On the generation of collision-probability matrices with Serpent 2

Serpent User Group Meeting 2018
Espoo, Finland
Eric Dorval
VTT Technical Research Centre of Finland Ltd

Overview

- Motivation and overview of the CP method.
- Generation of collision probability matrices in Serpent 2.
- CP matrix structure – examples.
- Application of CP matrices to simplified problems.
- Future prospects.

Motivation

Motivation

- To develop a(nother) low-order neutronics solver in the Kraken framework.
- Exploit the unique capabilities of MC techniques for tracking in complex geometries.
- Relax some of the assumptions of the collision-probability method:
  - Flat source approximation
  - Isotropic neutron sources
  - Isotropic/cosine angular currents
- Combine the CP-method with homogenized XS data generated by Serpent $\rightarrow$ drastically reduce the number of energy groups.
- Decompose the problem into nodes coupled by boundary currents. Solve one CP problem inside every node.
- Explore the application of 3-D CP-methods to, e.g., ex-core radiation transport.

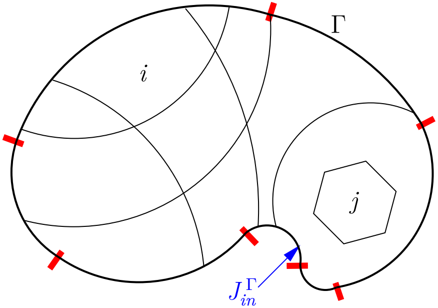
An overview of the CP method

The Collision Probability method

- Partition $V = \{V_1, V_2, \ldots, V_i, \ldots, V_N\}$; and Γ
- Evaluate collision probability matrices:
  - P : *first-flight* collision probabilities
  - E : *escape* collision probabilities
  - G : cell *opacity*
  - T : surface segment *transmission* (or*transparency*)
- Solve the systems:

$$\Sigma^i_{\mathrm{t},g}\Phi^i_g V_i = \sum_{\forall j}\left[P_{j\to i} Q^g_{\mathrm{t},g} V_j\right] + \sum_{\forall s}\left[G_{s\to i} J^{g,s}_{in} A_s\right]$$

$$J^{g,s}_{out} A_s = \sum_{\forall j}\left[E_{j\to s} Q^g_{\mathrm{t},g} V_j\right] + \sum_{\forall s'}\left[T_{s'\to s} J^{g,s'}_{in} A_{s'}\right]$$



- $Q^g_{\mathrm{t},g} = Q^g_{\mathrm{ext},g} + Q^g_{\mathrm{fis},g} + Q^g_{\mathrm{down},g} + \Sigma^i_{\mathrm{s},g\to g}\Phi^i_g + Q^g_{\mathrm{up},g}$
- Can replace $\Phi^i_g V_i$ and $J^{g,s}_{in} A_s$ by volume- and surface-integrated values

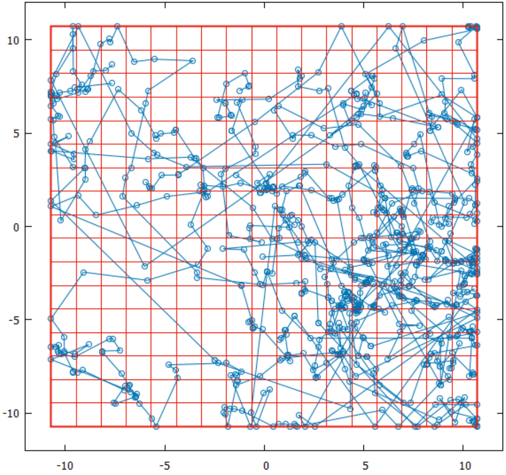
Implementation in Serpent

Implementation in Serpent

- Implemented in Serpent version 2.1.29.
- Scores various arrays upon source, (physical) collision and outer boundary crossing events.
- Enforces

