Lecture Notes for the Course on
NUMERICAL METHODS FOR NUCLEAR REACTORS

Prof. WALTER AMBROSINI
University of Pisa, Italy

Unit-4 - Problems of Dynamics of Fission Reactors

NOTICE: These notes were prepared by Prof. Ambrosini mainly **on the basis of the material adopted by Prof. Bruno Montagnini for the lectures he held up to years ago**, when he left to Prof. Ambrosini the charge of the course held for the Degree in Nuclear Engineering at the University of Pisa. This material is freely distributed to Course attendees or to anyone else requesting it. It has not the worth of a textbook and it is not intended to be an official publication. It was conceived as the notes that the teacher himself would take of his own lectures in the paradoxical case he could be both teacher and student at the same time (sometimes space and time stretch and fold in strange ways). It is also used as slides to be projected during lectures to assure a minimum of uniform, constant quality lecturing, regardless of the teacher's good and bad days. As such, the material contains reference to classical textbooks and material whose direct reading is warmly recommended to students for a more accurate understanding. In the attempt to make these notes as original as feasible and reasonable, considering their purely educational purpose, most of the material has been completely re-interpreted in the teacher's own view and personal preferences about notation. In this effort, errors in details may have been introduced which will be promptly corrected in further versions after discovery. Requests of clarification, suggestions, complaints or even sharp judgements in relation to this material can be directly addressed to Prof. Ambrosini at the e-mail address: walter.ambrosini@ing.unipi.it

MULTIGROUP NEUTRON KINETICS EQUATIONS

- **We consider the G energy group neutron kinetic equations with N_d delayed neutron groups:**

$$\frac{1}{v_g}\frac{\partial \phi_g}{\partial t} = div\, D_g(\vec{r})\, grad_{\vec{r}}\phi_g(\vec{r},t) - \Sigma_{r,g}(\vec{r})\phi_g(\vec{r},t) + \sum_{g'<g}\Sigma_{s,g'\to g}(\vec{r})\phi_{g'}(\vec{r},t)$$

$$+(1-\beta)\chi_g^p\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r})\phi_{g'}(\vec{r},t) + \sum_{d=1}^{N_d}\lambda_d\chi_g^d C_d(\vec{r},t) \quad (\vec{r}\in V,\ g=1,...,G)$$

$$\frac{\partial C_d}{\partial t} = -\lambda_d C_d(\vec{r},t) + \beta_d\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r})\phi_{g'}(\vec{r},t) \quad (\vec{r}\in V,\ d=1,...,N_d)$$

The initial and boundary conditions can be, e.g.:

$$\phi_g(\vec{r},t)=0 \qquad C_d(\vec{r},t)=0 \qquad (\vec{r}\in \partial V)$$

$$\phi_g(\vec{r},0)=\phi_g^0(\vec{r}) \qquad C_d(\vec{r},0)=C_d^0(\vec{r}) \qquad (\vec{r}\in V)$$

- **With respect to the eigenvalue calculations, *it is now necessary to discretise also with respect to time.* In this purpose we can write the equations in the following general form**

$$\frac{\partial \boldsymbol{\psi}}{\partial t} = \mathbf{A}\,\boldsymbol{\psi}$$

Time discretisation gives:

- ***explicit scheme***

$$\frac{\boldsymbol{\psi}^{n+1}-\boldsymbol{\psi}^n}{\Delta t} = \mathbf{A}\,\boldsymbol{\psi}^n \quad \Rightarrow \quad \boldsymbol{\psi}^{n+1} = (\mathbf{I}+\Delta t\,\mathbf{A})\boldsymbol{\psi}^n$$

- ***implicit scheme***

$$\frac{\boldsymbol{\psi}^{n+1}-\boldsymbol{\psi}^n}{\Delta t} = \mathbf{A}\,\boldsymbol{\psi}^{n+1} \quad \Rightarrow \quad (\mathbf{I}-\Delta t\,\mathbf{A})\boldsymbol{\psi}^{n+1} = \underline{\boldsymbol{\psi}}^n$$

- ***weighted scheme*** ($0\le\theta\le 1$)

$$\frac{\boldsymbol{\psi}^{n+1}-\boldsymbol{\psi}^n}{\Delta t} = \mathbf{A}\left[(1-\theta)\boldsymbol{\psi}^n + \theta\,\boldsymbol{\psi}^{n+1}\right] \quad \Rightarrow \quad (\mathbf{I}-\theta\Delta t\,\mathbf{A})\boldsymbol{\psi}^{n+1} = \left[\mathbf{I}+(1-\theta)\Delta t\,\mathbf{A}\right]\boldsymbol{\psi}^n$$

These relationships must be spatially discretised discussing some of the properties of the obtained schemes

CONSISTENCY, STABILITY AND CONVERGENCE OF FINITE DIFFERENCE METHOS

Space and time discretisation

While discretising a partial differential equation, we substitute to a given differential problem a corresponding problem expressed in terms of finite differences (no matter the adopted discretisation scheme, i.e., finite differences, finite volumes or finite elements):

$$\textbf{PDE} \rightarrow \textbf{FDE}$$

(PDE = Partial Differential Equation; FDE = Finite Difference Equations)

It is then necessary to assure that the obtained numerical scheme possesses some fundamental properties making it mathematically sound and useful in practice.

For instance, we saw that the neutron kinetic equations, discretised with respect to energy, have the general form:

$$\frac{\partial \psi}{\partial t} = A\,\psi$$

where A **e** ψ **in our case represent a linear (matrix) differential operator and a vector function. We will omit their explicit indication in that form, for the sake of generality.**

If we proceed by discretising with respect to time and to each space coordinate, we have:

$$B_1\,\psi^{n+1} = B_0\,\psi^n$$

where B_1 **e** B_0 **are** ***finite difference linear operators depending, among the other variables, on the adopted space and time increments***

$$B_0 = B_0(\Delta t, \Delta x, \Delta y, \Delta z, \ldots) \qquad B_1 = B_1(\Delta t, \Delta x, \Delta y, \Delta z, \ldots).$$

We have to assume that the inverse operator of B_1 **does exist: if not it would be impossible to advance the calculation; so, any reasonable scheme should satisfy this condition. In such a case, it is**

$$\psi^{n+1} = B_1^{-1} B_0\,\psi^n = C\,\psi^n$$

where we defined $C = B_1^{-1} B_0 = C(\Delta t, \Delta x, \Delta y, \Delta z, \ldots)$**.**

Though this is not strictly mandatory, it is customary in some treatments (see e.g., Richtmyer and Morton, 1967) to assume that each space increments are related to the time increments in order to establish the speed at which they will vanish for vanishing Δt :

$$\Delta x = g_x(\Delta t) \qquad \Delta y = g_y(\Delta t) \qquad \Delta z = g_z(\Delta t)$$

This allows to assume that

$$\lim_{\Delta t \to 0} \Delta x = \lim_{\Delta t \to 0} g_x(\Delta t) = 0 \qquad \lim_{\Delta t \to 0} \Delta y = \lim_{\Delta t \to 0} g_y(\Delta t) = 0$$

$$\lim_{\Delta t \to 0} \Delta z = \lim_{\Delta t \to 0} g_z(\Delta t) = 0$$

So, we can use the notation:

$$C = C\left(\Delta t, g_x(\Delta t), g_y(\Delta t), g_z(\Delta t)\right) = C(\Delta t)$$

Convergence

Assuming to calculate the numerical solution of algebraic equations without *"round-off"* errors, the *discretization error* at the time level *n* with respect to the exact solution ψ_e^n is defined as:

$$\delta^n = \psi_e^n - \psi^n$$

The *total error*, instead, will include also round-off.

An approximate solution can be obviously considered reliable if δ^n keeps limited and can be reduced below a selected threshold by an appropriate choice of the increments.

Therefore, by selecting a time $\bar{t}$ at which we wish to advance the calculation and any desired n such that $n\Delta t \le \bar{t}$, it must be:

$$\lim_{\Delta t \to 0} \left\| \delta^n \right\| = 0$$

where $\|\circ\|$ is an appropriate *norm* operating on the functional domain of functions ψ (e.g.,: $\sqrt{\int_a^b |f(x)|^2 dx}$). This can be expressed by saying that:

a numerical scheme is said to be "convergent"   ***if its solution tends to the exact solution of the differential problem***   ***in the limit of small time and space increments***

The error δ^n is determined by the approximation adopted for evaluating the differential operator (replaced by the difference

operator), i.e. from *truncation*, and by the *propagation of the error affecting the data at the previous step*. In fact:

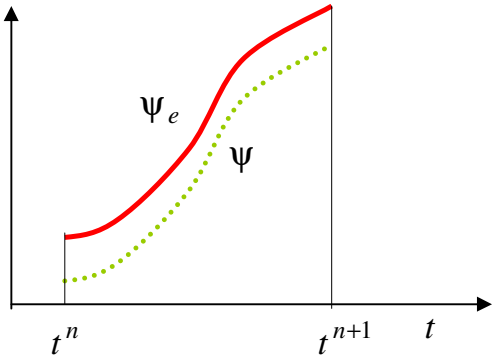
$$\delta^n = \psi_e^n - C(\Delta t)\psi^{n-1} = \psi_e^n - C(\Delta t)\psi_e^{n-1} + C(\Delta t)\psi_e^{n-1} - C(\Delta t)\psi^{n-1}$$

or

$$\delta^n = \underbrace{\psi_e^n - C(\Delta t)\psi_e^{n-1}}_{\text{error due to truncation}} + \underbrace{C(\Delta t)\delta^{n-1}}_{\substack{\text{propagation of the error} \\ \text{at the previous time step}}}$$

The partition of the contributions to errors into two terms brings to the consideration of two properties strictly related with convergence:

- ***consistency*, expressing at what extent the difference operator is a reasonable approximation of the differential one**
- ***stability*, defining the necessary and/or sufficient conditions in order to avoid excessive propagation of the error encountered at a given advancement step**



Consistency

Considering the relation

$$\psi^{n+1} = C(\Delta t)\psi^n$$

it can be noted this is equivalent to writing

$$\frac{\psi^{n+1} - \psi^n}{\Delta t} = \frac{C(\Delta t) - I}{\Delta t}\psi^n$$

where I is the identity operator. Comparing the above relation with the differential equation

$$\frac{\partial \psi}{\partial t} = A\psi$$

and considering that it is

$$\frac{\psi^{n+1} - \psi^n}{\Delta t} \to \frac{\partial \psi}{\partial t} \quad per\ \Delta t \to 0$$

it must be expected that it should be also

$$\frac{C(\Delta t) - I}{\Delta t}\psi(t) \to A\psi(t) \quad per\ \ \Delta t \to 0 \quad t \le \bar{t}$$

In other words, we need that the assigned difference equation be consistent with the given differential equation

$$\frac{\psi(t+\Delta t)-\psi(t)}{\Delta t}-\frac{C(\Delta t)-I}{\Delta t}\psi(t)\rightarrow\frac{\partial\psi(t)}{\partial t}-A\psi(t)\ \ per\ \ \Delta t\rightarrow 0,\ t\leq\bar{t}$$

This can be expressed as follows:

a numerical scheme is said to be "consistent" with the assigned differential problem if its difference equations tend to those of the differential problem in the limit of small increments

This means:

$$\lim_{\Delta t\rightarrow 0} FDE = PDE$$

(remember that we have assumed that if the time increment vanishes, the same will be for the spatial ones).

Consistency is therefore a property pertaining to the form of the numerical scheme.

Every method allowing transforming the differential problem into a corresponding difference problem that is consistent with it is said a "*discretization method*".

The difference TE=FDE-PDE is said *truncation error*. In particular, if ψ_e is the exact solution of the assigned original differential problem for some initial condition, the *local truncation error* is sometimes introduced *("local TE")* in the form

$$LTE_{n+1}=\frac{\psi_e^{n+1}-\psi_e^n}{\Delta t}-\frac{C(\Delta t)-I}{\Delta t}\psi_e^n-\underbrace{\left(\frac{\partial\psi_e}{\partial t}-A\psi_e\right)}_{=0}=\frac{\psi_e^{n+1}-C(\Delta t)\psi_e^n}{\Delta t}$$

being obtained by applying the difference scheme to ψ_e. It must be recognized that in literature different definitions of the LTE are also available with respect to the one preferred here.

Developing the difference equation by the use of the Taylor expansion it is possible to obtain an expression of the truncation error as a function of powers of the space and time increments.

Example: The dimensionless diffusion equation

$$\frac{\partial \psi}{\partial t} - \frac{\partial^2 \psi}{\partial x^2} = 0$$

is approximated as (*i* indicates spatial location and *n* the time level)

$$\frac{\psi_i^{n+1} - \psi_i^n}{\Delta t} - \frac{\psi_{i+1}^n - 2\psi_i^n + \psi_{i-1}^n}{\Delta x^2} = 0$$

Considering that

$$\psi_i^{n+1} = \psi_i^n + \left.\frac{\partial \psi}{\partial t}\right|_i^n \Delta t + \left.\frac{\partial^2 \psi}{\partial t^2}\right|_i^n \frac{(\Delta t)^2}{2} + \left.\frac{\partial^3 \psi}{\partial t^3}\right|_i^n \frac{(\Delta t)^3}{6} + \left.\frac{\partial^4 \psi}{\partial t^4}\right|_i^n \frac{(\Delta t)^4}{24} + \ldots$$

$$\psi_{i\pm1}^n = \psi_i^n \pm \left.\frac{\partial \psi}{\partial x}\right|_i^n \Delta x + \left.\frac{\partial^2 \psi}{\partial x^2}\right|_i^n \frac{(\Delta x)^2}{2} \pm \left.\frac{\partial^3 \psi}{\partial x^3}\right|_i^n \frac{(\Delta x)^3}{6} + \left.\frac{\partial^4 \psi}{\partial x^4}\right|_i^n \frac{(\Delta x)^4}{24} + \ldots$$

it is

$$\frac{\psi_i^{n+1} - \psi_i^n}{\Delta t} = \left.\frac{\partial \psi}{\partial t}\right|_i^n + \left.\frac{\partial^2 \psi}{\partial t^2}\right|_i^n \frac{\Delta t}{2} + \left.\frac{\partial^3 \psi}{\partial t^3}\right|_i^n \frac{(\Delta t)^2}{6} + \left.\frac{\partial^4 \psi}{\partial t^4}\right|_i^n \frac{(\Delta t)^3}{24} + \ldots$$

$$\frac{\psi_{i+1}^n - 2\psi_i^n + \psi_{i-1}^n}{\Delta x^2} = \left.\frac{\partial^2 \psi}{\partial x^2}\right|_i^n + \left.\frac{\partial^4 \psi}{\partial x^4}\right|_i^n \frac{(\Delta x)^2}{12} + \ldots$$

Then, substituting into the difference equation, we get:

$$\left.\frac{\partial \psi}{\partial t}\right|_i^n - \left.\frac{\partial^2 \psi}{\partial x^2}\right|_i^n + \left.\frac{\partial^2 \psi}{\partial t^2}\right|_i^n \frac{\Delta t}{2} - \left.\frac{\partial^4 \psi}{\partial x^4}\right|_i^n \frac{(\Delta x)^2}{12} + \ldots = 0$$

or

$$\left.\frac{\partial \psi}{\partial t}\right|_i^n - \left.\frac{\partial^2 \psi}{\partial x^2}\right|_i^n = O(\Delta t) + O\left(\Delta x^2\right)$$

It can be noted that:

- **the scheme is consistent, since the truncation error vanishes for vanishing increments (even if we put $\Delta x = r\,\Delta t$)**
- **a *first order* truncation error in Δt and a *second order* truncation error in Δx are obtained**
- **as a consequence of the truncation error, whenever finite increments are used, the solved differential equation is no more the original one, but it is shown to be a *"modified"* equation containing higher order derivatives**
- **in general, for a consistent numerical scheme the truncation error can be expressed in terms of powers of Δt and space increments**

$$LTE_{n+1} = \frac{\psi_e^{n+1} - C(\Delta t)\psi_e^n}{\Delta t} = O\left(\Delta t^p\right) + O\left(\prod_i \Delta x_i^{q_i}\right) + \ldots$$

Putting a relation between space and time increments, it will be

$$LTE_{n+1} = \frac{\psi_e^{n+1} - C(\Delta t)\psi_e^n}{\Delta t} = O\left(\Delta t^l\right)$$

Stability

In short:

a numerical scheme is said to be "stable" if the unavoidable round of and truncation errors and the errors in the initial data are not amplified during the calculations

Since it is

$$\delta^1 = \psi_e^1 - C(\Delta t)\psi_e^0 + C(\Delta t)\psi_e^0 - C(\Delta t)\psi^0 = \Delta t\, LTE_1 + C(\Delta t)\delta^0$$

$$\delta^2 = \Delta t\, LTE_2 + C(\Delta t)\delta^1 = \Delta t\, LTE_2 + \Delta t\, C(\Delta t) LTE_1 + C^2(\Delta t)\delta^0$$

.....

$$\delta^n = \Delta t \sum_{i=1}^{n} C^{n-i}(\Delta t) LTE_i + C^n(\Delta t)\delta^0$$

it can be noted that stability is related to the *boundedness* of the difference operators $C^n(\Delta t)$ over the functional domain D_ψ in which they are applied

It is therefore requested that any operator $C^n(\Delta t)$ obtainied by varying Δt in an admissible interval $(0,\tau)$ (so that $n\Delta t \le \bar{t}$) be *uniformly bounded* on D_ψ, i.e., a real number K must exist such that for any $\psi \in \mathsf{D}_\psi$ it is

$$\left\| C^n(\Delta t)\psi \right\| \le K \left\| \psi \right\|$$

where the adopted norm is the one selected to be applied in this context.

Then, for any n it will be also

$$\left\| C^n(\Delta t)\delta^0 \right\| \le K \left\| \delta^0 \right\|$$

Lax's Equivalence Theorem

Given a properly posed initial-value linear differential problem and a finite difference approximation to it that satisfies the consistency condition, stability is the necessary and sufficient condition for convergence

A *properly posed problem* has the following characteristics:

- **its solution exists and is unique**
- **the solution depends in a continuous manner from the initial data.**

Just to provide an illustration of the above concepts, we note that

$$\delta^n = \Delta t \sum_{i=1}^{n} C^{n-i}(\Delta t) LTE_i + C^n(\Delta t)\delta^0$$

and, shifting to norms, we have

$$\left\| \delta^n \right\| \le \Delta t \sum_{i=1}^{n} \left\| C^{n-i}(\Delta t) LTE_i \right\| + \left\| C^n(\Delta t)\delta^0 \right\|$$

For a stable scheme, we have

$$\left\| \delta^n \right\| \le \Delta t\, K \sum_{i=1}^{n} \left\| LTE_i \right\| + K \left\| \delta^0 \right\|$$

If the scheme is also consistent, it is

$$\left\|\delta^n\right\| \le n\Delta t\, K\, O\left(\Delta t^l\right) + K\left\|\delta^0\right\| \le \bar{t} K\, O\left(\Delta t^l\right) + K\left\|\delta^0\right\|$$

where we used the relation $n\Delta t \le \bar{t}$. It can be therefore noted that, if the initial error is zero

$$\lim_{\Delta t \to 0}\left\|\delta^n\right\| = 0$$

meaning that the scheme is also convergent.

Consistency and stability* are therefore the properties that must be possessed by a numerical scheme to be also *convergent.

Checking consistency is relatively easy; for stability, instead, specific criteria can be applied

Specific forms of stability criteria

Let us consider a difference operator having the form

$$\boldsymbol{\psi}^{n+1} = \mathbf{T}\boldsymbol{\psi}^n + \mathbf{c}$$

with $\mathbf{T}$ a matrix and ψ e c vectors.

In the absence of initial and round-off errors it is:

$$\boldsymbol{\psi}^1 = \mathbf{T}\boldsymbol{\psi}^0 + \mathbf{c} \qquad \boldsymbol{\psi}^2 = \mathbf{T}\boldsymbol{\psi}^1 + \mathbf{c} = \mathbf{T}^2\boldsymbol{\psi}^0 + \mathbf{T}\mathbf{c} + \mathbf{c}$$

$$\boldsymbol{\psi}^n = \mathbf{T}^n\boldsymbol{\psi}^0 + \mathbf{T}^{n-1}\mathbf{c} + \ldots + \mathbf{T}\mathbf{c} + \mathbf{c}$$

On the contrary, if the initial vector ψ^0 is perturbed by a vector δ^0 it is

$$\boldsymbol{\psi}^n + \boldsymbol{\delta}^n = \mathbf{T}^n\left(\boldsymbol{\psi}^0 + \boldsymbol{\delta}^0\right) + \mathbf{T}^{n-1}\mathbf{c} + \ldots + \mathbf{T}\mathbf{c} + \mathbf{c}$$

and then

$$\boldsymbol{\delta}^n = \mathbf{T}^n\boldsymbol{\delta}^0$$

Applying the norms, it is:

$$\left\|\boldsymbol{\delta}^n\right\| = \left\|\mathbf{T}^n\boldsymbol{\delta}^0\right\| \le \left\|\mathbf{T}^n\right\|\left\|\boldsymbol{\delta}^0\right\|$$

On the basis of this formulation we can apply two different stability criteria.

