Development of new Monte Carlo methods for reactor physics applications

NORDITA

Jan Dufek

Nuclear Reactor Technology Division
Department of Physics, School of Engineering Sciences, KTH

October 7, 2014

Outline

STORM in MC criticality calculations

STORM in MC reactor physics calculations

The goal

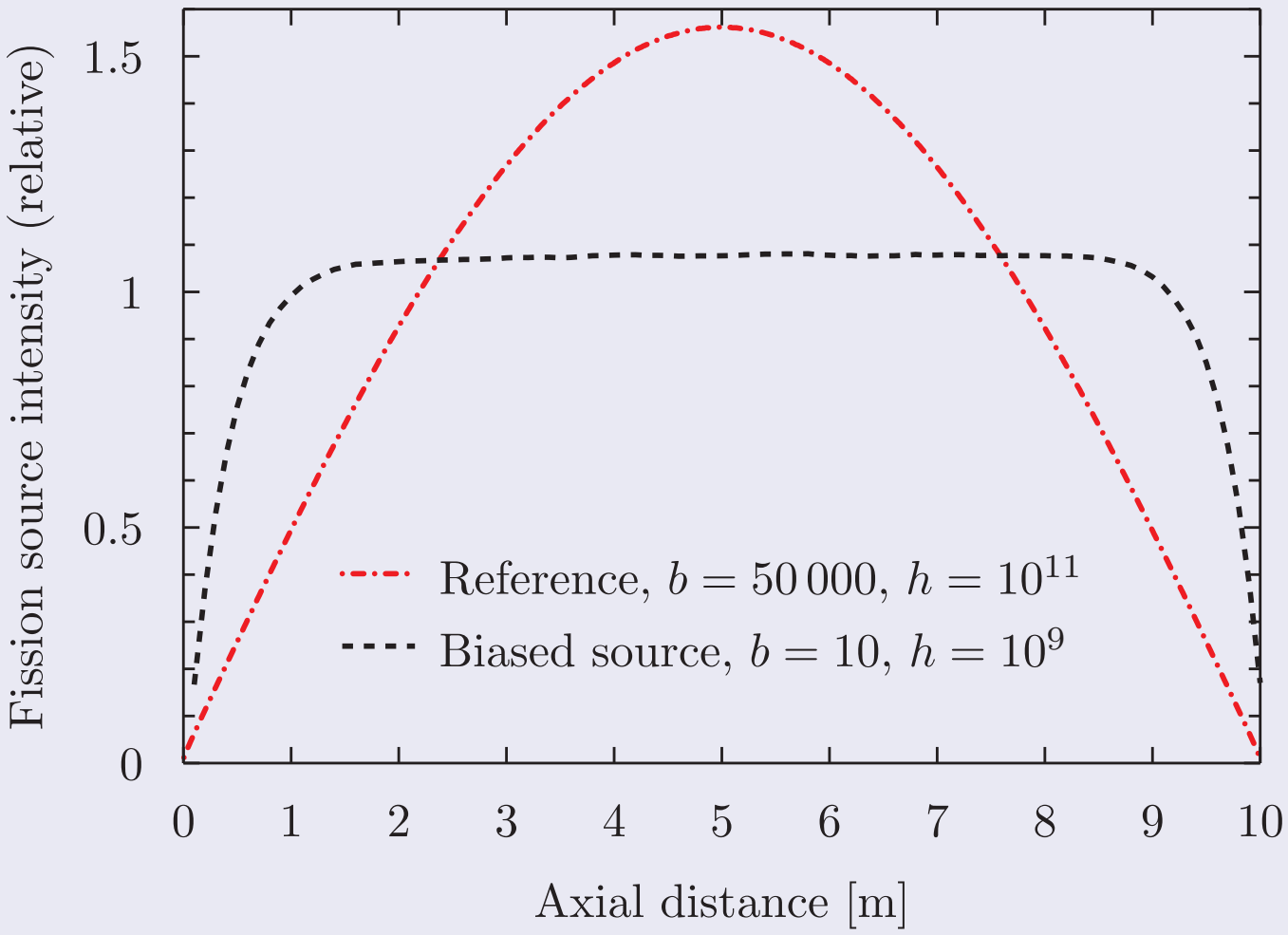
Improving the efficiency of MC criticality calculations.

