une version hybride tabou/gradient permettrait de détecter des minima locaux avant de finaliser l'optimisation.

Dans nos calculs, nous n'avons utilisé qu'une seule zone d'enrichissement. Sans pour autant aller jusqu'à un enrichissement par canal, l'utilisation de quelques zones d'enrichissement permettrait peut-être de réduire la fonction de coût.

À plus long terme, deux aspects de la gestion du combustible pourrait être optimisés. Premièrement, la gestion du combustible à l'équilibre du rechargement comporte deux volets. Le premier, l'optimisation du patron de référence, a été traité dans notre étude. Un deuxième volet est aussi très important: la séquence des canaux à recharger à chaque jour. Les méthodes de gradients ne se prêtent malheureusement pas à l'optimisation de cette partie de la gestion du combustible, qui a pour variables de décision des valeurs combinatoires. Les méthodes méta-heuristiques telles qu'utilisées dans les PWR et BWR sont plus indiquées. Les développements réalisés par notre étude ne s'appliquent donc pas tous dans ce cas. Cependant, plusieurs parties "utilitaires" de la méthode de recherche tabou pourraient être réutilisées pour de l'optimisation de variables combinatoires ou discontinues. Nous devons néanmoins souligner que l'optimisation de la séquence de rechargement des canaux sur une longue période perd un peu de son sens. En effet, plusieurs paramètres d'ingénierie (ex: maintenance d'équipements) ne peuvent être pris en compte.

Toujours à plus long terme, en supposant qu'un module pour le choix automatique des sites de rechargements soit disponible, les méthodes d'optimisation que nous avons développées pourraient être utilisées pour l'optimisation de la charge initiale. En effet, dans ce cas, les variables de décision sont continues (zones d'enrichissement ou d'appauvrissement). La fonction objectif serait de minimiser la charge initiale par exemple. La séquence de rechargement calculée automatiquement pour la période de transition peut ne pas être celle qui sera exactement suivie en pratique, elle a pour but principal de permettre l'évaluation de la fonction objectif liée à la distribution initiale de l'enrichissement du cœur neuf.

RÉFÉRENCES

- [1] M. Boubcher and R. Chambon, “Sensitivity of acr-700 core calculations to the number of energy group,” Tech. Rep. 108- 03140- 401-002, Atomic Energy Canadian limited (AECL), Toronto, Canada, 2002.
- [2] R. Chelouah and P. Siarry, “A hybrid method combining continuous tabu search and nelder-mead simplex algorithms for the global optimization of multimima functions,” *European Journal of Operational Research*, vol. 161, pp. 636–654, 2005.
- [3] A. F. Henry, *Nuclear-Reactor Analysis*. The MIT Press, 1975.
- [4] D. Rozon, *Introduction à la Cinétique des Réacteurs Nucléaires*. Presses internationales Polytechnique, 1992.
- [5] B. Rouben, *CANDU Lectures - Reactor Core and Fuel Management*. Toronto: Atomic Energy of Canada limited (AECL), 1984. AECL-8333(F).
- [6] D. Rozon and M. Beaudet, “Canada deuterium uranium reactor design optimization using three-dimensional generalized perturbation theory,” *Nuclear Science and Engineering*, vol. 111, pp. 1–20, 1992.
- [7] E. Varin, A. Hébert, R. Roy, and J. Koclas, “A user’s guide for donjon,” Tech. Rep. IGE-208, École Polytechnique de Montréal, Institut de Génie Nucléaire, 2003.
- [8] V. G. Toshinsky, H. Sekimoto, and G. I. Toshinsky, “Multiobjective fuel management optimization for self-fuel-providing lmfbr using genetic algorithms,” *Annals of Nuclear Energy*, vol. 26, pp. 738–802, 1999.

- [9] J. Tajmouati, *Optimisation de la gestion du combustible enrichi d'un réacteur CANDU avec prise en compte des paramètres locaux*. Thèse de Doctorat, École Polytechnique de Montréal, Montréal, 1993.
- [10] B. Arsenault, "General introduction to methods of physics analysis and physics computer codes." <http://canteach.candu.org/library/20054302.pdf>, 2000.
- [11] C. J. Jeong, "An on-line fuel management method using in-core detector readings for candu reactors," *Annals of Nuclear Energy*, vol. 29, pp. 1827–1836, 2002.
- [12] D. Rozon and M. Beaudet, "Application de la théorie des perturbations à l'étude des réacteurs candu," tech. rep., E.D.F., 1990. Bulletin de la Direction des Études et Recherches, Série A.
- [13] M. Melice, "Pressurized water reactor optimal core management and reactivity profiles," *Nuclear Science and Engineering*, vol. 37, pp. 451–477, 1969.
- [14] D. Rozon, A. Hébert, and D. McNabb, "The application of generalized perturbation theory and mathematical programming to equilibrium refueling studies of a CANDU reactor," *Nuclear Science and Engineering*, vol. 78, pp. 211–226., 1981.
- [15] D. Rozon, *Optimization of in-Core Fuel Management in CANDU Nuclear Reactors*. Thèse de Doctorat, McMaster University, Canada, 1985.
- [16] M. Beaudet, *Optimisation des barres de compensation dans le réacteur de Gentilly-2*. Mémoire de maîtrise, École Polytechnique de Montréal, Montréal, 1987.

- [17] M. Beaudet, *Application de la programmation non-linéaire aux calculs de design et de gestion du combustible d'un réacteur CANDU*. Thèse de Doctorat, École Polytechnique de Montréal, Montréal, 1991.
- [18] D. Rozon, "Gestion du combustible," 2000. Notes de cours chap. 4.5.
- [19] D. Rozon and W. Shen, "A parametric study of the dupic fuel cycle to reflect pwr fuel management strategy," *Nuclear Science and Engineering*, vol. 138, pp. 1–25, 2001.
- [20] W. Stacey, *Variational Methods in Nuclear Reactor Physics*. New York: Academic Press, 1974. 181p.
- [21] W. Stacey, "Variational estimates of reactivity worths and reaction rate ratios in critical nuclear reactors," *Nuclear Science and Engineering*, vol. 48, pp. 444–458, 1972.
- [22] W. Stacey, "Variational estimates and generalized perturbation theory for the ratios of linear and bilinear functionals," *Nuclear Transaction*, vol. 13, no. 8, pp. 1119–1125, 1972.
- [23] C. E. Lemke, "Bimatrix equilibrium points and mathematical programming," *Management Science*, vol. 11, pp. 681–689, 1965.
- [24] J. A. Ferland, "A linear programming problem with an additional quadratic constraint solved by parametric linear complementarity," Tech. Rep. Rapport 497, Université de Montréal Département d'Informatique et de Recherche Opérationnelle, 1984.
- [25] S. B. Alaoui, *Étude du Rechargement Optimal d'un Reacteur CANDU: Développement du Code OPTEX-2D*. Thèse de Doctorat, École Polytechnique de Montréal, Montréal, 1985.

- [26] R. Roy, “The cle-2000 tool-box,” Tech. Rep. IGE-163, École Polytechnique de Montréal, Institut de Génie Nucléaire, 1999.
- [27] F. Ceschino and J. Kuntzmann, *Problèmes Différentiels de Conditions Initiales Méthodes Numériques*. Dunod, 1963.
- [28] A. I. Wight and P. Girouard, “Optimum burnup distribution in a continuously fueled reactor,” *Nuclear Science and Engineering*, vol. 68, pp. 61–72, 1978.
- [29] A. Faisant, *TP et TD de Topologie générale*. Paris: Hermann, 1973.
- [30] J. Gauvin, *Leçons de programmation mathématique*. Montréal, Canada: Édition de l’École Polytechnique, 1995. 140p.
- [31] F. Delbos and J. C. Gilbert, “Global linear convergence of an augmented lagrangian algorithm for convex quadratic optimization problems,” *Journal of Convex Analysis*, vol. 12, no. 1, pp. 45–69, 2005. http://www.optimization-online.org/DB_HTML/2003/12/790.html.
- [32] R. Chambon, E. Varin, and D. Rozon, “Candu fuel management optimization using generalized perturbation theory in the donjon diffusion code,” in *Canadian Nuclear Society Twenty-Sixth Annual Conference*, vol. on CD, 2005.
- [33] R. Chambon, E. Varin, and D. Rozon, “Optimization of candu fuel management with gradient methods using generalized perturbation theory,” in *Mathematics and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Applications*, vol. on CD, 2005.
- [34] P. Chan, K. Tsang, and D. Buss, “Reactor physics of ng-candu,” in *Canadian Nuclear Society Twenty-Second Annual Conference*, vol. on CD, 2001.
- [35] M. Boubcher, H. C. Chow, and R. Chambon, “Acr-700 core calculations with various number of energy groups,” in *Canadian Nuclear Society Twenty-fourth Annual Conference*, vol. on CD, p. 11p, 2003.

- [36] J. A. Favorite and W. Stacey, “Variational estimates of point-kinetics parameters,” *Nuclear Science and Engineering*, vol. 121, pp. 353–370, 1995.
- [37] P. Dufour, *Implantation et application de la théorie d’équivalence dans la chaîne de calcul DRAGON/DONJON-NDF*. Mémoire de maîtrise, École Polytechnique de Montréal, Montréal, 2004.
- [38] “World nuclear association, web site.” <http://world-nuclear.org/info/>, 2006.
- [39] “Point lepreau generating station.” www.nuclear.nbpower.com/en/pointlepreaurefurb/PLGS_History_factsheet_sept2005.pdf, 2002.
- [40] E. F. Author, *Parallel Metaheuristics : New Class of Algorithms* (. Wiley-Interscience, 2005.
- [41] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Don Mill, Ont.: Addison-Wesley, 1989. 412p.
- [42] G. T. Parks, “Multiobjective pwr reload core design by nondominated genetic algorithm search,” *Nuclear Science and Engineering*, vol. 124, pp. 178–187, 1996.
- [43] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, 4598, pp. 671–680, 1983.
- [44] V. Cerny, “Thermodynamical approach to the traveling salesman problem: An efficient simulation approach,” *Journal of Optimization Theory and Applications*, vol. 45 1, pp. 41–51, 1985.
- [45] F. Glover and M. Laguna, *Tabu Search*. Boston: Kluwer Academic Publishers, 1997. 382p.

- [46] M. D. DeChaine and M. A. Feltus, "Fuel management using genetic algorithms and expert knowledge," *Nuclear Science and Engineering*, vol. 124, pp. 188–196, 1996.
- [47] Y. Kobayashi and E. Aiyoshi, "Optimization of bwr loading pattern using improved genetic algorithm," in *Proceedings of the 2001 IEEE International Symposium on Intelligent Control*, pp. 383–390, 2001.
- [48] Y. Kobayashi and E. Aiyoshi, "Integrative two-stage optimization for core design of bwr using genetic algorithm," in *Proceedings of the 2003 IEEE Conference on Control Applications*, pp. 109–114, 2003.
- [49] A. Yamamoto and H. Hashimoto, "Application of the distribution genetic algorithm for in-core fuel optimization problems under parallel computational environment," *Journal of Nuclear Science and Technology*, vol. 39, no. 12, pp. 1281–1288, 2002.
- [50] C. M. N. A. Pereira and C. M. F. Lapa, "Coarse-grained parallel genetic algorithm applied to a nuclear reactor core design optimization problem," *Annals of Nuclear Energy*, vol. 30, pp. 555–565, 2003.
- [51] A. Yamamoto and H. Hashimoto, "Application of temperature parallel simulated annealing to loading pattern optimization of pwr," *Nuclear Science and Engineering*, vol. 136, pp. 247–257, 2000.
- [52] A. Castillo, J. J. Ortiz, G. Alonso, L. B. Morales, and E. D. Valle, "Bwr control rod design using tabu search," *Annals of Nuclear Energy*, vol. 32, pp. 151–161, 2005. www.sciencedirect.com.
- [53] A. Castillo, G. Alonso, L. B. Morales, C. M. D. Campo, J. L. Francois, and E. D. Valle, "Bwr fuel reloads design using tabu search technique," *Annals*

- of Nuclear Energy*, vol. 31, pp. 151–161, 2004. http://www.optimization-online.org/DB_HTML/2003/12/790.html.
- [54] S. Jagawa, T. Yoshii, and A. Fukao, “Boiling water reactor loading pattern optimization using simple linear perturbation and modified tabu search methods,” *Nuclear Science and Engineering*, vol. 138, pp. 67–77, 2001.
- [55] I. A. B. Hmaida, J. N. Carter, C. R. E. D. Oliviera, A. J. H. Goddard, and G. T. Parks, “Nuclear in-core fuel management optimisation using the tabu search method,” in *Mathematics and Computation, Reactors Physics and Environmental Analysis in Nuclear Applications*, (Madrid, Spain), pp. 1658–1666, 1999.
- [56] P. Siarry and G. Berthiau, “Fitting of tabu search to optimize functions of continuous variables,” *International Journal for numerical Methods in Engineering*, vol. 40, pp. 2449–2457, 1997.
- [57] R. Chelouah and P. Siarry, “Tabu search applied to global optimization,” *European Journal of Operational Research*, vol. 123, pp. 256–270, 2000.
- [58] J. A. Nelder and R. Mead, “A simplex method for function minimization,” *The Computer journal*, vol. 7, pp. 308–313, 1965.
- [59] R. Chelouah, *Adaptation aux problèmes variables continues de plusieurs métaheuristiques d’optimisation combinatoire : la méthode tabou, les algorithmes génétiques et les méthodes hybrides. Application en contrôle non destructif*. Thèse de Doctorat, Université de Cergy-Pontoise, Cergy-Pontoise, France, 1999.
- [60] R. Chambon, D. Rozon, and E. Varin, “Étude préliminaire de la physique du ng-candu,” in *27th CNS/CNA Student Conference*, 2002.

- [61] E. Varin, G. Marleau, and R. Chambon, “Transport-diffusion coupling for candu reactor core follow-up,” in *International Conference on Supercomputing in Nuclear Applications SNA*, 2003.
- [62] “Aecl web site.” <http://www.aecl.ca>, 2006.
- [63] G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell, *Engineering optimization, methods and applications*. New York: Wiley, 1983.
- [64] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*. New York: The McGraw-Hill Companies, Inc., 1996. 692p.
- [65] R. E. Griffiths and R. A. Stewart, “A nonlinear programming technique for the optimization of continuous processing systems,” *Management Science*, vol. 7, pp. 379–392, 1961.
- [66] R. W. Cottle and G. B. Danzig, “Complementarity pivot theory of mathematical programming,” *Mathematical programming*, vol. 3, pp. 210–224, 1968.
- [67] R. W. Cottle, “Monotone solutions of the parametric linear complementarity problem,” *Linear Algebra and its Applications*, vol. 1, pp. 103–125, 1972.
- [68] H. D. S. M. S. Bazaraa and C. M. Shetty, *Nonlinear programming: theory and algorithms*. New York: Wiley, 1993.
- [69] R. W. H. Sargent, “An efficient implementation of the lemke algorithm and its extension to deal with upper and lower bounds,” *Mathematical Programming Study* 7, pp. 36–54, 1978.
- [70] T. Falcon, “Étude et méthode de résolution d’un problème de programmation quasi-linéaire,” Tech. Rep. GAN-52, École Polytechnique de Montréal, Groupe d’analyse Nucléaire, 1983.

- [71] S. K. Park and K. W. Miller, “Random number generators: Good ones are hard to find,” *Communications of the ACM*, vol. 31, no. 10, pp. 1192–1201, 1988.

ANNEXE I

**ALGORITHMES DÉTAILLÉS DES MÉTHODES DE GRADIENTS
D'OPTEX**

Chacune des méthodes présentées dans cette annexe a été largement décrite dans la littérature. Le but de cette annexe est de rassembler les notions théoriques (équations, algorithmes, sous-problèmes, ...) des approches implantées dans le logiciel DONJON. Parallèlement, une compréhension plus précise du code permet de mieux anticiper les limites d'utilisation de celui-ci. L'idée principale de toutes ces méthodes étant une linéarisation, les approximations faites peuvent entraîner un échec des différents algorithmes. Nous mettrons donc une emphase sur les arrêts possibles et les solutions envisagées le cas échéant.

Le problème à résoudre est donné par les équations suivantes:

$$\min_{\Delta \vec{X}} \nabla f^k \cdot \Delta \vec{X} \quad \text{avec} \quad \begin{cases} \nabla h_i^k \cdot \Delta \vec{X} & = & b_i - h_i(\vec{X}^k) \\ \nabla g_j^k \cdot \Delta \vec{X} & \leq & c_j - g_j(\vec{X}^k) \end{cases} \quad (2.22)$$

$$\text{et} \quad \begin{cases} X_n^{INF} - X_n^k \leq \Delta X_n \leq X_n^{SUP} - X_n^k \\ \sum w_n \cdot \Delta X_n^2 \leq S^2 \end{cases} \quad (2.23)$$

Les trois méthodes présentées sont conçues pour ne traiter que des variables positives. Aussi le changement de variable suivant est effectué:

$$x_n = \Delta X_n - \Delta X_{n,min} = \Delta X_n - \max\{X_n^{INF} - X_n^k, -\sqrt{w_n}S\} \quad (I.1)$$

En employant une notation matricielle plus générale, le problème d'optimisation

peut alors s'écrire sous la forme suivante:

$$\min_x c^T x \quad \text{avec} \quad \begin{cases} A_{eq}x = b_{eq} \\ A_{in}x \leq b_{in} \\ 0 \leq x_n \leq x_n^{max} \end{cases} \quad (\text{I.2})$$

$$\text{et} \quad x^T P x \leq S^2 \quad (\text{I.3})$$

I.1 SIMPLEX

La méthode du SIMPLEX, inventée par Dantzig en 1947, est décrite par différents auteurs avec des approches plutôt pratiques ([30]) ou plus théoriques ([63], [64]). Cette méthode est très répandue et sert de base à la méthode MAP que nous décrirons par la suite. Elle n'accepte que des contraintes d'égalité. Les contraintes d'inégalité (exceptées $x \geq 0$) sont donc transformées en contraintes d'égalité par l'utilisation de variables d'écart $x_{in} \geq 0$. Les contraintes se présentent alors sous la forme matricielle suivante:

$$Ax' = b \quad \text{avec} \quad A = \begin{bmatrix} A_{eq} & 0 \\ A_{in} & I \end{bmatrix}, \quad x' = [x_n \quad x_{in}] \quad \text{et} \quad b = \begin{bmatrix} b_{eq} \\ b_{in} \end{bmatrix}$$

où I est la matrice identité.

La méthode du SIMPLEX ne peut pas tenir compte de l'équation (I.3), car cette dernière n'est pas linéaire. Or pour essayer de maintenir la validité à l'approximation linéaire, le pas d'avance est quand même limité. L'hyper-ellipsoïde est alors remplacé par un hyper-rectangle défini par:

$$-\sqrt{w_n}S \leq \Delta X_n \leq \sqrt{w_n}S \quad (\text{I.4})$$

Supposons dans un premier temps que dans notre cas, le domaine réalisable n'est pas vide. Alors, ce dernier est fini et peut être représenté par un polyèdre, car les variables de décision sont bornées. Pour un tel problème, la théorie [64] démontre que l'optimum est sur un sommet du polyèdre (à moins d'une dégénérescence au quel cas l'optimum n'est pas unique et est représenté par une arête ou une face du polyèdre). Une représentation géométrique de l'algorithme du SIMPLEX montre que la solution se déplace, à chaque itération, à un sommet adjacent en améliorant la solution.

Le résumé de l'algorithme du SIMPLEX présenté par Gauvin est reproduit ici pour sa bonne représentation des différents choix à faire lors de la résolution d'un problème classique. Quelques explications supplémentaires ont été rajoutées pour expliquer les notations choisies par l'auteur.

Le problème primal est donné par: et son dual par:

$$\min_x c^T x \text{ avec } Ax = b \qquad \max_\lambda \lambda^T b \text{ avec } \lambda^T A \leq c^T$$

λ est la variable duale. Nous supposons que la résolution des problèmes primal et dual conduisent à la même solution.

1. $A = [B \ N]$ où B représente une matrice carrée formée par certaines colonnes de la matrice A , de tel sorte à ce qu'elle soit également une base. N est donnée par le reste des colonnes de la matrice A . Les variables de décision hors base \bar{x}_N sont nulles. Donc en remplaçant A par $[B \ N]$ dans l'équation des contraintes, nous obtenons pour les variables de décision de la base, qui sont positives par définition: $\bar{x}_B = B^{-1}b \geq 0$. Les variables duales de la base sont alors données par: $\lambda_B^T = c_B^T B^{-1}$

Si les coûts réduits $c_N^T - \lambda_B^T N \geq 0$, alors \bar{x}_B et λ_B^T sont les solutions optimales pour le primal et le dual; sinon λ_B^T est non admissible pour le dual. Une variable de décision d'indice q hors base (colonne $\in N$) remplace une de la base (colonne $\in B$).