- ***Stability according to Lax and Richtmyer***

Assigning a relation between Δt and spatial increments, we require that at a given time level $\bar{t} = n\Delta t$ there exist a positive number M independent from n, Δt e and from spatial increments, such that

$$\left\| \mathbf{T}^n \right\| \leq M$$

In order to have this criterion satisfied for any *n*, it must be:

$$\left\| \mathbf{T} \right\| \leq 1$$

However, it must be noted that if the solution of the differential problem increases with time (e.g., a reactor with $k_{eff} > 1$) we accept that

$$\left\| \mathbf{T} \right\| \leq 1 + O(\Delta t)$$

since in this *"non-dissipative"* problem amplification of perturbations (as well as of the solution) does occur even in the exact solution

- ***Matrix Stability***

In this case, we assign values of Δt and of space increments *separately* and we progress in calculation advancing time (so, the dimensionality of the problem does not increase).

It is therefore evident that, in order to get

$$\lim_{n\to\infty} \boldsymbol{\delta}^n = \lim_{n\to\infty} \mathbf{T}^n \boldsymbol{\delta}^0 = 0$$

with $\mathbf{T}$ a matrix of constant order it is:

$$\rho(\mathbf{T}) \leq 1$$

In case of an exact solution that is anyway divergent, we have

$$\rho(\mathbf{T}) \leq 1 + O(\Delta t)$$

This criterion is similar to the previous one:

$$\rho(\mathbf{T}) \leq 1 \quad \textbf{instead of} \quad \left\| \mathbf{T} \right\| > 1$$

- ***A further definition: stability according to von Neumann***

Given the case of a partial differential equation in the unknown function $\psi(x,t)$, we assume that the error with respect to the

numerical solution obtained by *non-perturbed* previous time-step data can be expressed in complex exponential form as:

$$\delta(x,t) = \psi(x,t) - \psi_{np}(x,t) = \delta_{ref} e^{\alpha(t-t_{ref})} e^{i\beta(x-x_{ref})} \qquad \alpha \in \mathbf{C},\ \beta \in \mathbf{R}$$

Defining $C(\Delta t, \Delta x)$ as the difference operator, it is:

$$\psi_{np}(x,t+\Delta t) = C(\Delta t, \Delta x)\psi_{np}(x,t)$$

$$\psi(x,t+\Delta t) = C(\Delta t, \Delta x)\psi(x,t)$$

it is also (subtract the above relationships side by side):

$$\delta(x,t+\Delta t) = C(\Delta t, \Delta x)\delta(x,t)$$

The ratio

$$G(\Delta t, \Delta x) = \frac{\delta(x,t+\Delta t)}{\delta(x,t)}$$

is said *amplification factor* and allows discussing stability.

<u>NB</u>: $G(\Delta t, \Delta x)$ is a ratio of numbers, while $C(\Delta t, \Delta x)$ is an operator

The stability criterion is, in particular:

$$\left|G(\Delta t, \Delta x)\right| = \left|e^{\alpha \Delta t}\right| \le 1 \qquad \textit{(von Neumann condition)}$$

and implies discussing the real part of α as a function of β.

<u>Note</u>: Assuming a complex α and a real β implies that:

- **the evolution of the error in time *is in general oscillatory "damped" or "amplified"*; as a degenerate case (Im(α)=0), it can be simply amplified or damped; in fact:**

$$e^{\alpha(t-t_{ref})} = e^{\mathrm{Re}(\alpha)(t-t_{ref})}\left\{\cos\left[\mathrm{Im}(\alpha)(t-t_{ref})\right] + i\sin\left[\mathrm{Im}(\alpha)(t-t_{ref})\right]\right\}$$

- **the trend of the error in space is oscillatory (unless $\beta = 0$)**

$$e^{i\beta(x-x_{ref})} = \cos\left[\beta(x-x_{ref})\right] + i\sin\left[\beta(x-x_{ref})\right]$$

So, checking stability involves considering $\mathrm{Re}(\alpha)$, i.e.:

$$\mathrm{Re}(\alpha) < 0 \Rightarrow \textit{stable} \quad \textbf{and} \quad \mathrm{Re}(\alpha) > 0 \Rightarrow \textit{unstable}$$

as a function of an arbitrary β. This has the meaning to ascertain if there is any *wave number* β at which the corresponding spatial perturbation is made to grow; since perturbations generally contain all spatial wavelengths (i.e., all wave numbers) it must be assumed that the system is unstable whenever there is a value of β at which $\mathrm{Re}(\alpha) > 0$.

In the case in which the exact solution of the differential equation is increasing in time, it is only requested that

$$\left|G(\Delta t, \Delta x)\right| \le 1 + K\,\Delta t \qquad \textit{(von Neumann necessary condition)}$$

Example 1: **Evaluate stability of the dimensionless diffusion equation for the simple explicit finite difference discretisation**

$$\frac{\partial \psi}{\partial t} - \frac{\partial^2 \psi}{\partial x^2} = 0 \qquad \rightarrow \qquad \frac{\psi_i^{n+1} - \psi_i^n}{\Delta t} - \frac{\psi_{i+1}^n - 2\psi_i^n + \psi_{i-1}^n}{\Delta x^2} = 0$$

In the presence of perturbations, it is

$$\frac{\psi_i^{n+1} + \delta_i^{n+1} - \psi_i^n - \delta_i^n}{\Delta t} - \frac{\psi_{i+1}^n + \delta_{i+1}^n - 2\left(\psi_i^n + \delta_i^n\right) + \psi_{i-1}^n + \delta_{i-1}^n}{\Delta x^2} = 0$$

Subtracting the two above equations it is:

$$\frac{\delta_i^{n+1} - \delta_i^n}{\Delta t} - \frac{\delta_{i+1}^n - 2\delta_i^n + \delta_{i-1}^n}{\Delta x^2} = 0$$

with

$$\delta_i^{n+1} = \delta_i^n e^{\alpha \Delta t} \qquad \delta_{i-1}^n = \delta_i^n e^{-i\beta \Delta x} \qquad \delta_{i+1}^n = \delta_i^n e^{i\beta \Delta x}$$

Therefore

$$\frac{e^{\alpha \Delta t} - 1}{\Delta t} - \frac{e^{i\beta \Delta x} - 2 + e^{-i\beta \Delta x}}{\Delta x^2} = 0 \rightarrow e^{\alpha \Delta t} = 1 + \frac{\Delta t}{\Delta x^2}\left[\left(e^{i\beta \Delta x} + e^{-i\beta \Delta x}\right) - 2\right]$$

$$|G| = \left|e^{\alpha\Delta t}\right| = \left|1 + \frac{2\Delta t}{\Delta x^2}\left[\cos(\beta\Delta x) - 1\right]\right| \le 1 \Rightarrow \frac{2\Delta t}{\Delta x^2} \le 1$$

So, stability of the explicit scheme is conditioned to the above criterion (*conditional stability*).

Example 2: **Evaluate stability of the dimensionless diffusion equation for the simple implicit finite difference discretisation**

$$\frac{\partial\psi}{\partial t} - \frac{\partial^2\psi}{\partial x^2} = 0 \qquad \rightarrow \qquad \frac{\psi_i^{n+1} - \psi_i^n}{\Delta t} - \frac{\psi_{i+1}^{n+1} - 2\psi_i^{n+1} + \psi_{i-1}^{n+1}}{\Delta x^2} = 0$$

It is now:

$$\frac{\delta_i^{n+1} - \delta_i^n}{\Delta t} - \frac{\delta_{i+1}^{n+1} - 2\delta_i^{n+1} + \delta_{i-1}^{n+1}}{\Delta x^2} = 0$$

where we put

$$\delta_i^n = \delta_i^{n+1} e^{-\alpha\Delta t} \qquad \delta_{i-1}^{n+1} = \delta_i^{n+1} e^{-i\beta\Delta x} \qquad \delta_{i+1}^{n+1} = \delta_i^{n+1} e^{i\beta\Delta x}$$

Then:

$$\frac{1 - e^{-\alpha\Delta t}}{\Delta t} - \frac{e^{i\beta\Delta x} - 2 + e^{-i\beta\Delta x}}{\Delta x^2} = 0 \rightarrow e^{-\alpha\Delta t} = 1 + \frac{\Delta t}{\Delta x^2}\left[2 - \left(e^{i\beta\Delta x} + e^{-i\beta\Delta x}\right)\right]$$

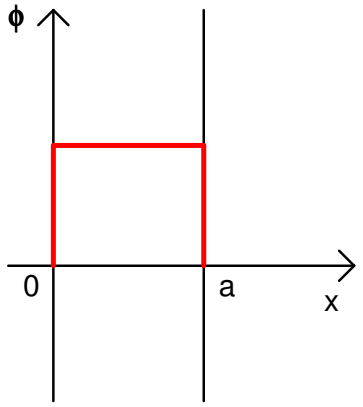
$$|G| = \left|e^{\alpha\Delta t}\right| = \left|\left\{1 + \frac{2\Delta t}{\Delta x^2}\left[1 - \cos\left(\beta\Delta x\right)\right]\right\}^{-1}\right| \le 1 \qquad (always\ satisfied)$$

So, *implicit schemes tend to be unconditionally stable, something that makes them preferable in cases in which an explicit scheme would limit too much the allowable time step*.

Anyway, stability is not the single consideration for selecting a numerical scheme. Of course, stability must be assured but, beyond that, the overall efficiency and accuracy of the calculation in a given application are often the most important considerations.

A FURTHER LOOK AT EXPLICIT AND IMPLICIT TIME ADVANCEMENT SCHEMES

Let us consider for simplicity a time-dependent problem of diffusion in a 1D absorbing layer. This problem is simpler than the neutron kinetics problem and is somehow similar to the "pulsed" experiments studied in Reactor Physics.



The neutron balance partial differential equation for a homogeneous layer is:

$$\frac{1}{v}\frac{\partial \phi}{\partial t} = D\frac{\partial^2 \phi}{\partial x^2} - \Sigma_a \phi \qquad \textbf{(1)}$$

with the simplified boundary and initial conditions

$$\phi(0,t) = \phi(a,t) = 0 \qquad \phi(x,0) = \phi_0(x) \qquad \textbf{(2)}$$

By discretizing the equation only in space, with the usual finite difference technique we get:

$$\frac{1}{v}\frac{d\phi_i}{dt} = D\frac{\phi_{i+1}(t) - 2\phi_i(t) + \phi_{i-1}(t)}{\Delta x^2} - \Sigma_a \phi_i(t) \quad (i = 1, \ldots, N)$$

This represents a problem of linear ordinary differential equations that can be written in vector form as:

$$\frac{d\boldsymbol{\phi}}{dt} = \mathbf{A}\boldsymbol{\phi} \qquad \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0 \qquad \textbf{(3)}$$

In similarity with the scalar version of such separable variable problem, we can calculate the exact solution of it as:

$$\boldsymbol{\phi}(t) = e^{t\mathbf{A}}\boldsymbol{\phi}_0 \qquad \textbf{(4)}$$

(see V.I. Arnold, Ordinary Differential Equations, MIT Press, 1995). Without going into complex mathematical details, we can note that:

- **the eigenvectors of the matrix $e^{t\mathbf{A}}$ are the same as the eigenvectors of matrix A: let's call them $\boldsymbol{\varphi}_h$ ($h=1,..,N$);**
- **the eigenvalues of the matrix $e^{t\mathbf{A}}$ will be $e^{\lambda_h t}$, where λ_h are the eigenvalues of A.**

For purpose of illustration of the above considerations, let's note that it is by definition:

$$e^{t\mathbf{A}} = \mathbf{I} + t\mathbf{A} + \frac{t^2}{2!}\mathbf{A}^2 + \ldots = \sum_{k=0}^{\infty}\frac{t^k\mathbf{A}^k}{k!} \qquad \textbf{(5)}$$

and then

$$e^{t\mathbf{A}}\boldsymbol{\varphi}_h = \boldsymbol{\varphi}_h + t\mathbf{A}\boldsymbol{\varphi}_h + \frac{t^2}{2!}\mathbf{A}^2\boldsymbol{\varphi}_h + \ldots = \left[\sum_{k=0}^{\infty}\frac{t^k\mathbf{A}^k}{k!}\right]\boldsymbol{\varphi}_h \qquad (h=1,\ldots,N) \qquad \textbf{(6)}$$

or

$$e^{t\mathbf{A}}\boldsymbol{\varphi}_h = \boldsymbol{\varphi}_h + t\lambda_h\boldsymbol{\varphi}_h + \frac{t^2}{2!}\lambda_h{}^2\boldsymbol{\varphi}_h + \ldots = \left[\sum_{k=0}^{\infty}\frac{(\lambda_h t)^k}{k!}\right]\boldsymbol{\varphi}_h = e^{\lambda_h t}\boldsymbol{\varphi}_h \quad (h=1,\ldots,N) \qquad \textbf{(7)}$$

Now let's assume that the matrix A has distinct eigenvalues and that:

$$\boldsymbol{\phi}_0 = \sum_{h=1}^{N} c_h^0\boldsymbol{\varphi}_h \qquad \textbf{(8)}$$

In such a case, the solution of (3) is written as:

$$\boldsymbol{\phi}(t) = e^{t\mathbf{A}}\boldsymbol{\phi}_0 = e^{t\mathbf{A}}\left[\sum_{h=1}^{N} c_h^0\boldsymbol{\varphi}_h\right] = \sum_{h=1}^{N} c_h^0 e^{\lambda_h t}\boldsymbol{\varphi}_h \qquad \textbf{(9)}$$

a result that is coherent with the known techniques for solving linear Ordinary Differential Equation (ODE) systems.

It is interesting now to review the explicit, implicit and Crank-Nicolson schemes starting from a different perspective. *Padé approximant functions* are rational fractional expressions that represent approximations of the exponential as:

$$e^{\theta} = \frac{1 + p_1\theta + p_2\theta^2 + \ldots + p_T\theta^T}{1 + q_1\theta + q_2\theta^2 + \ldots + q_S\theta^S} + c_{S+T+1}\theta^{S+T+1} + O(\theta^{S+T+2}) \tag{10}$$

where the right hand side can be expressed as

$$R_{S,T}(\theta) = \frac{1 + p_1\theta + \ldots + p_T\theta^T}{1 + q_1\theta + \ldots + q_S\theta^S} = \frac{P_T(\theta)}{Q_S(\theta)} \tag{11}$$

A table of some of these Padé rational approximations can be found in G. D. Smith, "Numerical Solution of Partial Differential Equations: Finite Difference Methods", Oxford Univ Press, 1986.

(S, T)	$R_{S,T}(\theta)$	Principal error term
$(0, 1)$	$1+\theta$	$\frac{1}{2}\theta^2$
$(0, 2)$	$1+\theta+\frac{1}{2}\theta^2$	$\frac{1}{6}\theta^3$
$(1, 0)$	$\frac{1}{1-\theta}$	$-\frac{1}{2}\theta^2$
$(1, 1)$	$\frac{1+\frac{1}{2}\theta}{1-\frac{1}{2}\theta}$	$-\frac{1}{12}\theta^3$
$(1, 2)$	$\frac{1+\frac{2}{3}\theta+\frac{1}{6}\theta^2}{1-\frac{1}{3}\theta}$	$-\frac{1}{72}\theta^4$
$(2, 0)$	$\frac{1}{1-\theta+\frac{1}{2}\theta^2}$	$\frac{1}{6}\theta^3$
$(2, 1)$	$\frac{1+\frac{1}{3}\theta}{1-\frac{2}{3}\theta+\frac{1}{6}\theta^2}$	$\frac{1}{72}\theta^4$
$(2, 2)$	$\frac{1+\frac{1}{2}\theta+\frac{1}{12}\theta^2}{1-\frac{1}{2}\theta+\frac{1}{12}\theta^2}$	$\frac{1}{720}\theta^5$

Focusing on the three selected approximations in the table, we can have the following cases:

1. **the *explicit scheme* can be considered as the consequence of selecting the approximant (0,1)**

$$\boldsymbol{\phi}^{n+1} = e^{\Delta t\mathbf{A}}\boldsymbol{\phi}^{n} \approx \left(\mathbf{I} + \Delta t\mathbf{A}\right)\boldsymbol{\phi}^{n} \tag{12}$$

   or, assuming equality,

$$\frac{\boldsymbol{\phi}^{n+1} - \boldsymbol{\phi}^{n}}{\Delta t} = \mathbf{A}\boldsymbol{\phi}^{n} \tag{13}$$

2. **the *implicit scheme* can be considered as the consequence of selecting the approximant (1,0)**

$$\boldsymbol{\phi}^{n+1} = e^{\Delta t\mathbf{A}}\boldsymbol{\phi}^{n} \approx \left(\mathbf{I} - \Delta t\mathbf{A}\right)^{-1}\boldsymbol{\phi}^{n} \tag{14}$$

   or, assuming equality,

$$\left(\mathbf{I} - \Delta t\mathbf{A}\right)\boldsymbol{\phi}^{n+1} = \boldsymbol{\phi}^{n} \Rightarrow \frac{\boldsymbol{\phi}^{n+1} - \boldsymbol{\phi}^{n}}{\Delta t} = \mathbf{A}\boldsymbol{\phi}^{n+1} \tag{15}$$

3. **the *Crank-Nicolson scheme* can be considered as the consequence of selecting the approximant (1,1)**

$$\boldsymbol{\phi}^{n+1} = e^{\Delta t\mathbf{A}}\boldsymbol{\phi}^{n} \approx \left(\mathbf{I} - \frac{\Delta t}{2}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{\Delta t}{2}\mathbf{A}\right)\boldsymbol{\phi}^{n} \tag{16}$$

   or, assuming equality

$$\left(\mathbf{I} - \frac{\Delta t}{2}\mathbf{A}\right)\boldsymbol{\phi}^{n+1} = \left(\mathbf{I} + \frac{\Delta t}{2}\mathbf{A}\right)\boldsymbol{\phi}^{n} \Rightarrow \frac{\boldsymbol{\phi}^{n+1} - \boldsymbol{\phi}^{n}}{\Delta t} = \frac{1}{2}\mathbf{A}\left(\boldsymbol{\phi}^{n} + \boldsymbol{\phi}^{n+1}\right) \tag{17}$$

It is therefore clear that the explicit and implicit approximations (forward and backward time differences respectively) have a larger truncation error with respect to the Crank-Nicolson one (being sort of centered difference scheme).

In relation to stability, let's consider the case in which all the eigenvalues of the matrix A have *a strictly negative real part*. *This is the case of a "dissipative" problem, i.e., a problem in which the components of the initial vector die out in time.*

***"Matrix stability"* will require for such a problem that the *spectral radius of the numerically discretized matrix is lower than unity*.**

For the explicit scheme, this means that putting

$$\boldsymbol{\phi}^{n+1} = \left(\mathbf{I} + \Delta t \mathbf{A}\right)\boldsymbol{\phi}^{n} \qquad \textbf{(18)}$$

it must be:

$$\rho\left(\mathbf{I} + \Delta t \mathbf{A}\right) < 1 \qquad \textbf{(19)}$$

Let us call λ_h the eigenvaules of A, such that $\mathrm{Re}(\lambda_h) < 0$. We see that the eigenvalues of the matrix $\mathbf{I} + \Delta t \mathbf{A}$ are

$$\mu_h = 1 + \Delta t \lambda_h \qquad \textbf{(20)}$$

So, it is:

$$\mathrm{Re}(\mu_h) = 1 + \Delta t\, \mathrm{Re}(\lambda_h) \qquad \mathrm{Im}(\mu_h) = \Delta t\, \mathrm{Im}(\lambda_h) \qquad \textbf{(21)}$$

and then

$$\left|\mu_h\right|^2 = \left[\mathrm{Re}(\mu_h)\right]^2 + \left[\mathrm{Im}(\mu_h)\right]^2 = \left[1 + \Delta t\, \mathrm{Re}(\lambda_h)\right]^2 + \left[\Delta t\, \mathrm{Im}(\lambda_h)\right]^2 \qquad \textbf{(22)}$$

or

$$\begin{aligned} \left|\mu_h\right|^2 &= 1 + 2\Delta t\, \mathrm{Re}(\lambda_h) + \Delta t^2 \left[\mathrm{Re}(\lambda_h)\right]^2 + \Delta t^2 \left[\mathrm{Im}(\lambda_h)\right]^2 \\ &= 1 + 2\Delta t\, \mathrm{Re}(\lambda_h) + \Delta t^2 \left|\lambda_h\right|^2 \end{aligned} \qquad \textbf{(23)}$$

So, the condition

$$\left|\mu_h\right|^2 < 1 \qquad \forall h = 1, \ldots, N \qquad \textbf{(24)}$$

can be satisfied *conditionally* , i.e., if:

$$
\begin{aligned}
&|\mu_h|^2 < 1 \Rightarrow 1 + 2\Delta t\,\mathrm{Re}(\lambda_h) + \Delta t^2 |\lambda_h|^2 < 1 \\
&\Rightarrow 2\Delta t\,\mathrm{Re}(\lambda_h) + \Delta t^2 |\lambda_h|^2 < 0 \\
&\Rightarrow 2\,\mathrm{Re}(\lambda_h) + \Delta t |\lambda_h|^2 < 0 \\
&\Rightarrow \Delta t < -\frac{2\,\mathrm{Re}(\lambda_h)}{|\lambda_h|^2}
\end{aligned}
\qquad \textbf{(25)}
$$

The explicit scheme is therefore only *"conditionally stable"*.

