The Synthesis Proper Orthogonalization Tracking method

A. Hébert

2024/07/22

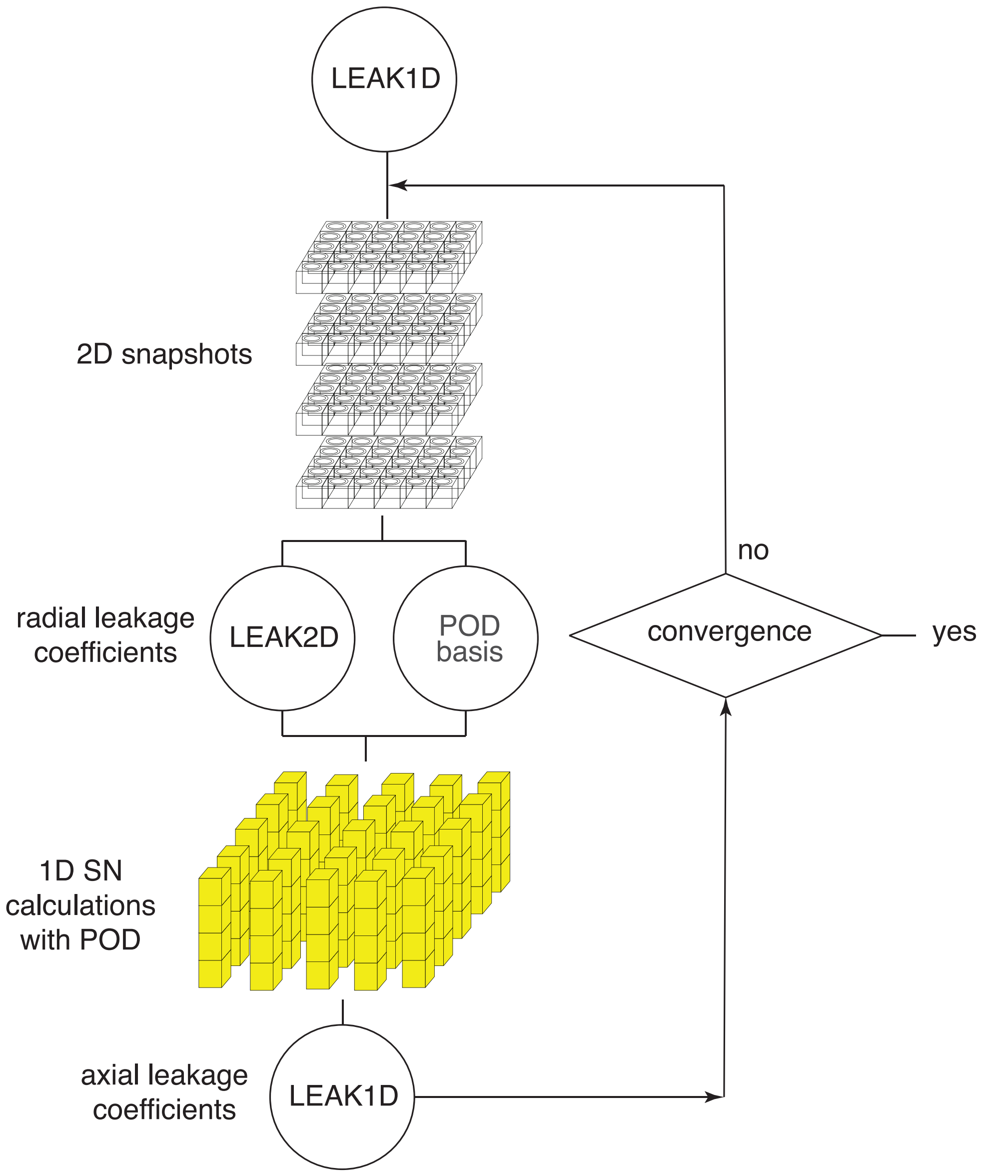
Table of contents

Introduction
Global SPOT iteration
Solution of the Boltzmann equation over a 2D slice
Axial solution of the Boltzmann equation
CLE-2000 implementation
DRAGON5 implementation
Ressources

Introduction

- Develop a 2D/1D synthesis method for lattice calculation over axially heterogeneous geometries
- Assume that all axial slices have the same location of elementary volumes and the same number of unknowns
- Use an existing solver (solution Door) with the method of discrete ordinates (SN) or the method of characteristics (MOC) in the radial direction
- Use a modified discrete ordinates solution in the axial direction
- Iterate between a 2D leakage distribution used in 1D calculations and a 1D leakage distribution used in 2D calculations