2. Choisir la colonne *entrante* $A_q \in N$ avec $c_q - \lambda_B^T A_q < 0$ et calculer $Y_q = B^{-1} A_q$.

3. Pour déterminer la colonne *sortante* $A_p \in B$, calculer:

$$\frac{\bar{x}_p}{y_{pq}} = \min \left\{ \frac{\bar{x}_i}{y_{iq}} \mid y_{iq} > 0 \ A_i \in B \right\}$$

4. On obtient la nouvelle base B dans laquelle A_q remplace A_p . Calculer B^{-1} et retourner à 1.

Le point de départ de l'algorithme est $x = 0$. Ce point n'étant pas forcément réalisable, un nouveau problème est défini comme suit:

$$\min_{x,v} \sum v_i \text{ avec } \begin{cases} Ax' + v = b \\ v \geq 0 \\ 0 \leq x_n \leq x_n^{max} \end{cases} \quad (\text{I.5})$$

En prenant $x = 0$ et $v = b$ comme point de départ de ce nouveau problème d'optimisation, nous obtenons avec la même méthode du SIMPLEX l'optimum (x^*, v^*) . Deux cas de figure sont alors possibles:

- $v^* = 0$, ce qui nous donne un point initial pour le problème d'origine (I.2),
- $v^* \neq 0$, ce qui veut dire que le problème d'origine n'a pas de solution.

La résolution de ce problème intermédiaire est généralement appelée la phase 1 du SIMPLEX.

I.2 MAP

La solution d'un problème linéaire tel que (I.2) par la méthode du SIMPLEX conduit forcément à une solution sur un des sommets formés par les contraintes. Pour supprimer cette 'extrémalité' de la solution, une contrainte quadratique (I.3) est rajoutée. La méthode MAP ("method of approximate programming") développée par Griffiths [65] permet de résoudre le problème défini par (I.2) et (I.3). Cette méthode est basée sur le SIMPLEX et procède par approximations successives. À chaque iteration interne m , la variable ΔX_n est remplacée par $\Delta X_n^m + \delta X_m$ où $\Delta X_n^m = \sum_{j=1}^{j=m-1} \delta X_j$. La nouvelle inconnue est donc δX_m , et le changement de variable est fait dans le problème (I.2). La contrainte quadratique est linéarisée en ne gardant que les termes du premier ordre, et un pas d'avance réduit (δ) est utilisé. La contrainte quadratique devient alors:

$$2 \sum w_n \delta X_m \Delta X_n^m \leq S^2 - \sum w_n \Delta X_n^{m2} \quad (\text{I.6})$$

et les variables de décision sont bornées:

$$-\delta_n \leq \sqrt{w_n} \delta X_m \leq \delta_n \quad (\text{I.7})$$

Le nouveau problème linéaire est résolu avec le SIMPLEX pour la variable δX_m . Une vérification du respect de la contrainte quadratique (non-linéarisée) est faite à la fin de chaque itération interne m avec ΔX_n^{m+1} . Si le nouveau point ne respecte pas la contrainte quadratique alors certaines bornes δ_n seront réduites. L'algorithme s'arrête lorsque δX_m est suffisamment petit. Une représentation graphique de cette approche est également donnée par Rozon [15].

I.3 Lemke

L'algorithme pour la résolution du problème défini par (I.2) et (I.3) par la méthode dite de Lemke tel qu'implanté dans le code est divisé en deux étapes. La première consiste à ne pas tenir compte de la contrainte quadratique. Deux choix sont alors possibles: la méthode du SIMPLEX ou celle des pivots complémentaires. La première a été décrite précédemment. La deuxième est basée sur les conditions de Kuhn-Tucker. La représentation géométrique de cette approche consiste à dire que si un point est optimum, alors soit le gradient de la fonction objectif est nul, soit une frontière du domaine réalisable est atteinte. Pour un problème linéaire tel que (I.2), cela se traduit par l'équation suivante [66]:

$$w = q + Mz \tag{I.8}$$

où $z = \begin{pmatrix} x \\ \lambda \end{pmatrix}$ avec λ qui représente les variables du dual du problème (I.2),
 $w = \begin{pmatrix} \lambda_{in} \\ x_{in} \end{pmatrix}$ avec x_{in} et λ_{in} les variables d'écart pour les contraintes d'inégalités

du primal ($A_{in}x + Ix_{in} = b$) et du dual ($A_{in}^T\lambda + I\lambda_{in} = b$) respectivement, $q = \begin{pmatrix} c \\ b \end{pmatrix}$,

$$M = \begin{pmatrix} 0 & A^T \\ -A & 0 \end{pmatrix} \text{ avec } A = \begin{pmatrix} A_{eq} \\ A_{in} \end{pmatrix}.$$

Par définition les variables x et λ sont positives, et les variables de d'écart x_{in} et λ_{in} aussi. Donc $(w, z) \geq 0$.

Si l'équation (I.8) admet une solution (w, z) telle que $w.z = 0$ ($w \geq 0$ et $z \geq 0$) alors cette solution, dite complémentaire, est aussi l'optimum du problème (I.2). La valeur optimale de la fonction objectif sera alors notée d^* .

Si la solution à la première étape de l'algorithme de Lemke satisfait la contrainte

quadratique alors le problème original défini par (I.2) et (I.3) est résolu. Si ce n'est pas le cas, Ferland [24] propose la seconde étape suivante. L'auteur souligne le fait que si le problème original a une solution alors la contrainte quadratique est forcément active. Partant de cette constatation, il propose de considérer le problème suivant:

$$\min_x \frac{1}{2} x^T P x \text{ avec } \begin{cases} Ax \leq b \\ c^T x \leq d^* + \tau \end{cases} \quad (\text{I.9})$$

où P est la matrice diagonale contenant les poids des variables de décision w_n et τ un paramètre.

Le problème d'origine revient alors à trouver la plus petite valeur $\tau^* \geq 0$ qui satisfait $\frac{1}{2} x^T P x = S^2$ et $c^T x = d^* + \tau^*$. Plusieurs problèmes (I.9) sont alors résolus en ajustant la valeur de τ , car la solution du problème (I.9) est une fonction monotone de τ [67]. L'algorithme choisi est encore basé sur les conditions de Kuhn-Tucker, et peut être directement résolu en trouvant la solution de l'équation suivante:

$$w = q + \tau p + Mz \quad (\text{I.10})$$

où $w = \begin{pmatrix} b - Ax \\ d^* + \tau - c^T x \end{pmatrix}$, $q = \begin{pmatrix} b \\ d^* \end{pmatrix}$, $p = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $M = \begin{pmatrix} AP^{-1}A^T & AP^{-1}c \\ c^T P^{-1}A^T & c^T P^{-1}c \end{pmatrix}$,
 $z = \begin{pmatrix} \nu \\ \pi \end{pmatrix}$ avec ν et π deux paramètres positifs qui proviennent de l'expression des conditions de Kuhn-Tucker pour les contraintes de l'équation (I.9). Le vecteur de décision est alors donné par [24]:

$$x = -P^{-1}A^T \nu - \pi P^{-1}c \quad (\text{I.11})$$

Il est à noter que cette notation pour w , q , M et z est plus compacte que la notation couramment employée pour les problèmes de programmation quadratique [66, 68].

La méthode des pivots complémentaires pour résoudre les équations (I.8) et (I.10) a été conçue par Lemke, et décrite avec une approche théorique dans [66], [69] et pratique dans l'annexe de [70] et [68] où elle est illustrée par un exemple. Nous résumerons rapidement ici cette méthode afin de permettre de mieux comprendre les arrêts/limites éventuels du code. La notation de l'équation (I.8): $w = q + Mz$ est utilisée par la suite. Pour l'équation (I.10), le terme τp est inclus dans le terme q .

Une initialisation du problème est faite dans un premier temps. Si $q \geq 0$, alors $(w, z) = (q, 0)$ est une solution qui respecte les conditions de complémentarité du problème, et ce dernier est donc résolu. Dans le cas contraire, la variable z_0 est définie comme l'opposée du plus petit élément de q , i.e. $z_0 = -q_s = \max\{-q_i\}$. Le problème est alors réécrit sous la forme suivante:

$$w = q + 1z_0 + Mz \quad (\text{I.12})$$

Avec la notation matricielle, les variables de w forment une base. La fin de l'initialisation consiste à faire rentrer z_0 dans la base à la place de w_s par un pivot. Les coefficients des vecteurs q , 1 et de la matrice M sont changés à chaque pivot. Les mêmes notations seront cependant gardées à chaque itération. Par la complémentarité des variables w et z , la première variable à faire rentrer dans la base est $y_s = z_s$.

La première étape est de trouver l'indice r de la variable sortante. Si toutes les composantes de la colonne d_s sous y_s sont négatives, alors il n'y a pas de solution au problème. Dans le cas contraire, l'indice r est donné par:

$$\frac{q_r}{d_{rs}} = \min_i \left\{ \frac{q_i}{d_{is}} \mid d_{is} > 0 \right\} \quad (\text{I.13})$$

Si la variable correspondant à l'indice r est z_0 alors le dernier pivot entre y_s et

z_0 est fait. La base obtenue est alors une solution du problème (I.8) (ou (I.10)). Sinon, la variable qui correspond à l'indice r est notée w_l (ou z_l).

La deuxième étape consiste à faire le pivot entre y_s et w_l (ou z_l). Étant donné la complémentarité entre les variables w et z , la prochaine variable entrante sera obligatoirement la complémentaire de celle qui vient de sortir, i.e. $y_s = z_l$ (ou $y_s = w_l$ respectivement). L'algorithme recommence à la première étape.

I.4 Limitations

Deux types d'erreurs communes aux méthodes du SIMPLEX et de Lemke sont possibles. Premièrement, l'algorithme détecte que le problème linéarisé n'a pas de solution. Ceci est dû à une trop grande approximation (S trop large) par rapport à la précision désirée lors d'une itération précédente. Deuxièmement, l'algorithme peut 'tourner en rond', signe d'une dégénérescence du problème linéarisé. Dans ce cas un nombre maximum d'itérations est fixé. Enfin, pour la méthode de Lemke, une autre erreur peut survenir s'il y a une impossibilité à interpoler le paramètre τ pour avoir $\frac{1}{2}x^T Px = S^2$. Ce cas de figure est encore dû à une trop grande approximation lors d'une itération précédente.

En cas d'erreur dans le code, nous recommandons de revenir à mi-chemin entre le dernier et l'avant dernier point valides pour la linéarisation de l'itération suivante.

ANNEXE II

GUIDE DE L'USAGER

Le guide de l'utilisateur de DONJON est en anglais. Le reste de cette annexe ne sera donc pas traduite.

II.1 Optimization modules

In this section, modules used for fuel management optimization are described.

II.1.1 The FOBJCT: module

The FOBJCT: module is used to define the different parameters for an optimization calculation. These parameters can be decision variables, constraint zone definitions, constraint limits, ... This module can also evaluate the objective function and the constraints values.

The calling specifications are:

Tableau II.1: Structure FOBJCT:

```
OPTIM := FOBJCT: [ OPTIM ] [ MAPFL ] [ FLUX [ FLUXP ] ] [ MACRO ] [ TRACK
INDEX ] :: (descfobjct)
```

where

OPTIM **character*12** name of the extended OPTIMIZE. If *OPTIM* appears on the RHS, the information previously stored in *OPTIM* is modified if necessary and stored.

MAPFL **character*12** name of the extended MAP. If *MAPFL* appears on the

RHS, the information in it will be read for many parameters initialization.

<i>FLUX</i>	character*12 name of the FLUX linked list. This object is used for the function evaluation. It is recommended to provide it even if no function evaluation is done for many parameter reading.
<i>FLUXP</i>	character*12 name of the FLUX linked list. This object is used for some function evaluation such as void reactivity.
<i>MACRO</i>	character*12 name of the MACROLIB linked list file containing fuel regions description and burnup informations. If it appears on RHS, it means it will be necessary for a function evaluation (objective or constraint).
<i>TRACK</i>	character*12 name of the TRACKING linked list file containing the tracking informations. If it appears on RHS, it means it will be necessary for a function evaluation (objective or constraint) or to memorize the average exit burnup or the fuel cost distribution.
<i>INDEX</i>	character*12 name of the INDEX linked list file containing the index informations. If it appears on RHS, it means it will be necessary for a function evaluation (objective or constraint) or to memorize the average exit burnup or the fuel cost distribution.
(descfobjct)	structure containing the data to module FOBJCT:.

II.1.1.1 Data input for module FOBJCT:

Tableau II.2: Structure (**descfobjct**)

```
[ EDIT iprint ]
[ CTRL-ZONE-DF (czdf_data) ]
[ FUEL-COST-DF (fcdf_data) ]
[ EXIT-B-DIST MEMORY]
[ CST-ZONE-DF (cstzdf_data) ]
[ EVAL-OBJ-CST (eval_data) ]
;
```

where

EDIT	key word used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module.
CTRL-ZONE-DF	key word used to define the decision variables and their zones of influence.
(czdf_data)	structure containing the data to the option CTRL-ZONE-DF.
FUEL-COST-DF	key word used to define the cost of the uranium in the core.
(fcdf_data)	structure containing the data to the option FUEL-COST-DF.
EXIT-B-DIST	key word used to specify that the distribution of the average exit burnup for each volume will be pre-calculated.
MEMORY	key word used to specify that the distribution of the average exit burnup will be stored in the <i>OPTIM</i> object.
CST-ZONE-DF	key word used to define the constraint (type, value, zone of influence).
(cstzdf_data)	structure containing the data to the option CST-ZONE-DF.
EVAL-OBJ-CST	key word used to define and evaluate the objective and / or constraints functions.
(eval_data)	structure containing the data to the option EVAL-OBJ-CST. This will be treated as a section in itself because other module will refer to it.

Tableau II.3: Structure (**czdf_data**)

<pre>[BURNUP-ZONE <i>burn_min</i> <i>burn_max</i> { ALL { { Y N }_{<i>i</i>,<i>i</i>=1,<i>nbz</i>} } }]</pre> <pre>[ENRICH-ZONE <i>enri_min</i> <i>enri_max</i> { { Y N }_{<i>i</i>,<i>i</i>=1,<i>nez</i>} }]</pre>

where

BURNUP-ZONE	key word used to specify that exit-burnup decision variables will be set. This exit-burnup zone were defined previously and are stored in the MAP.
-------------	--

<i>burn_{min}</i>	minimum value of the exit-burnup.
<i>burn_{max}</i>	maximum value of the exit-burnup.
ALL	key word used to specify that all burnup-exit zone will be a decision variable.
Y	key word used to specify that a specific burnup-exit zone will be a decision variable.
N	key word used to specify that a specific burnup-exit zone will not be a decision variable.
<i>nbz</i>	number of average exit burnup zone.
ENRICH-ZONE	key word used to specify that exit-burnup decision variables will be set. This exit-burnup zone were defined previously and are stored in the MAP.
<i>enri_{min}</i>	minimum value of the exit-burnup.
<i>enri_{max}</i>	maximum value of the exit-burnup.
Y	key word used to specify that a specific enrichment zone will be a decision variable.
N	key word used to specify that a specific enrichment zone will not be a decision variable.
<i>nez</i>	number of enrichment zone.

Tableau II.4: Structure (**fcdf_data**)

```
[ FIXED  $cost_i, i=1, nez$  )
| DEPENDANT  $\varepsilon_w C_{NU} C_S C_{FAB}$  interest  $t_{obt} t_{enr}$  ]
[ MEMORY ]
```

where

FIXED key word used to specify that the price of the fuel is fixed for each enrichment zone.

<i>cost</i>	cost of the fuel.
<i>nez</i>	number of enrichment zone.
DEPENDANT	key word used to specify that the price of the fuel is dependant of the enrichment for each enrichment zone.
ε_w	U^{235} concentration of waste uranium after the separation work.
C_{NU}	natural uranium cost (\$/kg).
C_S	cost of a separation work unit (\$/SWU).
C_{FAB}	cost of fabrication of the bundles (\$/kg).
<i>interest</i>	interest rate (y^{-1}).
t_{obt}	time to obtain uranium (y).
t_{enr}	time for enrichment (y).
MEMORY	key word used to specify that the distribution of the purchase cost of uranium actually in the reactor will be pre-calculated and stored in the <i>OPTIM</i> object.

Tableau II.5: Structure (**cstzdf_data**)

```

[ KEFF  $k_{eff}$  ]
[ MAXPOWER
... [[ ZONE-DEF [[ SURV-ZONE  $n_{surv\_zone}$  [[ PLAN  $i_{plan}$  {  $i_{zone,j,j=1,n_{cha}}$  | SAME  $j_{plan}$  } ] ]
.....| BUNDLE { ALL | [[ PLAN  $i_{plan}$  { { 0 | 1 } $_{j,j=1,n_{cha}}$  | SAME  $j_{plan}$  } ] ]
.....| CHANNEL { ALL | { 0 | 1 } $_{j,j=1,n_{cha}}$  } ] ]
... | VALUE-DEF { {  $i_{zone1}$   $cst_{lim}$  | RANGE  $i_{zone1}$   $i_{zone2}$  {  $cst_{lim,j,j=i_{zone1},i_{zone2}}$  | ALLSAME
 $cst_{lim}$  } } ] ]
... END-MAX-POW ]
[ VOID-REACT-FC  $\rho_{V,FC}$  ]
[ ANALYTIC-FCT  $cst_{type}$   $cst_{lim}$  ]

```

where

KEFF key word used to defined k_{eff} .

k_{eff}	neutron multiplication factor aimed (this is a constraint of type equal).
MAXPOWER	key word used to defined a maximum power in a zone (this is a constraint of type inferior).
ZONE-DEF	key word used to specify that the definition of the zone will be provided.
SURV-ZONE	key word used to specified that the zone will be defined manually.
n_{surv_zone}	total number of surveillance zone.
i_{zone}	number of the zone that the bundle is part of (0 if no surveillance zone for this bundle).
BUNDLE	key word to specified that surveillance zone are bundles.
CHANNEL	key word to specified that surveillance zone are channels.
PLAN	key word to specify that the definition of surveillance zone for i_{plan} will follow.
i_{plan}	numbers of the plan to be defined.
SAME	key word used to specify that the definition of surveillance zone in the plan i_{plan} will be the same one as in the plan j_{plan} .
j_{plan}	number of the plan already defined.
ALL	key word used to specify that the power in all bundles or channels will be a constraint.
n_{cha}	number of channels.
VALUE-DEF	key word used to specify that the limit of maximum zone power will be provided.
i_{zone1}	first number of surveillance zone.
cst_{lim}	constraint limit.
RANGE	key word used to specify that the constraint limit will be specified for several zones.
i_{zone2}	second number of surveillance zone.
ALLSAME	key word used to specify that all the constraint will have the same limit for the zone number between i_{zone1} and i_{zone2} .
END-MAX-POW	key word used to specify that the definition of the maximum power surveillance zones is finished.

VOID-REAC-FC	key word used to define the full core void reactivity.
$\rho_{V,FC}$	full core void reactivity.
ANALYTIC-FCT	key word used to specify that the corresponding constraint will be defined analytically.
cst_{type}	type of the analytic constraint (-1 for \leq , 0 for $=$ and 1 for \geq).

II.1.1.2 Data input for module functions definition

Because the functions definition is common with other modules, its description will be grouped in this section, as a independent part of the **FOBJCT**: module description.

The functions definition is based on inverted polish notation logic. Some functions are predefined, but if it is not the case, new functions can be defined manually. In this particular case, variables may be required. Some of them are predefined also, otherwise the user can get them with the same logic as the module **GREP**. For the functions representing the constraints the user need to specify its number. So it is important to know the order in which constraints where defined.

Tableau II.6: Structure (**eval_data**)

```
[[ { FOBJ | CONSTRAINT  $i_{cst1}$   $i_{cst2}$  } { [ VARDEF (vardef_data) ] (seq_data) |
FUNCT-PREDEF predef_func } ]]
```

where

FOBJ	key word used to specify that the objective function will be evaluated.
CONSTRAINT	key word used to specify that constraint functions between number i_{cst1} and i_{cst2} will be evaluated.
i_{cst1}	first number of constraint.
i_{cst2}	second number of constraint.
VARDEF	key word used to define the variables needed for the function evaluation.

(vardef_data)	structure containing the data to the option VARDEF.
(seq_data)	structure containing the data used to defined a function directly by the user.
FUNCT-PREDEF	key word used to specify that a predefined function will be evaluated.
<i>predef_func</i>	name of the predefined function. The predefined function name are : UCOST DX-UCOST DPHI-UCOST POWERLIMIT DX-POWER DPHI-POWER KEFF D-KEFF VOID-REAC-FC D-VOID-R-FC KEFF=KREF k_{ref} D-KEFF=KREF k_{ref} MINPCMAX $q_{moy} i_{cst1} i_{cst2}$ D-MINPCMAX $q_{moy} i_{cst1} i_{cst2}$

Where :

UCOST is defined by F_C :

$$F_C = \frac{\langle \frac{C_F(\varepsilon_j)}{B_j} \cdot H, \phi \rangle_{réacteur}}{\langle H, \phi \rangle_{réacteur}} \quad (\text{II.1})$$

DX-UCOST is defined by $\frac{\partial F_C}{\partial X_i}$:

$$\begin{aligned} \frac{\partial F_C}{\partial X_i} = & \frac{\langle \frac{\partial C_u}{\partial X_i}, \frac{H}{B} \phi \rangle_{V_i}}{\langle H, \phi \rangle_V} + \frac{\langle C_u \left(\frac{1}{B} \frac{\partial H}{\partial X_i} - \frac{H}{B^2} \frac{\partial B}{\partial X_i} \right), \phi \rangle_{V_i}}{\langle H, \phi \rangle_V} \\ & - F_C \cdot \frac{\langle \frac{\partial H}{\partial X_i}, \phi \rangle_{V_i}}{\langle H, \phi \rangle_V} \quad i \in (1, n_{var}) \end{aligned} \quad (\text{II.2})$$

DPHI-UCOST is defined by $\frac{\partial F_C}{\partial \phi}$:

$$\frac{\partial F_C}{\partial \phi} = S_{F_C}^* = \frac{C_u \cdot H(\vec{r}) - F_C \cdot H(\vec{r})}{\langle H, \phi \rangle_V} \quad (\text{II.3})$$

POWERLIMIT is defined by q_j :

$$q_j = ZPPF_j \cdot \frac{V \langle H, \phi \rangle_{V_j}}{V_j \langle H, \phi \rangle_V} \leq \frac{P_{lim}}{\bar{P}} = f_{lim} \quad (\text{II.4})$$

DX-POWER is defined by $\frac{\partial q_j}{\partial X_i}$:

$$\begin{aligned} \frac{\partial q_j}{\partial X_i} = & \frac{q_j}{ZPPF_j} \cdot \frac{\partial ZPPF_j}{\partial X_i} \cdot \delta_{ij} + \frac{q_j}{\langle H, \phi \rangle_{V_j}} \left\langle \frac{\partial H}{\partial X_i}, \phi \right\rangle_{V_j} \cdot \delta_{ij} \\ & - \frac{q_j}{\langle H, \phi \rangle_V} \left\langle \frac{\partial H}{\partial X_i}, \phi \right\rangle_V \quad i \in (1, I) \text{ et } j \in (1, n_{control-zone}) \end{aligned} \quad (\text{II.5})$$

DPHI-POWER is defined by $\frac{\partial q_j}{\partial \phi}$:

$$\frac{\partial q_j}{\partial \phi} = S_{q_j}^* = \frac{ZPPF_j \frac{V}{V_j} \cdot H(\vec{r}_j) - q_j \cdot H(\vec{r})}{\langle H, \phi \rangle_V} \quad j \in (1, n_{control-zone}) \quad (\text{II.6})$$

where $H(\vec{r}_j) = \begin{cases} H_j & \vec{r} \in V_j \\ 0 & \text{otherwise} \end{cases}$

KEFF is defined by k_{eff} : k_{eff} is directly taken in the *FLUX* object.