For the <u>implicit scheme</u>, instead, it is:

$$
(\mathbf{I} - \Delta t\mathbf{A})\boldsymbol{\phi}^{n+1} = \boldsymbol{\phi}^n \Rightarrow \boldsymbol{\phi}^{n+1} = (\mathbf{I} - \Delta t\mathbf{A})^{-1}\boldsymbol{\phi}^n \qquad \textbf{(26)}
$$

requiring that

$$
\rho\left[(\mathbf{I} - \Delta t\mathbf{A})^{-1}\right] < 1 \qquad \textbf{(27)}
$$

or

$$
|\mu_h| = \left|\frac{1}{1 - \Delta t\lambda_h}\right| < 1 \qquad (h = 1, \ldots, N) \qquad \textbf{(28)}
$$

With a development similar to the one made for the explicit case, we have that

$$
\begin{aligned}
&\frac{1}{|\mu_h|^2} = |1 - \Delta t\lambda_h|^2 = \left[1 - \Delta t\,\mathrm{Re}(\lambda_h)\right]^2 + \left[\Delta t\,\mathrm{Im}(\lambda_h)\right]^2 \\
&= 1 - 2\Delta t\,\mathrm{Re}(\lambda_h) + \Delta t^2\left[\mathrm{Re}(\lambda_h)\right]^2 + \Delta t^2\left[\mathrm{Im}(\lambda_h)\right]^2 > 1 \qquad (as\ \mathrm{Re}(\lambda_h) < 0)
\end{aligned}
\qquad \textbf{(29)}
$$

So, the implicit scheme is *unconditionally stable*.

In the <u>Crank-Nicolson scheme</u>, we have:

$$
\boldsymbol{\phi}^{n+1} = \left(\mathbf{I} - \frac{\Delta t}{2}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{\Delta t}{2}\mathbf{A}\right)\boldsymbol{\phi}^n \qquad \textbf{(30)}
$$

and

$$\left|\mu_h\right|^2 = \left|\frac{1+\lambda_h \Delta t/2}{1-\lambda_h \Delta t/2}\right|^2 = \frac{\left|1+\lambda_h \Delta t/2\right|^2}{\left|1-\lambda_h \Delta t/2\right|^2}$$

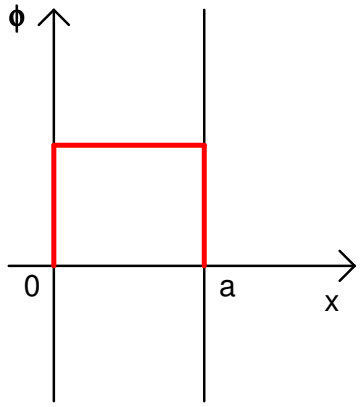
$$= \frac{\left[1+\mathrm{Re}\left(\lambda_h\right)\Delta t/2\right]^2 + \left[\mathrm{Im}\left(\lambda_h\right)\Delta t/2\right]^2}{\left[1-\mathrm{Re}\left(\lambda_h\right)\Delta t/2\right]^2 + \left[\mathrm{Im}\left(\lambda_h\right)\Delta t/2\right]^2} < 1 \qquad (as\ \mathrm{Re}\left(\lambda_h\right) < 0) \tag{31}$$

leading to an unconditionally stable scheme as well.

APPLICATION:
1D TRANSIENT DIFFUSION IN AN ABSORBING LAYER

- **Physical problem**

  Transient evolution of neutron flux in an absorbing layer, with zero flux conditions at the boundaries, starting from an initial flat distribution (with or without localized sources)



- **Mathematical formulation:**

$$\frac{1}{v}\frac{\partial \phi}{\partial t} = D\frac{\partial^2 \phi}{\partial x^2} - \Sigma_a \phi + S$$

$$\phi(0,t) = \phi(a,t) = 0 \qquad \phi(x,0) = 1$$

- **Parameter values:**

$$\Sigma_a = 0.08\ cm^{-1} \qquad D = 0.4\ cm$$

$$v = 2.2 \times 10^5\ cm/s \qquad a = 10\ cm$$

- **Numerical formulation:**

$$\frac{1}{v}\frac{\phi_i^{n+1}-\phi_i^{n}}{\Delta t}=\alpha\left(D\frac{\phi_{i+1}^{n+1}-2\,\phi_i^{n+1}+\phi_{i-1}^{n+1}}{\Delta x^2}-\Sigma_a\,\phi_i^{n+1}\right)$$

$$+(1-\alpha)\left(D\frac{\phi_{i+1}^{n}-2\,\phi_i^{n}+\phi_{i-1}^{n}}{\Delta x^2}-\Sigma_a\,\phi_i^{n}\right)+S_i$$

$$(i=1,\ldots,N)\quad;\quad(n=0,1,\ldots)$$

$$\phi_0^n=\phi_{N+1}^n=0\quad(n=0,1,\ldots)\;;\;=1\qquad(i=1,\ldots,N)$$

It is:

- $\alpha = 0$: explicit scheme
- $\alpha = 0.5$: Crank-Nicolson scheme
- $\alpha = 1$: implicit scheme

Putting:

$$a_i=c_i=-\frac{\alpha\,v\,\Delta t\,D}{\Delta x^2}\qquad b_i=1+\alpha\,v\,\Delta t\left(\frac{2\,D}{\Delta x^2}+\Sigma_a\right)$$

$$d_i=\phi_i^n+(1-\alpha)\,v\,\Delta t\left(D\frac{\phi_{i+1}^{n}-2\,\phi_i^{n}+\phi_{i-1}^{n}}{\Delta x^2}-\Sigma_a\,\phi_i^{n}\right)+v\,\Delta t\,S_i$$

It is:

$$a_i\,\phi_{i-1}^{n+1}+b_i\,\phi_i^{n+1}+c_i\,\phi_{i+1}^{n+1}=d_i\qquad(i=1,\ldots,N)$$

We put: $N = 39 \Rightarrow \Delta x = 0.25$ cm

In the case of the explicit scheme without source, it is:

$$\frac{1}{v}\frac{\phi_i^{n+1}-\phi_i^n}{\Delta t}=\left(D\frac{\phi_{i+1}^n-2\phi_i^n+\phi_{i-1}^n}{\Delta x^2}-\Sigma_a\phi_i^n\right)$$

By applying the von Neumann stability analysis, we find:

$$\frac{1}{v}\frac{\delta_i^{n+1}-\delta_i^n}{\Delta t}=\left(D\frac{\delta_{i+1}^n-2\delta_i^n+\delta_{i-1}^n}{\Delta x^2}-\Sigma_a\delta_i^n\right)$$

where the usual substitutions give:

$$\frac{1}{v}\frac{e^{\alpha\Delta t}-1}{\Delta t}=\left(D\frac{e^{i\beta\Delta x}-2+e^{-i\beta\Delta x}}{\Delta x^2}-\Sigma_a\right)$$

$$e^{\alpha\Delta t}=1+\left(D\frac{e^{i\beta\Delta x}-2+e^{-i\beta\Delta x}}{\Delta x^2}-\Sigma_a\right)v\Delta t$$

$$e^{\alpha\Delta t}=1+\left(D\frac{2\cos(\beta\Delta x)-2}{\Delta x^2}-\Sigma_a\right)v\Delta t$$

$$e^{\alpha\Delta t}=1-\left(2D\,v\Delta t\frac{1-\cos(\beta\Delta x)}{\Delta x^2}+\Sigma_a\,v\Delta t\right)$$

Obtaining the magnitude of the LHS being less than one we need that

$$0<\left(2D\,v\Delta t\frac{1-\cos(\beta\Delta x)}{\Delta x^2}+\Sigma_a\,v\Delta t\right)<2$$

Being $0<1-\cos(\beta\Delta x)<2$ the most limiting condition is

$$\left(2D\,v\Delta t\frac{2}{\Delta x^2}+\Sigma_a\,v\Delta t\right)<2$$

or

$$\Delta t<\frac{2\Delta x^2}{4vD+v\Sigma_a\Delta x^2}$$

Similar conclusions can be obtained by proving that the eigenvectors of the system matrix are

$$\phi_k = \left\{ C_k \sin \frac{k\pi i \Delta x}{a} \right\}_{i=1}^{N} \qquad (k = 1, \ldots, N)$$

and the corresponding eigenvalues are

$$\lambda_k = v\Sigma_a + vD \frac{2}{\Delta x^2} \left[1 - \cos\left(\frac{k\pi \Delta x}{a} \right) \right] \quad (k = 1, \ldots, N) \;;$$

imposing the matrix stability criterion

$$\Delta t < -\frac{2\,\mathrm{Re}(\lambda_h)}{|\lambda_h|^2}$$

the same conclusion obtained above are achieved.

With the adopted parameters values it is:

$$\Delta t < \frac{2\Delta x^2}{4vD + v\Sigma_a \Delta x^2} = \frac{2 \times (0.25)^2}{4 \times 2.2 \times 10^5 \times 0.4 + 2.2 \times 10^5 \times 0.8 \times (0.25)^2} \approx 3.54 \times 10^{-7}\, s$$

So, it can be expected that, with a larger time step, the explicit scheme will be unstable. The implicit and Crank-Nicolson ones are instead stable.

FORTRAN PROGRAM

```
c-----------------------------------------------------------------------c
c                                                                       c
c    1D Trasient Neutron diffusion with sources                         c
c    Program set up for teaching purposes only                          c
c    W. Ambrosini - Università di Pisa                                  c
c                                                                       c
c-----------------------------------------------------------------------c
      program nonsta
      implicit double precision (a-h,o-z)
      character*80 riga
c
      parameter (m = 50)
      dimension phi(0:m+1),x(0:m+1),xsour(10),sour(10),isour(10)
      dimension a(m),b(m),c(m),d(m),v(m),alef(m),bet(m)
c
      open (unit=5,file='nonsta.dat')
      open (unit=6,file='nonsta.txt')
c
      read (5,100) riga
      read (5,*) aleng,diff,sigma,vel,n
c
      read (5,100) riga
      read (5,*) dt,tend,alpha
c
      read (5,100) riga
      read (5,*) nsour
c
      read (5,100) riga
c
      do 5 is = 1,nsour
      read (5,*) xsour(is),sour(is)
    5 continue
c
      read (5,100) riga
      read (5,*) xbar,sigbar
c
      np1 = n + 1
      dx = aleng / dfloat (np1)
      cmpalp = 1.d00 - alpha
      veldt = vel * dt
      sigdt = sigma * dt
      dx2 = dx * dx
c
      phi(0) = 0.d00
      phi(np1) = 0.d00
      x(0) = 0.d00
      x(np1) = aleng
c
      do 10 i = 1,n
c
      phi(i) = 1.d00
      x(i) = dx * dfloat(i)
c
      halfdx = 0.5d00 * dx
c
         do j = 1,nsour
        if(dabs(xsour(j)-x(i)).le.halfdx) isour(j) = i
         enddo
c
        if(dabs(xbar-x(i)).le.halfdx) ibar = i
c
      auxsig = 0.d00
       if(i.eq.ibar) auxsig = sigbar
c
      a(i) = - veldt * alpha * diff / dx2
      b(i) = 1.d00 + veldt * alpha * ( 2.d00 * diff / dx2
     &                 + sigma + auxsig )
      c(i) = a(i)
```

```
c
   10 continue
c
c   Loop on time advancement
c
      time = 0.d00
      do 50 it = 1,5000000
      time = time + dt
c
      do 20 i = 1,n
      im1 = i - 1
      ip1 = i + 1
      d2phi = phi(ip1) - 2.d00 * phi(i) + phi(im1)
C
      auxsig = 0.d00
       if(i.eq.ibar) auxsig = sigbar
      d(i) = phi(i) + veldt * cmpalp * ( diff * d2phi / dx2
     &                      - ( sigma + auxsig ) * phi(i) )
c
       do j = 1,nsour
        if(i.eq.isour(j)) d(i) = d(i) + veldt * sour(j)
       enddo
c
   20 continue
c
      call tdma (a,b,c,d,v,alef,bet,n,m)
c
      phimax = 0.d00
      do 30 i = 0,np1
      if((i.ne.0).and.(i.ne.np1)) phi(i) = v(i)
      write(6,110) time,x(i),phi(i)
      if(phimax.lt.dabs(phi(i))) phimax = phi(i)
   30 continue
      if((time.ge.tend).or.(phimax.gt.10.d00)) goto 60
c
   50 continue
   60 continue
c
      stop
  100 format(a80)
  110 format(3(1x,e14.7))
      end
c-----------------------------------------------------------------------c
c                                                                       c
c   TDMA Algorithm                                                      c
c                                                                       c
c-----------------------------------------------------------------------c
      subroutine tdma (a,b,c,d,v,alef,bet,n,ld)
      implicit double precision (a-h,o-z)
      dimension a(ld),b(ld),c(ld),d(ld),v(ld),alef(ld),bet(ld)
      ub=1.d00/b(1)
      alef(1)=c(1)*ub
      bet(1)=d(1)*ub
      do 10 i=2,n
      l=i-1
      qz=b(i)-a(i)*alef(l)
      uqz=1.d00/qz
      alef(i)=c(i)*uqz
   10 bet(i)=(d(i)-a(i)*bet(l))*uqz
      nm1=n-1
      v(n)=bet(n)
      do 20 i=1,nm1
      ii=n-i
      l=ii+1
   20 v(ii)=bet(ii)-alef(ii)*v(l)
      return
       end
```

Proposed exercises

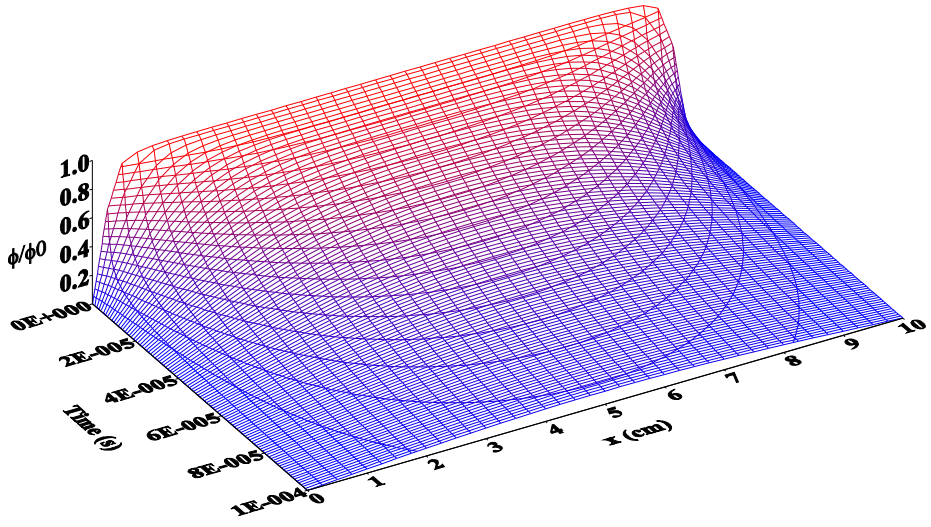
1. Study the transient concerning the neutron diffusion starting from a uniform distribution in the layer:
   a) with the explicit scheme ($\alpha = 0$) and time steps of 2×10^{-7} s;
   b) with the explicit scheme ($\alpha = 0$) and time steps of 4×10^{-7} s (higher than the stability boundary $\approx 3.54 \times 10^{-7}$ s);
   c) with the implicit ($\alpha = 1$) and the Crank-Nicolson ($\alpha = 0.5$) schemes and time steps of 1×10^{-6} s or greater.

   Make use of the supplied MATLAB file to visualize the results.

INPUT DECK

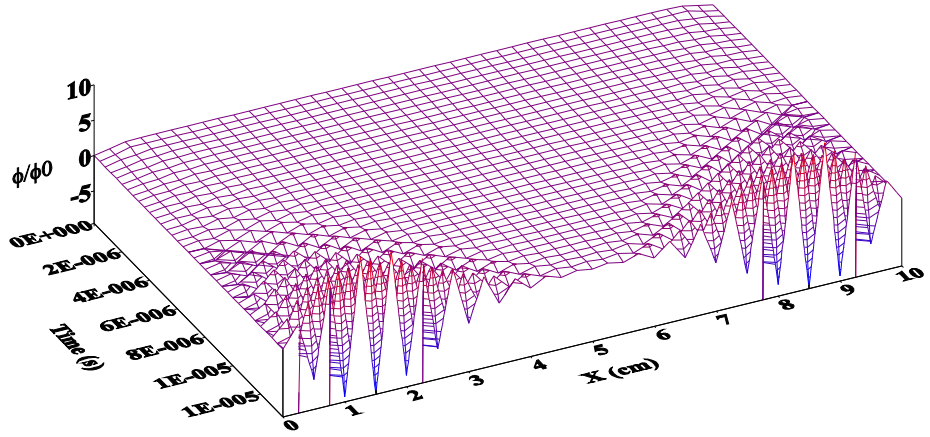
```
   ALENG       DIFF      SIGMA                VEL          N
      10.       0.4       0.08             2.2e5         39
      DT       TEND      ALPHA
4.0e-07     0.0002        1.0
  NSOUR
  4
     XSOUR      SOUR
     2.          0.0
     5.          0.0
     6.          0.0
     8.          0.0
     XBAR       SIGBAR
     5.          0.00
```

Expected results:



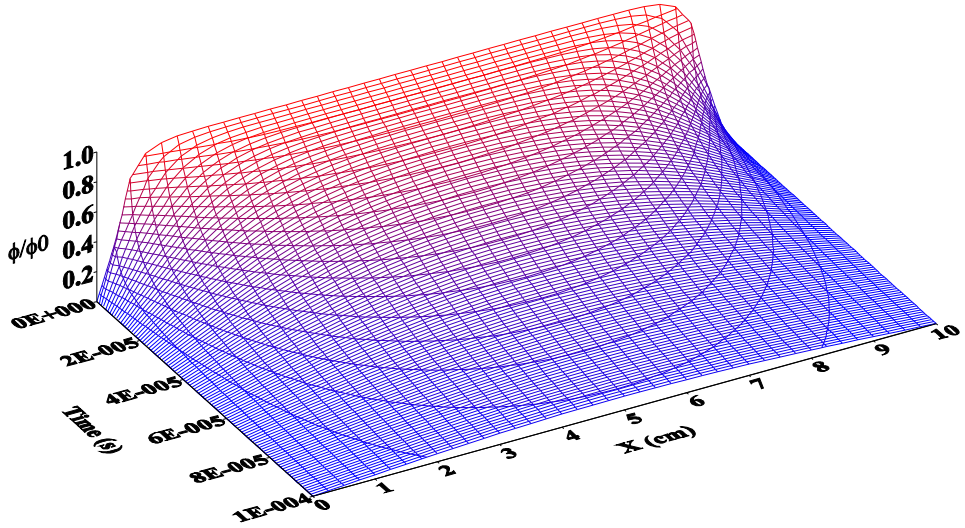
Explicit Method - Time step = 2.e-7 s

Stability Limit: Time step < 3.54e-7 s



Explicit Method Time step = 4.e-7 s

Stability Limit: Time step < 3.54e-7 s



Crank Nicolson Method - Time step = 1.e-6 s

Unconditionally Stable

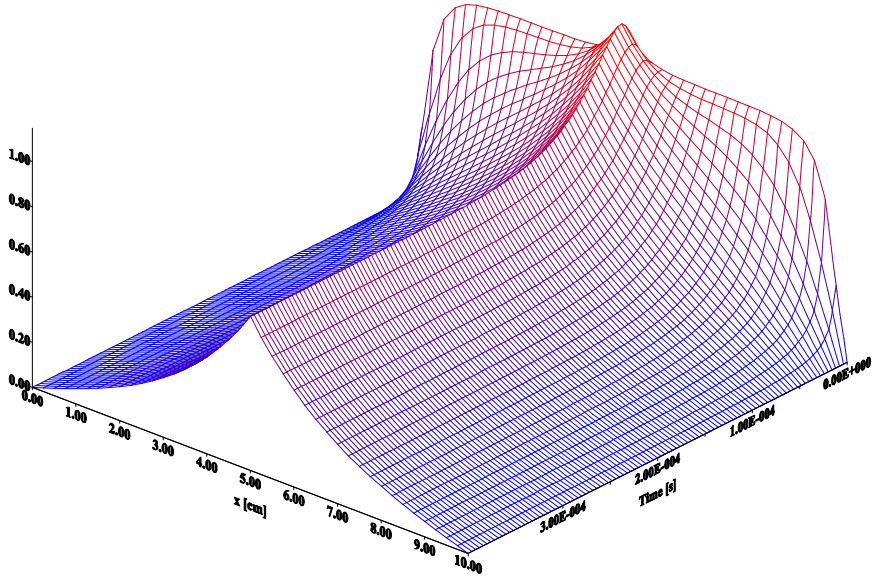
2. Study the transient related to neutron diffusion starting from a uniform flux distribution in the case of a point source (nearly a Dirac's delta function) placed in different positions in the layer