Global SPOT iteration



Solution of the Boltzmann equation over a 2D slice

- The 2D equation is:

$$\begin{aligned}\boldsymbol{\Omega}\cdot\nabla\phi_g(\boldsymbol{r},\boldsymbol{\Omega}_n)+\Sigma_g(\boldsymbol{r})\,\phi_g(\boldsymbol{r},\boldsymbol{\Omega}_n) &= \sum_{\ell=0}^{L}\frac{2\ell+1}{4\pi}\sum_{m=-\ell}^{\ell}R_\ell^m(\boldsymbol{\Omega}_n)\,\Sigma_{\mathrm{s},g\leftarrow g,\ell}(\boldsymbol{r})\,\phi_{g,\ell}^m(\boldsymbol{r})\\ &+ L_g^{\mathrm{1D}}(\boldsymbol{r})\,\phi_g(\boldsymbol{r},\boldsymbol{\Omega}_n)+Q_g^\diamond(\boldsymbol{r},\boldsymbol{\Omega}_n)\\ &\qquad ;\ 1\le g\le G \qquad (1)\end{aligned}$$

  where
  $\phi_g(\boldsymbol{r},\boldsymbol{\Omega}_n)$: discrete ordinates fluxes in group g,
  $\Sigma_g(\boldsymbol{r})$: macroscopic total cross sections in group g,
  $\Sigma_{\mathrm{s},g\leftarrow g,\ell}(\boldsymbol{r})$: Legendre moments of the macroscopic within-group scattering cross sections in group g,
  $\phi_{g,\ell}^m(\boldsymbol{r})$: spherical harmonics fluxes,
  $Q_g^\diamond(\boldsymbol{r},\boldsymbol{\Omega}_n)$: out-of-group and fission sources,
  $L_g^{\mathrm{1D}}(\boldsymbol{r})$: axial leakage coefficients,
  $R_\ell^m(\boldsymbol{\Omega}_n)$: real spherical harmonics.

- All G groups are solved independently. Source $Q_g^\diamond(\boldsymbol{r},\boldsymbol{\Omega}_n)$ is set before call to the solution Door and is not modified during the call.
- Solution of Eq. (1) requires GMRES-accelerated scattering iterations.
- Out of group and power iterations are done outside the solution Door.

Solution of the Boltzmann equation over a 2D slice

Radial leakage coefficients $L_g^{2\mathrm{D}}(\boldsymbol{r})$ stored in the LEAK2D data structure are obtained for each snapshot as

$$L_g^{2\mathrm{D}}(\boldsymbol{r}) = -\Sigma_g(\boldsymbol{r}) + \Sigma_{\mathrm{s},g\leftarrow g,0}(\boldsymbol{r}) - L_g^{1\mathrm{D}}(\boldsymbol{r}) + \frac{Q_g^{\diamond}(\boldsymbol{r})}{\phi_{g,0}^{0}(\boldsymbol{r})} \tag{2}$$

If the snapshot calculation is made in fundamental mode condition, the radial leakage coefficients sum up to zero over each energy group:

$$\langle L_g^{2\mathrm{D}} \phi_{g,0}^{0} \rangle = 0. \tag{3}$$

These leakage coefficients are assigned to each corresponding region of the "1D SN calculation with POD" based on Eq. (4).

Axial solution of the Boltzmann equation

- The modified 1D equation for each POD moment k is:

$$\mu_n \frac{\partial}{\partial z}\phi_g^{(k)}(z,\mu_n) + \sum_{\ell=1}^{K_g} \Sigma_g^{(k,\ell)}(z)\,\phi_g^{(\ell)}(z,\mu_n) = \sum_{m=0}^{L}\frac{2m+1}{2}P_m(\mu_n)\sum_{\ell=1}^{K_g}\Sigma_{\mathrm{s},g\leftarrow g,m}^{(k,\ell)}(z)\,\phi_{g,m}^{(\ell)}(z) + L_g^{\mathrm{2D}(k)}(z)\,\phi_g^{(k)}(z,\mu_n) + Q_g^{\diamond(k)}(z,\mu_n)$$
$$;\ 1\le k\le K_g \ \text{ and } \ 1\le g\le G \qquad (4)$$

where
$\phi_g^{(k)}(z,\mu_n)$: k-th POD moment of the discrete ordinates fluxes in group g,
$\Sigma_g^{(k,\ell)}(z)$: (k,ℓ)-th POD moment of the macroscopic total cross sections in group g,
$\Sigma_{\mathrm{s},g\leftarrow g,m}^{(k,\ell)}(z)$: (k,ℓ)-th POD moment of the Legendre moments of the macroscopic within-group scattering cross sections in group g,
$\phi_{g,m}^{(\ell)}(z)$: k-th POD moment and m-th Legendre moment fluxes,
$Q_g^{\diamond(k)}(z,\mu_n)$: k-th POD moment of the out-of-group and fission sources,
$L_g^{\mathrm{2D}(k)}(z)$: k-th POD moment of the axial leakage coefficients,
$P_m(\mu_n)$: Legendre polynomials.