D-KEFF is defined by $\frac{dk_{eff}}{dX_i}$:

$$\frac{dk_{eff}}{dX_i} = -k_{eff}^2 \frac{d\lambda}{dX_i} \quad (\text{II.7})$$

where $\frac{d\lambda}{dX_i}$ is the derivative of the eigenvalue previously calculated with the PERTUR: module.

VOID-REAC-FC is defined by ρ_V :

$$\rho_V = \lambda - \lambda_V = \frac{1}{k_{eff}} - \frac{1}{k_{eff,V}} \quad (\text{II.8})$$

where k_{eff} and $k_{eff,V}$ are directly taken in the *FLUX* and *FLUXP* object respectively.

D-VOID-R-FC is defined by $\frac{d\rho_V}{dX_i}$:

$$\frac{d\rho_V}{dX_i} = \frac{d\lambda}{dX_i} - \frac{d\lambda_V}{dX_i} \quad (\text{II.9})$$

where $\frac{d\lambda}{dX_i}$ is the derivative of the eigenvalue previously calculated with the PERTUR: module and $\frac{d\lambda_V}{dX_i}$ is the derivative of the eigenvalue previously calculated for a voided

reactor with the PERTUR: module.

KEFF=KREF is defined by Δk_{eff} :

$$\Delta k_{eff} = (k_{eff} - k_{ref})^2 \quad (\text{II.10})$$

where k_{ref} is the required reference multiplication factor.

D-KEFF=KREF is defined by

$$\frac{d\Delta k_{eff}}{dX_i} = -2 * (k_{eff} - k_{ref}) \cdot k_{eff}^2 \frac{d\lambda}{dX_i} \quad (\text{II.11})$$

where k_{ref} is the required reference multiplication factor and $\frac{d\lambda}{dX_i}$ is the derivative of the eigenvalue previously calculated with the PERTUR: module.

MINPCMAX is defined by

$$EPmoy = \sum_{j=i_{cst1}|q_j>q_{moy}}^{i_{cst2}} (q_j - q_{moy})^{2m} \quad (\text{II.12})$$

where q_{moy} is the average power zone. m can be changed with the module QLPUTL:. The sum is performed from constraint i_{cst1} to i_{cst2} .

D-MINPCMAX is defined by

$$\frac{d\Delta EPmoy}{dX_i} = \sum_{j=i_{cst1}|q_j>q_{moy}}^{i_{cst2}} 2m(q_j - q_{moy})^{2m-1} \frac{dq_j}{dX_i} \quad (\text{II.13})$$

where q_{moy} is the average power zone. m can be changed with the module QLPUTL:. The sum is performed from constraint i_{cst1} to i_{cst2} .

Tableau II.7: Structure (**vardef_data**)

```
[[ LOAD object [[ DOWN repertory ]] GREP data_name IN var_name ]]
[[ MSYS*FLUX object_sys { A | B } object_flux [ ADJOINT ] IN var_name ]]
[[ PREDEF predef_var ]]
```

where

LOAD	key word used to define the object where the data are stored.
<i>object</i>	name of the object.
DOWN	key word used to go in a sub-directory.
<i>repertory</i>	name of the repertory.
GREP	key word used to define the name of the data to load.
<i>data_name</i>	name of the data to load.
IN	key word used to define the name of the local variable.
<i>var_name</i>	name of the local variable name.
MSYS*FLUX	key word used to specify that a local variable will be calculated by the product of a system matrix and a flux (or adjoint).
<i>object_sys</i>	name of the object L_SYSTEM.
A	key word used to specify that the system matrix corresponding to the lost of the neutrons will be used. $(\mathbf{A} - \lambda\mathbf{B})\phi = 0$
B	key word used to specify that the system matrix corresponding to the production of the neutrons will be used. $(A - \lambda\mathbf{B})\phi = 0$
<i>object_flux</i>	name of the object L_FLUX.
ADJOINT	key word used to specify that the adjoint flux will be used instead of the flux (default value). In this case the adjoint system matrix are used automatically.
PREDEF	key word used to specify that a predefined variable will be load.
<i>predef_var</i>	name of the predefined variable. The predefined variable name are define below.

key word	contents	size
FLUX	neutron flux distribution	$nun \times ngrp$
AFLUX	adjoint flux distribution of the second provided	$nun \times ngrp$
	L_FLUX	
FLUX2	neutron flux distribution of the second provided	$nun \times ngrp$
	L_FLUX	
AFLUX2	adjoint flux distribution	$nun \times ngrp$
FLUX-AV	average flux distribution by channel	$nch \times ngrp$
FLUX-AX	axial average flux distribution	$n_k \times ngrp \times n_z$

DIFFX	diffusion coefficient along X abscisse	$nun \times ngrp$
DIFFY	diffusion coefficient along Y abscisse	$nun \times ngrp$
DIFFZ	diffusion coefficient along Z abscisse	$nun \times ngrp$
TOTAL	total cross-sections	$nun \times ngrp$
NFTOT	fission cross-sections	$nun \times ngrp$
NUSIGF	number of neutrons per fission time fission cross-sections	$nun \times ngrp$
H-FACTORS	fission cross section times the energy recovered by fission	$nun \times ngrp$
CHI	fission spectrum	$nun \times ngrp$
SIGW-0	isotropic component of the within group of the scattering of the scattering cross-sections	$nun \times ngrp$
SIGW-1	linearly anisotropic component of the within group of the scattering of the scattering cross-sections	$nun \times ngrp$
D-TOTAL	derivative of total cross-sections	$nun \times ngrp$
D-CHI	derivative of fission spectrum	$nun \times ngrp$
D-DIFFX	derivative of diffusion coefficients along X abscisse	$nun \times ngrp$
D-DIFFY	derivative of diffusion coefficients along Y abscisse	$nun \times ngrp$
D-DIFFZ	derivative of diffusion coefficients along Z abscisse	$nun \times ngrp$
D-NUSIGF	derivative of fission cross-sections	$nun \times ngrp$
D-NFTOT	derivative of number of neutrons per fission time fission cross-sections	$nun \times ngrp$
D-HFACT	derivative of fission cross section times the energy recovered by fission	$nun \times ngrp$
D-SIGW0	derivative of isotropic component of the within group of the scattering of the scattering cross-sections	$nun \times ngrp$
D-SIGW1	derivative of linearly anisotropic component of the within group of the scattering of the scattering cross-sections	$nun \times ngrp$
A*PHI	A system matrix times neutron flux vector	$nun \times ngrp$
B*PHI	B system matrix times neutron flux vector	$nun \times ngrp$
AP*PHI	perturbated A system matrix times neutron flux vector	$nun \times ngrp$
BP*PHI	perturbated B system matrix times neutron flux vector	$nun \times ngrp$
FUNCVALUE	value of the function, usually used when the derivative function is calculated (see the definition of the DX-UCOST predefined function for example).	$ncst+1$

FUNCZVOL	value of the volume on which the function is defined / integrated.	$ncst+1$
KEFF	k_{eff}	$nun \times ngrp$
KEFF-VOID	k_{eff} corresponding to a pertubated flux	$nun \times ngrp$
D-LAMBDA	derivative of the eigenvalue	$nun \times ngrp$
D-LAMBDA-V	derivative of the eigenvalue corresponding to a pertu- bated flux	$nun \times ngrp$

Tableau II.8: Structure (**seq_data**)

```

INIT
[[ (data)
| INTEGRAL { REACTOR | CORE | CST-ZONE | VAR-ZONE | DBL-ZONE | DISCRETE-ALL |
DISCRETE-COR | DISCRETE-CST }
... [[ (data) | ENERGY { ALL | gpr_from gpr_to } [[ (data) ]] END-ENERGY ]]
... END-INTEGRAL ]] ]]
END

```

where

INIT	key word used to specify that the function definition will follow.
(data)	structure containing the data used to defined parts of the function.
INTEGRAL	key word used to define an integral over one volume and energy.
REACTOR	key word used to specify that the integration volume is the reactor
CORE	key word used to specify that the integration volume is the core (all the bundles).
CST-ZONE	key word used to specify that the integration volume is a control zone of one constraint.
VAR-ZONE	key word used to specify that the integration volume is a zone where a decision variable applies.
DBL-ZONE	key word used to specify that the integration volume is on the intersection of a control zone of one constraint and a zone where a decision variable applies.

DISCRETE-ALL	key word used to specify that the integration will be performed only on the energy for every point of the reactor.
DISCRETE-COR	key word used to specify that the integration will be performed only on the energy for every point of the core.
DISCRETE-CST	key word used to specify that the integration will be performed only on the energy for every point of a control zone of one constraint.
ENERGY	key word used to define the energy part of the integration.
ALL	key word used to specify that the integration will be performed on all energy groups.
<i>grpfrom</i>	number of the first energy group for the integration.
<i>grpto</i>	number of the last energy group for the integration.
END-ENERGY	key word used to specify that the definition of the integral over energy is finished.
END-INTEGRAL	key word used to specify that the definition of the integral is finished.
END	key word used to specify that the definition of the function is finished.

Tableau II.9: Structure (**data**)

[[{ *real* | **VAR** *loc_var_name* | *operator* | **VARF** *loc_var_name* }]]

where

<i>real</i>	real number.
VAR	key word used to specify that a local variable will be used.
VARF	key word used to specify that a local variable which depend with the functional will be used (ex: zone volume).
<i>loc_var_name</i>	name of a local variable name. Note : it has to be loaded before.
<i>operator</i>	name of a numerical operator. The name must be one of these : PLUS, +, MINUS, -, TIMES, *, DIVISION, /, POWER, **, MAX, MIN, LOG, LN, EXP, SIN, COS, TAN, ABS, SQRT.

II.1.2 The QLPUTL: module

The QLPUTL: module is used to define the optimization options and tools. It is also used to do some pre-calcultaion.

The calling specifications are:

Tableau II.10: Structure QLPUTL:

```
OPTIM := QLPUTL: OPTIM [ FLUX ] [ MAPFL ] [ MACRO [ MACROP ] ] [ SYS
[SYSP] TRACK ] :: (descqlputl)
```

where

<i>OPTIMIZE</i>	character*12 name of the extended OPTIMIZE.
<i>FLUX</i>	character*12 name of the FLUX linked list. This object is used for the function evaluation. It is recommended to provide it even if no function evaluation is done for many parameters reading. file.
<i>MAPFL</i>	character*12 name of the extended MAP linked list file containing fuel regions description and burnup informations. If <i>MAPFL</i> appears on the RHS, the information in it will be red for many parameters initialisation.
<i>MACRO</i>	character*12 name of the MACROLIB linked list file containing the mixtures cross sections. If it appears on RHS, it means it will be necessary for a function evaluation (objective or constraint).
<i>MACROP</i>	character*12 name of the MACROLIB linked list file containing the mixtures perturbed cross sections. If it appears on RHS, it means it will be necessary for a function evaluation (objective or constraint).
<i>SYS</i>	character*12 name of the SYSTEM containing the reference system matrices. SYSTEM must be a linked list. If it appears on RHS, it will be necessary for 'system matrice times flux' calculations.
<i>SYSP</i>	character*12 name of the SYSTEM containing the perturbed system matrices. SYSTEM must be a linked list. If it appears on RHS, it will be necessary for 'perturbated system matrice times flux' calculations.

TRACK **character*12** name of the TRACK (type L_TRIVAC) containing the tracking informations. TRACK must be a linked list. If it appears on RHS, it will be necessary for 'system matrice times flux' calculations.

(descqlputl) structure containing the data to module PQLUTL:.

II.1.2.1 Data input for module QLPUTL:

Tableau II.11: Structure (**descqlputl**)

```
[ EDIT iprint ]
[ DEFINITION (def_data) ]
[ STEP-VALID [ TEST-CST-VLD ] >> test1 << ]
[ STEP-INTERP { PUT | RECOVER >> test2 << } ]
[ DX-METHOD { EPS epsilon | PREVIOUS } ]
[ NEW-VAL-UPDT ]
[ PERTURB-VAR { i_var1 | RESTORE } ]
[ BKP-MACRO-P i_var2 ]
[ MAT*FLUX { A*PHI | B*PHI | AP*PHI i_var3 | BP*PHI i_var3 } ]
[ LA-PNLT [ INITIAL ] [ F-EVAL ] [ COEF-UPDATE ] [ CONV-TEST >> conv << ]
... [ ALMOST-FSBLE >> feas << ] ]
[ HISTORY i_iter1 [ POWER-CHA ] [ K-EFFECTIVE ] [ QUAD-CST ] ]
... [ CONSTRAINT { ALL | RANGE << i_cst1 >> << i_cst2 >> | << i_cst1 >> } ] ... [ DIRECT
<< num >> { vali, i = 1, num } ] ] [ POWER-CHA#2 ] [ POWER-CHA#3 ]
;
```

where

EDIT key word used to set *iprint*.

iprint index used to control the printing in module.

DEFINITION key word used to define the optimization options.

(def_data) structure containing the data to the option **DEFINITION**.

STEP-VALID key word used to verify if the new $\{X_i\}$ ends with a better objective function.

TEST-CST-VLD	key word used to verify if the new $\{X_i\}$ respects the constraints.
<i>test</i>	logical value for the validation of the new decision variables. <i>test</i> equals <i>true</i> . if $F_C(X_i^{k+1})$ is better than $F_C(X_i^k)$.
STEP-INTERP	key word used to specify that an interpolation of the objective function for the middle point between $\{X_i^k\}$ and $\{X_i^{k+1}\}$ will be done.
PUT	key word used to calculate and store the middle value.
RECOVER	key word used to verify the middle value.
<i>test2</i>	logical value for the validation of interpolation. If $F_C(X_i^{k+\frac{1}{2}})$ is less than $F_C(X_i^{k+1})$ then the middle value is kept, otherwise the new value is restored.
DX-METHOD	key word used to define which method will be used to evaluate the perturbed cross-section.

$$\frac{d\Sigma}{dX_i^k} = \frac{\Sigma(X_{i,p}^k) - \Sigma(X_i^k)}{X_{i,p}^k - X_i^k} \quad (\text{II.15})$$

where $X_{i,p}^k$ is the perturbed decision variable.

EPS	key word used to define $X_{i,p}^k$ by $X_i^k * (1 + \epsilon)$.
<i>epsilon</i>	value of ϵ .
PREVIOUS	key word used to define $X_{i,p}^k$ by X_i^{k-1} .
NEW-VAL-UPDT	key word used to update the new decision variables.
PERTURB-VAR	key word used to perturbate a decision variable.
<i>i_{var1}</i>	number of the decision variable to perturbate.
RESTORE	key word used to restore the unperturbed decision variables.
BKP-MACRO-P	key word used to store the perturbed macroscopic cross-section. By default all cross-section are stored. To store only some of them, see PQLUTL/DEFINITION/BKP-MCR-P-XS.
<i>i_{var2}</i>	number of the decision variable for which the cross-section are perturbed and will be stored.
MAT*FLUX	key word used to precalculate the system matrice times the flux.
A*PHI	key word used to precalculate the $A.\phi$ (<i>i_{var3}</i> =0 is implicit).
B*PHI	key word used to precalculate the $B.\phi$ (<i>i_{var3}</i> =0 is implicit).

AP*PHI	key word used to precalculate the $A_p.\phi$.
BP*PHI	key word used to precalculate the $B_p.\phi$.
i_{var3}	number of the step directory where the $A.\phi$ and $B.\phi$ will be stored. i_{var3} represents the decision variable for which the system matrices were perturbed and will be multiplied by the flux for optimization. The result will be stored in L_OPTIMIZE/'STEP//HSIGN ' with WRITE(HSIGN,I8) i_{var3} .
LA-PNLT	key word used to specify that a task related to the augmented lagrangian or penalty method is performed.
INITIAL	key word used to initialize the constraint weight (if not already done) and the lagrangian coefficient (when augmented lagrangian method is used).
F-EVAL	key word used to calculate the augmented lagrangian or penalty function. It can be also used with tabu search to evaluate the corresponding objective function.
COEF-UPDATE	key word used to update the constraint weights and lagrangian coefficients (if necessary) in an external iteration.
CONV-TEST	key word used to specify that a convergence test for external iteration will be performed.
<i>conv</i>	logical value representing the result of the external convergence test.
ALMOST-FSBLE	key word used to specify that a test will be performed to check if the current point is 'almost feasible'.
<i>feas</i>	logical value representing the result of the 'almost feasible' test. The maximum error allowed to set <i>feas</i> to .true. is a relative difference between prescribed and current constraint values lower than the convergence criterion.
HISTORY	key word used to store the decision vector and the functional values for iteration i_{iter1} .
i_{iter1}	integer representing the iteration number.
POWER-CHA	key word used to specify that the channel power distribution is also stored.
K-EFFECTIVE	key word used to specify that k_{eff} value is also stored.
QUAD-CST	key word used to specify that quadratic constraint limit is also stored.
CONSTRAINT	key word used to specify that some constraint values are also stored.

ALL	key word used to specify that all constraint values are stored.
RANGE	key word used to specify that values of a range of constraint are stored.
i_{cst1}	integer representing the first or only number of constraint.
i_{cst2}	integer representing the second number of constraint.
DIRECT	key word used to specify that values provided by the user are stored.
num	integer representing the number of values provided by the user.
val_i	real representing the values provided by the user.
POWER-CHA#2	same key word as POWER-CHA. It can be used when channel power distribution for a perturbed state of the reactor is also stored.
POWER-CHA#3	same key word as POWER-CHA#2.

Tableau II.12: Structure (**def.data**)

```

[ METHOD { SIMPLEX | LEMKE | MAP | AUG-LAGRANG | PENAL-METH } ]
[ { MAXIMIZE | MINIMIZE } ]
[ INN-STEP-LIM step ]
[ VAR-WEIGHT { TYP-BURNUP weight | TYP-ENRICH weight } ]
[[ CST-WEIGHT {  $i_{cst1}$  weight | RANGE  $i_{cst1}$   $i_{cst2}$  { ALLSAME weight |  $weight_j, j=i_{cst1}, i_{cst2}$  } } ] ]
[ OUT-STEP-LIM step ]
[ INN-STEP-NMX  $n_{max}$  ]
[ OUT-STEP-NMX  $n_{max}$  ]
[ INN-STEP-EPS  $\epsilon_{ext}$  ]
[ OUT-STEP-EPS  $\epsilon_{inn}$  ]
[ STEP-REDUCT { HALF | PARABOLIC } ]
[ CST-QUAD-LIM epsilon4 ]
[ BKP-MCR-P-XS { ADD | NEW } [[ XS_name ] ] ]
[ F-C-VOLUME [FOBJ {REACTOR | CORE}] [CONSTRAINT  $i_{cst1}$   $i_{cst2}$  {REACTOR | CORE | ZONE}] ]
[ CST-WGT-MFAC  $\alpha$  ]
[ CST-VIOL-EPS  $\epsilon_{cst}$  ]
[ MIN(PCMX) ^2N  $m$  ]

```

where

METHOD	key word used to define the quasi-linear programming method.
SIMPLEX	key word used to specify that the SIMPLEX method will be used.
LEMKE	key word used to specify that the LEMKE method will be used.
MAP	key word used to specify that the MAP method will be used.
AUG-LAGRANG	key word used to specify that the augmented lagrangian method will be used.
PENAL-METH	key word used to specify that the penalty method will be used.
MAXIMIZE	key word used to specify that the optimization problem will be a maximization.
MINIMIZE	key word used to specify that the optimization problem will be a minimization (default).
INN-STEP-LIM	key word used to limit the inner step of the optimization problem.
<i>step</i>	limit for a step.
VAR-WEIGHT	key word used to set the weight of the different types of the decision variables for the quadratic limit of the outer step of the optimization problem.
	$\sum w_i \cdot X_i^2 \leq S_k \quad (\text{II.16})$
CST-WEIGHT	key word used to set the weight of the constraints.
<i>i_{cst1}</i>	number of the (first) constraint to set the weight.
<i>weight</i>	weight of the constraint(s).
RANGE	key word used to specify that several constraint weights will be set.
<i>i_{cst2}</i>	number of the last constraint to set the weight.
ALLSAME	key word used to specify that the several constraint weights will be identical.
TYP-BURNUP	key word used to set a limit for a burnup type decision variable.
TYP-ENRICH	key word used to set a limit for a enrichment type decision variable.
<i>weight</i>	weight for the decision variable.
OUT-STEP-LIM	key word used to limit the outer step of the optimization problem.
INN-STEP-NMX	key word used to set the maximum of inner iteration of the optimization problem.