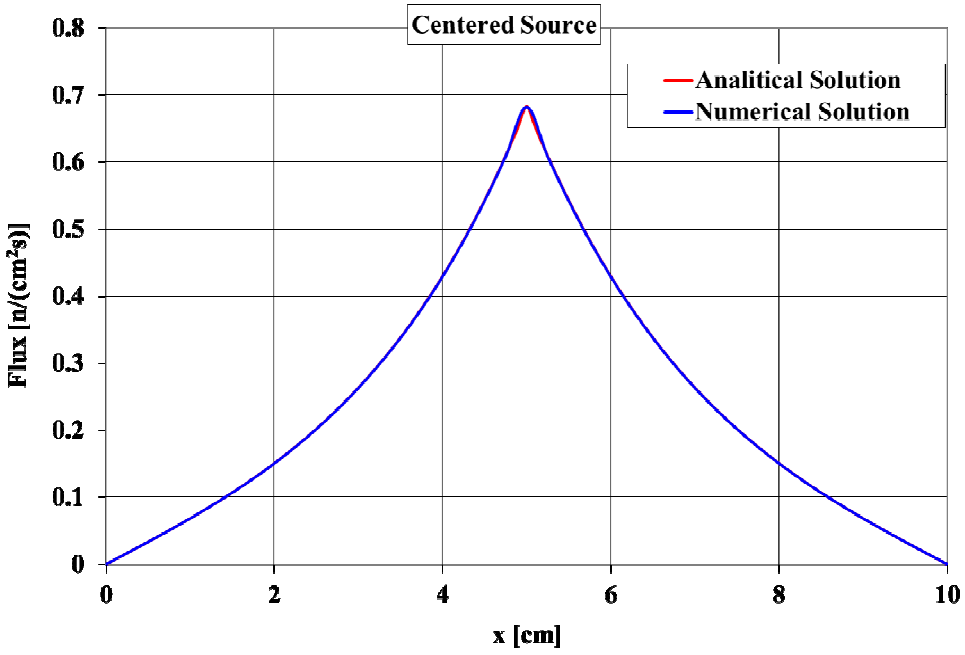
$$\phi^{+} = \frac{S}{\frac{D}{L}\sinh\frac{a}{L}}\sinh\frac{x'}{L}\sinh\frac{a-x}{L}, \qquad 0 \le x' \le x \le a$$

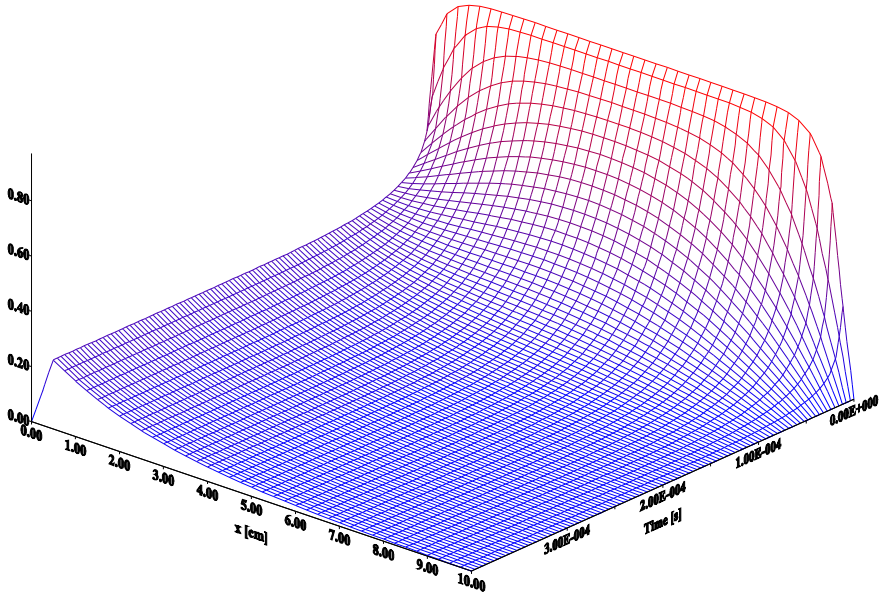
$$\phi^{-} = \frac{S}{\frac{D}{L}\sinh\frac{a}{L}}\sinh\frac{a-x'}{L}\sinh\frac{x}{L}, \qquad 0 \le x \le x' \le a$$

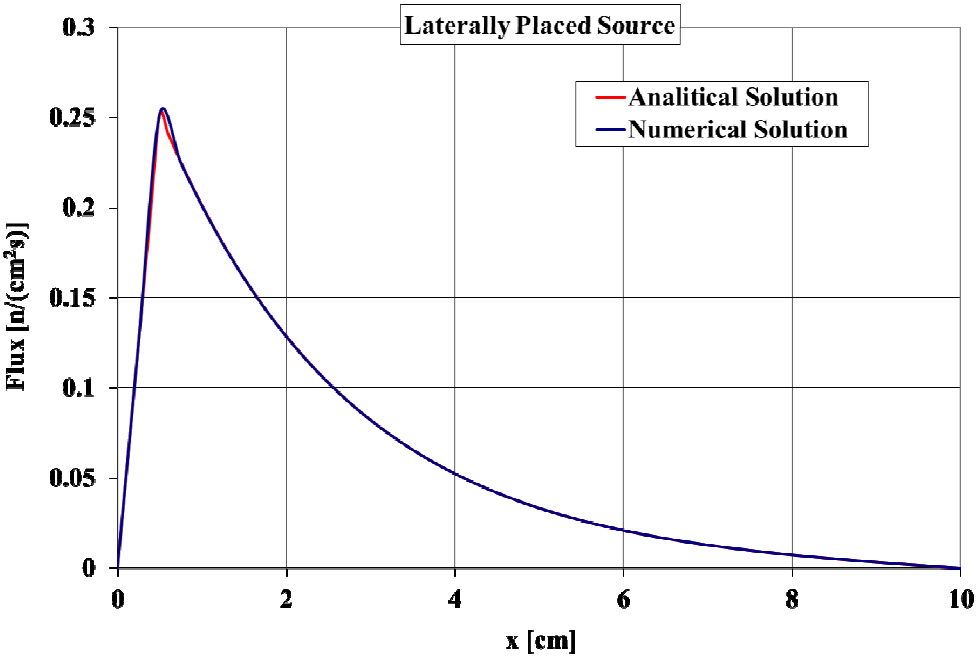
Expected results:

a) source located in the centre









3. Modify the program for different source types (ad es.,uniform, sinusoidal) or placing “control rods”.

4. Suggest how to change the equations and the program for including a fission source (without delayed neutrons).

STIFFNESS OF NEUTRON KINETICS EQUATIONS

Let us consider a differential problem defined as

$$\frac{d\boldsymbol{\psi}}{dt} = \mathbf{F}(\boldsymbol{\psi}, t)$$

and let us consider a "fixed point", i.e., a vector that satisfies the steady-state condition:

$$\mathbf{F}(\boldsymbol{\psi}_0) = 0$$

Around that point, the problem can be linearized as follows:

$$\frac{d}{dt}(\delta\boldsymbol{\psi} + \boldsymbol{\psi}_0) = \mathbf{F}(\delta\boldsymbol{\psi} + \boldsymbol{\psi}_0) \approx \underset{=0}{\mathbf{F}(\boldsymbol{\psi}_0)} + \left.\frac{\partial \mathbf{F}}{\partial \boldsymbol{\psi}}\right|_{\boldsymbol{\psi}_0} \delta\boldsymbol{\psi}$$

where $\delta\boldsymbol{\psi}$ **is a perturbation vector and** $\left.\frac{\partial \mathbf{F}}{\partial \boldsymbol{\psi}}\right|_{\boldsymbol{\psi}_0}$ **indicates a Jacobian matrix; we write:**

$$\frac{d}{dt}(\delta\boldsymbol{\psi}) = \left.\frac{\partial \mathbf{F}}{\partial \boldsymbol{\psi}}\right|_{\boldsymbol{\psi}_0} (\delta\boldsymbol{\psi})$$

Assuming that the eigenvalues, $\lambda_h \ (h = 1, \ldots, N)$ **of the Jacobian matrix have negative real part**

$$\mathrm{Re}(\lambda_h) < 0$$

the problem is said "stiff" if the (negative) real part of the eigenvalues spans over a wide interval, so that the ratio

$$R = \frac{\max_h \{|\mathrm{Re}(\lambda_h)|\}}{\min_h \{|\mathrm{Re}(\lambda_h)|\}}$$

is large.

In other words, ***the system contains time constants that are largely different*****, so that fulfilling a criterion for stability of an explicit numerical scheme will result particularly limiting as**

the "shortest" time constant will be the most constraining, even if we are interested at the transient progression over larger time scales.

Neutron kinetics equations have a "stiff" character, since the time constants that come into play are the "prompt neutron life time", with can be very short (10^{-5} – 10^{-3} s), and the delayed neutron precursor lifetimes, which can range up to several tens of seconds.

As in many other "stiff" problems, this obliges to discard explicit discretisations, because they would lead to the use of very small time-steps. On the other hand, implicit discretisations could be heavy from a computational point of view, proposing again the scheme of "inner" and "outer" iterations also in transient analyses. In fact:

- ***inner iterations*** **are necessary to advance each time step by any numerical scheme by the solution of linear systems:**

$$\boldsymbol{\phi}^{n+1} = \left(\mathbf{I} - \frac{\Delta t}{2}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{\Delta t}{2}\mathbf{A}\right)\boldsymbol{\phi}^{n}$$

$$\Rightarrow \left(\mathbf{I} - \frac{\Delta t}{2}\mathbf{A}\right)\boldsymbol{\phi}^{n+1} = \left(\mathbf{I} + \frac{\Delta t}{2}\mathbf{A}\right)\boldsymbol{\phi}^{n} = \mathbf{s}^{n} \qquad \textbf{(32)}$$

- ***outer iterations*** **are actually the replacement of the old with the new vector.**

So, using totally implicit schemes may be too limiting in terms of efficiency and "partially implicit ones" could be more conveniently used.

In the following we will see some examples of ADI, ADE, and ADE-ADI methods.

Alternating Direction Implicit Scheme (ADI)

Let us consider the problem

$$\begin{cases} \dfrac{d\boldsymbol{\phi}}{dt} = \mathbf{A}\boldsymbol{\phi} \\ \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0 \end{cases}$$

and let's divide the matrix A into the x and y contributions of the discretized Laplace operator plus absorption:

$$\mathbf{A} = \mathbf{X} + \mathbf{Y} + \boldsymbol{\Sigma} = \underbrace{\left(\mathbf{X} + \frac{1}{2}\boldsymbol{\Sigma}\right)}_{\mathbf{X}_1} + \underbrace{\left(\mathbf{Y} + \frac{1}{2}\boldsymbol{\Sigma}\right)}_{\mathbf{Y}_1} = \mathbf{X}_1 + \mathbf{Y}_1$$

Let us also assume that the matrices $\mathbf{X}_1$ and $\mathbf{Y}_1$ commute:

$$\mathbf{X}_1\mathbf{Y}_1 = \mathbf{Y}_1\mathbf{X}_1$$

The time advancement operator for a time step Δt can be written as follows:

$$e^{\Delta t\mathbf{A}} = e^{\Delta t(\mathbf{X}_1+\mathbf{Y}_1)} = e^{\Delta t\mathbf{X}_1}e^{\Delta t\mathbf{Y}_1} = e^{\frac{\Delta t}{2}\mathbf{X}_1}e^{\frac{\Delta t}{2}\mathbf{X}_1}e^{\frac{\Delta t}{2}\mathbf{Y}_1}e^{\frac{\Delta t}{2}\mathbf{Y}_1}$$

$$\approx \left(I - \frac{\Delta t}{2}\mathbf{X}_1\right)^{-1}\left(I + \frac{\Delta t}{2}\mathbf{X}_1\right)\left(I - \frac{\Delta t}{2}\mathbf{Y}_1\right)^{-1}\left(I + \frac{\Delta t}{2}\mathbf{Y}_1\right)$$

$$= \left(I - \frac{\Delta t}{2}\mathbf{Y}_1\right)^{-1}\left(I + \frac{\Delta t}{2}\mathbf{X}_1\right)\left(I - \frac{\Delta t}{2}\mathbf{X}_1\right)^{-1}\left(I + \frac{\Delta t}{2}\mathbf{Y}_1\right)$$

The last passage was possible since the matrices commute. Using this (Crank-Nicolson) approximation for the exponential is equivalent to a two-step advancement scheme

$$\begin{cases} \boldsymbol{\phi}^{n+1/2} = \left(I - \dfrac{\Delta t}{2}\mathbf{X}_1\right)^{-1}\left(I + \dfrac{\Delta t}{2}\mathbf{Y}_1\right)\boldsymbol{\phi}^{n} \\ \boldsymbol{\phi}^{n+1} = \left(I - \dfrac{\Delta t}{2}\mathbf{Y}_1\right)^{-1}\left(I + \dfrac{\Delta t}{2}\mathbf{X}_1\right)\boldsymbol{\phi}^{n+1/2} \end{cases}$$

or

$$\begin{cases} \left(I - \dfrac{\Delta t}{2}\mathbf{X}_1 \right)\boldsymbol{\phi}^{n+1/2} = \left(I + \dfrac{\Delta t}{2}\mathbf{Y}_1 \right)\boldsymbol{\phi}^{n} \\ \left(I - \dfrac{\Delta t}{2}\mathbf{Y}_1 \right)\boldsymbol{\phi}^{n+1} = \left(I + \dfrac{\Delta t}{2}\mathbf{X}_1 \right)\boldsymbol{\phi}^{n+1/2} \end{cases}$$

where it is noted that the two matrices in the LHS have *a three point structure*, that can be treated by the Thomas algorithm (TDMA) in similarity with LOR and ADI techniques for steady-state problems.

It is interesting to see that this is equivalent to write

$$\begin{cases} \dfrac{\boldsymbol{\phi}^{n+1/2} - \boldsymbol{\phi}^{n}}{\Delta t/2} = \mathbf{X}_1\boldsymbol{\phi}^{n+1/2} + \mathbf{Y}_1\boldsymbol{\phi}^{n} \\ \dfrac{\boldsymbol{\phi}^{n+1} - \boldsymbol{\phi}^{n+1/2}}{\Delta t/2} = \mathbf{Y}_1\boldsymbol{\phi}^{n+1} + \mathbf{X}_1\boldsymbol{\phi}^{n+1/2} \end{cases}$$

In the first step, the unknown vector components, $\boldsymbol{\phi}^{n+1/2}$, are "in the lines" and in the second step the unknowns $\boldsymbol{\phi}^{n+1}$ are "in the columns".

In fact, in double index notation for the simple case of the finite difference method with equally spaced nodes it is:

$$\begin{cases} \dfrac{1}{v}\dfrac{\phi_{i,j}^{n+1/2} - \phi_{i,j}^{n}}{\Delta t/2} = \dfrac{D}{h_x^2}\phi_{i+1,j}^{n+1/2} - \left(\dfrac{2D}{h_x^2} + \dfrac{\Sigma_a}{2} \right)\phi_{i,j}^{n+1/2} + \dfrac{D}{h_x^2}\phi_{i-1,j}^{n+1/2} + \dfrac{D}{h_y^2}\phi_{i,j+1}^{n} - \left(\dfrac{2D}{h_y^2} + \dfrac{\Sigma_a}{2} \right)\phi_{i,j}^{n} + \dfrac{D}{h_y^2}\phi_{i,j-1}^{n} \\ \dfrac{1}{v}\dfrac{\phi_{i,j}^{n+1} - \phi_{i,j}^{n+1/2}}{\Delta t/2} = \dfrac{D}{h_y^2}\phi_{i,j+1}^{n+1} - \left(\dfrac{2D}{h_y^2} + \dfrac{\Sigma_a}{2} \right)\phi_{i,j}^{n+1} + \dfrac{D}{h_y^2}\phi_{i,j-1}^{n+1} + \dfrac{D}{h_x^2}\phi_{i+1,j}^{n+1/2} - \left(\dfrac{2D}{h_x^2} + \dfrac{\Sigma_a}{2} \right)\phi_{i,j}^{n+1/2} + \dfrac{D}{h_x^2}\phi_{i-1,j}^{n+1/2} \end{cases}$$

Alternating Direction Explicit Scheme (ADE)

In this case, it is chosen to evaluate the "already known components" at the new time level and the others at the old time level:

$$\begin{cases} \dfrac{1}{v}\dfrac{\phi_{i,j}^{n+1/2}-\phi_{i,j}^{n}}{\Delta t/2} = \dfrac{D}{h_x^2}\phi_{i+1,j}^{n} - \left(\dfrac{2D}{h_x^2}+\dfrac{\Sigma_a}{2}\right)\phi_{i,j}^{n+1/2} + \dfrac{D}{h_x^2}\phi_{i-1,j}^{n+1/2} + \dfrac{D}{h_y^2}\phi_{i,j+1}^{n} - \left(\dfrac{2D}{h_y^2}+\dfrac{\Sigma_a}{2}\right)\phi_{i,j}^{n+1/2} + \dfrac{D}{h_y^2}\phi_{i,j-1}^{n+1/2} \\ \dfrac{1}{v}\dfrac{\phi_{i,j}^{n+1}-\phi_{i,j}^{n+1/2}}{\Delta t/2} = \dfrac{D}{h_y^2}\phi_{i,j+1}^{n+1/2} - \left(\dfrac{2D}{h_y^2}+\dfrac{\Sigma_a}{2}\right)\phi_{i,j}^{n+1} + \dfrac{D}{h_y^2}\phi_{i,j-1}^{n+1} + \dfrac{D}{h_x^2}\phi_{i+1,j}^{n+1/2} - \left(\dfrac{2D}{h_x^2}+\dfrac{\Sigma_a}{2}\right)\phi_{i,j}^{n+1} + \dfrac{D}{h_x^2}\phi_{i-1,j}^{n+1} \end{cases}$$

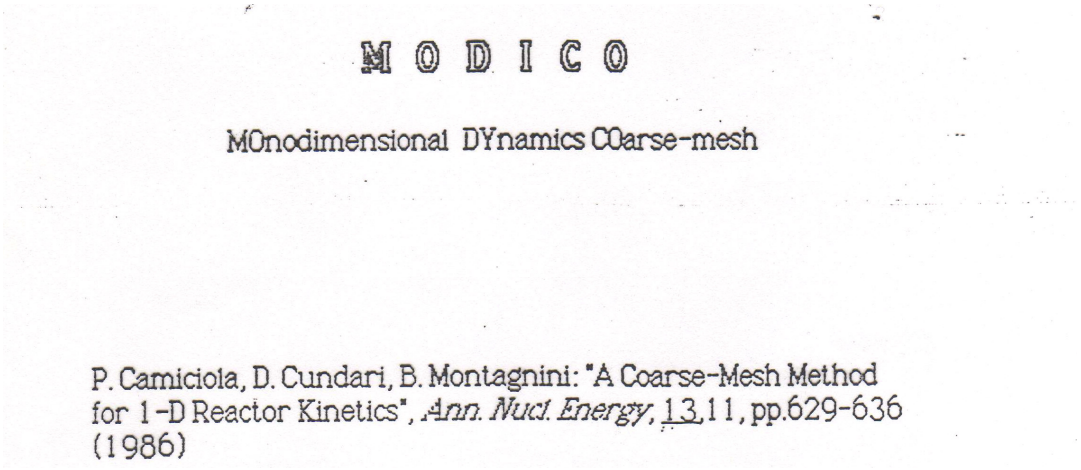
It can be noted that the formulation is completely explicit, though it results in a sort of Crank-Nicolson scheme. The value of the fluxes are "explicitly" evaluated at their most recent value.