Solution of the Boltzmann equation over a 2D slice

- The POD moments of cross sections and source terms are

$$\Sigma_g^{(k,\ell)}(z) = \sum_{i=1}^{N_{\rm reg}} U_{i,k}\, U_{i,\ell}\, \Sigma_{i,g}(z) \tag{5}$$

where $N_{\rm reg}$ is the number of regions in a 2D slice, and

$$\Sigma_{{\rm s},g\leftarrow g,m}^{(k,\ell)}(z) = \sum_{i=1}^{N_{\rm reg}} U_{i,k}\, U_{i,\ell}\, \Sigma_{{\rm s},i,g\leftarrow g,m}(z) \tag{6}$$

$$L_g^{{\rm 2D}(k)}(z) = \sum_{i=1}^{N_{\rm reg}} U_{i,k}\, L_{i,g}^{\rm 2D}(z) \tag{7}$$

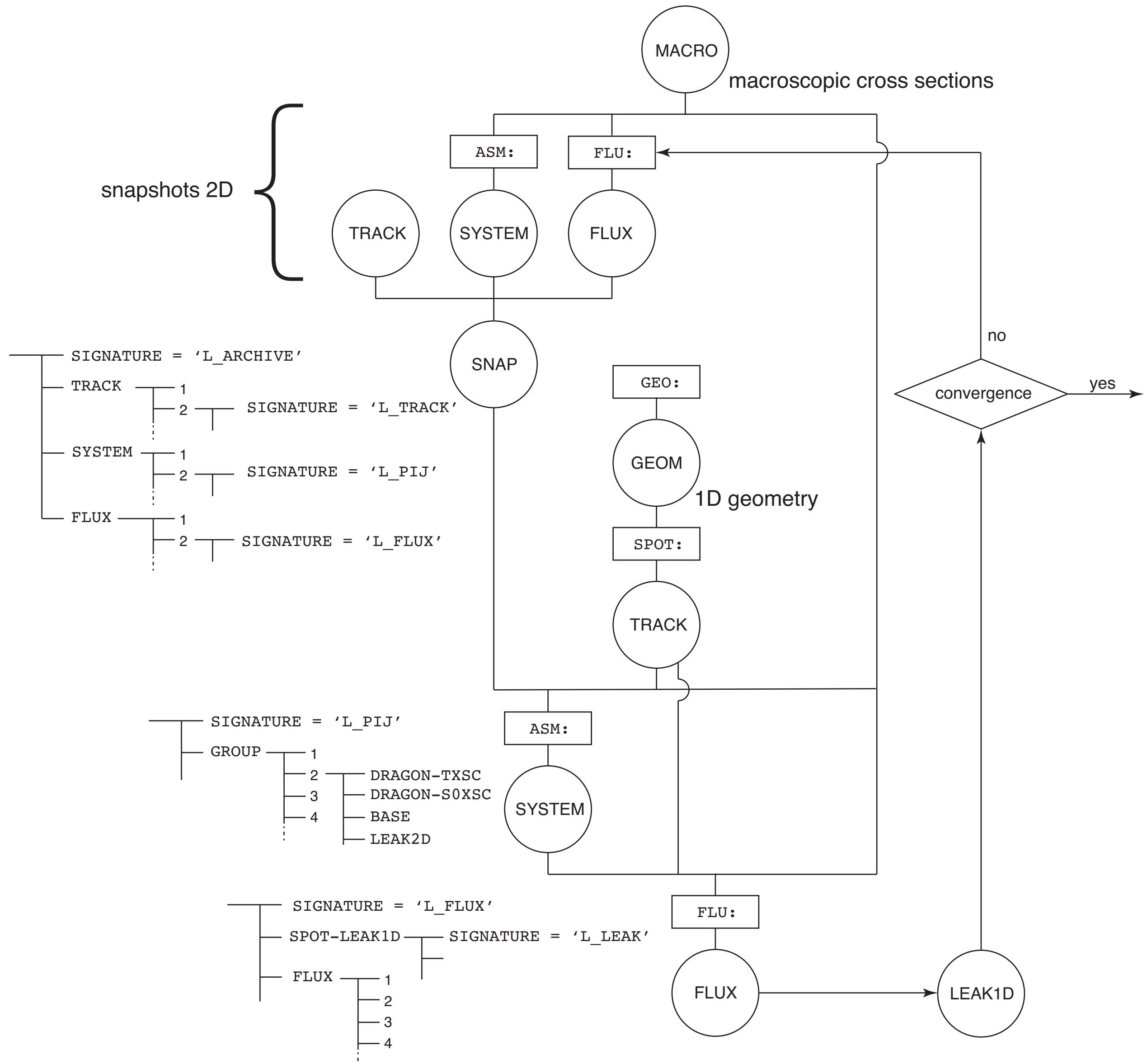
$$Q_g^{\diamond(k)}(z,\mu_n) = \sum_{i=1}^{N_{\rm reg}} U_{i,k}\, Q_{i,g}^{\diamond}(z,\mu_n) \tag{8}$$