Problem

- The neutron batch size (the number of neutrons per cycle) affects the efficiency, due to two effects:
  - presence of bias (large bias when batch size is small)
  - speed of convergence (faster convergence when batch size is small)
- To achieve the best efficiency, the batch size must be optimised to balance the two effects.

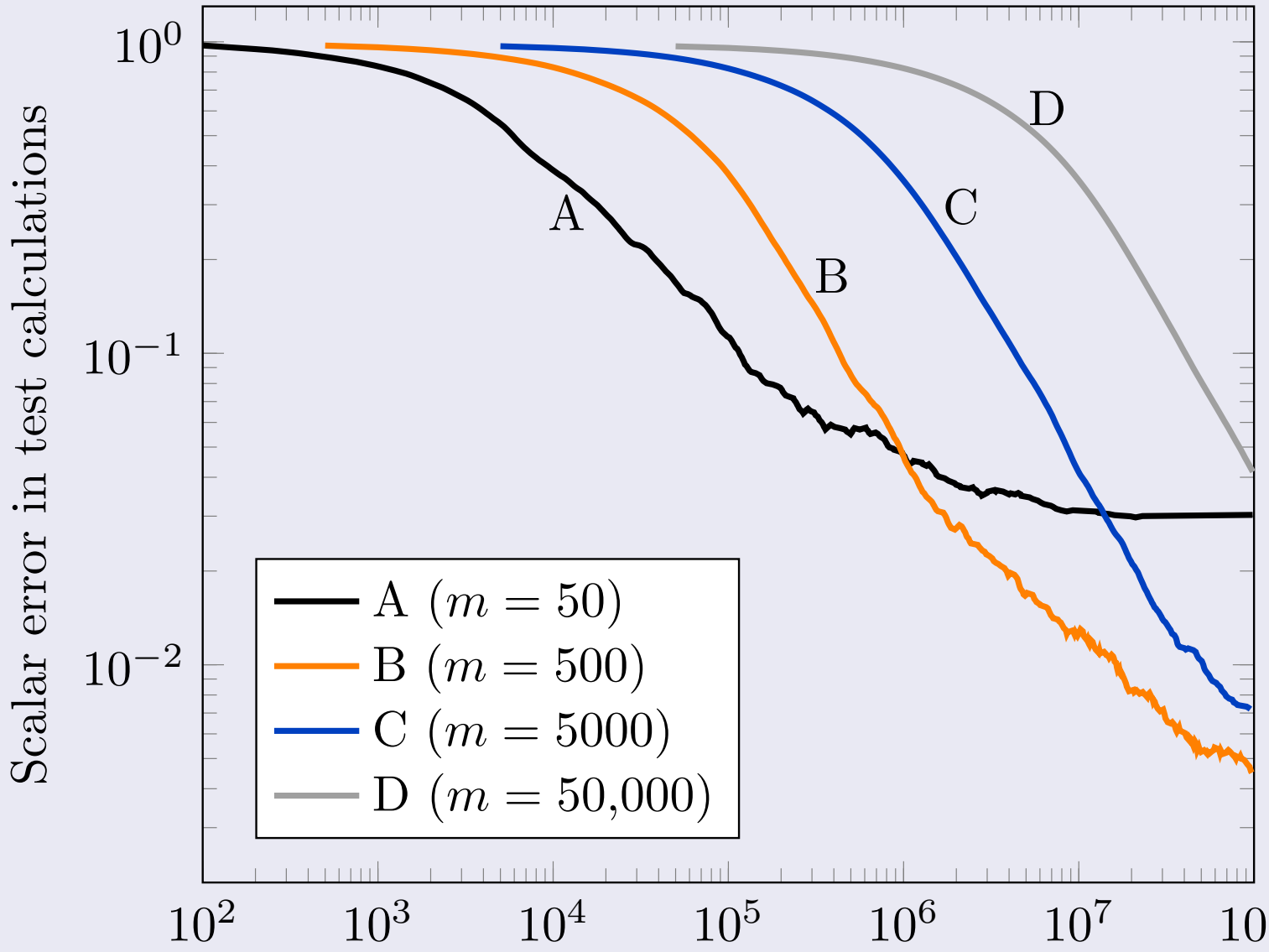
STORM in MC reactor physics calculations