ADE-ADI

$$\begin{cases} \dfrac{1}{v}\dfrac{\phi_{i,j}^{n+1/2}-\phi_{i,j}^{n}}{\Delta t/2} = \alpha\left[\dfrac{D}{h_x^2}\phi_{i+1,j}^{n+1/2} - \left(\dfrac{2D}{h_x^2}+\dfrac{\Sigma_a}{2}\right)\phi_{i,j}^{n+1/2} + \dfrac{D}{h_x^2}\phi_{i-1,j}^{n+1/2}\right] + (1-\alpha)\left[\dfrac{D}{h_x^2}\phi_{i+1,j}^{n} - \left(\dfrac{2D}{h_x^2}+\dfrac{\Sigma_a}{2}\right)\phi_{i,j}^{n} + \dfrac{D}{h_x^2}\phi_{i-1,j}^{n}\right] \\ \qquad + \dfrac{D}{h_y^2}\phi_{i,j+1}^{n} - \left(\dfrac{2D}{h_y^2}+\dfrac{\Sigma_a}{2}\right)\phi_{i,j}^{n+1/2} + \dfrac{D}{h_y^2}\phi_{i,j-1}^{n+1/2} \\ \dfrac{1}{v}\dfrac{\phi_{i,j}^{n+1}-\phi_{i,j}^{n+1/2}}{\Delta t/2} = \alpha\left[\dfrac{D}{h_y^2}\phi_{i,j+1}^{n+1} - \left(\dfrac{2D}{h_y^2}+\dfrac{\Sigma_a}{2}\right)\phi_{i,j}^{n+1} + \dfrac{D}{h_y^2}\phi_{i,j-1}^{n+1}\right] + (1-\alpha)\left[\dfrac{D}{h_y^2}\phi_{i,j+1}^{n+1/2} - \left(\dfrac{2D}{h_y^2}+\dfrac{\Sigma_a}{2}\right)\phi_{i,j}^{n+1/2} + \dfrac{D}{h_y^2}\phi_{i,j-1}^{n+1/2}\right] \\ \qquad + \dfrac{D}{h_x^2}\phi_{i+1,j}^{n+1/2} - \left(\dfrac{2D}{h_x^2}+\dfrac{\Sigma_a}{2}\right)\phi_{i,j}^{n+1} + \dfrac{D}{h_x^2}\phi_{i-1,j}^{n+1} \end{cases}$$

As it can be noted, both the characteristics of ADI (implicit evaluation along a line/column) and ADE (explicit evaluation of the fluxes at their most recent values) appear in this mixed algorithm

AN EXAMPLE OF APPLICATION TO 1D NEUTRON KINETICS OF COARSE-MESH SCHEMES



Multi-group 1D Kinetics Equations

$$\frac{1}{v^g}\frac{\partial \phi^g}{\partial t} = \frac{\partial}{\partial z} D^g \frac{\partial \phi^g}{\partial z} - \Sigma_r^g \phi^g + \sum_{g'<g} \Sigma_s^{g'\to g} \phi^{g'} + (1-\beta)\chi_p^g \sum_{g'=1}^{G} (\nu\Sigma_f)^{g'} \phi^{g'} + \sum_{r=1}^{R} \lambda_r \chi_d^g C_r$$

$$\frac{\partial C_r}{\partial t} = -\lambda_r C_r + \beta_r \sum_{g=1}^{G} (\nu\Sigma_f)^g \phi^g \qquad (1')$$

$$\phi^g(0,t) = \phi^g(H,t) = 0$$
$$C_r(0,t) = C_r(H,t) = 0$$

$$\phi^g(z,0) = \phi_0^g(z)$$
$$C_r(z,0) = C_{r0}(z)$$

Useful definitions

$$\underline{\phi}(z,t) = \begin{bmatrix} \phi^1(z,t) \\ \vdots \\ \phi^G(z,t) \end{bmatrix}, \quad \underline{C}(z,t) = \begin{bmatrix} C_1(z,t) \\ \vdots \\ C_R(z,t) \end{bmatrix}$$

$$\mathbb{V} = diag\left[v^1, \dots, v^G\right], \quad \mathbb{D} = diag\left[D^1, \dots, D^G\right]$$

$$\Sigma_r = diag\left[\Sigma_r^1, \dots, \Sigma_r^G\right], \quad \Lambda = diag\left[\lambda_1, \dots, \lambda_R\right]$$

$$\Sigma_s = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \Sigma_s^{1\to 2} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \Sigma_s^{1\to G} & \Sigma_s^{2\to G} & \dots & 0 \end{bmatrix}$$

$$\mathbb{F} = \begin{bmatrix} \chi_p^1(\nu\Sigma_f)^1 & \dots & \chi_p^1(\nu\Sigma_f)^G \\ \dots & \dots & \dots \\ \chi_p^G(\nu\Sigma_f)^1 & \dots & \chi_p^G(\nu\Sigma_f)^G \end{bmatrix}$$

$$\mathbb{B} = \begin{bmatrix} \beta_1(\nu\Sigma_f)^1 & \dots & \beta_1(\nu\Sigma_f)^G \\ \dots & \dots & \dots \\ \beta_R(\nu\Sigma_f)^1 & \dots & \beta_R(\nu\Sigma_f)^G \end{bmatrix}$$

$$\mathbb{\Gamma} = \begin{bmatrix} \chi_d^1\lambda_1 & \dots & \chi_d^1\lambda_R \\ \dots & \dots & \dots \\ \chi_d^G\lambda_1 & \dots & \chi_d^G\lambda_R \end{bmatrix}$$

$$\mathbb{E} = -\Sigma_r + \Sigma_s + (1-\beta)\mathbb{F}$$

COMPACT FORM OF THE KINETICS EQUATIONS

$$\mathbb{V}^{-1}\frac{\partial \underline{\phi}}{\partial t} = \left(\frac{\partial}{\partial z}\mathbb{D}\frac{\partial}{\partial z} + \mathbb{E}\right)\underline{\phi} + \mathbb{\Gamma}\cdot\underline{c} \qquad (2a)$$

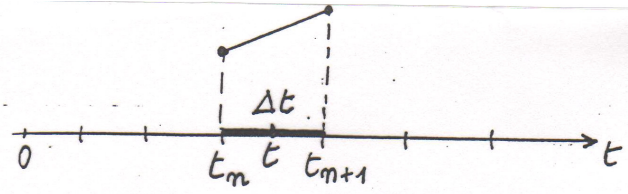
$$\frac{\partial \underline{c}}{\partial t} = \mathbb{B}\underline{\phi} - \mathbb{\Lambda}\underline{c} \qquad (2b)$$

where

$\mathbb{V}$, $\mathbb{D}$, $\mathbb{E}$ e $\mathbb{B}$

dependent on time and space in general

The assumption is made of a linear variation in time of flux and diffusion coefficients



$$D^g(z,t) = D^g(z,t_m) + \left[D^g(z,t_{m+1}) - D^g(z,t_m)\right]\frac{t-t_m}{\Delta t}$$

$$= \overset{m}{D}{}^g(z) + \left[\overset{m+1}{D}{}^g(z) - \overset{m}{D}(z)\right]\frac{t-t_m}{\Delta t}$$

Matrix form

$$\mathbb{D}(z,t) = \overset{m}{\mathbb{D}}(z) + \left[\overset{m+1}{\mathbb{D}}(z) - \overset{m}{\mathbb{D}}(z)\right]\frac{t-t_m}{\Delta t}$$

$$\underline{\phi}(z,t) = \overset{m}{\underline{\phi}}(z) + \left[\overset{m+1}{\underline{\phi}}(z) - \overset{m}{\underline{\phi}}(z)\right]\frac{t-t_m}{\Delta t}$$

The precursor concentrations are integrated directly

$$\underline{c}(z,t) = e^{-(t-t_m)\Lambda}\,\overset{m}{\underline{c}}(z) + \int_{t_m}^{t} e^{-(t'-t_m)\Lambda}\,\mathbb{B}(z,t')\,\underline{\phi}(z,t')\,dt' \qquad (3)$$

appross. lineare

With the assumption of the linear variation of flux and the other quantities, a "TIME INTEGRATED FORMULATION" is obtained

$$\overset{m+\frac{1}{3}}{\mathbb{D}} = \frac{2}{3}\overset{m}{\mathbb{D}} + \frac{1}{3}\overset{m+1}{\mathbb{D}}, \qquad \overset{m+\frac{2}{3}}{\mathbb{D}} = \frac{1}{3}\overset{m}{\mathbb{D}} + \frac{2}{3}\overset{m+1}{\mathbb{D}}$$

$\overset{m+\frac{1}{3}}{\mathbb{Q}}$ = matrice data da una espressione complicata in cui figurano

$\overset{m}{\mathbb{E}}, \overset{m+1}{\mathbb{E}}, \overset{m}{\mathbb{V}}, \overset{m+1}{\mathbb{V}}, \overset{m}{\mathbb{B}}, \overset{m+1}{\mathbb{B}}, \Gamma, \Lambda$

$\overset{m+\frac{2}{3}}{\mathbb{Q}}$ = matrice analoga

$$\mathbb{Z} = \frac{2}{\Delta t}\Gamma\Lambda^{-1}\left(1 - e^{-\Delta t\Lambda}\right)$$

$$\frac{d}{dz}\mathbb{D}^{n+\frac{2}{3}}\frac{d\underline{\phi}^{n+1}}{dz} + \mathbb{Q}^{n+\frac{2}{3}}\underline{\phi}^{n+1} = -\frac{d}{dz}\mathbb{D}^{n+\frac{1}{2}}\frac{d\underline{\phi}^{n}}{dz} - \mathbb{Q}^{n+\frac{1}{3}}\underline{\phi}^{n} - \mathbb{Z}\underline{c}^{n} \qquad (4a)$$

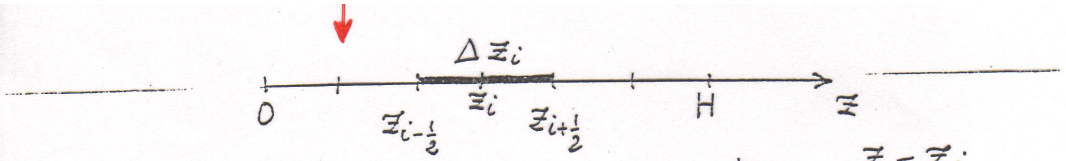
TIME INTEGRATED FORM OF KINETICS EQUATION

Complex analytical formulation for the precursor concentration

$$\underline{c}^{n+1} = e^{-\Delta t\Lambda}\underline{c}^{n} + (Pol\,\Lambda^{-1})\mathbb{B}^{n}\underline{\phi}^{n} + (Pol'\Lambda^{-1})(\mathbb{B}^{n}\underline{\phi}^{n+1} + \mathbb{B}^{n+1}\underline{\phi}^{n}) + (Pol''\Lambda^{-1})\mathbb{B}^{n+1}\underline{\phi}^{n+1} \qquad (4b)$$

$$(Pol\,\Lambda^{-1}) = \frac{2}{(\Delta t)^2}\Lambda^{-2} - \left[\mathbb{I} + \frac{2}{\Delta t}\Lambda^{-1} + \frac{2}{(\Delta t)^2}\Lambda^{-2}\right]e^{-\Delta t\Lambda} \quad \text{ecc.}$$

Space discretization by a piecewise cubic polynomial



$$\zeta = \frac{z - z_i}{\Delta z_i}$$

$$\phi^{g}(\zeta) = \phi_i^{g} + a_i^{g}\zeta + b_i^{g}\left(\zeta^2 - \frac{1}{12}\right) + c_i^{g}\left(\zeta - 4\zeta^3\right)$$

THE METHOD OF WEIGHTED RESIDUALS IS THEN APPLIED WITH:

- **the subdomain method**
- **the collocation method with** $\delta(\zeta - 1/2) - \delta(\zeta + 1/2)$

$$6D_i^{n+\frac{2}{3}g}\frac{\phi_{i+\frac{1}{2}}^{n+1g}-2\bar{\phi}_i^{n+1g}+\phi_{i-\frac{1}{2}}^{n+1g}}{(\Delta z_i)^2}+\sum_{g'=1}^{G}Q_i^{n+\frac{2}{3}gg'}\bar{\phi}_i^{n+1g'}$$

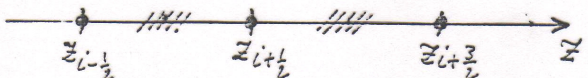
Three point equation in the interface fluxes

$$=-6D_i^{n+\frac{1}{3}g}\frac{\phi_{i+\frac{1}{2}}^{ng}-2\bar{\phi}_i^{ng}+\phi_{i-\frac{1}{2}}^{ng}}{(\Delta z_i)^2}-\sum_{g'=1}^{G}Q_i^{n+\frac{1}{3}gg'}\bar{\phi}_i^{ng'}-(\chi\bar{c}_i)^g$$

Continuity conditions at the interface

$$\phi_{i+\frac{1}{2}}^{n+1g}=\phi_{(i+1)-\frac{1}{2}}^{n+1g}$$

$$-D_i^{n+\frac{2}{3}}\frac{d\phi^{n+1g}}{dz}\Big|_{z=z_{i+\frac{1}{2}}}=-D_{i+1}^{n+\frac{2}{3}}\frac{d\phi^{n+1g}}{dz}\Big|_{z=z_{(i+1)-\frac{1}{2}}}$$



The final balance equations are written in terms of interfacial fluxes

$$X_i^{n+1}\underline{\phi}_{i-\frac{1}{2}}^{n+1}+\left(Y_i^{n+1}+Y_{i+1}^{n+1}\right)\underline{\phi}_{i+\frac{1}{2}}^{n+1}+X_{i+1}^{n+1}\underline{\phi}_{i+\frac{3}{2}}^{n+1}=\underline{\tilde{b}}_i^{n}$$

THE COARSE-MESH 3D NEUTRON KINETICS OF DPOL3DA

Neutron kinetics equations

$$\frac{1}{v^g}\frac{\partial \varphi^g}{\partial t} = \nabla \cdot D^g \nabla \varphi^g - \Sigma_r^g \varphi^g + \sum_{g'=1}^{G} \Sigma_s^{g' \to g} \varphi^{g'}$$

$$+(1-\beta)\chi_p^g \sum_{g'=1}^{G} \nu\Sigma_f^{g'} \varphi^{g'} + \sum_{l=1}^{L} \lambda^l \chi_{dl}^g c^l,$$

$$\frac{\partial c^l}{\partial t} = -\lambda^l c^l + \beta^l \sum_{g=1}^{G} \nu\Sigma_f^g \varphi^g \qquad g = 1, \cdots, G,$$

$$l = 1, \cdots, L$$

Boundary and initial conditions

$$\sigma^g \varphi^g + \tau^g \left.\frac{\partial \varphi^g}{\partial u}\right|_{\partial V} = 0,$$

$$\varphi^g(\mathbf{r},0) = \varphi_0^g(\mathbf{r})$$

$$c^l(\mathbf{r},0) = c_0^l(\mathbf{r})$$

Compact form of the equations

$$\begin{cases} \mathbf{V}^{-1}\dfrac{\partial \Phi}{\partial t} = (\nabla \cdot \mathbf{D}\nabla + \mathbf{E})\Phi + \Gamma \mathbf{c} \\ \dfrac{\partial \mathbf{c}}{\partial t} = \mathbf{B}\Phi - \Lambda \mathbf{c} \end{cases}$$

where it was put

$$\Phi(\mathbf{r},t)=\mathrm{col}\left[\varphi^1,\ldots,\varphi^G\right], \qquad \mathbf{c}(\mathbf{r},t)=\mathrm{col}\left[c^1,\ldots,c^L\right],$$

$$\mathbf{V}=\mathrm{diag}\left[\nu^1,\ldots,\nu^G\right], \qquad \mathbf{D}(\mathbf{r},t)=\mathrm{diag}\left[D^1,\ldots,D^G\right],$$

$$\mathbf{S}=\mathrm{diag}\left[\sigma^1,\ldots,\sigma^G\right], \qquad \mathbf{T}=\mathrm{diag}\left[\tau^1,\ldots,\tau^G\right],$$

$$\mathbf{G}(\mathbf{r},t)=\mathrm{diag}\left[\Sigma^1,\ldots,\Sigma^G\right], \qquad \Lambda=\mathrm{diag}\left[\lambda^1,\ldots,\lambda^L\right]$$

$$\mathbf{G}_s(\mathbf{r},t)=\begin{bmatrix}0 & 0 & \cdots & 0\\ \Sigma_{sl}^{1\to 2} & 0 & \cdots & 0\\ \cdots & \cdots & \cdots & \cdots\\ \Sigma_{sl}^{1\to G} & \Sigma_{sl}^{2\to G} & \cdots & 0\end{bmatrix}, \qquad \mathbf{F}(\mathbf{r},t)=\begin{bmatrix}\chi_p^1\nu\Sigma_f^1 & \cdots & \chi_p^1\nu\Sigma_f^G\\ \cdots & \cdots & \cdots\\ \chi_p^G\nu\Sigma_f^1 & \cdots & \chi_p^G\nu\Sigma_f^G\end{bmatrix},$$

$$\mathbf{B}(\mathbf{r},t)=\begin{bmatrix}\beta^1\nu\Sigma_f^1 & \cdots & \beta^1\nu\Sigma_f^G\\ \cdots & \cdots & \cdots\\ \beta^L\nu\Sigma_f^1 & \cdots & \beta^L\nu\Sigma_f^G\end{bmatrix}, \qquad \Gamma=\begin{bmatrix}\chi_{d_1}^1\lambda^1 & \cdots & \chi_{d_L}^1\lambda^L\\ \cdots & \cdots & \cdots\\ \chi_{d_1}^G\lambda^1 & \cdots & \chi_{d_L}^G\lambda^L\end{bmatrix}$$

$$\mathbf{E}(\mathbf{r},t)\equiv-\mathbf{G}(\mathbf{r},t)+\mathbf{G}(\mathbf{r},t)_s+(1-\beta)\mathbf{F}(\mathbf{r},t)$$

Time integrated approach: at each step the linear variation in time of the flux and the diffusion coefficients are assumed

$$\Phi(\mathbf{r},t)=\Phi^n(\mathbf{r})+\frac{t-t_n}{\Delta t}\left[\Phi^{n+1}(\mathbf{r})-\Phi^n(\mathbf{r})\right], \quad t\in\left[t_n,t_{n+1}\right].$$

$$\mathbf{D}(\mathbf{r},t)=\mathbf{D}^n(\mathbf{r})+\frac{t-t_n}{\Delta t}\left[\mathbf{D}^{n+1}(\mathbf{r})-\mathbf{D}^n(\mathbf{r})\right], \quad t\in\left[t_n,t_{n+1}\right]$$

Direct integration of the equation of precursor concentrations

$$\mathbf{c}(\mathbf{r},t) = \exp\left[-(t-t_n)\Lambda\right]\mathbf{c}^n(\mathbf{r}) + \int_{t_n} \exp\left[-(t-t')\Lambda\right]\mathbf{B}(\mathbf{r},t')\Phi(\mathbf{r},t')dt'$$

$$= \exp\left[-(t-t_n)\Lambda\right]\mathbf{c}^n(\mathbf{r}) + \int_{t_n}^{t} \exp\left[-(t-t')\Lambda\right]\times$$

$$\times\left\{\mathbf{B}^n(\mathbf{r}) + \frac{t'-t_n}{\Delta t}\left[\mathbf{B}^{n+1}(\mathbf{r}) - \mathbf{B}^n(\mathbf{r})\right]\right\}\times\left\{\Phi^n(\mathbf{r}) + \frac{t'-t_n}{\Delta t}\left[\Phi^{n+1}(\mathbf{r}) - \Phi^n(\mathbf{r})\right]\right\}dt',$$

$$t \in \left[t_n, t_{n+1}\right]$$

Time integration of the kinetics equation

$$\nabla\cdot\mathbf{D}^{n+2/3}\nabla\Phi^{n+1} + \mathbf{Q}^{n+2/3}\Phi^{n+1} = -\nabla\cdot\mathbf{D}^{n+1/3}\nabla\Phi^n - \mathbf{Q}^{n+1/3}\Phi^n - \mathbf{W}\mathbf{c}^n$$

Definition of appearing constants

$$\mathbf{D}^{n+1/3} \equiv (2/3)\mathbf{D}^n + (1/3)\mathbf{D}^{n+1}$$

$$\mathbf{D}^{n+2/3} \equiv (1/3)\mathbf{D}^n + (2/3)\mathbf{D}^{n+1}$$

$$\mathbf{Q}^{n+1/3} \equiv \mathbf{E}^{n+1/3} + (1/\Delta t)\left[\left(\mathbf{V}^n\right)^{-1} + \left(\mathbf{V}^{n+1}\right)^{-1}\right] + \Gamma\Lambda^{-1}\left(\mathbf{P}_3\mathbf{B}^n + \mathbf{P}_1\mathbf{B}^{n+1}\right)$$

$$\mathbf{Q}^{n+2/3} \equiv \mathbf{E}^{n+2/3} - (1/\Delta t)\left[\left(\mathbf{V}^n\right)^{-1} + \left(\mathbf{V}^{n+1}\right)^{-1}\right] + \Gamma\Lambda^{-1}\left(\mathbf{P}_1\mathbf{B}^n + \mathbf{P}_2\mathbf{B}^{n+1}\right)$$

$$\mathbf{E}^{n+1/3} \equiv (2/3)\mathbf{E}^n + (1/3)\mathbf{E}^{n+1} \qquad \mathbf{E}^{n+2/3} \equiv (1/3)\mathbf{E}^n + (2/3)\mathbf{E}^{n+1}$$

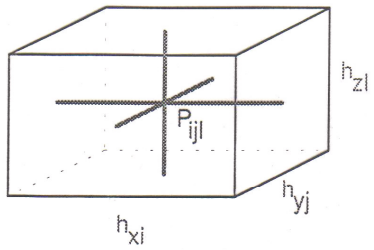
Closed expression of the precursor concentration

$$
\begin{aligned}
\mathrm{c}^{n+1} &= \exp(-\Delta t\Lambda)\mathrm{c}^{n} \\
&+(\Delta t)\left\{\frac{2}{(\Delta t)^3}\Lambda^{-3}-\left[\frac{1}{\Delta t}\Lambda^{-1}+\frac{2}{(\Delta t)^2}\Lambda^{-2}+\frac{2}{(\Delta t)^3}\Lambda^{-3}\right]\exp(-\Delta t\Lambda)\right\}\mathbf{B}^{n}\Phi^{n} \\
&+(\Delta t)\left\{\frac{1}{(\Delta t)^2}\Lambda^{-2}-\frac{2}{(\Delta t)^3}\Lambda^{-3}+\left[\frac{1}{(\Delta t)^2}\Lambda^{-2}+\frac{2}{(\Delta t)^3}\Lambda^{-3}\right]\exp(-\Delta t\Lambda)\right\}\times \\
&\times\left(\mathbf{B}^{n}\Phi^{n+1}+\mathbf{B}^{n+1}\Phi^{n}\right) \\
&+(\Delta t)\times\left[\frac{1}{\Delta t}\Lambda^{-1}-\frac{2}{(\Delta t)^2}\Lambda^{-2}+\frac{2}{(\Delta t)^3}\Lambda^{-3}\left(\mathbf{I}-\exp(-\Delta t\Lambda)\right)\right]\mathbf{B}^{n+1}\Phi^{n+1}
\end{aligned}
$$

General form of the time integrated kinetics equation

$$
\nabla\cdot\mathbf{D}^{n+2/3}\nabla\Phi^{n+1}+\mathbf{Q}^{n+2/3}\Phi^{n+1}+\mathrm{s}^{n+1}=0
$$