- After solution of Eq. (4) has converged, axial leakage coefficients $L_g^{\rm 1D}(\boldsymbol{r})$ stored in the LEAK1D data structure are written in terms of interface net currents defined along the axial axis, obtained from the solution od Eq. (4):

$$L_{i,g}^{\rm 1D}(z_j) = \frac{1}{\langle \phi_{i,g}\rangle_{V_j}} \left[J_{i,g}(z_{j+1/2}) - J_{i,g}(z_{j-1/2}) \right] \tag{9}$$

where the axial mesh j is defined in the interval $z_{j-1/2} \le z \le z_{j+1/2}$.
At first iteration, $L_{i,g}^{\rm 1D}(z_j)$ are set to zero in Eq. (1) and are re-evaluated using Eq. (9) after each SPOT iteration.

CLE-2000 implementation



CLE-2000 implementation

A CLE-2000 procedure `SpotProc.c2m` was written to encapsulate the SPOT computational scheme. Snapshot 2D calculations are performed in fundamental mode conditions with `TYPE L`. Axial SPOT calculations are `TYPE K` cases.

SpotProc.c2m

```
****************************************************************
*                                                              *
* Procedure : SpotProc.c2m                                     *
* Purpose   : Synthetic Proper Orthogonal Tracking procedure   *
*             in fundamental mode condition                    *
* Author    : A. Hebert                                        *
*                                                              *
* CALL      :                                                  *
*  SPOFLX := SpotProc MACRO SNAP SPOTRK ;                      *
*                                                              *
*  Input objects:                                              *
*    SNAP     : archive containing snapshot radial trackings   *
*    MACRO    : macrolib                                       *
*    SPOTRK   : axial SPOT tracking                            *
*                                                              *
*  Output object:                                              *
*    SPOFLX   : axial SPOT fluxes                              *
*    SNAP     : archive containing snapshot assembly objects   *
*               and radial fluxes                              *
*                                                              *
****************************************************************
PARAMETER  SPOFLX SNAP MACRO SPOTRK ::
  ::: LINKED_LIST SPOFLX SNAP MACRO SPOTRK ; ;
LINKED_LIST TRACK SYSTEM FLUX LEAK1D ;
MODULE ASM: FLU: SPOT: BACKUP: RECOVER: GREP: DELETE: UTL: ABORT: END: ;
REAL eps := 1.0E-4 ; ! criterion for SPOT leakage convergence
*
INTEGER isnap nsnap ;
GREP: SNAP :: GETVAL 'LISTDIM' 1 >>nsnap<< ;
REAL errspo := 1.0 ;
INTEGER iter := 0 ;
WHILE  errspo eps > DO
  EVALUATE iter := iter 1 + ;
  ECHO "----------------------------------" ;
  ECHO "SpotProc: SPOT leakage iteration" iter ;
  ECHO "----------------------------------" ;
  EVALUATE isnap := 0 ;
```