OUT-STEP-NMX	key word used to set the maximum of outer iteration of the optimization problem.
n_{max}	maximum number of iterations.
INN-STEP-EPS	key word used to set the tolerance of inner iteration convergence criterion of the optimization problem.
ϵ_{ext}	tolerance for convergence of inner iterations (real).
OUT-STEP-EPS	key word used to set the tolerance of outer iteration convergence criterion of the optimization problem.
ϵ_{inn}	tolerance for convergence of external iterations (real).
STEP-REDUCT	key word used to define the method of the reduction of the outer step.
HALF	key word used to specify that the step will be reduced by a factor 2.
PARABOLIC	key word used to specify that the step will be reduced with the parabolic method.
CST-QUAD-LIM	key word to set the parameter ϵ_4 for the quadratic limit of the step.
ϵ_4	parameter ϵ_4 .
BKP-MCR-P-XS	key word used to specify which of the perturbed macroscopic cross-section will be stored on a backup repertory of the L_OPTIMIZE object. (for complementary information see PQLUTL/BKP-MACRO-P)
ADD	key word used to add name of cross-section to be stored.
NEW	key word used to define a new list of name of cross-section to be stored.
XS_name	name of the cross-section to be stored. The list of available name is: DIFFX, DIFFY, DIFFZ, TOTAL, NFTOT, NUSIGF, H-FACTORS, CHI, SIGW-0, SIGW-1, SCAT-0, SCAT-1, CHI, FIXE.
F-C-VOLUME	key word used to specify that the volume where the functionals apply will be calculated.
FOBJ	key word used to specify that the volume corresponding to the objective function will be computed.
CONSTRAINT	key word used to specify that the volume corresponding to the constraints between number i_{cst1} and i_{cst2} will be computed.
i_{cst1}	number of the first constraint for which the volume will be calculated.
i_{cst2}	number of the last constraint for which the volume will be calculated.

REACTOR	key word used to specify that the volume of the functional is the whole reactor.
CORE	key word used to specify that the volume of the functional is the core represented by all the fuel channels.
ZONE	key word used to specify that the volume of the functional is its corresponding zone.
CST-WGT-MFAC	key word used to set the multiplication factor α for the constraint weight update.
α	multiplication factor for the constraint weight update.
CST-VIOL-EPS	key word used to set the precision ϵ_{cst} when the validation of a new point is done with the constraint validity.
ϵ_{cst}	precision for the constraint validity.
MIN(PCMX)^2N	key word used to set the coefficient m of the power distribution optimization problem.
m	coefficient for channel having power greater than the average.

II.1.3 The PERTUR: module

The PERTUR: module is used to compute gradients of function using the first order of perturbation theory. Then it can be used to calculate the variation of reactivity of one reactor with a small perturbation of the cross-sections. There is two different approaches to solve the problem of reactivity.

The first method uses in fact the module 'SORKEF:' of the previous version. This part of the module computes source terms based on a first order perturbation theory over diffusion equation. The direct diffusion equation for system matrix perturbations ΔA and ΔB can be written for a linear perturbation of the flux $\phi = \phi_o + \Delta\phi$:

$$(A_o - \lambda_o B_o)\Delta\phi = -(\Delta A - \lambda_o \Delta B - \Delta\lambda B_o)\phi_o \quad (\text{II.17})$$

The direct source term is then simply $(\Delta A - \lambda_o \Delta B - \Delta\lambda B_o)\phi_o$ where $\Delta\lambda$ is the first order estimate of the eigenvalue variation, Rayleigh formulation.

The adjoint source terms are easily obtained from a similar expression of the adjoint diffusion equation.

The second method is a part of the optimization modules package. To calculate $\frac{\Delta\lambda}{\Delta X_i}$, the user has to precalculate system matrices * flux. It can be done easily and automatically by using the module PQLUTL: with the key word 'MAT*FLUX'. For the specific case of the reactivity, the variation of the inverse of k-effective is given by the following equation:

$$\frac{\partial\lambda}{\partial X_i} = \lambda \left(\frac{\langle \phi^*, \frac{\partial A}{\partial X_i} \phi \rangle}{\langle \phi^*, A \phi \rangle} - \frac{\langle \phi^*, \frac{\partial B}{\partial X_i} \phi \rangle}{\langle \phi^*, B \phi \rangle} \right) \quad (\text{II.18})$$

$$\frac{\Delta\lambda}{\Delta X_i} = \lambda \left(\frac{\langle \phi^*, \frac{A_p}{\Delta X_i} \phi \rangle}{\langle \phi^*, A \phi \rangle} - \frac{\langle \phi^*, \frac{B_p}{\Delta X_i} \phi \rangle}{\langle \phi^*, B \phi \rangle} \right) \quad i \in (1, n_{var}) \quad (\text{II.19})$$

The calling specifications are:

Tableau II.13: Structure (**PERTUR:**)

```
OPTIMIZE := PERTUR: OPTIMIZE FLUX [ SYS [ SYSP ] TRACK ] [ MACRO [
MACROP ] ] :: (pertur_data)
```

where

<i>GPT</i>	character*12 name of the SOURCE containing the source terms. If <i>GPT</i> appears on the RHS, the previous values will be updated.
<i>FLUX</i>	character*12 name of the FLUX containing the unperturbed flux, direct or adjoint.
<i>SYS</i>	character*12 name of the SYSTEM containing the reference system matrices. SYSTEM must be a linked list.
<i>SYSP</i>	character*12 name of the SYSTEM containing the perturbation of the system matrices.
<i>TRACK</i>	character*12 name of the TRACK (type L.TRIVAC) containing the tracking informations. TRACK must be a linked list.
<i>OPTIMIZE</i>	character*12 name of the OPTIMIZE containing the optimization informations. <i>GPT</i> must appear on the RHS to be able to updated the previous values.
(pertur_data)	structure containing the data to the second choice for the module PERTUR:.

II.1.3.1 Data input for module PERTUR:

Tableau II.14: Structure (**pertur_data**)

```
[ EDIT iprint ]
[ VARMUN { ivar1 ivar2 | ALL }
... {D-LAMBDA | D-LAMBDA/DX | D-LAMBDA-V | D-LAMBDA-V/DX |(eval_data) } ] ;
```

where

EDIT	key word used to set <i>i</i> print.
<i>i</i> print	index used to control the printing in module.
VARNUM	key word used to define the decision variable for which the perturbation theory calculations will be done.

i_{var1}	number of the first decision variable.
i_{var2}	number of the second decision variable.
ALL	key word used to specify that the perturbation theory calculations will be done for all the decision variables.
D-LAMBDA	key word used to specify that absolute variation of the eigenvalue perturbation will be calculated for the corresponding perturbed decision variables.
D-LAMBDA/DX	key word used to specify that derivative of the eigenvalue perturbation will be calculated for the corresponding perturbed decision variables.
D-LAMBDA-V	key word used to specify that absolute variation of the eigenvalue perturbation will be calculated for the corresponding perturbed decision variables and that the provided system matrix correspond to the voided reactor (or an other configuration of the reactor).
D-LAMBDA-V/DX	key word used to specify that derivative of the eigenvalue perturbation will be calculated for the corresponding perturbed decision variables and that the provided system matrix correspond to the voided reactor (or an other configuration of the reactor).
(eval_data)	see explanations in the module FOBJCT: key word 'EVAL-OBJ-CST' II.1.1.2. Some predefined function are described too.

II.1.4 The GPTSRC: module

The GPTSRC: module is used to calculate the sources terms (direct and / or adjoint) for generalized perturbation theory.

The calling specifications are:

Tableau II.15: Structure GPTSRC:

```
{ GPT := GPTSRC: [ GPT ] OPTIMIZE FLUX [ SYS [ SYSP ] TRACK ] [ MACRO
] [ MAPFL ] :: (gptsrc_data)
```

where

<i>GPT</i>	character*12 name of the GPT linked list file containing fuel regions description and burnup informations. If <i>GPT</i> appears on the RHS, the information previously stored in <i>GPT</i> is modified if necessary and stored.
<i>OPTIMIZE</i>	character*12 name of the extended OPTIMIZE linked list.
<i>FLUX</i>	character*12 name of the FLUX linked list. This object is used for the function evaluation. It is recommended to provide it even if no function evaluation is done for many parameters reading. file.
<i>MACRO</i>	character*12 name of the MACROLIB linked list file containing fuel regions description and burnup informations. If it appears on RHS, is means it will be necessary for a function evaluation (objective or constraint).
<i>MAPFL</i>	character*12 name of the extended MAP. If <i>MAPFL</i> appears on the RHS, the information in it will be red for many parameters initialisation.
<i>TABFL</i>	character*12 name of the TABLE linked list file containing fuel regions description and burnup informations. If it appears on RHS, is means it will be necessary for a function evaluation (objective or constraint).
(gptsrc_data)	structure containing the data to module GPTSRC:.

II.1.4.1 Data input for module GPTSRC:

Tableau II.16: Structure `gptsrc_data`

```
[ EDIT iprint ]
[[ DIRECT { ivar1 ivar2 | ALL } ]]
[[ ADJOINT (eval_data) ]]
[[ OTHER { DIRECT | ADJOINT } ivar1 (eval_data) ]] ;
```

where

EDIT key word used to set *iprint*.

iprint index used to control the printing in module.

DIRECT key word used to calculate a direct source term for decision variables S_i .

$$S_i = \frac{\partial (A - \lambda B)}{\partial X_i} \cdot \phi = \frac{A_p \phi - A \phi}{\Delta X_i} - \frac{\Delta \lambda}{\Delta X_i} \cdot B \phi - \lambda \cdot \frac{B_p \phi - B \phi}{\Delta X_i} \quad (\text{II.20})$$

i_{var1} number of the first decision variable.

i_{var2} number of the second decision variable.

ALL key word used to specify that the direct source terms calculations will be done for all the decision variables.

ADJOINT key word used to calculate a adjoint source term for decision variables S_j^* .

$$S_j^* = \frac{\partial G_j}{\partial \phi} \quad (\text{II.21})$$

(**eval_data**) see explanations in the module FOBJCT: key word 'EVAL-OBJ-CST'. Some predefined function are described too.

II.1.5 The GPTFLU: module

The GPTFLU: module is used to compute the solution to a fixed source eigenvalue problem corresponding to a set of unperturbed system matrices and sources vectors.

If S is the source term of the explicit generalized adjoint equation, this module will solve:

$$(A_o - \lambda_o B_o)\Gamma_i = S_i \quad (\text{II.22})$$

If S is the source term of the implicit generalized adjoint equation, this module will solve:

$$(A_o^* - \lambda_o B_o^*)\Gamma_j^* = S_j^* \quad (\text{II.23})$$

The calling specifications are:

Tableau II.17: Structure GPTFLU:

DFLUX := GPTFLU: [*DFLUX*] *GPT FLUX SYS TRACK* :: (**gptflu_data**)

where

<i>DFLUX</i>	character*12 name of the FLUX containing the solution. If <i>DFLUX</i> appears on the RHS, the solution previously stored in <i>DFLUX</i> is used to initialize the new iterative process; otherwise, a uniform unknown vector is used.
<i>GPT</i>	character*12 name of the SOURCE containing the sources previously computed with SORKEF: module or any other source construction.
<i>FLUX</i>	character*12 name of the FLUX containing the unperturbed flux.
<i>SYS</i>	character*12 name of the SYSTEM containing the reference system matrices. SYSTEM must be a linked list.
<i>TRACK</i>	character*12 name of the TRACK (type L_TRIVAC) containing the tracking informations. TRACK must be a linked list.
(gptflu_data)	structure containing the data to module GPTFLU:.

II.1.5.1 Data input for module GPTFLU:

Data input for module GPTFLU: shows many similarities with data input for module GPTFLU:. The common options are defined only in [7], section FLUD:.

Tableau II.18: Structure `gptflu_data`

```
[ EDIT iprint ]
[ HIST imph ]
[ EXTE maxx0 ]
[ VAR1 icl1 icl2 ]
[ ADI nadi ]
[ PREC eps2 ]
[ { EXPLICIT | IMPLICIT } ]
FROM-TO i_src1 i_src2
;
```

where

EXPLICIT	key word used to obtain the solution of an direct fixed source eigenvalue problem.
IMPLICIT	key word used to obtain the solution of an adjoint fixed source eigenvalue problem. If neither 'EXPLICIT' nor 'IMPLICIT' are provided the default value will be chosen as a function of n_{var} and $n_{cst} + 1$.
FROM-TO	key word used to specify the numbers of the sources for which a generalized adjoint will be calculated.
i_{src1}	number of the first source.
i_{src2}	number of the last source.

For other keyword see module FLUD.

II.1.6 The GPTGRD: module

The GPTGRD: module is used to compute the gradient of functions using the generalized perturbation theory. To do that the user must precalculate the sources terms (module GPTSRC) and the generalized adjoints (module GPTFLU).

The GPTGRD: module also allows to define directly values of gradient of functions.

The calling specifications are:

Tableau II.19: Structure GPTGRD:

```
OPTIMIZE := GPTGRD: OPTIMIZE FLUXP [ SYS [ SYSP [ SYS2 [ SYS2P ] ] ]
TRACK ] MACRO [ FLUX ] :: [ (direct_data) ] (gptgrd_data)
OPTIMIZE := GPTGRD: OPTIMIZE :: (direct_data)
```

where

<i>OPTIMIZE</i>	character*12 name of the OPTIMIZE containing the optimization informations. <i>GPT</i> must appear on the RHS to be able to updated the previous values.
<i>FLUXP</i>	character*12 name of the FLUX containing the generalized adjoint flux, explicit or implicit.
<i>MACRO</i>	character*12 name of the MACROLIB linked list file containing fuel regions description and burnup informations.
<i>FLUX</i>	character*12 name of the FLUX containing the unperturbed flux, direct or adjoint. If it appears on RHS, is means it will be necessary for a function evaluation (objective or constraint).
<i>TABFL</i>	character*12 name of the TABLE linked list file containing fuel regions description and burnup informations. If it appears on RHS, is means it will be necessary for a function evaluation (objective or constraint).
<i>GPT</i>	character*12 name of the GPT linked list file containing fuel regions description and burnup informations. If it appears on RHS, is means it will be necessary for a function evaluation (objective or constraint).

Tableau II.21: Structure `gptgrd_data`

```
GPT [[ DIRECT { ivar1 ivar2 | ALL } (eval_data) ]]
... [[ INDIRECT [ { EXPLICIT | IMPLICIT } ] { ivar1 ivar2 | ALL } ] { FOBJ | CONSTRAINT
ifcn1 ifcn2 } ]]
;
```

where

DIRECT	key word used to specify that the direct part of the gradient will be calculated.
<i>i_{var1}</i>	first decision variable for which the gradient will be defined.
<i>i_{var2}</i>	last decision variable for which the gradient will be defined.
ALL	key word used to specify that the gradient will be calculated for all decision variables.
(eval_data)	see explanations in the module FOBJCT: key word 'EVAL-OBJ-CST'. Some predefined function are described too.
INDIRECT	key word used to specify that the indirect part of the gradient will be calculated.
EXPLICIT	key word used to obtain the solution of an direct fixed source eigenvalue problem.
IMPLICIT	key word used to obtain the solution of an adjoint fixed source eigenvalue problem. If neither 'EXPLICIT' nor 'IMPLICIT' are provided the default value will be chosen as a function of n_{var} and $n_{cst} + 1$.
FOBJ	key word used to specify that the gradient of the objective function will be defined.
CONSTRAINT	key word used to specify that the gradient of constraints will be defined.
<i>i_{fcn1}</i>	first constraint for which the gradient will be defined.
<i>i_{fcn2}</i>	last constraint for which the gradient will be defined.

II.1.7 The QLP: module

The QLP: module is used to solve the quasi-linear programming problems. The gradients of the functions are calculated with the modules GPTGRD: and PERTUR:. The options and parameters for the different method to solve the problem are defined in the QLPUTL: module.

The calling specifications are:

Tableau II.22: Structure QLP:

```
OPTIMIZE := QLP: OPTIMIZE :: (qlp_data)
```

where

OPTIMIZE **character*12** name of the OPTIMIZE containing the optimization informations. *OPTIMIZE* must appear on the RHS to be able to updated the previous values.

(qlp_data) structure containing the data to the module QLP:.

II.1.7.1 Data input for module PQL:

Tableau II.23: Structure qlp_data

```
[ EDIT iprint ]
[ WARNING-ONLY ]
CALCUL-DX [ NO-STORE-OLD ]
[ COST-EXTRAP >> ecost << ]
[ CONV-TEST >> lconv << ]
;
```

where

EDIT	key word used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module.
WARNING-ONLY	key word used to specify that only a warning will be used when no valid previous decision vectors can be recall in case of error of the mathematical programming.
CALCUL-DX	key word used to specify that the new step will be calculated.
NO-STORE-OLD	key word used to specify that the old value of decision variables and gradients will not be stored in the L_OPTIMIZE/'OLD-VALUE ' directory.
COST-EXTRAP	key word used to calculate the extrapolated cost <i>ecost</i> .
<i>ecost</i>	extrapolated cost.
CONV-TEST	key word used to calculate if the external convergence has been reached.
l_{conv}	=1 means that external convergence has been reached, =0 otherwise.

II.1.8 The TABU: module

The TABU: module is used to define options and data storage for the tabu search optimization algorithm.

The calling specifications are:

Tableau II.24: Structure TABU:

```
TABUSH [ OPTIM ] := TABU: [ TABUSH ] OPTIM :: (desctabu)
```

where

TABUSH **character*12** name of the extended TABU linked list file.

OPTIM **character*12** name of the extended OPTIMIZE linked list file. If *OPTIM* appears on the LHS, decision variables or their limits (for example) may be changed for further evaluation of the objective function and constraint.

(desctabu) structure containing the data to module TABU:.

II.1.8.1 Data input for module TABU:

Tableau II.25: Structure (desctabu)

```
[ EDIT iprint ]
[ DEFINITION (def_data) ]
[ NEIGHB-CREAT ]
[ NEIGHB-CHOIC [ INIT-PRO-LIST ] [ NELDER-MEAD ] ineig ]
[ NEIGHB-EVAL [ INIT-PRO-LIST ] [ NELDER-MEAD ] ineig ]
[ NEIGHB-BEST [ CONV-TEST >> Lconv << ] [ PROMISE-TEST [ NO-THRESHOLD ] >>
Lpro << ] ]
[ PROMISE-AREA [ NELDER-MEAD ] { CREATION | UPDATE } ]
```

continued on next page

Structure (**desctabu**)

continued from last page

```
[ NELDER-MEAD (nelder_data) ]
;
```

where

EDIT	key word used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module.
DEFINITION	key word used to define the tabu search optimization options.
(def_data)	structure containing the data to the option DEFINITION.
NEIGHB-CREAT	key word used to create the neighborhood for the decision variable set stored as the current one in the <i>TABUSH</i> object.
NEIGHB-CHOIC	key word used to specify the number i_{neig} within the neighbors which will be evaluated. The corresponding decision variable values are copied in the <i>OPTIM</i> object as the current decision variables.
NEIGHB-EVAL	key word used to specify the number i_{neig} within the neighbors which have been evaluated. The corresponding functional values and the tabu function result are stored in the <i>TABU</i> object.
INIT-PRO-LIST	key word used to specify the initial elements of the promising list are selected and evaluated (and not the neighbors).
NELDER-MEAD	key word used to specify the initial elements of the polytope for the Nelder-Mead simplex algorithm are selected and evaluated (and not the neighbors).
i_{neig}	integer value for a neighbor point to be / which has been evaluated.
NEIGHB-BEST	key word used to check the neighbors results. The best neighbor result is compared to the fittest solution ever found. An update is performed if necessary. Tests for global convergence and promising area detection can be done. The tabu list is updated.
CONV-TEST	key word used to verify if global convergence is achieved.
<i>Lconv</i>	logical value for the global convergence. <i>Lconv</i> equals <i>.true.</i> if <i>Nit</i> is greater than Nit_{max} .
PROMISE-TEST	key word used to verify if a promising area has been detected.

NO-THRESHOLD	key word used to specify that no threshold limits the acceptance of promising areas.
L_{conv}	logical value for the promising area detection.
PROMISE-AREA	key word used to specify that calculation based on gradient methods will be performed on a promising area previously detected.
NELDER-MEAD	key word used to specify the Nelder-Mead simplex algorithm is used instead of the gradient method.
CREATION	key word used to define the area for the local gradient method optimization algorithm. A backup of original decision variable limits is done in <i>TABUSH</i> object and new smaller ones are stored in <i>OPTIM</i> object.
UPDATE	key word used to set the gradient method result for the promising area as the new current decision variable. An update of the best point ever found is done is necessary. The promising list is also updated.
NELDER-MEAD	key word used to specify the Nelder-Mead simplex algorithm is selected.
(def_data)	structure containing the data to the option NELDER-MEAD corresponding to the different geometric transformations.