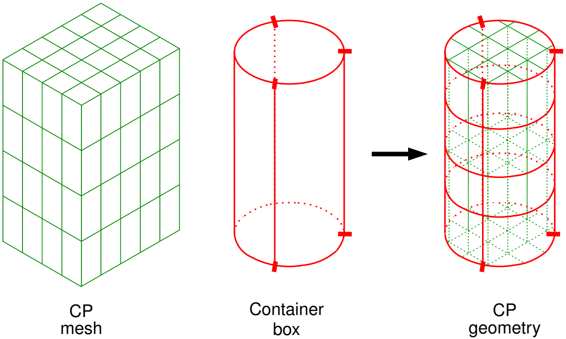
```
DATA_STOP_AT_BOUNDARY = YES
```

- Only works with *reflective* boundary conditions.
- For depletion problems, generates CP matrices (P, E, G, and T) per burnup step.
- Data output in Octave/MatLab format.
- Face numbering for efficient coupling in second step (outside solver).
- About geometry...



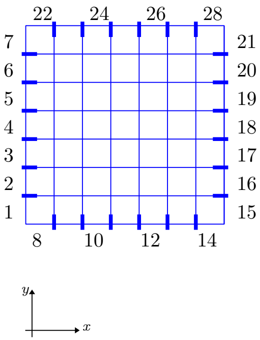
CP – geometry

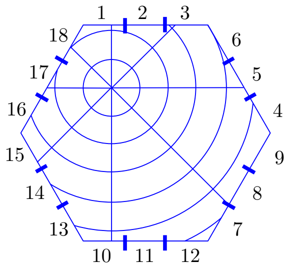
- Combines a CP mesh (V) and a container box (Γ).
- CP mesh used for scoring all P elements, domain of E, image of G.
- “Extended” container box for scoring all T elements, domain of G, image of E.
- Axial discretization from the CP mesh.
- Radial *face* discretization from container box.
- Top and bottom face discretization given by the CP mesh.



Radial segment numbering

- Per axial level, the discretization follows the container box **SEGFACE** value, **not** the one of the CP mesh.
- Face sweeping order is given by DFPos().
- Sweeping order per mesh is related to Cartesian coordinates.
- Top and bottom faces (not shown) follow the –shifted– numbering of the CP meshes, as in standard mesh detectors.





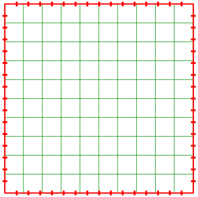
cpmat cards

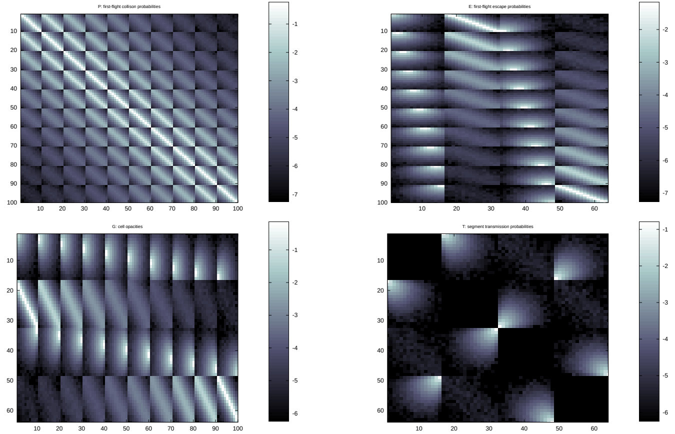
```
cpmat   NAME   SURF   FACESEG
             [ PART ]
             [ dx XMIN XMAX NX ]
             [ dy YMIN YMAX NY ]
             [ dz ZMIN ZMAX NZ ]
             [ dn  -1  MIN1 MAX1 N1 MIN2 MAX2 N2 MIN3 MAX3 N3 ]
             [ dh TYPE X0 Y0 PITCH N1 N2 ZMIN ZMAX NZ ]
             [ 2d ]
```

- Do not use periodic BCs.
- Group structure is taken from `set nfg` card.
- Container box (`SURF`) needs not be a part of the geometry, but consistent with the `2d` radial flag.
- Good practice: CP mesh fully contained within the (whole) problem geometry.