CLE-2000 implementation

SpotProc.c2m (cont'n)

```
  WHILE isnap nsnap < DO
    EVALUATE isnap := isnap 1 + ;
    ECHO "----------------------------" ;
    ECHO "SpotProc: Process snapshot" isnap ;
    ECHO "----------------------------" ;
    TRACK := RECOVER: SNAP :: ITEM <<isnap>> ;
    IF iter 10 > THEN
      ECHO "SpotProc: maximum number of SPOT leakage iterations reached" ;
      ABORT: ;
    ELSEIF iter 1 = THEN
      SYSTEM := ASM: MACRO TRACK :: EDIT 2 ARM ;
      SNAP := BACKUP: SNAP SYSTEM :: ITEM <<isnap>> ;
      FLUX :=  FLU: MACRO TRACK SYSTEM ::
        EDIT 1 TYPE L P0 SIGS EXTE 200 1.0E-5 ;
    ELSE
      SYSTEM := RECOVER: SNAP :: ITEM <<isnap>> ;
      LEAK1D := SPOFLX :: STEP UP 'SPOT-LEAK1D' STEP AT <<isnap>> ;
      SYSTEM := SYSTEM LEAK1D ;
      SYSTEM := UTL: SYSTEM :: DIR CREA STATE-VECTOR 14 14 = 1 ;
      FLUX := RECOVER: SNAP :: ITEM <<isnap>> ;
      FLUX :=  FLU: FLUX MACRO TRACK SYSTEM ::
        EDIT 1 TYPE L P0 SIGS EXTE 200 1.0E-5 ;
      LEAK1D := DELETE: LEAK1D ;
    ENDIF ;
    SNAP := BACKUP: SNAP FLUX :: ITEM <<isnap>> ;
    TRACK SYSTEM FLUX := DELETE: TRACK SYSTEM FLUX ;
  ENDWHILE ;
  *----
  * Solve the axial SPOT system
  *----
  SYSTEM := ASM: MACRO SPOTRK SNAP :: EDIT 2 ;
  IF iter 1 = THEN
    SPOFLX :=  FLU: MACRO SPOTRK SYSTEM ::
      EDIT 2 TYPE K P0 SIGS EXTE 100 5.0E-8 SPOT >>errspo<< ;
  ELSE
    SPOFLX :=  FLU: SPOFLX MACRO SPOTRK SYSTEM ::
      EDIT 2 TYPE K P0 SIGS EXTE 100 5.0E-8 SPOT >>errspo<< ;
  ENDIF ;
  SYSTEM := DELETE: SYSTEM ;
  ECHO "SpotProc: LEAK1D error step at iteration" iter "=" errspo ;
ENDWHILE ;
ECHO "-------------------------------------------------" ;
ECHO "SpotProc: SPOT converged in" iter "leakage iterations" ;
ECHO "-------------------------------------------------" ;
END: ;
```

DRAGON5 implementation

The following DRAGON5 subroutines are modified (M) or added (?):

```
M        ASM.f       call SPOASM
M        B1HOM.f     include SPOT leakage from SPOBAL in BN leakage model
M        DOORFV.f    call SPOF
M        FLU.f       |
M        FLUDRV.f    |  pass PICK,ISPOT variables
M        FLUGPT.f    |
M        FLU2DR.f    call SPOSOU and SPODB2
M        FLUBAL.f    include SPOT leakage in rebalancing factors
M        FLUGPI.f    read SPOT keyword in module FLU:
M        KDRDRV.F    call SPOT
?        SPOASM.f    |
?        SPOBAL.f    |
?        SPODB2.f    |
?        SPOF.f      |
?        SPOFLV.f    | new SPOT-related subroutines
?        SPOMRE.f    |
?        SPOSOU.f    |
?        SPOT.f      |
?        SPOT1P.f    |
```

DRAGON5 implementation

New subroutines were added in DRAGON5 to support the SPOT computational scheme:

`tracking`

SPOT.f This is the tracking operator. Perform unknown ordering for the SPOT method.

module `ASM:`

SPOASM.f Compute radial leakage in eash snapshot, save it in the `pij` LCM object and compute the reduce order (SVD) orthogonal base in each energy group.

module `FLU:`

SPOBAL.f Compute the volume-integrated radial or axial leakage cross section for the SPOT method.

SPODB2.f Compute axial flux and axial leakage cross sections in each snapshot for the SPOT method, save it in the `flux` LCM object and save the axial leakage accuracy in a CLE-2000 variable.

SPOSOU.f Compute the source density for the radial or axial SPOT leakage.

SPOF.f Flux vectorial door for the SPOT method. Call subroutine `SPOFLC`.

SPOFLV.f Solve the SPOT equations for obtaining axial fluxes over the 3D domain for a set of `NGEFF` energy groups.

SPOMRE.f GMRES acceleration of the SPOT equations.

SPOT1P.f Solution of the SN equations for the SPOT method using Proper Orthogonal Decomposition (POD).

Ressources

- Academic:
  Alain Hébert (alain.hebert@polymtl.ca)
- Merlin website:
  DRAGON5/DONJON5: `http://merlin.polymtl.ca`
- Archives website:
  - Access to Dragon5-related information
  - Academic contributions

  `http://merlin.polymtl.ca/archives.htm`
- SPOT initial prototype:

```
curl -O http://merlin.polymtl.ca/downloads/archive_Version5_spot_ev3500.tgz
```

- Textbook:
  A. Hébert, Applied Reactor Physics, Third Edition, PIP, 2020.
- Seminars (in French)
  Moodle site