Tableau II.26: Structure **(def_data)**

```

[ NEIGHBOR-NB  $ngh$  ]
[ NEIGHBOR-TYP { RECTANGLE | BALL } ]
[ NEIGHBOR-DIS { GEOMETRIC  $fact$  | LINEAR | ISOVOLUME } ]
[ NEIGHBOR-RAD  $R_n$  ]
[ TABU-RAD  $R_t$  ]
[ PROMIS-RAD  $R_p$  ]
[ NIT-MAX-CONV  $Nit_{max}$  ]
[ TABU-LIST-LG { ALL [ $Lg_t$ ] } ]
[ PROM-LIST-LG { ALL [ $Lg_p$ ] } ]
[ GET-CURRENT [ COMPLETE ] ]
[ PUT-CURRENT ]
[ INITIALIZE ]
[ INIT-PRO-LIST ]
[ RESET-BEST ]

```

continued on next page

Structure (**def_data**)

continued from last page

[BEST-AS-CURR]
 [NELDER-EPS ϵ_{ned}]

where

- NEIGHBOR-NB key word used to define the number of neighbors ngh .
- ngh integer value for the number of neighbors (default 5).
- NEIGHBOR-TYP key word used to specify the type of the neighborhood.
- RECTANGLE key word used to specify that the neighborhood will be hyperrectangle crowns (default).
- BALL key word used to specify that the neighborhood will be hypersphere crowns.
- NEIGHBOR-DIS key word used to specify that the type of discretisation within the neighborhood.
- GEOMETRIC key word used to specify that the radius of the crowns are given by a geometric serie. The radius are given by:

$$r_i = R_n \frac{1}{fact^{ngh-i}} \text{ with } i \in \{1, ngh\}$$

- $fact$ real number (> 1) for the geometric serie for the radius determination.
- LINEAR key word used to specify that the radius of the crowns are given by a linear serie (default). The radius are given by:

$$r_i = R_n \frac{i}{ngh} \text{ with } i \in \{1, ngh\}$$

- ISOVOLUME key word used to specify that the radius of the crowns are chosen to have a constant volume for all crowns. The radius are given by:

$$r_i = R_n \sqrt[ngh]{\frac{i}{ngh}} \text{ with } i \in \{1, ngh\}$$

- NEIGHBOR-RAD key word used to set the radius R_n of the neighborhood.

R_n	real number for neighborhood radius. This radius is a fraction of the total limits and then must be between 0.0 and 1.0.
TABU-RAD	key word used to set the radius R_t of the hyperrectangle / ball around tabu values. All the points within this small domain are tabu as well.
R_t	real number for tabu list radius. This radius is a fraction of the total limits and then must be between 0.0 and 1.0.
PROMIS-RAD	key word used to set the radius R_p of the hyperrectangle / ball around tabu values. All the points within this small domain are tabu as well.
R_p	real number for promising list radius. This radius is a fraction of the total limits and then must be between 0.0 and 1.0.
NIT-MAX-CONV	key word used to specify the number Nit_{max} of required external iteration without improvement of the best solution ever found for global convergence achievement.
Nit_{max}	integer value of required iterations for global convergence.
TABU-LIST-LG	key word used to specify the maximum length of the tabu list.
PROM-LIST-LG	key word used to specify the maximum length of the promising area list.
ALL	key word used to specify the values entering in a list are kept until the end of the optimization procedure.
Lg_t	integer value of maximum tabu list length.
Lg_p	integer value of maximum promising area list length.
GET-CURRENT	key word used to specify the decision variable set in <i>OPTIM</i> object will be stored as the current one in <i>TABUSH</i> object.
COMPLETE	key word used to specify the objective function, the constraints and the penalty functions will also be stored with the current values.
PUT-CURRENT	key word used to specify the current decision variable set in <i>TABU</i> object will be stored as the variable values in <i>OPTIM</i> object.
INITIALIZE	key word used to initialize (chose a random value) the starting decision variables in the decision space.
INIT-PRO-LIST	key word used to specify that the initial promising list is created.
RESET-BEST	key word used to reset the best value of the tabu search objective function.
BEST-AS-CURR	key word used to set the best value as the current value.

NELDER-EPS	key word used to set the convergence limit for the Nelder-Mead simplex algorithm.
ϵ_{ned}	real value for the Nelder-Mead convergence criterium (default 0.01).

Tableau II.27: Structure (**nelder_data**)

```
[ FIND-NEW >> $i_{worst}$ << ]
[ COMPARE-NEW >> $L_{exp}$ << >> $L_{ctt}$ << >> $L_{conv}$ << ]
[ EXPAN-VLD >> $L_{conv}$ << ]
[ CONTRACTION >> $L_{mctt}$ << >> $i_{best}$ << >> $L_{conv}$ << ]
```

where

FIND-NEW	key word used to find the worst point of the polytope for the Nelder-Mead simplex algorithm, and compute its reflected point.
i_{worst}	integer value for the index of the worst point of the polytope.
COMPARE-NEW	key word used to compare the reflected point of the worst point of the polytope with the other points. Next geometrical transformation is decided according to the results of the comparison.
L_{exp}	logical value for the expansion move.
L_{ctt}	logical value for the contraction move.
L_{conv}	logical value for convergence.
EXPAN-VLD	key word used to validate the expansion point (comparison of its results with the reflection point).
CONTRACTION	key word used to compare the contraction point (comparison of its results with the reflection point). Next geometrical transformation is decided according to the results of the comparison.
L_{ctt}	logical value for the multi-contraction move.
i_{best}	integer value for the index of the best point of the polytope.

II.2 Other modules

In this section, input of output data treatment modules are explained. Modified modules are also described.

II.2.1 The ADDOBJ: module

The ADDOBJ: module is used to perform the differences between two objects or to add two objects. For the MACROLIB and FLUX, this is possible only if they contain the same energy group and material mixture numbers.

The calling specifications are:

Tableau II.28: Structure ADDOBJ:

```
MACNEW := ADDOBJ: [ MACNEW ] MACRO1 MACRO2 :: (addmac_data)
FLUNEW := ADDOBJ: [ FLUNEW ] FLU1 FLU2 :: (addflu_data)
```

where

<i>MACNEW</i>	character*12 name of the MACROLIB containing either the nuclear increments, from the calculation of <i>MACRO1</i> - <i>MACRO2</i> or the sum of properties from <i>MACRO1</i> + <i>MACRO2</i> . Be aware the order of MACROLIB is important, even for addition option.
<i>MACRO1</i>	character*12 name of a MACROLIB.
<i>MACRO2</i>	character*12 name of a MACROLIB. When addition is performed, it must contain incremental nuclear properties.
(addmac)	structure containing the data to module ADDOBJ: with the options for MACROLIB operations.
<i>FLUNEW</i>	character*12 name of the FLUX which will be the result of the addition or the subtraction of the two old one. This object has to be in create mode only.
<i>FLU1</i>	character*12 name of the first FLUX.
<i>FLU2</i>	character*12 name of the second FLUX.

(addflu) structure containing the data to module ADDOBJ: with the options for FLUX addition.

II.2.1.1 Data input for module ADDOBJ:

Tableau II.29: Structure (**addmac_data**)

```
[ EDIT iprt ]
[ STEP ilev ]
[ { ADD [ frac1 [ frac2 ] ] | SUB } ]
[ FROM-MP ]
[ { DDIF | NODIF } ]
;
```

where

EDIT	key word used to set <i>iprt</i> .
<i>iprt</i>	index used to control the printing. ≤ 2 minimum printing; > 3 macroscopic differences are printed.
ADD	keyword to specify that the two objects will be added. If only <i>frac1</i> is specified, $MACRO1 + frac1.MACRO2$ will be performed using the options for the diffusion coefficients. If <i>frac1</i> and <i>frac2</i> are specified, $frac1.MACRO1 + frac2.MACRO2$ will be performed even for the diffusion coefficients.
<i>frac1</i>	first real that will multiply the value of the first object. (default 1.0)
<i>frac2</i>	second real that will multiply the value of the second object. (default 1.0)
SUB	keyword to specify that the two objects will be subtracted. This is the default option.
FROM-MP	keyword to specify that the two objects have been partially calculated by different CPU. An addition will be performed to calculate the 'complete' object (<i>frac1</i> and <i>frac2</i> equal 1.0).

STEP	key word used to set <i>ilev</i> .
<i>ilev</i>	number of the perturbed level in MACROLIB. In case of subtraction of two MACROLIB: If a single set of increments is stored, it must be equal to 1. This step is used to later compute perturbation system matrices. If this information is absent, incremental cross sections are stored on root directory. In case of addition of two MACROLIB: <i>ilev</i> specifies the perturbed level where informations is stored in the two initial MACROLIB. If both MACROLIB have a perturbed level, it must be the same. The resulting properties will be stored on root directory.
DDIFF	keyword to specify a correct treatment of diffusion coefficients. If SUB is specified, the resulting incremental diffusion coefficient will be:

$$\Delta D = \frac{1}{\frac{1}{D_1} - \frac{1}{D_2}}$$

where D_1 is taken from the first MACROLIB and D_2 from the second. If ADD is specified, the resulting diffusion coefficient will be:

$$D = \frac{1}{\frac{1}{D_1} + \frac{1}{\Delta D_2}}$$

where D_1 is taken from the first MACROLIB and ΔD_2 from the second.

NODIF	keyword to specify that no addition or subtraction of diffusion coefficients will be done. This is the default option.
-------	--

Tableau II.30: Structure (**addflu_data**)

```
[ EDIT iprt ]
{
[ { ADD [ frac1 frac2 ] | SUB } ] [ F-ADJOINT [ ADJ-ONLY ] ]
|
FROM-2-PARTS { FLUX | AFLUX |DFLUX iga1 |ADFLUX iga1 } frac1
..... { FLUX | AFLUX |DFLUX iga2 |ADFLUX iga2 } frac2
..... IN { FLUX | AFLUX |DFLUX iga2 |ADFLUX iga2 }
};
```

where

EDIT	key word used to set <i>iprt</i> .
<i>iprt</i>	index used to control the printing. => 1 structure of the resulting object is printed;
ADD	keyword to specify that the two objects will be added. This is the default option.
<i>frac1</i>	first real that will multiply the value of the first object. (default 1.0)
<i>frac2</i>	second real that will multiply the value of the second object. (default 1.0)
SUB	keyword to specify that the two objects will be subtracted.
F-ADJOINT	keyword to specify that the same numerical operation will performed on flux and adjoint flux.
ADJ-ONLY	keyword to specify that the numerical operation will be performed on the adjoint flux only.
FROM-2-PARTS	keyword to specify that the two objects contain one part of the result flux. An addition will be performed to calculate the 'complete' object. This option can not be done with other options (except EDIT) in the same call of the module.
FLUX	keyword to specify that the flux will be used or the result of the addition.
AFLUX	keyword to specify that the adjoint flux will be used or the result of the addition.
DFLUX	keyword to specify that the explicit generalized adjoint will be used or the result of the addition.
ADFLUX	keyword to specify that the implicit generalized adjoint will be used or the result of the addition.
<i>iga1</i>	number of the first generalized adjoint if applicable.
<i>iga2</i>	number of the second generalized adjoint if applicable.
IN	keyword to specify that the type of flux for the result of the addition.
<i>iga3</i>	number of the third generalized adjoint if applicable.

II.2.2 The XSFUEL: module

The XSFUEL: module is used to create or modify an extended MACROLIB containing set of interpolated nuclear properties from a TABLE linked list or XSM file. The construction of an extended MACROLIB for fuel properties is made directly from TABLE information with respect to an evolution parameter (burnup or neutron exposure) contained in the fuel map object. The resulting number of mixtures in MACROLIB is the total number of fuel bundles. The TABLE information are obtained by the XSCONS: or AFM: module.

The calling specifications are:

Tableau II.31: Structure XSFUEL:

MACFL := XSFUEL: [*MACFL*] *TABFL* *MAPFL* :: (**descxsfuel**)

where

<i>MACFL</i>	character*12 name of the extended MACROLIB. If <i>MACFL</i> appears on the RHS, the information previously stored in <i>MACFL</i> is kept.
<i>TABFL</i>	character*12 name of the TABLE linked list or XSM file containing burnup dependent fuel properties.
<i>MAPFL</i>	character*12 name of the MAP linked list or XSM file containing fuel regions description and burnup informations.
(descxsfuel)	structure containing the data input to module XSFUEL: for MACROLIB construction from TABLE files.

II.2.2.1 Data input for module XSFUEL:

Tableau II.32: Structure (**descxsfuel**)

```
[ EDIT iprint ]
[[ TABLE TABFL (descinfo) ]]
;
```

where

EDIT	key word used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module XSFUEL:. =0 for no print; =1 for minimum printing (default value); larger values produce increasing amounts of output.
TABLE	key word used to select a TABLE and to set the interpolation information.
<i>TABFL</i>	character*12 name of the selected fuel TABLE.
(descinfo)	structure containing the interpolation data for TABLE file.

Tableau II.33: Structure (**descinfo**)

```
[[ MIX matnum HTYPE [ DERIV ] [ DERIV-ENR ]
{ BURNUP | N/KB } { HOMG | TAVG | TAVGC | DIRECTC | DIRECT }
[ DERIV-PAR { AVG-EX-BURN | ENRICH } ivar ] ] ENDMIX
```

where

MIX	key word used to set <i>matnum</i> .
<i>matnum</i>	identifier for the material mixture to be created. The maximum number of identifiers permitted is <i>nmixt</i> and the maximum value that <i>matnum</i> may have is <i>nmixt</i> . For TABLE interpolation, it identifies the fuel type as defined in the fuel map MAP.

<i>HTYPE</i>	name of the material mixture. Each name refers to a type of nuclear data that is stored on a directory in the <i>TABLE</i> linked list or <i>XSM</i> file.
<i>DERIV</i>	key word used to compute the derivative of the <i>MACROLIB</i> information with respect to burnup. By default, the <i>MACROLIB</i> information is not differentiated.
<i>DERIV-ENR</i>	key word used to compute the derivative of the <i>MACROLIB</i> information with respect to enrichment. By default, the <i>MACROLIB</i> informations are not enrichment dependent, unless a specific table has been created in the <i>XSCONS:</i> module.
<i>BURNUP</i>	key word used to set the burnup as the interpolation parameter in <i>TABLE</i> file.
<i>N/KB</i>	key word used to set the neutron exposure as the interpolation parameter in <i>TABLE</i> file.
<i>HOMG</i>	key word used to specify an homogeneous interpolation of fuel properties according to burnup values stored in <i>MAPFL</i> . The method should be compatible with the one set in <i>MAP</i> data structure in a call to <i>INIRES:</i> module.
<i>TAVG</i>	key word used to specify a time-average interpolation per combustion zone of fuel properties according to burnup values stored in <i>MAPFL</i> . The method should be compatible with the one set in <i>MAP</i> data structure in a call to <i>INIRES:</i> module.
<i>TAVGC</i>	key word used to specify a time-average interpolation of fuel properties according to burnup values stored in <i>MAPFL</i> . The method should be compatible with the one set in <i>MAP</i> data structure in a call to <i>INIRES:</i> module.
<i>DIRECT</i>	key word used to specify a single interpolation per combustion zone of fuel properties according to burnup values stored in <i>MAPFL</i> . The method should be compatible with the one set in <i>MAP</i> data structure in a call to <i>INIRES:</i> module.
<i>DIRECTC</i>	key word used to specify a single interpolation of fuel properties according to burnup values stored in <i>MAPFL</i> . The method should be compatible with the one set in <i>MAP</i> data structure in a call to <i>INIRES:</i> module.
<i>DERIV-PAR</i>	key word used to compute the partial derivative of the <i>MACROLIB</i> information with respect to average exit burnup, enrichment for the decision variable <i>ivar</i> in an optimisation calculation. By default, the <i>MACROLIB</i> information is not differentiated.

AVG-EX-BURN	key word used to specify that the type of the decision variable is the average exit burnup.
ENRICH	key word used to specify that the type of the decision variable is enrichment .
<i>ivar</i>	identifier for the number of the decision variable for an optimisation problem.
ENDMIX	end of specification key word for the material mixture.

II.2.3 The `MATLAB:` module

The `MATLAB:` module is used to create an ASCII file executable by MATLAB. Two options are available. First one is used to create a file to draw the gradients of functions calculated in a optimisation problem. Second option allows to draw maps of the flux distribution.

The calling specifications are:

Tableau II.34: Structure `MATLAB:`

```
{ ASCII-MAT := MATLAB: OPTIM :: (descmatlgrd) |
  ASCII-MAT := MATLAB: FLUX TRACK INDEX GEOM :: (descmatlfu) }
```

where

<i>ASCII-MAT</i>	<code>character*12</code> name of the ASCII file executable by MATLAB.
<i>OPTIM</i>	<code>character*12</code> name of the OPTIMISATION linked list or XSM file containing the gradients and the stored perturbed values of the functions. Such file is obtained using the module <code>GPTVRF:</code> .
(descmatlgrd)	structure containing the data input to module <code>MATLAB:</code> for gradients plotting.
<i>FLUX</i>	<code>character*12</code> name of the FLUX linked list or XSM file containing the flux or adjoint or generalized adjoints or harmonics to be mapped.
<i>TRACK</i>	<code>character*12</code> name of the TRACK linked list or XSM file containing the tracking datas (TRIVAA is the only type of tracking compatible).
<i>INDEX</i>	<code>character*12</code> name of the INDEX linked list or XSM file containing the index datas.
<i>GEOM</i>	<code>character*12</code> name of the GEOM linked list or XSM file containing the geometry description.
(descmatlfu)	structure containing the data input to module <code>MATLAB:</code> for flux distribution's mapping.

II.2.3.1 Data input for module MATLAB:

Tableau II.35: Structure (**descmatlgrd**)

```
[ EDIT iprint ]
OPT-GRAD-VRF
;
```

where

EDIT key word used to set *iprint*.

iprint index used to control the printing in module XSFUEL:. =0 for no print (default value); =1 for minimum printing; larger values produce increasing amounts of output.

OPT-GRAD-VRF key word used to select the gradients verification and plotting option and produce the *ASCII-MAT* file. An exemple of the results is presented on the following figure.

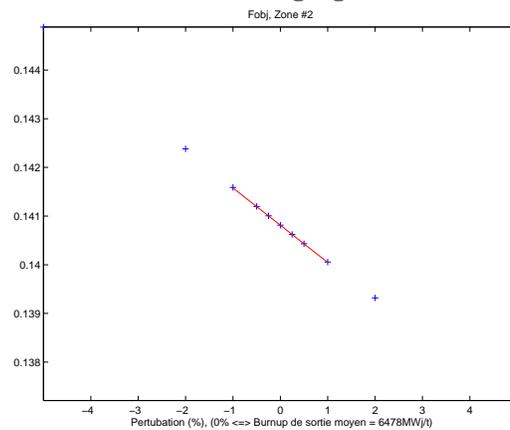


Tableau II.36: Structure (`descmatlfu`)

```
[ EDIT iprint ]
MAP-FLUX {FLUX | ADJOINT | GPT-FLU isrc | GPT-ADJOINT isrc | HARMONIC ihrm }
{ NO-GRID | CENTER-GRID | CHANEL-GRID }
;
```

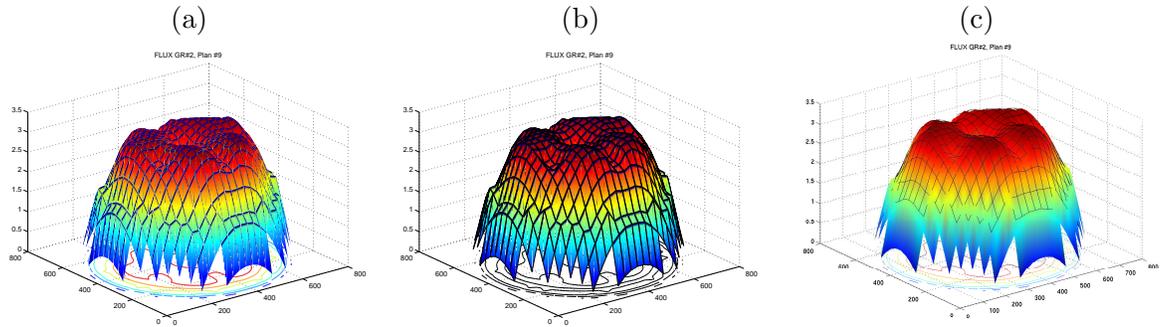
where

EDIT	key word used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module XSFUEL:. =0 for no print (default value); =1 for minimum printing; larger values produce increasing amounts of output.
MAP-FLUX	key word used to select the flux distribution mapping option and produce the <i>ASCII-MAT</i> file.
FLUX	key word used to specify that the distribution of the flux will be drawn.
ADJOINT	key word used to specify that the distribution of the adjoint will be drawn.
GPT-FLU	key word used to specify that the distribution of the explicit generalized adjoint corresponding to the source number <i>isrc</i> will be drawn.
GPT-ADJOINT	key word used to specify that the distribution of the implicit generalized adjoint corresponding to the source number <i>isrc</i> will be drawn.
<i>isrc</i>	identifier for the source number corresponding to generalized adjoint to be drawn.
HARMONIC	key word used to specify that the distribution of the flux corresponding to the harmonic number <i>ihrm</i> will be drawn.
<i>ihrm</i>	identifier for the harmonic number corresponding to the flux to be drawn.
NO-GRID	key word used to specify that no grid will be add on the map of the flux distribution. An exemple of the results is presented on the figure a.
CENTER-GRID	key word used to specify that a grid will be add on the map of the flux distribution. The nodes of the grid correspond to the center of

the volumes where the flux are calculated. An example of the results is presented on the figure b.