Example of bias in the cumulative fission source



STORM in MC reactor physics calculations

Example of source convergence



STORM in MC reactor physics calculations

Origin of errors in the cumulative fission source

- Statistical errors
- Bias
- Error coming from initial fiss. source (from the first cycle)

Two new methods of estimating the error in the cumulative fission source

- Estimation via evaluating the difference between the fission source and the fundamental-mode eigenvector of the fission matrix (published in ANE).
- Simplified model suggested (next slide).

STORM in MC reactor physics calculations

Simplified model of the error in the cumulative fission source

$$\hat{\varepsilon} = \frac{m\varepsilon_0}{h}\left(1 - \frac{k_1}{k_0}\right)^{-1} + \frac{1}{m} + \frac{1}{\sqrt{h}} \tag{1}$$

where

- m is the batch size
- ε_0 is the error in the initial source
- h is the total number of neutrons to be simulated
- $\frac{k_1}{k_0}$ is the dominance ratio

STORM in MC reactor physics calculations

The computing efficiency (figure of merit)

$$\mathrm{FOM} = \frac{1}{\hat{\varepsilon}^2 h}$$

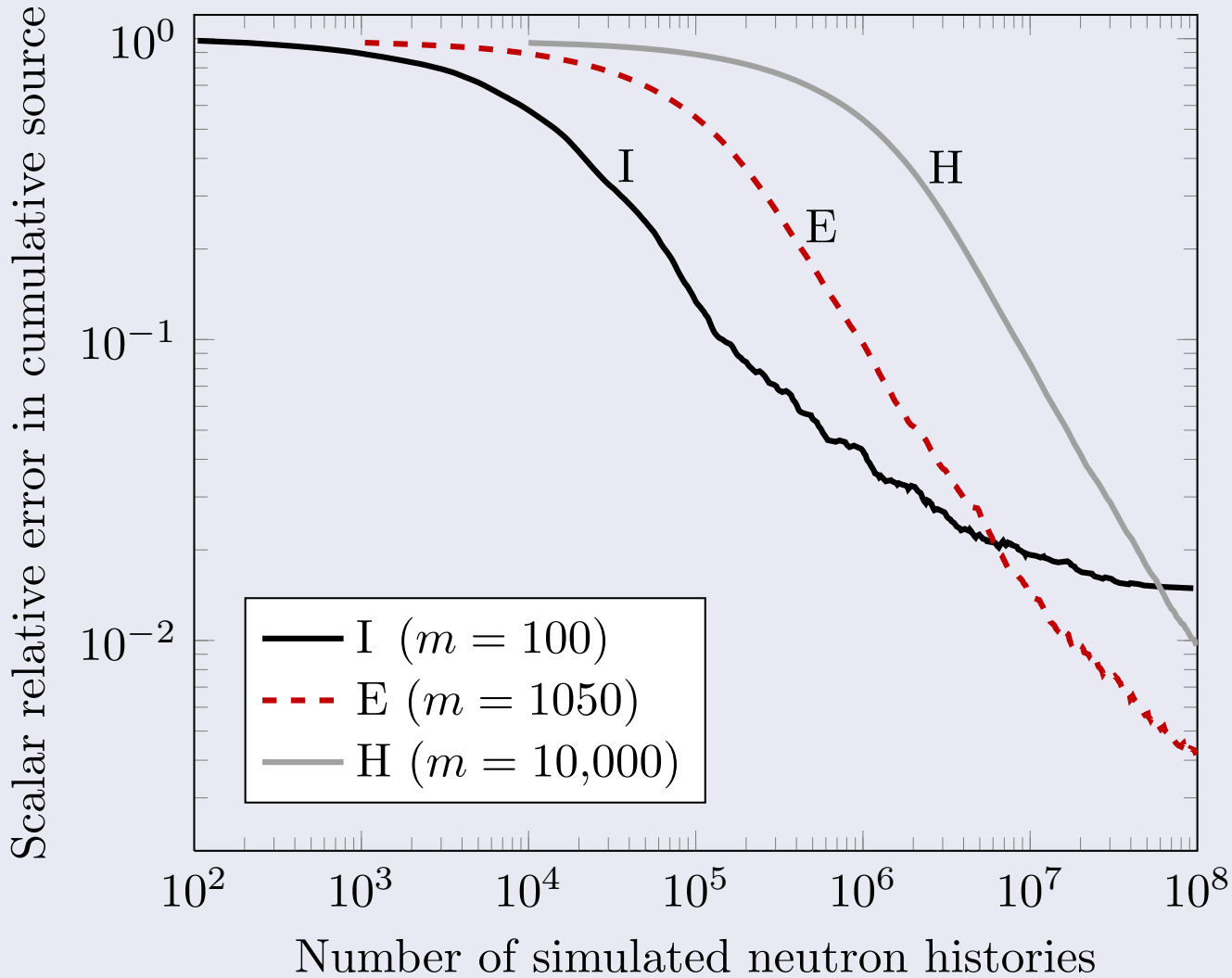
Optimising the batch size for best computing efficiency