Spatial discretization with the coarse-mesh technique



Polynomial expansion in space

$$\xi = \frac{x - x_i}{h_{x,i}},\ \eta = \frac{y - y_j}{h_{y,j}},\ \zeta = \frac{z - z_k}{h_{z,k}}$$

$$\varphi^{n,g}(x,y,z) = \varphi^{n,g}_{i,j,k}(\xi,\eta,\zeta) = \overline{\varphi}^{n,g}_{i,j,k} + a^{n,g}_{1,x}\xi + a^{n,g}_{2,x}\left(\xi^2 - \frac{1}{12}\right) + a^{n,g}_{3,x}\left(\xi - 4\xi^3\right)$$
$$+ a^{n,g}_{1,y}\eta + a^{n,g}_{2,y}\left(\eta^2 - \frac{1}{12}\right) + a^{n,g}_{3,y}\left(\eta - 4\eta^3\right)$$
$$+ a^{n,g}_{1,z}\zeta + a^{n,g}_{2,z}\left(\zeta^2 - \frac{1}{12}\right) + a^{n,g}_{3,z}\left(\zeta - 4\zeta^3\right)$$

Relations among constants and interfacial fluxes

$$a^{n,g}_{x,1} = \varphi^{n,g}_{i+1/2,jk} - \varphi^{n,g}_{i-1/2,jk}$$

$$a^{n,g}_{x,2} = 3(\varphi^{n,g}_{i+1/2,jk} - 2\overline{\varphi}^{n,g}_{ijk} + \varphi^{n,g}_{i-1/2,jk}).$$

Applications of the weighted residuals method

$$\mathbf{w^g = 1}$$

$$\mathbf{w^g_\xi = (2p+3)\ 2^{2p+2}\ \xi^{2p+3}} \qquad \mathbf{w^g_\eta = (2p+3)\ 2^{2p+2}\ \eta^{2p+3}}$$

$$\mathbf{w^g_\zeta = (2p+3)\ 2^{2p+2}\ \zeta^{2p+3}} \qquad \mathbf{(g = 1, ..., G)}$$

limit for p → ∞ on the weighting

$$a_{3,x}^{n+1,g} = \frac{h_{x,i}^2}{24 D_{ijk}^{n+2/3,g}} \times \sum_{g'=1}^{G} Q_{ijk}^{n+2/3,gg'} \left(\varphi_{i+1/2,jk}^{n+1,g'} - \varphi_{i-1/2,jk}^{n+1,g'} \right) + R_{ijk}^{n,g}$$

obtaining

$$2 D_{ijk}^{n+2/3,g} \left(\frac{a_{2,x}^{n+1,g}}{h_{x,i}^2} + \frac{a_{2,y}^{n+1,g}}{h_{y,j}^2} + \frac{a_{2,z}^{n+1,g}}{h_{z,k}^2} \right) + \sum_{g'=1}^{G} Q_{ijk}^{n+2/3,gg'} \overline{\varphi}_{ijk}^{n+1,g'} =$$

$$= -2 D_{ijk}^{n+1/3,g} \left(\frac{a_{2,x}^{n,g}}{h_{x,i}^2} + \frac{a_{2,y}^{n,g}}{h_{y,j}^2} + \frac{a_{2,z}^{n,g}}{h_{z,k}^2} \right) - \sum_{g'=1}^{G} Q_{ijk}^{n+1/3,gg'} \overline{\varphi}_{ijk}^{n,g'} - \sum_{l=1}^{L} W^{g,l} c_{ijk}^{l}$$

$$\left[(\mathsf{X}+\mathsf{Y}+\mathsf{Z}) \Phi^{n+1} \right]_{ijk} + \mathsf{Q}_{ijk}^{*n+1} \Phi_{ijk}^{n+1} = -\mathsf{s}_{ijk}^{*n+1} \qquad (*)$$

$$\mathsf{Q}_{ijk}^{*n+1} = \frac{1}{12} \left(\mathsf{D}_{ijk}^{n+2/3} \right)^{-1} \mathsf{Q}_{ijk}^{n+1/3}$$

$$(\mathsf{X}\Phi)_{ijk} = \frac{\Phi_{i+1/2,jk}^{n+1} - 2\Phi_{ijk}^{n+1} + \Phi_{i-1/2,jk}^{n+1}}{2 h_{x,i}^2}$$

$$(\mathsf{Y}\Phi)_{ijk} = \frac{\Phi_{i,j+1/2,k}^{n+1} - 2\Phi_{ijk}^{n+1} + \Phi_{i,j-1/2,k}^{n+1}}{2 h_{y,j}^2}$$

$$(\mathsf{Z}\Phi)_{ijk} = \frac{\Phi_{ij,k+1/2}^{n+1} - 2\Phi_{ijk}^{n+1} + \Phi_{ij,k-1/2}^{n+1}}{2 h_{z,k}^2}$$

EXAMPLE OF 2-GROUP NEUTRON KINETICS FOR A 1D REACTOR WITH VARIABLE COMPOSITION

1. General Two-Group Equations of Kinetics with Explicit Source

The equations of kinetics with a source are written as:

$$\begin{aligned}
&\frac{1}{v_1}\frac{\partial \phi_1}{\partial t} = \nabla \cdot D_1 \nabla \phi_1 - \Sigma_{a,1}\phi_1 - \Sigma_{s,1\to 2}\phi_1 + (1-\beta)\left(\nu\Sigma_{f,1}\phi_1 + \nu\Sigma_{f,2}\phi_2\right) + \sum_{d=1}^{N_d}\lambda_d C_d + S_1 \\
&\frac{1}{v_2}\frac{\partial \phi_2}{\partial t} = \nabla \cdot D_2 \nabla \phi_2 - \Sigma_{a,2}\phi_1 + \Sigma_{s,1\to 2}\phi_1 + S_2 \\
&\frac{\partial C_d}{\partial t} = -\lambda_d C_d + \beta_d\left(\nu\Sigma_{f,1}\phi_1 + \nu\Sigma_{f,2}\phi_2\right) \qquad (d = 1,\ldots,N_d)
\end{aligned} \tag{1}$$

where it has been implicitly assumed that all fission neutrons are produced in the fast energy group.

For a 1D slab the equations are written as:

$$\begin{aligned}
&\frac{1}{v_1}\frac{\partial \phi_1}{\partial t} = \frac{\partial}{\partial x}D_1\frac{\partial \phi_1}{\partial x} - \Sigma_{a,1}\phi_1 - \Sigma_{s,1\to 2}\phi_1 + (1-\beta)\left(\nu\Sigma_{f,1}\phi_1 + \nu\Sigma_{f,2}\phi_2\right) + \sum_{d=1}^{N_d}\lambda_d C_d + S_1 \\
&\frac{1}{v_2}\frac{\partial \phi_2}{\partial t} = \frac{\partial}{\partial x}D_2\frac{\partial \phi_2}{\partial x} - \Sigma_{a,2}\phi_2 + \Sigma_{s,1\to 2}\phi_1 + S_2 \\
&\frac{\partial C_d}{\partial t} = -\lambda_d C_d + \beta_d\left(\nu\Sigma_{f,1}\phi_1 + \nu\Sigma_{f,2}\phi_2\right) \qquad (d = 1,\ldots,N_d)
\end{aligned} \tag{2}$$

2. Eigenvalue problem

In steady-state conditions, assuming that no source is present, the equations of 1D kinetics reduce to the following form

$$\begin{aligned}
&\frac{\partial}{\partial x}D_1\frac{\partial}{\partial x}\phi_1 - \Sigma_{a,1}\phi_1 - \Sigma_{s,1\to 2}\phi_1 + (1-\beta)\left(\nu\Sigma_{f,1}\phi_1 + \nu\Sigma_{f,2}\phi_2\right) + \sum_{d=1}^{N_d}\lambda_d C_d = 0 \\
&\frac{\partial}{\partial x}D_2\frac{\partial}{\partial x}\phi_2 - \Sigma_{a,2}\phi_2 + \Sigma_{s,1\to 2}\phi_1 = 0 \\
&\lambda_d C_d = \beta_d\left(\nu\Sigma_{f,1}\phi_1 + \nu\Sigma_{f,2}\phi_2\right) \qquad (d = 1,\ldots,N_d)
\end{aligned} \tag{3}$$

Reformulating the above by considering the latter expression for the delayed precursors' activity, and *introducing the eigenvalue*, it is:

$$\frac{\partial}{\partial x} D_1 \frac{\partial \phi_1}{\partial x} - \Sigma_{a,1}\phi_1 - \Sigma_{s,1\to 2}\phi_1 + \frac{1}{k}\left(\nu\Sigma_{f,1}\phi_1 + \nu\Sigma_{f,2}\phi_2\right) = 0$$
$$\frac{\partial}{\partial x} D_2 \frac{\partial \phi_2}{\partial x} - \Sigma_{a,2}\phi_2 + \Sigma_{s,1\to 2}\phi_1 = 0 \qquad \textbf{(4)}$$

3. Numerical discretization of transient equations

By subdividing the range of the *x* variable (0,a) into *N* intervals, in each interval we have:

$$\left(\phi_{1,i}^{n+1} - \phi_{1,i}^{n}\right)\frac{\Delta x_i}{v_1 \Delta t} = -D_{1,i}\frac{\phi_{1,i}^{n,n+1} - \phi_{1,i-1/2}^{n,n+1}}{x_i - x_{i-1/2}} + D_{1,i}\frac{\phi_{1,i+1/2}^{n,n+1} - \phi_{1,i}^{n,n+1}}{x_{i+1/2} - x_i} - \Sigma_{a,1,i}\phi_{1,i}^{n,n+1}\Delta x_i - \Sigma_{s,1\to 2}\phi_{1,i}^{n,n+1}\Delta x_i$$
$$+ (1-\beta)\left(\nu\Sigma_{f,1}\phi_{1,i}^{n,n+1} + \nu\Sigma_{f,2}\phi_{2,i}^{n,n+1}\right)\Delta x_i + \sum_{d=1}^{N_d}\lambda_d C_{d,i}^{n,n+1}\Delta x_i + S_1^n \Delta x_i$$

$$\left(\phi_{2,i}^{n+1} - \phi_{2,i}^{n}\right)\frac{\Delta x_i}{v_2 \Delta t} = -D_{2,i}\frac{\phi_{2,i}^{n,n+1} - \phi_{2,i-1/2}^{n,n+1}}{x_i - x_{i-1/2}} + D_{2,i}\frac{\phi_{2,i+1/2}^{n,n+1} - \phi_{2,i}^{n,n+1}}{x_{i+1/2} - x_i} - \Sigma_{a,2,i}\phi_{2,i}^{n,n+1}\Delta x_i + \Sigma_{s,1\to 2}\phi_{1,i}^{n,n+1}\Delta x_i + S_2^n \Delta x_i$$

$$C_d^{n+1} = C_d^n\, e^{-\lambda_d \Delta t} + \frac{\beta_d}{\lambda_d}\left(\nu\Sigma_{f,1}\phi_{1,i}^{n,n+1} + \nu\Sigma_{f,2}\phi_{2,i}^{n,n+1}\right)\left(1 - e^{-\lambda_d \Delta t}\right) \qquad (d = 1,\ldots,N_d)$$

(5)

where the superscript *n,n+1* indicates an appropriate means between the values of the variable at the time t^n and t^{n+1}. As it can be noted, a semi-analytic expression was used for integrating the neutron precursors' equations. Moreover, it is assumed:

$$\phi_{1,i}^{n,n+1} = (1-\theta)\,\phi_{1,i}^{n} + \theta\,\phi_{1,i}^{n+1} \qquad \textbf{(6)}$$

where $0 \le \theta \le 1$; similar relationships hold for the other variables.

Concerning the fractional index fluxes, the usual relationships implying the continuity of neutron current allows eliminating them. It is:

$$-D_{1,i}\frac{\phi_{1,i}^{n,n+1}-\phi_{1,i-1/2}^{n,n+1}}{x_i-x_{i-1/2}}=-D_{1,i-1}\frac{\phi_{1,i-1/2}^{n,n+1}-\phi_{1,i-1}^{n,n+1}}{x_{i-1/2}-x_{i-1}} \qquad \textbf{(7)}$$

and then

$$-D_{1,i}\frac{\phi_{1,i}^{n,n+1}}{x_i-x_{i-1/2}}+D_{1,i}\frac{\phi_{1,i-1/2}^{n,n+1}}{x_i-x_{i-1/2}}=-D_{1,i-1}\frac{\phi_{1,i-1/2}^{n,n+1}}{x_{i-1/2}-x_{i-1}}+D_{1,i-1}\frac{\phi_{1,i}^{n,n+1}}{x_{i-1/2}-x_{i-1}}$$

$$\left[\frac{D_{1,i}}{x_i-x_{i-1/2}}+\frac{D_{1,i-1}}{x_{i-1/2}-x_{i-1}}\right]\phi_{1,i-1/2}^{n,n+1}=\left[\frac{D_{1,i-1}}{x_{i-1/2}-x_{i-1}}\phi_{1,i-1}^{n,n+1}+\frac{D_{1,i}}{x_i-x_{i-1/2}}\phi_{1,i}^{n,n+1}\right]$$

$$\left[\frac{2D_{1,i}}{\Delta x_i}+\frac{2D_{1,i-1}}{\Delta x_{i-1}}\right]\phi_{1,i-1/2}^{n,n+1}=\left[\frac{2D_{1,i-1}}{\Delta x_{i-1}}\phi_{1,i-1}^{n,n+1}+\frac{2D_{1,i}}{\Delta x_i}\phi_{1,i}^{n,n+1}\right]$$

$$\phi_{1,i-1/2}^{n,n+1}=\frac{\dfrac{D_{1,i-1}}{\Delta x_{i-1}}}{\dfrac{D_{1,i-1}}{\Delta x_{i-1}}+\dfrac{D_{1,i}}{\Delta x_i}}\phi_{1,i-1}^{n,n+1}+\frac{\dfrac{D_{1,i}}{\Delta x_i}}{\dfrac{D_{1,i-1}}{\Delta x_{i-1}}+\dfrac{D_{1,i}}{\Delta x_i}}\phi_{1,i}^{n,n+1} \qquad \textbf{(8)}$$

Taking into account the above relationship, it is:

$$-D_{1,i}\frac{\phi_{1,i}^{n,n+1}-\phi_{1,i-1/2}^{n,n+1}}{x_i-x_{i-1/2}}=-\frac{2D_{1,i}}{\Delta x_i}\phi_{1,i}^{n,n+1}+\frac{2D_{1,i}}{\Delta x_i}\phi_{1,i-1/2}^{n,n+1}$$

$$=-\frac{2D_{1,i}}{\Delta x_i}\phi_{1,i}^{n,n+1}+\frac{2D_{1,i}}{\Delta x_i}\left[\frac{\dfrac{D_{1,i-1}}{\Delta x_{i-1}}}{\dfrac{D_{1,i-1}}{\Delta x_{i-1}}+\dfrac{D_{1,i}}{\Delta x_i}}\phi_{1,i-1}^{n,n+1}+\frac{\dfrac{D_{1,i}}{\Delta x_i}}{\dfrac{D_{1,i-1}}{\Delta x_{i-1}}+\dfrac{D_{1,i}}{\Delta x_i}}\phi_{1,i}^{n,n+1}\right]$$

$$=-\frac{2D_{1,i}}{\Delta x_i}\phi_{1,i}^{n,n+1}+\frac{\dfrac{2D_{1,i}}{\Delta x_i}\dfrac{D_{1,i-1}}{\Delta x_{i-1}}}{\dfrac{D_{1,i-1}}{\Delta x_{i-1}}+\dfrac{D_{1,i}}{\Delta x_i}}\phi_{1,i-1}^{n,n+1}+\frac{\dfrac{2D_{1,i}}{\Delta x_i}\dfrac{D_{1,i}}{\Delta x_i}}{\dfrac{D_{1,i-1}}{\Delta x_{i-1}}+\dfrac{D_{1,i}}{\Delta x_i}}\phi_{1,i}^{n,n+1}$$

$$=\frac{\dfrac{2D_{1,i}}{\Delta x_i}\dfrac{D_{1,i-1}}{\Delta x_{i-1}}}{\dfrac{D_{1,i-1}}{\Delta x_{i-1}}+\dfrac{D_{1,i}}{\Delta x_i}}\phi_{1,i-1}^{n,n+1}-\frac{\dfrac{2D_{1,i}}{\Delta x_i}\dfrac{D_{1,i-1}}{\Delta x_i}}{\dfrac{D_{1,i-1}}{\Delta x_{i-1}}+\dfrac{D_{1,i}}{\Delta x_i}}\phi_{1,i}^{n,n+1}=-\frac{2\dfrac{D_{1,i-1}}{\Delta x_{i-1}}\dfrac{D_{1,i}}{\Delta x_i}}{\dfrac{D_{1,i-1}}{\Delta x_{i-1}}+\dfrac{D_{1,i}}{\Delta x_i}}\left(\phi_{1,i}^{n,n+1}-\phi_{1,i-1}^{n,n+1}\right)$$

Therefore, putting

$$\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1} = \frac{2\dfrac{D_{1,i-1}}{\Delta x_{i-1}}\dfrac{D_{1,i}}{\Delta x_i}}{\dfrac{D_{1,i-1}}{\Delta x_{i-1}} + \dfrac{D_{1,i}}{\Delta x_i}} \qquad \textbf{(9)}$$

it is

$$-D_{1,i}\frac{\phi_{1,i}^{n,n+1} - \phi_{1,i-1/2}^{n,n+1}}{x_i - x_{i-1/2}} = -\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1}\left(\phi_{1,i}^{n,n+1} - \phi_{1,i-1}^{n,n+1}\right) \qquad \textbf{(10)}$$

Similarly, it is also:

$$D_{1,i}\frac{\phi_{1,i+1/2}^{n,n+1} - \phi_{1,i}^{n,n+1}}{x_i - x_{i-1/2}} = \overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i+1}\left(\phi_{1,i+1}^{n,n+1} - \phi_{1,i}^{n,n+1}\right) \qquad \textbf{(11)}$$

and

$$-D_{2,i}\frac{\phi_{2,i}^{n,n+1} - \phi_{2,i-1/2}^{n,n+1}}{x_i - x_{i-1/2}} = -\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i-1}\left(\phi_{2,i}^{n,n+1} - \phi_{2,i-1}^{n,n+1}\right) \qquad \textbf{(12)}$$

$$D_{2,i}\frac{\phi_{2,i+1/2}^{n,n+1} - \phi_{2,i}^{n,n+1}}{x_i - x_{i-1/2}} = \overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i+1}\left(\phi_{2,i+1}^{n,n+1} - \phi_{2,i}^{n,n+1}\right) \qquad \textbf{(13)}$$

With the above definitions it is:

$$\left(\phi_{1,i}^{n+1} - \phi_{1,i}^{n}\right)\frac{\Delta x_i}{v_1 \Delta t} = -\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1}\left(\phi_{1,i}^{n,n+1} - \phi_{1,i-1}^{n,n+1}\right) + \overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i+1}\left(\phi_{1,i+1}^{n,n+1} - \phi_{1,i}^{n,n+1}\right) - \Sigma_{a,1,i}\phi_{1,i}^{n,n+1}\Delta x_i - \Sigma_{s,1\to 2}\phi_{1,i}^{n,n+1}\Delta x_i$$
$$+ (1-\beta)\left(\nu\Sigma_{f,1}\phi_{1,i}^{n,n+1} + \nu\Sigma_{f,2}\phi_{2,i}^{n,n+1}\right)\Delta x_i + \sum_{d=1}^{N_d}\lambda_d C_{d,i}^{n,n+1}\Delta x_i + S_1^n \Delta x_i$$

$$\left(\phi_{2,i}^{n+1} - \phi_{2,i}^{n}\right)\frac{\Delta x_i}{v_2 \Delta t} = -\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i-1}\left(\phi_{2,i}^{n,n+1} - \phi_{2,i-1}^{n,n+1}\right) + \overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i+1}\left(\phi_{2,i+1}^{n,n+1} - \phi_{2,i}^{n,n+1}\right) - \Sigma_{a,2,i}\phi_{2,i}^{n,n+1}\Delta x_i + \Sigma_{s,1\to 2}\phi_{1,i}^{n,n+1}\Delta x_i + S_2^n \Delta x_i$$

$$C_d^{n+1} = C_d^n e^{-\lambda_d \Delta t} + \frac{\beta_d}{\lambda_d}\left(\nu\Sigma_{f,1}\phi_{1,i}^{n,n+1} + \nu\Sigma_{f,2}\phi_{2,i}^{n,n+1}\right)\left(1 - e^{-\lambda_d \Delta t}\right) \qquad (d = 1, \ldots, N_d)$$

(14)