CP matrix structure – examples

Example 1 – 2-D: all in order





P, E ↗

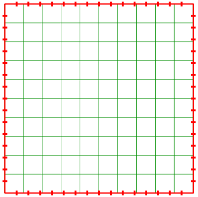
G, T →

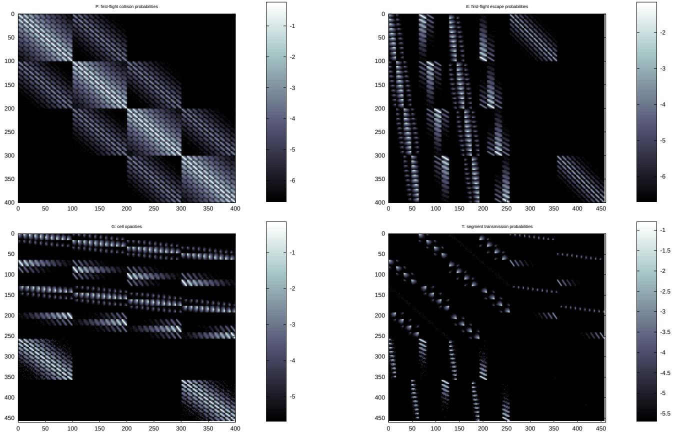
`cpmat` cards in Serpent ⟶

```
cpmat  ex_1   115                 16
       dx    -10.71      10.71   10
       dy    -10.71      10.71   10
       dz      0.0      365.76    1
       2d     yes

surf   115    rect    -10.71   10.71   -10.71   10.71
```

Example 2 – 3-D: all in order





P, E ↗

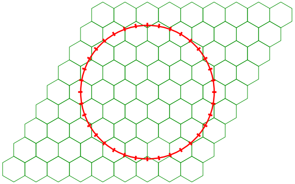
G, T →

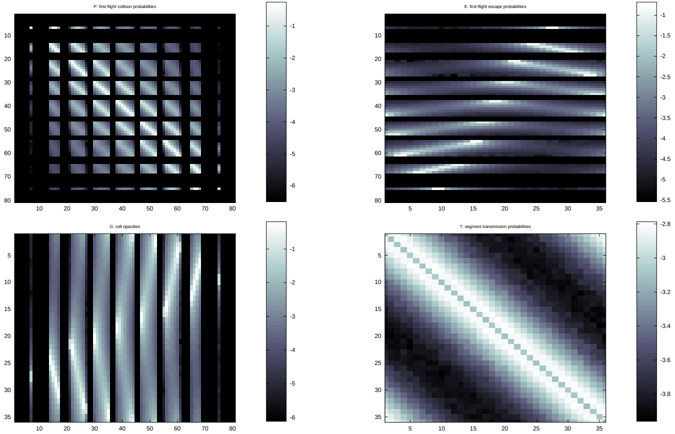
`cpmat` cards in Serpent ⟶

```
cpmat  ex_2   116                    16
       dx    -10.71      10.71   10
       dy    -10.71      10.71   10
       dz     0.0       365.76    4
       2d     no

surf   116    cuboid      -10.71    10.71    -10.71   10.71     0.0   365.76
```

Example 3 – 2-D: cylindrical container box





P, E ↗
G, T →

`cpmat` cards in Serpent ⟶

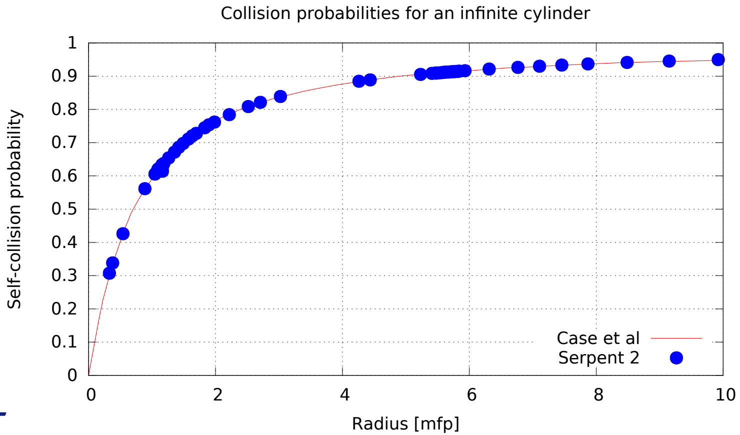
```
cpmat  ex_3   106    36
       dh     2  0.0 0.0   2.0    9    9    0.0  365.76   1
       2d     yes

surf   106    cyl    0.0   0.0    6.000
```

