The Abdus Salam
International Centre for Theoretical Physics

United Nations
Educational, Scientific and
Cultural Organization

IAEA
International Atomic Energy Agency

1858-19

School on Physics, Technology and Applications of Accelerator Driven Systems (ADS)

19 - 30 November 2007

ADS Dynamics
"Accelerator-Driven System Dynamics - MACISTE Multi Amplitude Code In Subcritical Transient Evaluation"

Piero RAVETTO
Politecnico di Torino, Dipt. di Energetica
10129 Torino
Italy

MACISTE

Multi Amplitude Code In Subcritical Transient Evaluation

Introduction

MACISTE was developed to solve the time-dependent multi-group diffusion equations in a 1D slab system. It allows the comparison of the different approximations of dynamics neutron equations:

- full inversion of the problem (SK): it is considered as the benchmark solution;
- point kinetics (PK): it is the conventional approach used for the dynamics;
- multipoint kinetics (MPK): in this case the phase space is divided into region and each of them evolves with a different amplitude.

The code is self-consistent, since it provides the steady-state solution. Both the direct and the adjoint problems are solved and from these shapes the kinetics parameters for PK and coupling coefficient for MPK are computed. An eigenvalue tool is implemented to allow the calculation of the variation of the reactivity and to adjust the initial subcriticality.

Input

As a input, the code wants several files. In the following a description of each of them is provided.

‘input_mpk.txt’ : defines the name of the other files in which there are the properties of the system. It is recommended not to change this file.

‘geometry.txt’ : is used to define both the multi-slab geometry (width in cm) and the spatial discretization (number of mesh per region). The last line refers to the boundary condition, which can be either of vacuum or reflection.

‘nucl_propr.txt’ : here nuclear properties are defined; the last line refers to the value of *keff* desired by the user (i.e. fission cross section is adjust to being such that the system has the right multiplication constant).

‘source.txt’ : the region in which there is the source neutron emission and the intensity (which is recommended to be one at the beginning of the transient).

‘perturbation.txt’ : defines the cause of the transient, which may be a variation of source intensity or the perturbation of a cross section in a region. Firstly, the spatial behaviour of the induced perturbation must be introduced (it can be a step or a ramp); secondly, the source intensity is set (if ‘/’ no change). Finally the material perturbation is defined: after having written the region in which it is localized (in cm), the value must be introduced as a ratio of the perturbation over the initial quantity.

‘dynamics.txt’ : set the parameter for the time dependent calculation. The first number refers to the length of the transient, whereas the second to the time step. In the next section there is the definition of the detector (1,2,3 are suggested; if 2 is chosen, pay attention that each MPK zone contains a part with fission cross section different from zero). Then the time mesh is specified (i.e. the number of time steps each interval defined in ‘perturbation.txt’). The next input required refers to the choice of a convex division of the phase space: is 1 is typed, the MPK region will be such that the perturbation evolves separated from the remaining system. As an alternative, one can introduce directly the division in cm. Lastly, there is the possibility of improving the results of the approximated dynamics with the recalculation of the shape on a coarser mesh (IQS).

Output

Outputs are organized in two folder, one related to the stationary calculation and the other to the time-dependend solution. The ‘dinamica’ folder contain a file, named ‘OpzioniSimulazione.txt’, which contain the parameter of simulation used. During the execution, on the command windows will appear some data of interest of the dynamics, like the reactivity of the system after the perturbation.

Test cases

Two test cases are provided, whose input are in ‘InputA’ and ‘InputB’. The procedure to run case A is the following:

- copy the folder of the ‘InputA’ and rename it as ‘Input’;
- run the code from the command windows;
- read the output and plot the results (for instance using MATLAT or Excel).

The system under analysis is a 1D multislab three-group Masurca-like reactor.

<u>Case A</u>
‘InputA’ refers to a case in which an intensity variation of the source emission is introduced. As an example, a switch-off is considered. Two points are considered, source plus driver.

Exercise:

a) analyse the results of the code: are there significant differences between PK and MPK?
b) Try to simulate an oscillation of the source (hint: change file ‘perturbation.txt’ introducing other intervals).
c) Introduce an increase of reactivity (operating either on fission or on absorption cross section) and reduce the rise of the power by operating on the source. Control the final reactivity of the system.

<u>Case B</u>
‘InputB’ refers to a case in which there is a variation of fission cross section in a localized region. The phase space division is made such that the perturbed region is separated form the rest of the system (3 MPK regions).

Exercise:

a) analyse the results of the code: which between MPK and PK are conservative compared to SK?
b) Plot the shape of the flux computed with SK, PK, MPK.

c) Try different borders of spatial subdivision. Plot the MPK results on a graphs to see the influence.
d) Change the number of regions: 4 instead of 3. Decide which might be the best option;
e) Different intensity of the cross section perturbation. Decide the energy subdivision depending on the perturbation;
f) Update of the shape: IQS. Compare with the case in which there is no IQS.
g) Change the number of recalculations of the shape and plot the curves on a graphs to see the influence of recalculation on the accuracy of the solution. Keep record of the calculation time.

Case C

'InputC' refers to a source switch-off followed by a .

Exercise:

a) analyse the results of the code: are there significant differences between PK and MPK?
b) Try different switch-off periods (in general: the longer the better PK performs!).