By elaborating the equation for the first energy group, we have:

$$\left(\phi_{1,i}^{n+1}-\phi_{1,i}^{n}\right)\frac{\Delta x_i}{v_1\Delta t}=\theta\left[-\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1}\left(\phi_{1,i}^{n+1}-\phi_{1,i-1}^{n+1}\right)+\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i+1}\left(\phi_{1,i+1}^{n+1}-\phi_{1,i}^{n+1}\right)-\Sigma_{a,1,i}\phi_{1,i}^{n+1}\Delta x_i-\Sigma_{s,1\to 2}\phi_{1,i}^{n+1}\Delta x_i\right.$$
$$\left.+(1-\beta)\left(\nu\Sigma_{f,1}\phi_{1,i}^{n+1}+\nu\Sigma_{f,2}\phi_{2,i}^{n+1}\right)\Delta x_i+\sum_{d=1}^{N_d}\lambda_d C_{d,i}^{n+1}\Delta x_i\right]$$
$$+(1-\theta)\left[-\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1}\left(\phi_{1,i}^{n}-\phi_{1,i-1}^{n}\right)+\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i+1}\left(\phi_{1,i+1}^{n}-\phi_{1,i}^{n}\right)-\Sigma_{a,1,i}\phi_{1,i}^{n}\Delta x_i-\Sigma_{s,1\to 2}\phi_{1,i}^{n}\Delta x_i\right.$$
$$\left.+(1-\beta)\left(\nu\Sigma_{f,1}\phi_{1,i}^{n}+\nu\Sigma_{f,2}\phi_{2,i}^{n}\right)\Delta x_i+\sum_{d=1}^{N_d}\lambda_d C_{d,i}^{n}\Delta x_i\right]+S_1^{n}\Delta x_i$$

or

$$-\theta\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1}\phi_{1,i-1}^{n+1}$$
$$+\left\{\frac{\Delta x_i}{v_1\Delta t}+\theta\left[\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1}+\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i+1}+\Sigma_{a,1,i}\Delta x_i+\Sigma_{s,1\to 2}\Delta x_i-(1-\beta)\nu\Sigma_{f,1}\Delta x_i\right]\right\}\phi_{1,i}^{n+1}$$
$$-\theta\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i+1}\phi_{1,i+1}^{n+1}$$
$$=\theta\left[(1-\beta)\left(\nu\Sigma_{f,2}\phi_{2,i}^{n+1}\right)\Delta x_i+\sum_{d=1}^{N_d}\lambda_d C_{d,i}^{n+1}\Delta x_i\right]$$
$$+\frac{\Delta x_i}{v_1\Delta t}\phi_{1,i}^{n}+(1-\theta)\left[-\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1}\left(\phi_{1,i}^{n}-\phi_{1,i-1}^{n}\right)+\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i+1}\left(\phi_{1,i+1}^{n}-\phi_{1,i}^{n}\right)-\Sigma_{a,1,i}\phi_{1,i}^{n}\Delta x_i-\Sigma_{s,1\to 2}\phi_{1,i}^{n}\Delta x_i\right.$$
$$\left.+(1-\beta)\left(\nu\Sigma_{f,1}\phi_{1,i}^{n}+\nu\Sigma_{f,2}\phi_{2,i}^{n}\right)\Delta x_i+\sum_{d=1}^{N_d}\lambda_d C_{d,i}^{n}\Delta x_i\right]+S_1^{n}\Delta x_i$$

(15)

Therefore, putting:

$$\boxed{a_{1,i}=-\theta\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1}} \qquad \textbf{(16)}$$

$$\boxed{b_{1,i}=\frac{\Delta x_i}{v_1\Delta t}+\theta\left[\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1}+\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i+1}+\Sigma_{a,1,i}\Delta x_i+\Sigma_{s,1\to 2}\Delta x_i-(1-\beta)\nu\Sigma_{f,1}\Delta x_i\right]} \qquad \textbf{(17)}$$

$$\boxed{c_{1,i}=-\theta\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i+1}} \qquad \textbf{(18)}$$

$$
\begin{aligned}
& d_{1,i} = \theta\left[(1-\beta)\left(\nu\Sigma_{f,2}\phi_{2,i}^{n+1}\right)\Delta x_i + \sum_{d=1}^{N_d} \lambda_d C_{d,i}^{n+1}\Delta x_i \right] \\
& + \frac{\Delta x_i}{v_1 \Delta t}\phi_{1,i}^{n} + (1-\theta)\left[-\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1}\left(\phi_{1,i}^{n} - \phi_{1,i-1}^{n}\right) + \overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i+1}\left(\phi_{1,i+1}^{n} - \phi_{1,i}^{n}\right) - \Sigma_{a,1,i}\phi_{1,i}^{n}\Delta x_i - \Sigma_{s,1\to 2}\phi_{1,i}^{n}\Delta x_i \right. \\
& \left. + (1-\beta)\left(\nu\Sigma_{f,1}\phi_{1,i}^{n} + \nu\Sigma_{f,2}\phi_{2,i}^{n}\right)\Delta x_i + \sum_{d=1}^{N_d} \lambda_d C_{d,i}^{n}\Delta x_i \right] + S_1^{n}\Delta x_i
\end{aligned}
\tag{19}
$$

the first energy group equation takes the form

$$
a_{1,i}\,\phi_{1,i-1}^{n+1} + b_{1,i}\,\phi_{1,i}^{n+1} + c_{1,i}\,\phi_{1,i+1}^{n+1} = d_{1,i} \tag{20}
$$

though this three-point equation hides in the known term the presence of $\phi_{2,i}^{n+1}$ and $C_{d,i}^{n+1}$, requiring a proper treatment by iterations.

Similarly, elaborating the second energy group equation, it is:

$$
\begin{aligned}
& -\theta\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i-1}\phi_{2,i-1}^{n,n+1} \\
& + \left\{ \frac{\Delta x_i}{v_2 \Delta t} + \theta\left[\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i-1} + \overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i+1} + \Sigma_{a,2,i}\Delta x_i \right] \right\}\phi_{2,i}^{n,n+1} \\
& -\theta\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i+1}\phi_{2,i+1}^{n,n+1} \\
& = \frac{\Delta x_i}{v_2 \Delta t}\phi_{2,i}^{n} + \theta\,\Sigma_{s,1\to 2}\phi_{1,i}^{n+1}\Delta x_i \\
& (1-\theta)\left[-\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i-1}\left(\phi_{2,i}^{n} - \phi_{2,i-1}^{n}\right) + \overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i+1}\left(\phi_{2,i+1}^{n} - \phi_{2,i}^{n}\right) - \Sigma_{a,2,i}\phi_{2,i}^{n}\Delta x_i + \Sigma_{s,1\to 2}\phi_{1,i}^{n}\Delta x_i \right] + S_2^{n}\Delta x_i
\end{aligned}
$$

and then, putting

$$
a_{2,i} = -\theta\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i-1} \tag{21}
$$

$$
b_{2,i} = \frac{\Delta x_i}{v_2 \Delta t} + \theta\left[\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i-1} + \overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i+1} + \Sigma_{a,2,i}\Delta x_i \right] \tag{22}
$$

$$
c_{2,i} = -\theta\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i+1} \tag{23}
$$

$$d_{2,i} = \frac{\Delta x_i}{v_2 \Delta t}\phi_{2,i}^n + \theta \Sigma_{s,1\to 2}\phi_{1,i}^{n+1}\Delta x_i$$
$$+(1-\theta)\left[-\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i-1}\left(\phi_{2,i}^n - \phi_{2,i-1}^n\right) + \overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i+1}\left(\phi_{2,i+1}^n - \phi_{2,i}^n\right) - \Sigma_{a,2,i}\phi_{2,i}^n\Delta x_i + \Sigma_{s,1\to 2}\phi_{1,i}^n\Delta x_i \right] + S_2^n \Delta x_i$$

(24)

the formulation is reached

$$a_{2,i}\,\phi_{2,i-1}^{n+1} + b_{2,i}\,\phi_{2,i}^{n+1} + c_{2,i}\,\phi_{2,i+1}^{n+1} = d_{2,i} \qquad \textbf{(25)}$$

where the three point equation again hides the presence of $\phi_{1,i}^{n+1}$ in the known term, requiring iterations.

For the equation precursors' concentrations, the closed form solution

$$C_d^{n+1} = C_d^n\, e^{-\lambda_d \Delta t} + \frac{\beta_d}{\lambda_d}\left(\nu\Sigma_{f,1}\phi_{1,i}^{n,n+1} + \nu\Sigma_{f,2}\phi_{2,i}^{n,n+1}\right)\left(1 - e^{-\lambda_d \Delta t}\right) \qquad (d = 1,\ldots,N_d)$$

(26)

is already available for integration adopted.

4. <u>Numerical discretization of the eigenvalue problem</u>

Adopting the same spatial discretization used in the case of the transient equation, it is

$$-\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1}\phi_{1,i-1} + \left[\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1} + \overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i+1} + \Sigma_{a,1,i}\Delta x_i + \Sigma_{s,1\to 2,i}\Delta x_i\right]\phi_{1,i} - \overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i+1}\phi_{1,i+1} = \frac{1}{k}\left(\nu\Sigma_{f,1,i}\phi_{1,i} + \nu\Sigma_{f,2,i}\phi_{2,i}\right)\Delta x_i$$
$$-\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i-1}\phi_{2,i-1} + \left[\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i-1} + \overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i+1} + \Sigma_{a,2,i}\Delta x_i\right]\phi_{2,i} - \overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i+1}\phi_{2,i+1} = \Sigma_{s,1\to 2,i}\,\phi_{1,i}\Delta x_i$$

(27)

The following external (power) iteration process can be therefore considered

$$-\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1}\phi_{1,i-1}^{(n+1)} + \left[\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1} + \overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i+1} + \Sigma_{a,1,i}\Delta x_i + \Sigma_{s,1\to 2,i}\Delta x_i\right]\phi_{1,i}^{(n+1)} - \overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i+1}\phi_{1,i+1}^{(n+1)} = \frac{1}{k^{(n)}}\left(\nu\Sigma_{f,1,i}\phi_{1,i}^{(n)} + \nu\Sigma_{f,2,i}\phi_{2,i}^{(n)}\right)\Delta x_i$$
$$-\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i-1}\phi_{2,i-1}^{(n+1)} + \left[\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i-1} + \overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i+1} + \Sigma_{a,2,i}\Delta x_i\right]\phi_{2,i}^{(n+1)} - \overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i+1}\phi_{2,i+1}^{(n+1)} = \Sigma_{s,1\to 2,i}\,\phi_{1,i}^{(n+1)}\Delta x_i$$

(28)

The generational formulation for advancing the eigenvalue is taken as:

$$k^{(n+1)} = k^{(n)} \frac{\sum_{i=1}^{N} \left(\nu\Sigma_{f,1,i}\phi_{1,i}^{(n+1)} + \nu\Sigma_{f,2,i}\phi_{2,i}^{(n+1)} \right) \Delta x_i}{\sum_{i=1}^{N} \left(\nu\Sigma_{f,1,i}\phi_{1,i}^{(n)} + \nu\Sigma_{f,2,i}\phi_{2,i}^{(n)} \right) \Delta x_i} \qquad \textbf{(29)}$$

Similar positions as in the transient case can be adopted for solving the two-group equations by the classical internal-external iteration scheme.

$$\underbrace{-\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1}}_{a_{1,i}} \phi_{1,i-1}^{(n+1)} + \underbrace{\left[\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i-1} + \overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i+1} + \Sigma_{a,1,i}\Delta x_i + \Sigma_{s,1\to 2,i}\Delta x_i \right]}_{b_{1,i}} \phi_{1,i}^{(n+1)} \underbrace{-\overline{\left(\frac{D_1}{\Delta x}\right)}_{i,i+1}}_{c_{1,i}} \phi_{1,i+1}^{(n+1)} = \underbrace{\frac{1}{k^{(n)}} \left(\nu\Sigma_{f,1,i}\phi_{1,i}^{(n)} + \nu\Sigma_{f,2,i}\phi_{2,i}^{(n)} \right) \Delta x_i}_{d_{1,i}}$$

$$\underbrace{-\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i-1}}_{a_{2,i}} \phi_{2,i-1}^{(n+1)} + \underbrace{\left[\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i-1} + \overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i+1} + \Sigma_{a,2,i}\Delta x_i \right]}_{b_{2,i}} \phi_{2,i}^{(n+1)} \underbrace{-\overline{\left(\frac{D_2}{\Delta x}\right)}_{i,i+1}}_{c_{2,i}} \phi_{2,i+1}^{(n+1)} = \underbrace{\Sigma_{s,1\to 2,i}\phi_{1,i}^{(n+1)}\Delta x_i}_{d_{2,i}}$$

(30)

MATLAB Program

```
function Kinetics1D
%============================================================
%============================================================
% Program for neutron kinetics in a 1D slab nuclear reactor
% Walter Ambrosini, March 26th, 2008
%============================================================
%============================================================
clc, clear;
%============================c
% Input data are assigned
%============================c
% NB: Lengths are expressed in cm
%----------------------------c
% Left reflector parameters
%----------------------------c
nrefl = 20 ;
dxrefl = 20. ;
%
dxrl = dxrefl / nrefl ;
n1 = nrefl ;
%
  for i=1:n1
  dx(i) = dxrl ;
  xm(i) = ( i - 0.5 ) * dxrl ;
%
  v1(i) = 1.25e07 ;
  d1(i) = 1.634 ;
  sa1(i) = 0.00266  ;
  ansf1(i) = 0. ;
  ss1to2(i) = 0.0276 ;
  v2(i) = 2.5e05 ;
  d2(i) = 0.264 ;
  sa2(i) = 0.0494 ;
  ansf2(i) = 0. ;
%
  sou1(i) = 0. ;
  sou2(i) = 0. ;
%
  end
%
x1 = xm(n1) + 0.5 * dxrl ;
%
%----------------------------------c
% Multiplicating medium parameters
%----------------------------------c
nmult = 200 ;
dxmult = 200. ;
%
dxmu = dxmult / nmult;
n2 = nrefl + nmult;
%
  for i=n1+1:n2
  dx(i) = dxmu ;
  xm(i) = x1 + ( i - n1 - 0.5 ) * dxmu ;
%
  v1(i) = 1.25e07 ;
  d1(i) = 1.426 ;
  sa1(i) = 0.01099  ;
  ansf1(i) = 0.0075 ;
  ss1to2(i) = 0.0176 ;
  v2(i) = 2.5e05 ;
  d2(i) = 0.350 ;
  sa2(i) = 0.09926 ;
  ansf2(i) = 0.1378 ;
%
  sou1(i) = 0. ;
  sou2(i) = 0. ;
%
  end
%
x2 = xm(n2) + 0.5 * dxmu ;
%
%----------------------------c
% Right reflector parameters
%----------------------------c
nrefr = 20 ;
```

```
dxrefr = 20. ;
%
dxrr = dxrefr / nrefr ;
n3 = nrefl + nmult + nrefr ;
ntot = n3 ;
%
  for i=n2+1:n3
  dx(i) = dxrr ;
  xm(i) = x2 + ( i - n2 - 0.5 ) * dxrr ;
%
  v1(i) = 1.25e07 ;
  d1(i) = 1.634 ;
  sa1(i) = 0.00266  ;
  ansf1(i) = 0. ;
  ss1to2(i) = 0.0276 ;
  v2(i) = 2.5e05 ;
  d2(i) = 0.264 ;
  sa2(i) = 0.0494 ;
  ansf2(i) = 0. ;
%
  sou1(i) = 0. ;
  sou2(i) = 0. ;
%
  end
%
x3 = xm(n3) + 0.5 * dxrr ;
%
alef(1:n3) = 0. ;
bet(1:n3) = 0. ;
%
%-----------------------------------c
% Parameters for kinetic calculations
%-----------------------------------c
%
  alambd(1) = 0.0127 ;
  alambd(2) = 0.0317 ;
  alambd(3) = 0.115 ;
  alambd(4) = 0.311 ;
  alambd(5) = 1.4 ;
  alambd(6) = 3.87 ;
%
  beta(1) = 0.000247 ;
  beta(2) = 0.0013845 ;
  beta(3) = 0.001222 ;
  beta(4) = 0.0026455 ;
  beta(5) = 0.000832 ;
  beta(6) = 0.000169 ;
%
   bttot = 0. ;
   for id = 1:6
       bttot = bttot + beta(id) ;
   end
%
disp('Value of beta = '); disp(bttot);
%
%---------------------------------c
% Averaged D/dx coefficients
%---------------------------------c
%
  d1ovdx(1) = 2. * d1(1) / dx(1) ;
  d2ovdx(1) = 2. * d2(1) / dx(1) ;
%
  for i=2:n3
  aux1 = d1(i-1) / dx(i-1) ;
  aux2 = d1(i) / dx(i) ;
  denom = aux1 + aux2 ;
  d1ovdx(i) = 2. * aux1 * aux2 / denom ;
%
  aux1 = d2(i-1) / dx(i-1) ;
  aux2 = d2(i) / dx(i) ;
  denom = aux1 + aux2 ;
  d2ovdx(i) = 2. * aux1 * aux2 / denom ;
  end
%
  d1ovdx(n3+1) = 2. * d1(n3) / dx(n3) ;
  d2ovdx(n3+1) = 2. * d2(n3) / dx(n3) ;
%
%-------------- Search for criticality  -----
```

```
disp('Criticality Search ');
%-----------------------------------------
%
% Interactive input for control rods
%
disp('Number of nodes in the left reflector = '); disp(n1);
disp('Number of nodes in the multiplicating medium = '); disp(n2-n1);
disp('Number of nodes in the right reflector = '); disp(n3-n2);
%
  ncrod=input('Assign the number of control rods ');
    if(ncrod>0)
       for ic=1:ncrod
       disp('Control rod number = ');
       disp(ic);
       ncr=input('Enter the node index of the control rod ');
       disp('Present values of Sigma1 and Sigma2 = ');
       disp(sa1(ncr));
       disp(sa2(ncr));
       sa1(ncr)=input('New value of Sigma1 ');
       sa2(ncr)=input('New value of Sigma2 ');
       end
    end;
%
%==============================
% Criticality search
%==============================
%
% Initial value of keff and fission source
%
akeff = 1.;
fissou(1:n1) = 0. ;
fissou(n1+1:n2) = 1.;
fissou(n2+1:n3) = 0.;
%
% Outer Iteration loop
%
  for iter=1:10000
%-----------------------------------------------------------------------c
% construction of the vectors a, b, c and d for the 1st energy group
%-----------------------------------------------------------------------c
    for i=1:n3
    a(i) = - d1ovdx(i) ;
    b(i) = d1ovdx(i) + d1ovdx(i+1) + ( sa1(i) + ss1to2(i) ) * dx(i) ;
    c(i) = - d1ovdx(i+1) ;
    d(i) = fissou(i) / akeff ;
    end
%
%  Solution of the TDM system for the 1st energy group
%
  t(1:ntot) = 0. ;
%
  [t]=tdma(a,b,c,d,alef,bet,ntot);
%
    for i=1:n3
    phi1(i) = t(i) ;
    end
%-----------------------------------------------------------------------c
% construction of the vectors a, b, c and d for the 2nd energy group
%-----------------------------------------------------------------------c
    for i=1:n3
    a(i) = - d2ovdx(i) ;
    b(i) = d2ovdx(i) + d2ovdx(i+1) + sa2(i) * dx(i) ;
    c(i) = - d2ovdx(i+1) ;
    d(i) = ss1to2(i) * dx(i) * phi1(i) ;
    end
%
%  Solution of the TDM system for the 1st energy group
%
  t(1:ntot) = 0. ;
%
  [t]=tdma(a,b,c,d,alef,bet,ntot);
%
    for i=1:n3
    phi2(i) = t(i) ;
    end
%-------------------------------------
%  Updating keff and Fission source
%-------------------------------------
```

```
  oldkef = akeff ;
  sumold = 0. ;
  sumfis = 0. ;
  amxph1 = -1. ;
  amxph2 = - 1. ;
%
    for i=1:n3
    oldfis(i) = fissou(i) ;
    fissou(i) = ( ansf1(i) * phi1(i) + ansf2(i) * phi2(i) ) * dx(i) ;
%
    sumold = sumold + oldfis(i) ;
    sumfis = sumfis + fissou(i) ;
%
      if (amxph1<phi1(i))
          amxph1 = phi1(i) ;
      end;
      if (amxph2<phi2(i))
        amxph2 = phi2(i) ;
      end;
%
    amxphi = amxph1 ;
      if (amxphi<amxph2)
         amxphi = amxph2 ;
      end;
    end
%
  akeff = oldkef * sumfis / sumold ;
%  disp('Iteration number and keff = '),iter,akeff;

  chk = num2str(akeff,'%10.5f');
  chiter = num2str(iter,'%10.0f');
%
  erkeff = abs( akeff - oldkef ) ;
  if (erkeff<1.e-06),break,end
%
  end
%-------------------------------------------------
%---------  end of criticality search loop ------
%-------------------------------------------------
% A plot of fluxes is generated
%
  plot(xm(1:n3),phi1(1:n3),'b.',xm(1:n3),phi2(1:n3),'r.')
    grid
    xlabel('x [cm]')
    ylabel('Normalised Neutron Flux')
    axis([0., x3, 0., 2*amxphi])
    title(['Iter = ',chiter,'     Keff = ',chk])
    h = legend('Fast Neutron Flux','Thermal Neutron Flux',1);
%
%-----------------------------------------------------------------
% Preparing the transient calculation
%-----------------------------------------------------------------
%
% Interactive input on starting criticality level
%
  aknew=input('Assign the new Keff ');
%
% The fission cross sections are reassigned
%
  fact = aknew / akeff ;
%
  sumfis = 0. ;
       for i=1:ntot
       ansf1(i) = ansf1(i) * fact ;
       ansf2(i) = ansf2(i) * fact ;
       fissou(i) = fissou(i) * fact ;
       sumfis = sumfis + fissou(i) ;
       end
%
%-----------------------------------------------------------------
% Transient calculation
%-----------------------------------------------------------------
%
disp('Transient Calculation ');
%
%-------------------------------------------------------------------------
%  Level of implicit calculation
%-------------------------------------------------------------------------
```

```
    theta=input('Enter theta [0.=Explicit; 0.5=C-N; 1.= Implicit] ');
%
    cmptht = 1. - theta ;
    cmpbet = 1. - bttot ;
%
% Interactive input on localised sources
%
disp('Number of nodes in the left reflector = '); disp(n1);
disp('Number of nodes in the multiplicating medium = '); disp(n2-n1);
disp('Number of nodes in the right reflector = '); disp(n3-n2);
%
  nsour=input('Assign the number of localised sources ');
    if(nsour>0)
       for is=1:nsour
       disp('Localised source number = ');
       disp(is);
       nsr=input('Enter the node index of the localised sources ');
       disp('Present values of Sou1 and Sou2 = ');
       disp(sou1(nsr));
       disp(sou2(nsr));
       sou1(nsr)=input('New value of Sou1 ');
       sou2(nsr)=input('New value of Sou2 ');
       end
    end;
%
%  Assigning the delayed neutron precursors concentrations at steady-state
%
      for id = 1:6
          for i = 1:ntot
          cd(id,i) = beta(id) * fissou(i) / alambd(id) ;
          end
      end
%
time = 0. ;
tread = 0. ;
%---------------------------------
%  Time advancement loop - start
%---------------------------------
%
for itime=1:100000
%
% Assigning the old fluxes and precursor concentrations
%
   for i = 1:ntot
     oldph1(i) = phi1(i) ;
     oldph2(i) = phi2(i) ;
     for id = 1:6
         oldcd(id,i) = cd(id,i) ;
     end
   end
%
% Interactive input for control rod movement
%
    if(time>=tread)
      disp('Number of nodes in the left reflector = '); disp(n1);
      disp('Number of nodes in the multiplicating medium = '); disp(n2-n1);
      disp('Number of nodes in the right reflector = '); disp(n3-n2);
%
      ncrod=input('Assign the number of control rods ');
       if(ncrod>0)
          for ic=1:ncrod
          disp('Control rod number = ');
          disp(ic);
          ncr=input('Enter the node index of the control rod ');
          disp('Present values of Sigma1 and Sigma2 = ');
          disp(sa1(ncr));
          disp(sa2(ncr));
          sa1(ncr)=input('New value of Sigma1 ');
          sa2(ncr)=input('New value of Sigma2 ');
          end
       end;
       tread=input('Assign the new break time [s] ');
       dt=input('Assign the time step [s] ');
    end;
%
    time = time + dt ;
    tv(itime) = time ;
%    disp(time);
```

```
%
% Iteration on the implict terms on the RHS
%
   for iter = 1:1000
%    disp(iter);
%
%-----------------------------------------------------------------------c
% construction of the vectors a, b, c and d for the 1st energy group
%-----------------------------------------------------------------------c
         for i=1:ntot
%
         a(i) = - theta * d1ovdx(i) ;
%
         aux1 = dx(i) / ( v1(i) * dt ) ;
         aux2 = d1ovdx(i) + d1ovdx(i+1) ;
         aux3 = ( sa1(i) + ss1to2(i) - cmpbet * ansf1(i) ) * dx(i) ;
         b(i) = aux1 + theta * ( aux2 + aux3 ) ;
%
         c(i) = - theta * d1ovdx(i+1) ;
%
         aux1 = theta * cmpbet * ansf2(i) * phi2(i) * dx(i) ;
         aux2 = 0. ;
            for id = 1:6
                aux2 = aux2 + alambd(id) * cd(id,i) ;
            end
         aux2 = aux2 * theta * dx(i) ;
%
         aux3 = dx(i) / ( v1(i) * dt ) * oldph1(i) ;
%
         aux4 = - ( d1ovdx(i) + d1ovdx(i+1) ) * oldph1(i) ;
           if(i>1)
             aux4 = aux4 + d1ovdx(i) * oldph1(i-1) ;
           end;
           if(i<ntot)
             aux4 = aux4 + d1ovdx(i+1) * oldph1(i+1) ;
           end;
         aux4 = aux4 - ( sa1(i) + ss1to2(i) ) * dx(i) * oldph1(i) ;
         auxfis = ansf1(i) * oldph1(i) + ansf2(i) * oldph2(i) ;
         aux4 = aux4 + cmpbet * auxfis * dx(i) ;
            for id = 1:6
                aux4 = aux4 + alambd(id) * oldcd(id,i) * dx(i);
            end
         aux4 = aux4 * cmptht ;
%
         aux5 = sou1(i) * dx(i) ;
%
         d(i) = aux1 + aux2 + aux3 + aux4 + aux5 ;
         end
%
%  Solution of the TDM system for the 1st energy group
%
    t(1:ntot) = 0. ;
%
    [t]=tdma(a,b,c,d,alef,bet,ntot);
%
         for i=1:ntot
         phi1(i) = t(i) ;
         end
%-----------------------------------------------------------------------c
% construction of the vectors a, b, c and d for the 2nd energy group
%-----------------------------------------------------------------------c
         for i=1:ntot
         a(i) = - theta * d2ovdx(i) ;
%
         aux1 = dx(i) / ( v2(i) * dt ) ;
         aux2 = d2ovdx(i) + d2ovdx(i+1) ;
         aux3 = sa2(i) * dx(i) ;
         b(i) = aux1 + theta * ( aux2 + aux3 ) ;
%
         c(i) = - theta * d2ovdx(i+1) ;
%
         aux1 = dx(i) / ( v2(i) * dt ) * oldph2(i) ;
         aux2 = theta * ss1to2(i) * phi1(i) * dx(i) ;
%
         aux3 = - ( d2ovdx(i) + d2ovdx(i+1) ) * oldph2(i) ;
         if(i>1)
             aux3 = aux3 + d2ovdx(i) * oldph2(i-1) ;
         end;
```