Verification case: infinite cylinder

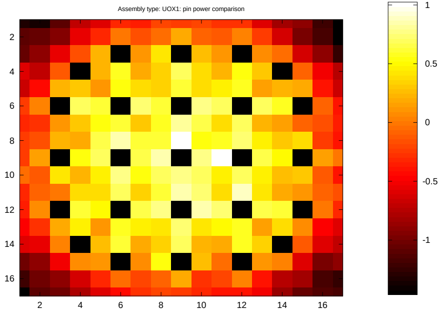
Infinite cylinder

- Infinite cylinder with uniform, isotropic source with 1 MeV neutrons.
- Serpent input for a water region. Data generated in 40 energy groups.



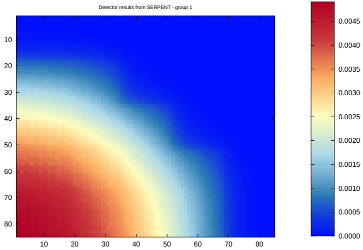
Example cases in Cartesian geometry

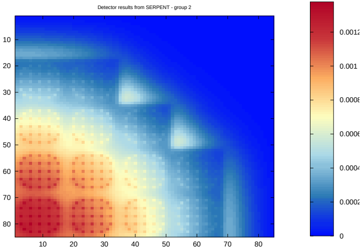
UOX1 PWR assembly from KAERI benchmark 1B

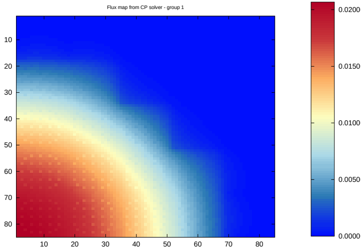


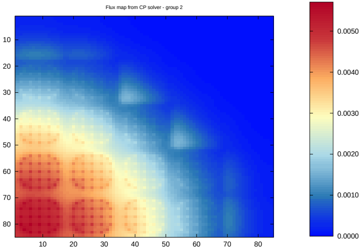
- XS and CP data generated in two energy groups for 2 pin types:
  - guide tubes
  - UOX 2.2 %
- $\Delta_{\rho} = 3$ pcm
- Maximum pin power difference against Serpent: 1.5 %

Simplified version of the KAERI benchmark 1B problem: fluxes

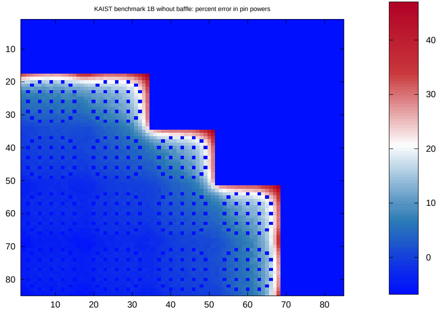








Simplified KAERI benchmark 1B problem: general



- Much to be improved:
  - $\Delta_\rho = 1250$ pcm
  - Maximum difference in pin-powers with Serpent: 47 %
- Other cases suggest that the problem is in the reflectors
- Possible solutions:
  - Model reflector XS data with spatial variations
  - Increase energy resolution
  - Replace reflectors by segment-wise albedos (available from CP generation routines)
  - Application of equivalence through SPH factors

Ongoing and future work

Ongoing and future work

- Support for binary print of CP data in Serpent.
- Development of a comprehensive, parallel, CP-based solver for fixed/adjoint source and eigenvalue calculations.
- Implementation of CP matrix perturbation models for SPH factor computations.
- Development of Coarse Mesh Collision Probability (CMCP) acceleration scheme.
- Testing on shielding problems.

www.vttresearch.com
#vttpeople / @VTTFinland