CHANEL-GRID

key word used to specify that a grid will be add on the map of the flux distribution. The squares of the grid correspond to the limits of the channels. An example of the results is presented on the figure c.



II.2.4 The GPTVRF: module

The GPTVRF: module is used to verify the computation of the gradients with classical and generalized perturbation theories. The analytical gradients calculated with the modules PERTUR: and GPTGRD: are compared with numerical gradients calculated with the Ceshino method using several values of the functionals for pertubated values of the decision variables.

The calling specifications are:

Tableau II.37: Structure GPTVRF:

$OPTIMV := GPTVRF: [OPTIMV] OPTIM :: (gptvrf_data)$

where

OPTIMV **character*12** name of the OPTIMIZE containing the pertubated and unpertubated values of the decision variables and of the functionals. If *OPTIMV* is the same as *OPTIM*, then the pertubated datas will be stored in the same OPTIMIZE object.

OPTIM **character*12** name of the OPTIMIZE containing the curent values of the decision variables and of the functionals.

(gptvrf_data) structure containing the data to module GPTVRF:.

II.2.4.1 Data input for module GPTVRF:

Tableau II.38: Structure gptvrf_data

[EDIT *iprint*]
[SAVE-PERT *i_{var}* ϵ]

continued on next page

Structure `gptvrf_data`

continued from last page

```
[ SAVE-GPT { ABS | REL  $\epsilon_{GPT}$  } ]
[ COMPARE {  $i_{var1}$   $i_{var2}$  | ALL } {  $i_{fct1}$   $i_{fct2}$  | ALL } ]
;
```

where

EDIT	key word used to set <i>iprint</i> .
<i>iprint</i>	index used to control the printing in module.
SAVE-PERT	key word used to store the current values of the functionals for the corresponding perturbation ϵ of the decision variable i_{var} .
i_{var}	number of the perturbed decision variable.
ϵ	amount of the perturbation (ex: $\epsilon=0.01$ corresponds to 1% of perturbation).
SAVE-GPT	key word used to specify that the analytical gradient will be stored.
ABS	key word used to specify that the stored analytical gradient correspond to a absolute perturbation or that the derivative are already calculated.
REL	key word used to specify that the stored analytical gradient will be divided by the corresponding amount of perturbation to obtain the relative gradient (the derivative).
ϵ_{GPT}	amount of the perturbation (ex: $\epsilon_{GPT}=0.01$ corresponds to 1% of perturbation).
COMPARE	key word used to specify that the numerical gradient will be computed and compared to the analytical gradients.
i_{var1}	number of the first decision variable for which the comparason will be done.
i_{var2}	number of the last decision variable for which the comparason will be done.
i_{fct1}	number of the first functional for which the comparason will be done.
i_{fct2}	number of the last functional for which the comparason will be done.

ALL key word used to specify that the comparason will be done for all decision variables and/or all functionals.

II.2.5 The XSCONS: module

The XSCONS: module is used to create a TABLE object containing burnup-dependent fuel nuclear properties from COMPO linked lists or XSM files. The calling specifications are:

Tableau II.39: Structure XSCONS:

```
TABFL := XSCONS: [[ CPO ]] :: (descxsc) ;
```

where

TABFL **character*12** name of the TABLE linked list or XSM file that will contain the fuel properties.

CPO **character*12** name of the COMPO linked list or XSM file.

(descxsc) structure containing the data to module XSCONS:.

II.2.5.1 Data input for module XSCONS:

Tableau II.40: Structure (descxsc)

```
[ EDIT iprint ]
[ NENRICH nε ]
READ { ENR iε εval | } {COMPO CPO (descint) }
[ D-ENRICH NAMTAB ]
;
```

where

EDIT key word used to set *iprint*.

<i>iprint</i>	index used to control the printing in module CRE :. =0 for no print; =1 for minimum printing (default value); Larger values produce increasing amounts of output.
NENRICH	key word used to specify that fuel nuclear properties are dependent of the burnup and the enrichment. It is also used to set n_ε .
n_ε	number of different enrichment values used to create burnup and enrichment dependent tables.
READ	key word used to read the TABLE specifications (burnup, neutron exposure, number densities) from the input data file.
ENR	key word used to set n_ε and ε_{val} . This keyword is required only if the number of enrichment values has been specified with the NENRICH keyword. Otherwise, using this keyword leads to an error of the program.
i_ε	index of the enrichment value. Index values must be between 1 and n_ε . They also must correspond to increasing enrichment values.
ε_{val}	enrichment value.
COMPO	key word used to select a COMPO and to set the interpolation information.
CPO	character*12 name of the selected COMPO linked list or XSM file.
(descint)	structure containing the interpolation and access data.
D-ENRICH	key word used to specify that the derivative of fuel nuclear properties with respect to the enrichment will be computed. It is also used to set NAMTAB .
NAMTAB	name of the common directories in the TABLE where the fuel properties are stored. The same name should have been provided earlier when tables have been created(cf. description of NAMTAB in the (descint) structure).

The **(descint)** structure is defined as

Tableau II.41: Structure **(descint)**

[[**TYPE** **NAMTAB** **HTYPE**

continued on next page

Structure (**descint**)

continued from last page

```
[ MICRO { [[ HISO { conc | * } ] ] | ALL } ]
[ UPS ] { BURNUP | N/KB } [ INTRPL delta ] ENDTYP ]]
```

where

TYPE	key word used to set <i>NAMTAB</i> .
<i>NAMTAB</i>	name of the directory in the TABLE where fuel properties will be stored. In general this name depend on the COMPO file specified. If fuel properties are enrichment dependent, a new <i>NAMTAB</i> name will be built from the concatenation <i>NAMTAB//IENR</i> where only the first 9 characters of <i>NAMTAB</i> are used and a three digit suffix with value ' 1', ' 2', ' 3', etc., indicating the enrichment value index.
<i>HTYPE</i>	name of the material mixture. Each name refers to a type of nuclear data that is stored on a directory in the COMPO linked list or XSM file. <i>HTYPE</i> is a character*12 name built from the concatenation <i>HCOMPO//HIORD</i> where <i>HCOMPO</i> is an ASCII name with a maximum of 8 characters and <i>HIORD</i> is a four digit suffix with value ' 1', ' 2', ' 3', etc., indicating the material mixture index.
MICRO	key word used to set the number densities of the extracted isotopes present in the COMPO linked list or XSM file. By default, the extracted isotopes are not added to the resulting TABLE.
<i>HISO</i>	character*12 name of an extracted isotope.
<i>conc</i>	user-defined value of the number density (in 10^{24} particles per cm^3) of the extracted isotope.
*	the value of the number density for isotope <i>HISO</i> is recovered from the COMPO.
ALL	all the values for the number densities are recovered from the COMPO.
UPS	key word used to compute properties with no up-scattering contribution.
BURNUP	key word used to recover burnup information and to set it as an interpolation parameter. This is the default option.

N/KB	key word used to recover neutron exposure or irradiation information and to set it as an interpolation parameter. This option can be useful to compare results with other diffusion codes where neutron exposure is the only dependent parameter for fuel properties.
INTRPL	keyword used to order a linear interpolation of fuel properties with respect to burnup.
<i>delta</i>	fuel depletion variable increment used in the property interpolation. If fuel burnup is the interpolation parameter, this value will be in MWd/t ; If fuel irradiation is the interpolation parameter, this value will be in neutron/kb.
ENDTYP	end of specification key word for the fuel type.

II.3 Data structure description

II.3.1 Contents of a /optimize/ data structure

The /optimize/ specification is used to store the optimization variables and functions values and definitions, limits and options. It is also used in a particular case with the module **GPTVRF**: to store the functions for many perturbed values of the decision variables and the gradients calculated numerically and analytically.

In any case, the signature variable for this data structure must be `SIGNA=L_OPTIMIZE_LLL`. The dimensioning parameters for this data structure, which are stored in the state vector \mathcal{S}_i^o , represents:

- The number of decision variables $N_{var} = \mathcal{S}_1^o$.
- The number of constraints $N_{cst} = \mathcal{S}_2^o$.
- The type of optimization. \mathcal{S}_3^o where

$$\mathcal{S}_3^o = \begin{cases} 1 & \text{minimization} \\ -1 & \text{maximization} \end{cases}$$

- The test for external convergence \mathcal{S}_4^o . where

$$\mathcal{S}_4^o = \begin{cases} 0 & \text{not converged} \\ 1 & \text{converged} \end{cases}$$

- The number of external iterations \mathcal{S}_5^o .
- The type of reduction for external step \mathcal{S}_6^o . where

$$\mathcal{S}_6^o = \begin{cases} 1 & \text{half} \\ 2 & \text{parabolic} \end{cases}$$

- The number of inner iterations \mathcal{S}_7^o .
- The number of outer iterations \mathcal{S}_8^o .
- The resolution's method for the linearized problem \mathcal{S}_9^o . where

$$\mathcal{S}_9^o = \begin{cases} 1 & \text{SIMPLEX/LEMKE} \\ 2 & \text{LEMKE/LEMKE} \\ 3 & \text{MAP} \\ 4 & \text{Augmented Lagragian} \\ 5 & \text{Penalty Method} \end{cases}$$

- The type of perturbation for the decision variables \mathcal{S}_{10}^o . where

$$\mathcal{S}_{10}^o = \begin{cases} 1 & \text{epsilon} \\ 2 & \text{previous} \end{cases}$$

- The type of fuel cost definition \mathcal{S}_{11}^o . where

$$\mathcal{S}_{11}^o = \begin{cases} 1 & \text{dependent of the enrichment} \\ 2 & \text{fixed} \end{cases}$$

- The test for a realistic decision vector \mathcal{S}_{12}^o . where

$$\mathcal{S}_{12}^o = \begin{cases} 0 & \text{do not respect all constraints} \\ 1 & \text{do respect all constraints} \end{cases}$$

- A flag for unsuccessful resolution in QLP \mathcal{S}_{13}^o . where

$$\mathcal{S}_{13}^o = \begin{cases} 0 & \text{successful at last iteration} \\ 1 & \text{number of iteration with unsuccessful resolution} \end{cases}$$

- The number of regions $N_r = \mathcal{S}_{16}^o$.
- The number of channels in the core $N_{ch} = \mathcal{S}_{17}^o$.
- The number of bundles per channel $N_k = \mathcal{S}_{18}^o$.
- The number of unknowns per energy group $N_u = \mathcal{S}_{19}^o$.
- The number of energy groups $G = \mathcal{S}_{20}^o$.

Tableau II.42: Main records and sub-directories in /optimize/

Name	Type	Condition	Units	Comment
SIGNATURE _{UUU}	C*12			Signature of the data structure (SIGNA)
STATE-VECTOR	I(40)			Vector describing the various parameters associated with this data structure \mathcal{S}_i^o
VAR-VALUE _{UUU}	R(N_{var})			The values of the decision variables
VAR-TYPE _{UUUU}	R(N_{var})			The type of the decision variables Var_{type} .
{var-zone}	I($N_{z,v}$)			The definition of the zone where i -th decision variable have an influence on the material properties.
VAR-MAX-VAL _U	R(N_{var})			The maximum values of the decision variables can be.
VAR-MIN-VAL _U	R(N_{var})			The minimum values of the decision variables can be.
VAR-WEIGHT _{UU}	R(N_{var})			The weight of the decision variables w_i in the quadratic constraint.
CST-VOL-TYPE	I(N_{cst})			Record containing the type for the zone where the constraints apply.
CST-OBJ _{UUUUU}	R(N_{cst})			The limit value of the constraints. The units depends with the type of the constraint type.
CST-TYPE _{UUUU}	I(N_{cst})			The type of the constraints: =-1 for \geq ; =0 for =; =1 for \leq .
{cst-zone}	I($N_{z,c}$)			The definition of the zone where the constraint j apply.
CST-WEIGHT _{UU}	R(N_{cst})			The weight of the constraint η_j and γ_j for the duals and meta-heuristic methods.
FOBJ-CST-VAL	R($N_{cst} + 1$)			The actual values of the objective function (first value) and the constraints (the following values). The number of the constraints are assigned in the order they have been defined.

continued on next page

continued from last page

Name	Type	Condition	Units	Comment
OPT-PARAM-R _U	R(40)			The different limits and values for the iterative calculations of the optimization problem.
FUNC-ZON-VOL	R($N_{cst} + 1$)			The first value is the volume where the objective function applies and the following ones correspond to the volumes where the constraints apply.
BKP-PERT-XS _U	I(13)			The indexes of the perturbed material properties to backup.
FUEL-COST _{UUU}	R(dependant)			The fixed fuel cost of each enrichment zone if $S_{11}^o = -1.0$, or the parameters to calculate the enrichment dependant cost if $S_{11}^o = -2.0$.
FUEL-C-DIST _U	R(N_u)		$\$/kg$	The fuel cost distribution corresponding to each point of the neutrons flux unknowns.
BURN-C-DIST _U	R(N_u)		MWd/t	The average exit burnup distribution corresponding to each point of the neutrons flux unknowns.
D-LAMBDA _{UUUU}	R(N_{var})			The gradients or the absolute variation depending of the keyword used in the PERTUR: module of the eigenvalue with the decision variables.
D-LAMBDA-V _{UU}	R(N_{var})			The gradients or the absolute variation depending of the keyword used in the PERTUR: module of the eigenvalue for the coolant voided reactor with the decision variables.
GRADIENT _{UUUU}	R($N_{var}, N_{cst} + 1$)			The gradients of the objective function and the constraints. The gradients of the objective for all the decision variables are in first position, then follow the gradients of the constraints.

continued on next page

continued from last page

Name	Type	Condition	Units	Comment
OLD-VALUE _{UUU}	Dir			Directory containing different informations of the previous iterations. the values of the decision variables, the objective function, the constraints and the gradients of these functions for the previous external iteration. This repertory will be created by the module QLP: unless it is specified to not do.
BKP-VALUE _{UUU}	Dir			Directory containing a backup of the values of the decision variables, the objective function, the constraints and the gradients of these functions for this external iteration. This repertory will be created if the interpolation of the objective function in the middle point is calculated.
{/stepdir/}	Dir			Directory containing the perturbed properties and the $A_p\phi$ and $B_p\phi$ vectors.

Notes related with the decision variables definition:

The type of the decision variable are given by Var_{type} which is defined by:

$$Var_{type} = i_{type} * 1000 + i_{zone}$$

where

$$i_{type} = \begin{cases} 1 & \text{average exit burnup} \\ 2 & \text{enrichment} \\ 3 & \text{number of bundles shift} \\ 4 & \text{device thickness} \end{cases}$$

i_{zone} for the number of the corresponding zone in the L-MAP for the decision variable.

The zones definition information records {var-zone} will be composed using the following FORTRAN instructions

```
WRITE(var - zone, '(A8, I4)') 'VAR - ZONE', i
```

for $1 \leq i \leq N_{var}$. The size $N_{z,v}$ of the records depends of the type of the decision variable. For average exit burnup, enrichment and number of bundles shift, $N_{z,v} = N_{ch}$. For device thickness, $N_{z,v} = N_{ch} * N_k$.

Notes related with the constraints definition:

The type of each constraint zones is defined by the following index :

$$j_{type} = \begin{cases} 1 & \text{one value for the whole reactor (ex: } k_{eff}\text{)} \\ 2 & \text{one value for all channels (ex: total power)} \\ 3 & \text{one value for one channel (ex: channel power)} \\ 4 & \text{one value for one bundle (ex: bundle power)} \\ 5 & \text{one value for one zone defined by many bundles} \\ 6 & \text{void reactivity} \\ 7 & \text{analytic function} \end{cases}$$

The zones definition information records {cst-zone} will be composed using the following FORTRAN instructions

WRITE(cst - zone, '(A8, I4)') 'CST - ZONE', j

for $1 \leq j \leq N_{cst}$. The size $N_{z,c}$ of the records depends of the type of the constraint.

$$N_{z,c} = \begin{cases} 1 & \text{if } j_{type} = 3, \text{ the record contains the number of the channel} \\ 2 & \text{if } j_{type} = 4, \text{ the record contains the number of the channel and} \\ & \text{the plan} \\ N_{ch} \times N_k & \text{if } j_{type} = 5, \text{ the record contains one value for each bundle : } =0 \\ & \text{out of the zone } j, =1 \text{ in the zone } j. \end{cases}$$

If $j_{type} = 1$ or 2 , no record is necessary to define the volume because it is implicit with the type of the constraint (reactor or core respectively).

Notes related with the different limits and values for the iterative calculations of the optimization problem:

1st	S	external step limit. It is used for the quadratic constraint if applicable. (default: 1.0)
2nd	δ	internal step limit for MAP method. (default: 0.1)
3rd	ε_{ext}	limit for external convergence. (default: 10^{-3})
4th	ε_{int}	limit for internal convergence. (default: 10^{-3})
5th	ε_4	limit for convergence for the quadratic constraint.
6th		expected value for the objective function calculated with the linearized problem.
7th	$\varepsilon_{\Delta X}$	relative variation of the decision variables for the 'epsilon' method of perturbation. (default: 10^{-2})
8th	m	Exponent for a objective function : $\min \sum_j (P_j - \bar{P}^{2m})$. (default: 8.0)
9th	α	multiplication factor for constraint weight if $S_0^o = 4$ or 5 . (default: 2.0)
10th	ε_α	limit of the error of the constraints for which function penalty weight have to be adjust.
11th	V_{core}	Volume of the core.
12th	$V_{reactor}$	Volume of the reactor.

The other value of the record are not used and set to 0.0.

Notes related with the perturbed material properties:

In the record BKP-PERT-XS-, 1 and 0 means that the corresponding perturbed material properties will be backuped or not respectively. The available perturbed material properties are :

1st	DIFFX	diffusion coefficient along x axes (D_x)
2nd	DIFY	diffusion coefficient along y axes (D_y)
3rd	DIFZ	diffusion coefficient along z axes (D_z)
4th	TOTAL	total cross-section (Σ_t)
5th	SCAT0	compressed isotropic component of the scattering matrix
6th	SIGW0	isotropic component of the within group of the scattering of the scattering cross-section
7th	SCAT1	compressed linearly anisotropic component of the scattering matrix
8th	SIGW1	linearly anisotropic component of the within group of the scattering of the scattering cross-section
9th	NFTOT	fission cross-section (Σ_f)
10th	NUSIGF	product of the fission cross-section (Σ_f) and the steady-state number of neutron produced per fission (ν^{ss})
11th	H-FACTORS	energy production coefficient (H)
12th	CHI	the steady-state energy spectrum of the neutron emitted by fission
13th	FIXE	fixed sources

Notes related with the fuel cost:

If the cost of the fuel is fixed, the dimension of the record is given by the number of enrichment zones. Otherwise the record contains the 7 parameters necessary to calculate the fuel cost, which are :

- 1st concentration in ^{235}U of the waste uranium (ε_w).
- 2nd cost of natural uranium ($C_{NU}\$/kg$).
- 3rd cost of a separation work unit ($C_S\$/SWU$).
- 4th cost of the bundle fabrication ($C_{FAB}\$/kg$).
- 5th interest rate ($int\%/day$).
- 6th time to obtain uranium ($days$).
- 7th time for enrichment ($days$).

The fuel cost is given by the equation:

Notes related with the perturbed materials properties directory:

The directories {stepdir} will be composed using the following FORTRAN instructions

$$\text{WRITE}(\text{stepdir}, '(A4, I8)') \text{'STEP'}, i$$

for $1 \leq i \leq N_{var}$. Each directory contains also the result of the multiplication of the perturbed system matrix and the flux. $i = 0$ is used to stored the result of the unperturbed system matrix and the flux.

II.3.1.1 The sub-directory /OLD-VALUE/ in /optimize/

Tableau II.43: Main records and sub-directories in //OLD-VALUE//

Name	Type	Condition	Units	Comment
VAR-VALUE _{UUUU}	$R(N_{var})$			The values of the decision variables of the last valid iteration.
FOBJ-CST-VAL	$R(N_{cst} + 1)$			The values of the objective function and the constraints of the last valid iteration.
GRADIENT _{UUUU}	$R(N_{var}, N_{cst} + 1)$			The gradients of the objective function and the constraints of the last valid iteration.

continued on next page

continued from last page

Name	Type	Condition	Units	Comment
VAR-VALUE2 _{UUU}	R(N_{var})			The values of the decision variables of the second-last valid iteration.
BEST-VAR _{UUUU}	R(N_{var})			The values of the decision variables corresponding to the best valid solution ever found.
BEST-FCT _{UUUU}	R(1)			The value of the objective function corresponding to the best valid solution ever found.

II.3.1.2 The sub-directory /stepdir/ in /optimize/

Tableau II.44: Main records and sub-directories in //stepdir//

Name	Type	Condition	Units	Comment
{Aphi}	R(N_u)			The group-dependent vectors representing the multiplication of a system matrix A and the unperturbed flux.
{Bphi}	R(N_u)			The group-dependent vectors representing the multiplication of a system matrix B and the unperturbed flux.
{/grpdir/}	Dir			The group-dependent directory containing the perturbed properties. The content is the same as for the /macrolib/ but limited to the properties to be saved only (see. BKP-PERT-XS).