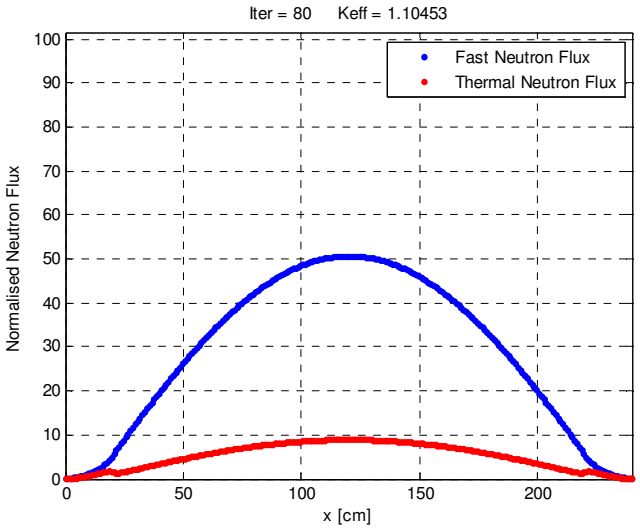
```
        if(i<ntot)
            aux3 = aux3 + d2ovdx(i+1) * oldph2(i+1) ;
        end;
        aux3 = aux3 - ( sa2(i)*oldph2(i) - ss1to2(i)*oldph1(i)) * dx(i)  ;
        aux3 = aux3 * cmptht ;
%
        aux4 = sou2(i) * dx(i) ;
%
        d(i) = aux1 + aux2 + aux3 + aux4 ;
        end
%
%  Solution of the TDM system for the 2nd energy group
%
    t(1:ntot) = 0. ;
%
    [t]=tdma(a,b,c,d,alef,bet,ntot);
%
        for i=1:ntot
        phi2(i) = t(i) ;
        end
%-------------------------------------------------------------
%  Advancing the precursors' concentrations
%-------------------------------------------------------------
%
   fisnew = 0. ;
%
       for i = 1:ntot
           auxfo = ansf1(i) * oldph1(i) + ansf2(i) * oldph2(i) ;
           auxfn = ansf1(i) * phi1(i) + ansf2(i) * phi2(i) ;
           fisnew = fisnew + auxfn * dx(i) ;
           for id =1:6
               expldt = exp ( - alambd(id) * dt ) ;
               cmpexp = 1. - expldt ;
               aux1 = oldcd(id,i) * expldt ;
               aux2 = auxfo * cmptht + auxfn * theta ;
               aux2 = aux2 * cmpexp * beta(id) / alambd(id) ;
               cd(id,i) = aux1 + aux2 ;
           end
       end
   fv(itime) = fisnew ;
%
%  Iteration stopping criterion
%
       if(iter==1)
           fisold = fisnew;
       else
           errel = abs (fisnew - fisold ) / fisnew ;
           if (errel<1.e-05), break,end
%           disp(errel);
           fisold = fisnew ;
       end;

   end
%
%  Calculating the reactor period
%
   if(itime>1)
      recper = ( fv(itime) - fv(itime-1) ) / ( fv(itime) * dt ) ;
   else
      recper = ( fv(itime) - sumfis ) / ( fv(itime) * dt ) ;
   end;
   per = 1. / recper ;
   pv(itime) = per ;
   rpv(itime) = recper ;
%
% Plots of fluxes and of the overall nuetron source is generated
%
  chtime = num2str(time,'%10.5f');
  chiter = num2str(iter,'%10.0f');
  chfv = num2str(fv(itime),'%10.3f');
  chpv = num2str(pv(itime),'%10.3f');
%
   subplot (1,3,1);
   plot(xm(1:n3),phi1(1:n3),'b.',xm(1:n3),phi2(1:n3),'r.')
    grid
    xlabel('x [cm]')
    ylabel('Normalised Neutron Flux')
    axis([0., x3,0.,2*amxphi])
```

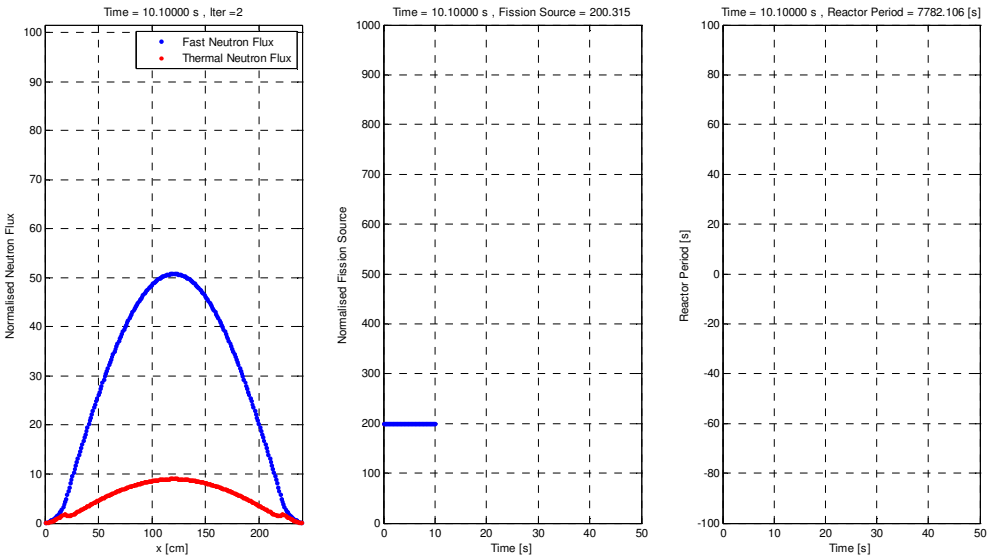
```
    title(['Time = ',chtime,' s , Iter =',chiter])
    h = legend('Fast Neutron Flux','Thermal Neutron Flux',1);
%
    subplot (1,3,2);
   plot(0.,sumfis,'r.',tv(1:itime),fv(1:itime),'b.')
    grid
    xlabel('Time [s]')
    ylabel('Normalised Fission Source')
    axis([0., 50.,0.,1000.])
    title(['Time = ',chtime,' s , Fission Source = ',chfv])
%
    subplot (1,3,3);
   plot(tv(1:itime),pv(1:itime),'b.')
    grid
    xlabel('Time [s]')
    ylabel('Reactor Period [s]')
    axis([0., 50.,-100.,100.])
    title(['Time = ',chtime,' s , Reactor Period = ',chpv,' [s]' ])
    pause (0.0001);
%
%--------------------------------
%  Time advancement loop - end
%--------------------------------
end
%----------------------------
%  End of Program
%----------------------------
end
%
%-------------------------------------------%
% Solution of the tridiagonal matrix system %
%-------------------------------------------%
function [t]=tdma(a,b,c,d,alef,bet,ntot);
%
alef(1:ntot)=0.;
bet(1:ntot)=0.;
t(1:ntot)=0.;
%
ub=1.d00/b(1);
alef(1)=c(1)*ub;
bet(1)=d(1)*ub;
%
    for i=2:ntot
      l=i-1;
      qz=b(i)-a(i)*alef(l);
      uqz=1.d00/qz;
      alef(i)=c(i)*uqz;
      bet(i)=(d(i)-a(i)*bet(l))*uqz;
    end
%
nm1=ntot-1;
t(ntot)=bet(ntot);
    for i=1:nm1
      ii=ntot-i;
      l=ii+1;
      t(ii)=bet(ii)-alef(ii)*t(l);
    end
%
end
```

Suggested activities:

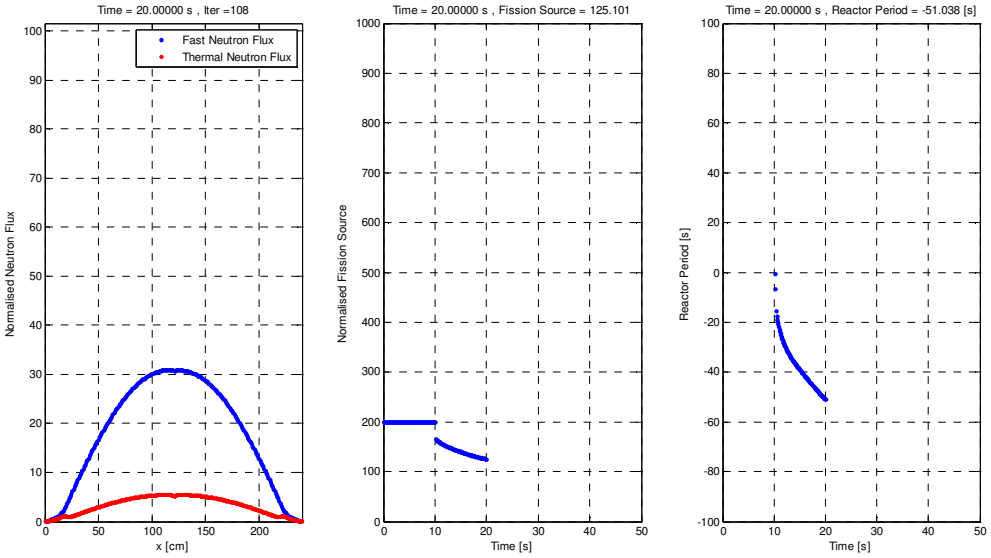
1. **Make a criticality calculation with the assigned values of parameters and discretization and with no control rod. The result should be as in the figure, where *the effect of the reflector on thermal flux can be clearly noted*.**



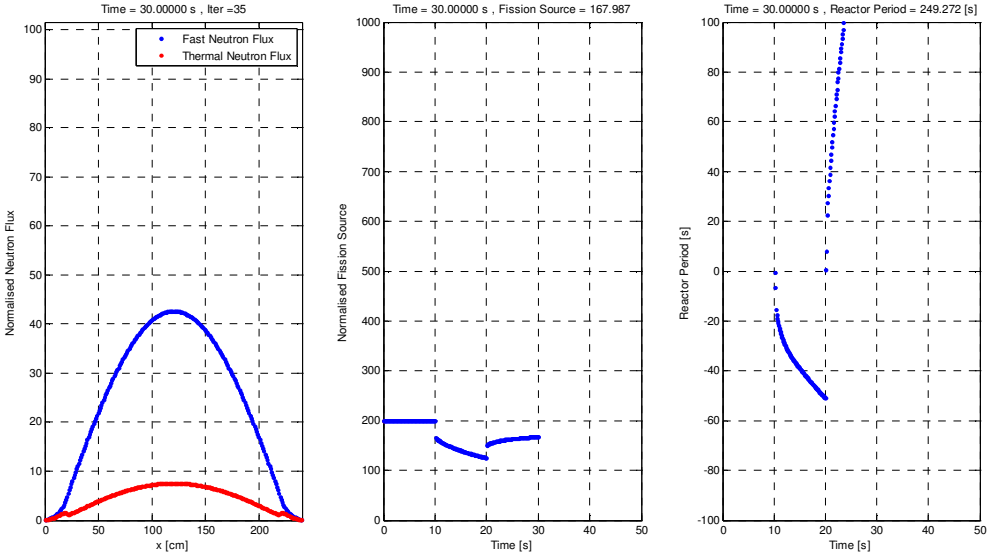
2. **Then, assign the new K_{eff} to unity and run an analysis of "steady" transient calculations (no control rods, no sources, for 10 seconds, implicit scheme, 0.1 s time step). No substantial change in normalized power is obtained.**



3. Now, try a new transient of 10 s (up to the next break time of 20 s) after inserting a control rod. Put it (e.g.) in the node 121 and increase the "sigma2" (thermal absorption cross section in the node. You will see the power decreasing smoothly after a clear negative "prompt jump".

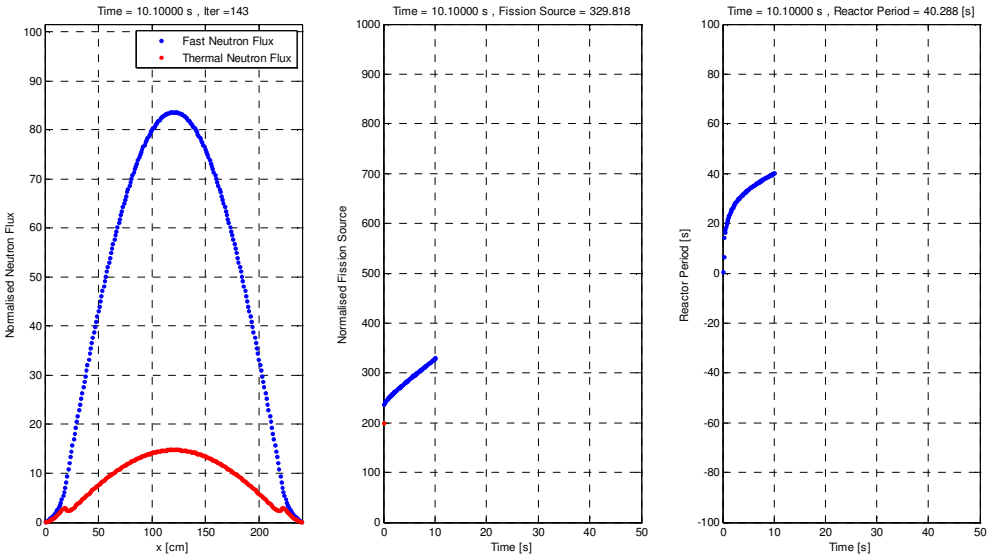


4. Now, try restoring a lower value of the thermal absorption cross section in the same node (121) trying to restore criticality.

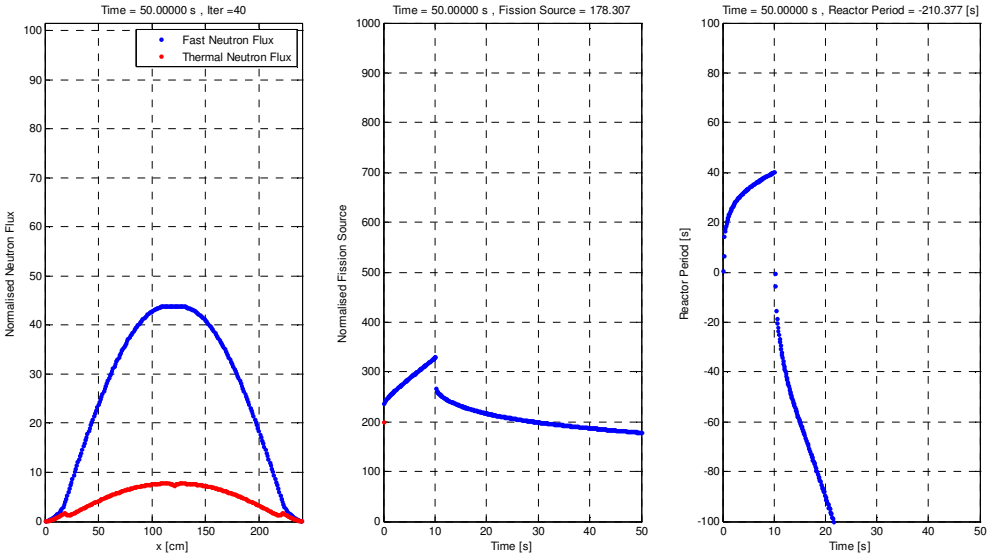


5. To end the analysis of this case, make any other change in the number and nature of control rod and try justifying the results you see.

6. Start again with another similar case but, after the eigenvalue calculation, assign a supercritical K_{eff}. The results will show the prompt jump and the subsequent effect of delayed neutron increase



7. Safely shut down the reactor by inserting a control rod and leave the reactor decrease its power up to 50 s.



Then, invent your own problems. Good luck !