The records {Aphi} and {Bphi} will be composed using the following FORTRAN in-

structions:

$$\text{WRITE}(\text{Aphi}, '(\text{A5}, \text{I7})') ' \text{A} * \text{PHI}', g$$

and

$$\text{WRITE}(\text{Bphi}, '(\text{A5}, \text{I7})') ' \text{B} * \text{PHI}', g$$

respectively where g represent the energy group.

II.3.1.3 Contents of a /optimize/ data structure in the particular case of the module GPTVRF :

When the module `GPTVRF` is used a new /optimize/ data structure can be created. This new structure contains a copy of the STATE-VECTOR record of the /optimize/ data structure used on the RHS of the module. Then only the bare minimum necessary data for gradients verification will be stored. This structure can also be deleted when a new point of the optimization procedure is calculated. If the same /optimize/ data structure is used in the module `GPTVRF` unnecessary data will be stored for the rest of the optimization calculations. This is why we recommend to use a new /optimize/ data structure on the LHS. In both cases, the /optimize/ data structure will contain the data described in the following table.

Tableau II.45: /optimize/ in the particular case of module `GPTVRF` :

Name	Type	Condition	Units	Comment
<code>SIGNATURE_</code>	<code>C*12</code>			Signature of the data structure (<code>SIGNA</code>)
<code>STATE-VECTOR</code>	<code>I(40)</code>			Vector describing the various parameters associated with this data structure S_i^o , same one or a copy of the previous description.
<code>VAR-VALUE-RF</code>	<code>R(N_{var})</code>			The references values of the decision variables for which the gradients are numerically and analytically calculated.

continued on next page

continued from last page

Name	Type	Condition	Units	Comment
GRADIENT-GPT	R($N_{var}, N_{cst} + 1$)			The gradients calculated analytically. This record is simply a copy of the record GRADIENT of the /optimize/ data structure on the RHS.
GRADIENT-EXP	R($N_{var}, N_{cst} + 1$)			The gradients calculated numerically of the objective function and the constraints. The gradients are stored the same way as for the record GRADIENT of the first description.
/varpertdir/	Dir			Directory containing the perturbation values and the corresponding values of the functions for a decision variable i .

Notes related with the perturbed decision variables directory:

The directories {varpertdir} will be composed using the following FORTRAN instructions

$$\text{WRITE}(\text{varpertdir}, '(A8, I4)') \text{'VAR - PERT', } i$$

for $1 \leq i \leq N_{var}$. Each directory contains also the records described in the following table.

Tableau II.46: The sub-directory /varpertdir/ in /optimize/

Name	Type	Condition	Units	Comment
EPSILON _{UUUUUU}	R($N_{\varepsilon, i}$)			The value of the perturbation of the decision variable i .
{fvalpert}	R($N_{cst} + 1$)			The values of the objective function and constraints for the perturbation p of the decision variable i .

The number of perturbation $N_{\varepsilon,i}$ can include the unperturbed case ($\varepsilon = 0.0$) and can be different for each decision variable i . Moreover, the values of the perturbation do not have to be in increasing or decreasing order.

The records {fvalpert} will be composed using the following FORTRAN instructions

```
WRITE(fvalpert,'(A9,I3)') 'F - C - VAL - P',p
```

for $1 \leq p \leq N_{\varepsilon,i}$.

II.3.2 Contents of a /tabu/ data structure

The /tabu/ specification is used to store the decision variable set used for a tabu search optimization method. The different options of this method are also stored in the /tabu/ data structure.

The signature variable for this data structure must be `SIGNA=L_TABU`. The dimensioning parameters for this data structure, which are stored in the state vector \mathcal{S}_i^t , represents:

- The number of neighbors $N_{eig} = \mathcal{S}_1^t$.
- The type of neighborhood \mathcal{S}_2^t where

$$\mathcal{S}_3^t = \begin{cases} 1 & \text{hyperrectangle} \\ 2 & \text{ball} \end{cases}$$

- The type of discretisation of the neighborhood. \mathcal{S}_3^t where

$$\mathcal{S}_3^t = \begin{cases} 1 & \text{geometric} \\ 2 & \text{linear} \\ 3 & \text{isovolume} \end{cases}$$

- The test for external convergence $Nit_{max} = \mathcal{S}_4^t$.
- The current number of iterations without improvement of the best value $Nit = \mathcal{S}_5^t$.
- The tabu list maximum size $L_t = \mathcal{S}_6^t$.
- The promising area list maximum size $L_p = \mathcal{S}_7^t$.
- The tabu list current size $L_t^c = \mathcal{S}_8^t$.
- The promising list current size $L_p^c = \mathcal{S}_9^t$.
- The number of decision variables $N_{var} = \mathcal{S}_{10}^t$.
- The number of constraints $N_{cst} = \mathcal{S}_{11}^t$.
- The first number for the random point generation algorithm \mathcal{S}_{39}^t (=TIME()).
- The number of generated random numbers \mathcal{S}_{40}^t .

Tableau II.47: Main records and sub-directories in /tabu/

Name	Type	Condition	Units	Comment
SIGNATURE _{UUU}	C*12			Signature of the data structure (SIGNA)
STATE-VECTOR	I(40)			Vector describing the various parameters associated with this data structure \mathcal{S}_i^t and the tabu search optimization integer options
TABU-PARAM-R	R(40)			Vector containing various tabu search optimization real options and several current values.
CURRENT-VAL _U	R($N_{var} + N_{cst} + 2$)			The values of the current decision variables, the corresponding functionals and the tabu function
BEST-VAL _{UUUUU}	R($N_{var} + N_{cst} + 2$)			The values of the decision variables, the corresponding functionals and the tabu function for the best point ever found
NEIGHBORHOOD	Dir			Directory containing the neighbor decision variable set.
TABU-LIST _{UUU}	Dir			Directory containing the tabu decision variable set.
PROMISE-LIST	Dir			Directory containing the decision variable set corresponding to promising areas.

Notes related with the different limits and values for the iterative calculations of the optimization problem:

1st	R_n	neighborhood radius (fraction $\in [0, 1]$ of the total decision space). (default: 0.01)
2nd	R_t	radius around tabu list decision variable set within point points are also tabu (fraction $\in [0, 1]$ of the total decision space). (default: 0.001)
3rd	R_p	radius around promising area list decision variable set within point points are also tabu (fraction $\in [0, 1]$ of the total decision space). (default: 0.002)
4th	$fact$	factor in case of a geometric discretisation of the neighborhood. (default: 2)
5th		best tabu search function value ever achieved.

The other value of the record are not used and set to 0.0.

II.3.2.1 The sub-directories in /optimize/

The sub-directories are NEIGHBORHOOD, TABU-LIST and PROMISE-LIST.

Tableau II.48: Main records and sub-directories in //stepdir//

Name	Type	Condition	Units	Comment
{Item}	$R(N_{var} + N_{cst} + 2)$			Decision variable set, its corresponding functional and tabu search function values.
TABU-F-VAL _{UU}	$R(N_{eig})$			The tabu search function values of the neighbors. This record is reinitialized to 0.0 each time a new neighborhood is created. NEIGHBORHOOD is the only directory to contain this record.

The records {Item} will be composed using the following FORTRAN instructions:

```
WRITE(Item, '(A4, I8)') 'ITEM', i
```

with i between

- 1 and N_{eig} for NEIGHBORHOOD directory
- 1 and $\min\{L_t^c, L_t\}$ for TABU-LIST directory
- 1 and $\min\{L_p^c, L_p\}$ for PROMISE-LIST directory

II.3.3 Contents of a /table/ data structure

The /table/ data structure is used to store burnup-dependent nuclear properties, including diffusion coefficients, reciprocal velocities and macroscopic cross sections. For each nuclear property, first derivative with respect to burnup can be also stored for further interpolation.

The signature variable for this data structure must be `SIGNA=L_TABLE_XXXX`. The dimensioning parameters for this data structure, which are stored in the state vector \mathcal{S}_i^t , represent:

- the maximum number of energy groups $\mathcal{S}_1^t = M_G$;
- the maximum number of burnup or irradiation steps $\mathcal{S}_2^t = M_b$;
- the maximum order of anisotropy of the scattering cross sections $\mathcal{S}_3^t = M_L$, ($L = 1$ for isotropic scattering in the laboratory system).
- the maximum number of coefficients for each property $\mathcal{S}_4^t = d$. If derivatives are included, $d = 2$, if not, $d = 1$. If fuel properties are enrichment dependent, $d = 4$.
- the number of enrichment values $\mathcal{S}_5^t = M_\epsilon$;
- the discontinuity factor flag $I_{df} = \mathcal{S}_7^t$:

$$I_{df} = \begin{cases} 0 & \text{no discontinuity factor information} \\ > 0 & \text{the discontinuity factors are available.} \end{cases}$$

Tableau II.49: Main records and sub-directories in /table/

Name	Type	Condition	Units	Comment
<code>SIGNATURE_XXXX</code>	C*12			Signature of the data structure (SIGNA)
<code>STATE-VECTOR</code>	I(40)			Vector describing the various parameters associated with this data structure \mathcal{S}_i^t
<code>{/nam_fuel/}</code>	Dir			Material table sub-directory containing burnup-dependent fuel properties for a specific type (Ex: Depleted, Natural, DUPIC ...)

Tableau II.50: Main records in a fuel type sub-directory

Name	Type	Condition	Units	Comment
STATE-VECTOR	I(40)			Vector describing the various parameters associated with this data structure \mathcal{S}_i
VOLUME _{UUUUUUU}	R(1)		cm ³	Cell volume from /compo/ informations.
ENERGY _{UUUUUUU}	R($G+1$)		eV	Energy groups limits E_g . Group g is defined as $E(g) < E \leq E(g-1)$.
BURNUP _{UUUUUUU}	R(N_b)		MW d T ⁻¹	Burnup reached at the different burnup steps B_k
N/KB _{UUUUUUUUU}	R(N_b)		Kb ⁻¹	Fuel irradiation reached at the different burnup steps w_k
DIFFX _{UUUUUUUUU}	R(N_b, G, d)		cm	Burnup ordered X -directed diffusion coefficients and derivatives with respect to burnup. Blocks DIFFY and DIFFZ are generally taken equal to DIFFX.
DIFFY _{UUUUUUUUU}	R(N_b, G, d)		cm	Burnup ordered Y -directed diffusion coefficients and derivatives with respect to burnup.
DIFFZ _{UUUUUUUUU}	R(N_b, G, d)		cm	Burnup ordered Z -directed diffusion coefficients and derivatives with respect to burnup.
TOTAL _{UUUUUUUUU}	R(N_b, G, d)		cm ⁻¹	Burnup ordered macroscopic total cross sections and derivatives with respect to burnup.
SCAT _{UUUUUUUUU}	R(N_b, L, G, G, d)		cm ⁻¹	Burnup ordered macroscopic scattering cross sections and derivatives with respect to burnup.
NFTOT _{UUUUUUUUU}	R(N_b, G, d)		cm ⁻¹	Burnup ordered macroscopic fission cross sections and derivatives with respect to burnup.

continued on next page

continued from last page

Name	Type	Condition	Units	Comment
NUSIGF _{UUUUUUUU}	$R(N_b, G, d)$		cm^{-1}	Burnup ordered $\nu(jg, matnum) \times \Sigma_f(jg, matnum)$ where $\nu(jg, matnum)$ is the number of neutrons per fission and $\Sigma_f(jg, matnum)$ is the macroscopic fission cross section and derivatives with respect to burnup.
H-FACTORS _{UU}	$R(N_b, G, d)$		$J cm^{-1}$	Burnup ordered macroscopic H-factors (i.e. product of the macroscopic fission cross section times the energy recovered by fission) and derivatives with respect to burnup.
CHI _{UUUUUUUUU}	$R(N_b, G, d)$			Mixture ordered fission spectrum and derivatives with respect to burnup.
OVERV _{UUUUUUUU}	$R(N_b, G, d)$		$cm^{-1}s$	Average reciprocal velocities ($1/v_g$ in $s cm^{-1}$) and derivatives with respect to burnup.
FLUX-INTG _{UUU}	$R(N_b, G, d)$		$n s^{-1}$	Burnup ordered volume integrated fluxes and derivatives with respect to burnup.
DISFACX _{UUUUU}	$R(N_b, 2, G, \mathcal{G}_7^k = 1)$		-	The X directed multigroup transport cross assembly discontinuity factors $f_{x\pm}^g$. By default, these factors are set equal to 1.0
DISFACY _{UUUUU}	$R(N_b, 2, G, \mathcal{G}_7^k = 1)$		-	The Y directed multigroup transport cross assembly discontinuity factors $f_{y\pm}^g$. By default, these factors are set equal to 1.0
DISFACZ _{UUUUU}	$R(N_b, 2, G, \mathcal{G}_7^k = 1)$		-	The Z directed multigroup transport cross assembly discontinuity factors $f_{z\pm}^g$. By default, these factors are set equal to 1.0.

The dimensioning parameters stored in a fuel type sub directory in the vector 'STATE-VECTOR' represent:

- the number of energy groups $\mathcal{S}_1 = G$ such that $G \leq M_G$;
- the number of burnup or irradiation steps $\mathcal{S}_2 = N_b$ such that $N_b \leq M_b$;
- the order of anisotropy of the scattering cross sections $\mathcal{S}_3 = L$, ($L = 1$ for isotropic scattering in the laboratory system), such that $L \leq M_L$.
- the number of coefficients for each property $\mathcal{S}_4 = d$. If derivatives are included, $d = 2$, if not, $d = 1$.

Stored fuel properties correspond to:

$$\begin{aligned}
 d = 1 & \quad \Sigma(B) \\
 d = 2 & \quad \Sigma(B), \frac{\partial \Sigma(B)}{\partial B} \\
 d = 4 & \quad \Sigma(B)|_\varepsilon, \frac{\partial \Sigma(B)}{\partial B}|_\varepsilon, \frac{\partial \Sigma(B)}{\partial \varepsilon}|_\varepsilon, \frac{\partial^2 \Sigma(B)}{\partial B \partial \varepsilon}|_\varepsilon
 \end{aligned}$$

The discontinuity factors are stored for f_- then f_+ for all burnup steps.

The scattering cross sections are not stored in a compressed mode because it is convenient mostly if derivatives are present. There is not much loss of memory space as few energy groups are used in diffusion theory and that thus the scattering matrix are almost full matrix.

II.3.4 The modifications in previous data structures

SIGNA=L_MAP_{LLLLLLLL}

- Number of Enrichment zones, $N_{ez} = \mathcal{S}_{13}^m$

Tableau II.51: Main records and sub-directories in /map/

Name	Type	Condition	Units	Comment
ENR-ZONE _{LLLLL}	I(N_{ch})			The definition of the enrichment zones for all channels. The greater number must be lower or equal to the number of enrichment zones N_{ez} .
ENR-VALUE _{LLLL}	R(N_{ez})			The value of the ^{235}U concentration of the new fuel in each zone.
POWER-CHA _{LLLL}	R(N_{cha})			The channel power distribution.

ANNEXE III

INTERPOLATION DES SECTIONS EFFICACES

Dans le code DONJON, les sections efficaces macroscopiques sont soit récupérées soit reconstruites à partir des fichiers COMPO dans les modules CRE: ou XSCONS: modules. Dans l'objet COMPO, les propriétés se présentent sous la forme suivante:

- les propriétés macroscopiques Σ
- les propriétés microscopiques σ_i de l'isotope ou pseudo-isotope i , ainsi que la densité N_i .

Les propriétés sont alors simplement données par:

$$\Sigma = \Sigma + \sum_i N_i \sigma_i \quad (\text{III.1})$$

Les propriétés ne sont données que pour différentes valeurs discrètes du burnup, qui correspondent au pas d'évolution dans le calcul de transport DRAGON.

Le but des modules CRE: ou XSCONS: avec XSFUEL: est d'interpoler les propriétés pour toutes les grappes de combustible. Le module CRE: interpole directement à partir des objets COMPO. Le module XSCONS: fait une pré-interpolation des propriétés ainsi que de plusieurs de leurs dérivées. Les résultats sont stockés pour le reste des calculs dans un objet TABLE. Ce dernier permet une interpolation plus rapide par la suite avec le module XSFUEL: lors de ses nombreux appels subséquents.

III.1 Méthode d'Hermite pour une interpolation à une variable

Avec la méthode d'Hermite, les propriétés ainsi que leur dérivées sont nécessaires. Ces dernières sont calculées à l'interne du module **CRE** : ou par le module **XSCONS** : en utilisant la méthode de Ceschino [27]. Les valeurs suivantes sont donc disponibles pour l'interpolation:

$$\Sigma(B_i) \text{ et } \frac{\partial \Sigma}{\partial B}(B_i)$$

où B_i correspond au pas d'évolution du calcul de transport.

Pour simplifier les notations et garder le caractère général de la méthode d'Hermite, nous utiliserons $f(x)$, où f représente les propriétés et x le burnup. Nous utiliserons un polynôme d'Hermite de degré 3. Pour une interpolation directe des propriétés, la fonction f pour tout x qui appartient à l'intervalle $[x_i, x_{i+1}]$ est donnée par:

$$f(x) = f(x_i)H_1(u) + f'(x_i)H_2(u) * \Delta x + f(x_{i+1})H_3(u) + f'(x_{i+1})H_4(u) * \Delta x \quad (\text{III.2})$$

où $u = \frac{1}{\Delta x} \left[x - \frac{1}{2}(x_i + x_{i+1}) \right]$, $\Delta x = (x_{i+1} - x_i)$ et les polynômes d'Hermite de degré 3 sont:

$$\begin{cases} H_1(u) = 3.0 * (0.5 - u)^2 - 2.0 * (0.5 - u)^3 \\ H_2(u) = (0.5 - u)^2 - (0.5 - u)^3 \\ H_3(u) = 3.0 * (0.5 + u)^2 - 2.0 * (0.5 + u)^3 \\ H_4(u) = -(0.5 + u)^2 + (0.5 + u)^3 \end{cases} \quad (\text{III.3})$$

Suivant les calculs fait avec le code **DONJON**, il est parfois nécessaire d'évaluer la dérivée ou l'intégrale des propriétés. En utilisant la formulation d'Hermite, la

dérivée de f en x qui appartient à l'intervalle $[x_i, x_{i+1}]$:

$$\begin{aligned} \frac{\partial f}{\partial x}(x) = & \frac{1}{\Delta x} (f(x_i)H'_1(u) + f'(x_i)H'_2(u) * \Delta x \\ & + f(x_{i+1})H'_3(u) + f'(x_{i+1})H'_4(u) * \Delta x) \end{aligned} \quad (\text{III.4})$$

où

$$\begin{cases} H'_1(u) = -6.0 * (0.5 - u) + 6.0 * (0.5 - u)^2 \\ H'_2(u) = -2.0(0.5 - u) - 3.0 * (0.5 - u)^2 \\ H'_3(u) = 6.0 * (0.5 + u) - 6.0 * (0.5 + u)^2 \\ H'_4(u) = -2.0(0.5 + u) + 3.0(0.5 + u)^2 \end{cases} \quad (\text{III.5})$$

Quant à l'intégrale des propriétés sur l'intervalle $[x_0, x_1]$ est séparée sur chaque intervalle de burnup $[x_i, x_{i+1}]$ de la manière suivante:

$$\int_{x_0}^{x_1} dx f(x) = \int_{x_0}^{x_{n_0}} dx f(x) + \sum_{i=n_0}^{n_1-1} \int_{x_i}^{x_{i+1}} dx f(x) + \int_{x_{n_1}}^{x_1} dx f(x) \quad (\text{III.6})$$

avec $x_{n_0-1} < x_0 \leq x_{n_0}$ et $x_{n_1} < x_1 \leq x_{n_1+1}$.

La valeur de chaque intégrale est donnée par la formule générale suivante:

$$\begin{aligned} \int_{x_A}^{x_B} dx f(x) = & C \sum_{j=1}^2 (f(x_{n_i})H_1(uu(j)) + f'(x_{n_i})H_2(uu(j)) * \Delta x \\ & + f(x_{n_{i+1}})H_3(uu(j)) + f'(x_{n_{i+1}})H_4(uu(j)) * \Delta x) \end{aligned} \quad (\text{III.7})$$

où $\Delta x = (x_{n_{i+1}} - x_{n_i})$, $C = 0.5 * \frac{(x_B - x_A)}{x_{n_{i+1}} - x_{n_i}}$, les polynomes d'Hermite sont donnés par l'équation (III.3) et

$$\begin{cases} u1 = \frac{1}{\Delta x} \left[x_A - \frac{1}{2}(x_{n_i} + x_{n_{i+1}}) \right] \\ u2 = \frac{1}{\Delta x} \left[x_B - \frac{1}{2}(x_{n_i} + x_{n_{i+1}}) \right] \\ uu(1) = 0.5 * (-(u2 - u1) * 1/\sqrt{3} + u1 + u2) \\ uu(2) = 0.5 * ((u2 - u1) * 1/\sqrt{3} + u1 + u2) \end{cases} \quad (\text{III.8})$$

avec x_A et x_B qui représentent les bornes inférieures et supérieures des intégrales de droite de l'équation III.6 respectivement, et donc $x_{n_i} \leq x_A < x_B \leq x_{n_{i+1}}$, où x_{n_i} et $x_{n_{i+1}}$ définissent une intervalle de burnup.

Les propriétés $\Sigma(B)$, leur dérivée $\frac{d\sigma}{dB}(B)$ ainsi que leur valeur moyennée dans le temps (modèle TA) $\bar{\Sigma}(B_e)$ (B_e est burnup moyen de sortie) peuvent être obtenue par les équations (III.2), (III.4) et (III.6).

Pour le cas particulier de l'optimisation de la gestion du combustible, la dérivé des propriétés moyennées dans le temps par rapport aux burnup de sortie est donné par l'équation (3.3). Elle est fonction de $\Sigma(B)$ et de $\bar{\Sigma}(B_e)$, et peut donc être obtenue indirectement avec les équations (III.2) et (III.6).

III.2 Méthode d'Hermite pour une interpolation à deux variables

Dans la pratique, les propriétés dépendent de nombreux paramètres en plus du burnup. Il est donc intéressant de pouvoir interpoler les propriétés pour plusieurs variables. Nous nous limiterons cependant ici à deux variables. Les méthodes d'interpolation d'Hermite peuvent être généralisées à des fonctions de deux variables, $f(x, y)$. Par convention, nous supposons que x représente le burnup, et y la deuxième variable (l'enrichissement dans notre application). De plus, le point (x, y) fait parti d'un domaine rectangulaire tel que $x \in [x_i, x_{i+1}]$ and $y \in [y_j, y_{j+1}]$. Nous définissons alors les polynomes bi-cubic d'Hermite $H(u) * H(v)$ où $u = \frac{1}{\Delta x} \left[x - \frac{1}{2}(x_i + x_{i+1}) \right]$, $\Delta x = (x_{i+1} - x_i)$, $v = \frac{1}{\Delta y} \left[y - \frac{1}{2}(y_j + y_{j+1}) \right]$, $\Delta y = (y_{j+1} - y_j)$, et H sont les polynomes d'Hermite définis par l'équation (III.3). La

fonction $f(x, y)$ est alors donnée par:

$$\begin{aligned}
f(x, y) = & H_1(u)H_1(v)f(x_i, y_j) + H_2(u)H_1(v) * \Delta x \frac{\partial f}{\partial x}(x_i, y_j) \\
& + H_1(u)H_2(v) * \Delta y \frac{\partial f}{\partial y}(x_i, y_j) + H_2(u)H_2(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_i, y_j) \\
& + H_3(u)H_1(v)f(x_{i+1}, y_j) + H_4(u)H_1(v) * \Delta x \frac{\partial f}{\partial x}(x_{i+1}, y_j) \\
& + H_3(u)H_2(v) * \Delta y \frac{\partial f}{\partial y}(x_{i+1}, y_j) + H_4(u)H_2(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_{i+1}, y_j) \\
& + H_1(u)H_3(v)f(x_i, y_{j+1}) + H_2(u)H_3(v) * \Delta x \frac{\partial f}{\partial x}(x_i, y_{j+1}) \\
& + H_1(u)H_4(v) * \Delta y \frac{\partial f}{\partial y}(x_i, y_{j+1}) + H_2(u)H_4(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_i, y_{j+1}) \\
& + H_3(u)H_3(v)f(x_{i+1}, y_{j+1}) + H_4(u)H_3(v) * \Delta x \frac{\partial f}{\partial x}(x_{i+1}, y_{j+1}) \\
& + H_3(u)H_4(v) * \Delta y \frac{\partial f}{\partial y}(x_{i+1}, y_{j+1}) \\
& + H_4(u)H_4(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_{i+1}, y_{j+1})
\end{aligned} \tag{III.9}$$

Comme pour le cas à une variable, les dérivés partielles selon x et y sont calculées par la méthode de Ceshino. Pour la dérivée croisée, elle est obtenue en calculant la dérivée selon y des dérivées selon x . Pour plus de précision, il serait préférable de calculer également la dérivée selon x des dérivées selon y et de faire la moyenne des deux méthodes. les calculs à deux variables ne sont possibles qu'avec les modules XSCONS: et XSFUEL:. Les valeurs suivantes sont donc précalculées par le module XSCONS: pour l'interpolation dans le module XSFUEL: :

$$\Sigma(B_i, \varepsilon_j), \frac{\partial \Sigma}{\partial B}(B_i, \varepsilon_j), \frac{\partial \Sigma}{\partial \varepsilon}(B_i, \varepsilon_j) \text{ et } \frac{\partial^2 \Sigma}{\partial B \partial \varepsilon}(B_i, \varepsilon_j)$$

où B_i correspond au pas d'évolution du calcul de transport, ε_j au différents enrichissement pour lesquels les calculs de transport ont été faits.

Pour que l'interpolation à deux variables soit valable, les intervalles doivent former un domaine cartésien.

Encore une fois, une interpolation de la dérivé ou l'intégrale des propriétés peut

être nécessaire suivant les calculs envisagés. Les dérivées partielles 'directes' sont facilement obtenues à partir de l'équation (III.9), étant donné que les variables sont découplées dans les polynômes bi-cubiques d'Hermite. Elles sont donc données par:

$$\begin{aligned}
\frac{\partial f}{\partial x}(x, y) = & \frac{1}{\Delta x}[H'_1(u)H_1(v)f(x_i, y_j) + H'_2(u)H_1(v) * \Delta x \frac{\partial f}{\partial x}(x_i, y_j) \\
& + H'_1(u)H_2(v) * \Delta y \frac{\partial f}{\partial y}(x_i, y_j) + H'_2(u)H_2(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_i, y_j) \\
& + H'_3(u)H_1(v)f(x_{i+1}, y_j) + H'_4(u)H_1(v) * \Delta x \frac{\partial f}{\partial x}(x_{i+1}, y_j) \\
& + H'_3(u)H_2(v) * \Delta y \frac{\partial f}{\partial y}(x_{i+1}, y_j) + H'_4(u)H_2(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_{i+1}, y_j) \\
& + H'_1(u)H_3(v)f(x_i, y_{j+1}) + H'_2(u)H_3(v) * \Delta x \frac{\partial f}{\partial x}(x_i, y_{j+1}) \\
& + H'_1(u)H_4(v) * \Delta y \frac{\partial f}{\partial y}(x_i, y_{j+1}) + H'_2(u)H_4(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_i, y_{j+1}) \\
& + H'_3(u)H_3(v)f(x_{i+1}, y_{j+1}) + H'_4(u)H_3(v) * \Delta x \frac{\partial f}{\partial x}(x_{i+1}, y_{j+1}) \\
& + H'_3(u)H_4(v) * \Delta y \frac{\partial f}{\partial y}(x_{i+1}, y_{j+1}) \\
& + H'_4(u)H_4(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_{i+1}, y_{j+1})]
\end{aligned} \tag{III.10}$$

et

$$\begin{aligned}
\frac{\partial f}{\partial y}(x, y) = & \frac{1}{\Delta y}[H_1(u)H'_1(v)f(x_i, y_j) + H_2(u)H'_1(v) * \Delta x \frac{\partial f}{\partial x}(x_i, y_j) \\
& + H_1(u)H'_2(v) * \Delta y \frac{\partial f}{\partial y}(x_i, y_j) + H_2(u)H'_2(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_i, y_j) \\
& + H_3(u)H'_1(v)f(x_{i+1}, y_j) + H_4(u)H'_1(v) * \Delta x \frac{\partial f}{\partial x}(x_{i+1}, y_j) \\
& + H_3(u)H'_2(v) * \Delta y \frac{\partial f}{\partial y}(x_{i+1}, y_j) + H_4(u)H'_2(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_{i+1}, y_j) \\
& + H_1(u)H'_3(v)f(x_i, y_{j+1}) + H_2(u)H'_3(v) * \Delta x \frac{\partial f}{\partial x}(x_i, y_{j+1}) \\
& + H_1(u)H'_4(v) * \Delta y \frac{\partial f}{\partial y}(x_i, y_{j+1}) + H_2(u)H'_4(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_i, y_{j+1}) \\
& + H_3(u)H'_3(v)f(x_{i+1}, y_{j+1}) + H_4(u)H'_3(v) * \Delta x \frac{\partial f}{\partial x}(x_{i+1}, y_{j+1}) \\
& + H'_3(u)H_4(v) * \Delta y \frac{\partial f}{\partial y}(x_{i+1}, y_{j+1}) \\
& + H_4(u)H'_4(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_{i+1}, y_{j+1})]
\end{aligned} \tag{III.11}$$

où H et H' sont définis par les équations (III.3) et (III.5).

L'intégrale des propriétés n'a de sens que pour les calculs en TA, et donc selon la

variable x (le burnup) pour une valeur donnée de y . Alors, l'intégrale des propriétés sur l'intervalle $[x_0, x_1]$ est séparée sur chaque intervalle de burnup $[x_i, x_{i+1}]$ de manière similaire au cas à une variable, c'est à dire:

$$\int_{x_0}^{x_1} dx f(x, y) = \int_{x_0}^{x_{n_0}} dx f(x, y) + \sum_{i=n_0}^{n_1-1} \int_{x_i}^{x_{i+1}} dx f(x, y) + \int_{x_{n_1}}^{x_1} dx f(x, y) \quad (\text{III.12})$$

avec $x_{n_0-1} < x_0 \leq x_{n_0}$ et $x_{n_1} < x_1 \leq x_{n_1+1}$.

Les variables étant découplées dans la fonction $f(x, y)$, la valeur de chaque intégrale est donnée par la formule générale suivante:

$$\begin{aligned} \int_{x_A}^{x_B} dx f(x, y) = & \\ C \sum_{k=1}^2 [& H_1(uu(k))H_1(v)f(x_i, y_j) + H_2(uu(k))H_1(v) * \Delta x \frac{\partial f}{\partial x}(x_i, y_j) \\ & + H_1(uu(k))H_2(v) * \Delta y \frac{\partial f}{\partial y}(x_i, y_j) + H_2(uu(k))H_2(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_i, y_j) \\ & + H_3(uu(k))H_1(v)f(x_{i+1}, y_j) + H_4(uu(k))H_1(v) * \Delta x \frac{\partial f}{\partial x}(x_{i+1}, y_j) \\ & + H_3(uu(k))H_2(v) * \Delta y \frac{\partial f}{\partial y}(x_{i+1}, y_j) + H_4(uu(k))H_2(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_{i+1}, y_j) \\ & + H_1(uu(k))H_3(v)f(x_i, y_{j+1}) + H_2(uu(k))H_3(v) * \Delta x \frac{\partial f}{\partial x}(x_i, y_{j+1}) \\ & + H_1(uu(k))H_4(v) * \Delta y \frac{\partial f}{\partial y}(x_i, y_{j+1}) + H_2(uu(k))H_4(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_i, y_{j+1}) \\ & + H_3(uu(k))H_3(v)f(x_{i+1}, y_{j+1}) + H_4(uu(k))H_3(v) * \Delta x \frac{\partial f}{\partial x}(x_{i+1}, y_{j+1}) \\ & + H_3(uu(k))H_4(v) * \Delta y \frac{\partial f}{\partial y}(x_{i+1}, y_{j+1}) \\ & + H_4(uu(k))H_4(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_{i+1}, y_{j+1})] \end{aligned} \quad (\text{III.13})$$

où H et uu sont définis par les équations (III.3) et (III.8) et, x_A et x_B qui sont définis à la suite de l'équation (III.8).

Pour le modèle TA, il peut être également nécessaire de connaître la dérivée partielle

selon y des propriétés moyennées dans le temps. Celle-ci est donnée par:

$$\begin{aligned}
\frac{\partial}{\partial y} [\int_{x_A}^{x_B} dx f(x, y)] = & \\
\frac{C}{\Delta y} \sum_{k=1}^2 [& H_1(uu(k))H_1'(v)f(x_i, y_j) + H_2(uu(k))H_1'(v) * \Delta x \frac{\partial f}{\partial x}(x_i, y_j) \\
& + H_1(uu(k))H_2'(v) * \Delta y \frac{\partial f}{\partial y}(x_i, y_j) + H_2(uu(k))H_2'(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_i, y_j) \\
& + H_3(uu(k))H_1'(v)f(x_{i+1}, y_j) + H_4(uu(k))H_1'(v) * \Delta x \frac{\partial f}{\partial x}(x_{i+1}, y_j) \\
& + H_3(uu(k))H_2'(v) * \Delta y \frac{\partial f}{\partial y}(x_{i+1}, y_j) + H_4(uu(k))H_2'(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_{i+1}, y_j) \\
& + H_1(uu(k))H_3'(v)f(x_i, y_{j+1}) + H_2(uu(k))H_3'(v) * \Delta x \frac{\partial f}{\partial x}(x_i, y_{j+1}) \\
& + H_1(uu(k))H_4'(v) * \Delta y \frac{\partial f}{\partial y}(x_i, y_{j+1}) + H_2(uu(k))H_4'(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_i, y_{j+1}) \\
& + H_3(uu(k))H_3'(v)f(x_{i+1}, y_{j+1}) + H_4(uu(k))H_3'(v) * \Delta x \frac{\partial f}{\partial x}(x_{i+1}, y_{j+1}) \\
& + H_3(uu(k))H_4'(v) * \Delta y \frac{\partial f}{\partial y}(x_{i+1}, y_{j+1}) \\
& + H_4(uu(k))H_4'(v) * \Delta x * \Delta y \frac{\partial^2 f}{\partial x \partial y}(x_{i+1}, y_{j+1})]
\end{aligned} \tag{III.14}$$

où H , H' et uu sont définis par les équations (III.3), (III.5) et (III.8) et, x_A et x_B qui sont définis à la suite de l'équation (III.8).

Similairement au cas à une variable, pour le cas particulier de l'optimisation de la gestion du combustible, la dérivé des propriétés moyennées dans le temps par rapport aux burnup de sortie est donné par l'équation (3.3). Elle peut donc être obtenue indirectement avec les équations (III.9) et (III.12).

ANNEXE IV

GÉNÉRATION DE NOMBRES ALÉATOIRES

Pour la génération de nombres aléatoires z , nous avons utilisé un des algorithmes proposé par Park et Miller dans [71]. Cet algorithme est basé sur une suite de nombre générés par récurrence. La fonction analytique utilisée consiste essentiellement en une suite géométrique (facteur multiplicateur a) dont la valeur modulo un chiffre prédéterminé m est prise. Dans la formule de récurrence proposée, le nouveau point ne dépend que du point de l'itération précédente ($z_{n+1} = f(z_n)$). Donc dès qu'un point revient, une boucle est obtenue. Le choix judicieux des paramètres assure donc d'avoir des boucles très longues. Le premier chiffre de la suite récursive est généré avec la fonction RAND de fortran.

Les premiers tests avec la méthode taboue ont été réalisés sur des fonctions analytiques. Lors de ces tests qui sont relativement rapides, et donc qui étaient très rapprochés dans le temps, nous avons remarqué que le premier point (relié au résultat de la fonction RAND) était proche d'un essai à l'autre. Cela se traduisait par un vecteur de décision initial toujours dans la même région de l'espace décision. Nous avons aussi remarqué que la suite des nombres aléatoires générés par l'algorithme de Park et Miller diffère rapidement d'un calcul à l'autre, même si le point de départ est proche. Aussi, pour contourner le problème du point de départ, nous avons rajouté une autre étape pour la génération des nombres aléatoires. Après avoir généré le premier point avec RAND, l'algorithme de Park et Miller est appelé k fois avant d'utiliser la valeur obtenue pour la suite des calculs. Ce nombre k est donné en prenant le résultat de la fonction TIME modulo 100. La fonction TIME donne le nombre de secondes écoulées entre le 1er janvier 1978 et

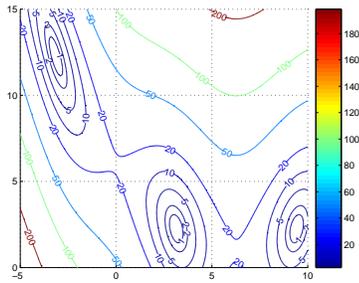
le moment actuel. Nous avons donc présumé que même si deux calculs sont lancés de manière rapprochés dans le temps, la valeur de k sera 'aléatoire', et donc par voie de conséquence le point initial pour la génération de nombres aléatoires également.

D'un point de vue pratique, la valeur du premier point et celle du point précédent doivent être gardés. Sachant que dans DONJON, une seule variable globale est utilisée, nous avons stocké ces deux valeurs dans l'objet L_TABU.

ANNEXE V

FONCTIONS ANALYTIQUES TESTS

Branin RCOS (RC):



$$RC(x_1, x_2) =$$

$$(x_2 - (5/(4\pi^2))x_1^2 + (5/\pi)x_1 - 6)^2 + 10(1 - (1/(8\pi)))\cos(x_1) + 10;$$

espace de recherche: $-5 < x_1 < 10$,
 $0 < x_2 < 15$;

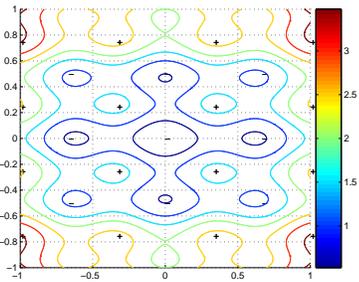
pas de minimum local;

3 minima globaux: $(x_1, x_2)^* = (-\pi, 49/4)$,

$(\pi, 9/4), (3\pi, 9/4)$;

$$RC((x_1, x_2)^*) = 5/(4\pi) = 0.397887$$

B2 (B2):



$$B2(x_1, x_2) = x_1^2 + 2x_2^2$$

$$-0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7;$$

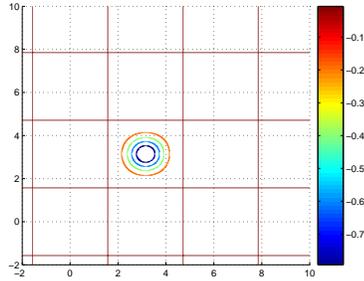
espace de recherche: $-1 < x_i < 1$, $i = 1, 2$;

plusieurs minimum locaux;

1 minimum global: $(x_1, x_2)^* = (0, 0)$;

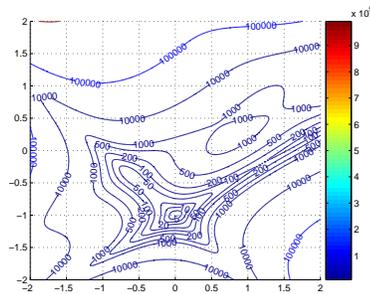
$$B2((x_1, x_2)^*) = 0.$$

Eason (ES):



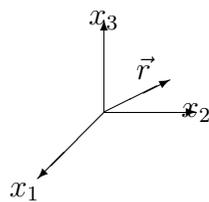
$ES(x_1, x_2) =$
 $-\cos(10x_1) \cos(10x_2)$
 $\times \exp(-((x_1 - \pi)^2 + (x_2 - \pi)^2));$
 espace de recherche: $-10 < x_i < 10, i = 1, 2;$
 plusieurs minimum locaux;
 1 minimum global: $(x_1, x_2)^* = (\pi, \pi);$
 $ES((x_1, x_2)^*) = -1.$

Goldstein Price (GP):



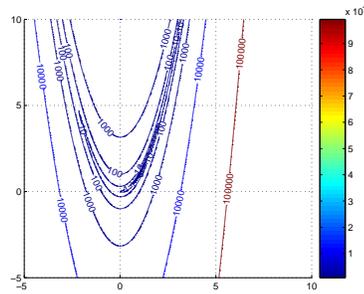
$GP(x_1, x_2) =$
 $[1 + (x_1 + x_2 + 1)^2$
 $\times (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$
 $\times [30 + (2x_1 - 3x_2)^2$
 $\times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)];$
 espace de recherche: $-2 < x_i < 2, i = 1, 2;$
 4 minima locaux;
 1 minimum global: $(x_1, x_2)^* = (0, -1);$
 $GP((x_1, x_2)^*) = 3$

De Jong (DJ):



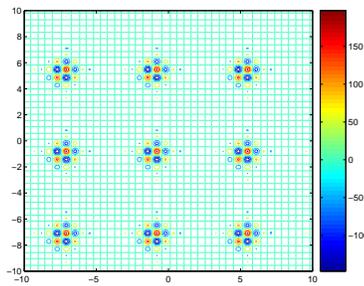
$DJ(x_1, x_2, x_3) =$
 $x_1^2 + x_2^2 + x_3^2;$
 espace de recherche: $-5.12 < x_i < 5.12;$
 pas minimum local;
 1 minimum global: $(x_1, x_2, x_3)^* = (0, 0, 0);$
 $DJ((x_1, x_2, x_3)^*) = 0$

Rosenbrock (Rn):



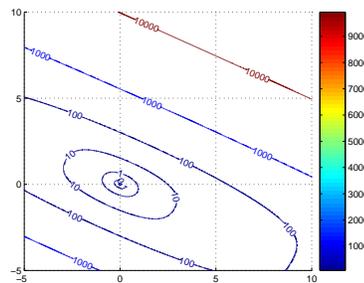
$Rn(x) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$;
 espace de recherche: $-5 < x_i < 10$;
 plusieurs minimum locaux;
 1 minimum global: $x^* = (1, \dots, 1)$;
 $Rn(x^*) = 0$

Shubert (SH):



$SH(x_1, x_2) = \{\sum_{j=1}^5 j \cos((j+1)x_1 + j)\}$
 $* \{\sum_{j=1}^5 j \cos((j+1)x_2 + j)\}$;
 espace de recherche: $-10 < x_i < 10, i = 1, 2$;
 760 minimum locaux;
 18 minimum global:
 $SH((x_1, x_2)^*) = -186.7309$

Zakharov (Zn):



$Zn(x) = (\sum_{i=1}^n x_i^2) + (\sum_{i=1}^n 0.5ix_i)^2 + (\sum_{i=1}^n 0.5ix_i)^4$;
 espace de recherche: $-5 < x_i < 10$;
 plusieurs minimum locaux;
 1 minimum global: $x^* = (0, \dots, 0)$;
 $Zn(x^*) = 0$