$$\frac{\partial(\hat{\varepsilon}^2 h)}{\partial m} = 0\,. \tag{2}$$

$$\Longrightarrow m_{opt} = \sqrt{\frac{h}{\varepsilon_0}\left(1 - \frac{k_1}{k_0}\right)} \tag{3}$$

STORM in MC reactor physics calculations

Numerical test results



Governing equations for MC burnup simulations

Governing equations

Quantities of interest

- $\vec{N}(\vec{r}, t)$ ... nuclide field
- $\phi(\vec{r}, E, t)$ ... neutron flux

The purpose of the Monte Carlo burnup calculations is to determine as to how these fields change during the fuel cycle.

Governing equations

This problem can be described by two **coupled** equations:

- **burnup equation** ... determines the nuclide field changes,
- **criticality equation** ... gives the neutron flux.

Governing equations

Burnup equation

The concentration of the fuel isotopes is described by the burnup equation

$$\frac{d\vec{N}(\vec{r},t)}{dt} = \mathbb{M}(\phi, T)\vec{N}(\vec{r},t), \tag{4}$$

where

$$\mathbb{M}[\phi(\vec{r},E,t), T(\vec{r},t)] = \int_0^\infty \phi(\vec{r},E,t)\mathbb{X}(T)\mathrm{d}E + \mathbb{D}, \tag{5}$$

where $\mathbb{X}(T)$ is a temperature dependant cross-section and yield matrix, and $\mathbb{D}$ is a decay matrix.

Governing equations

Formal solution of the burnup equation

The burnup equation has a formal solution

$$\vec{N}(\vec{r},t) = \exp[\mathbb{M}(\phi,T)(t-t_0)]\vec{N}|_{t_0}(\vec{r}) \tag{6}$$

for ϕ fixed in time.

Governing equations

Criticality equation

$$B(\vec{N},T)\phi \equiv [L(\vec{N},T) - \frac{1}{k}F(\vec{N},T)]\phi = 0, \quad (7)$$

where

- $L\phi$ represents the migration and loss of neutrons from $(\vec{r},\vec{\Omega},E)$,
- $F\phi$ accounts for neutron production in $(\vec{r},\vec{\Omega},E)$ due to fission

Formal solution

The criticality equation has many solutions. Let's denote the fundamental solution computed by the MC criticality code as ϕ_B

Numerical stability of existing MC burnup codes

Common coupling schemes in MC burnup codes

Common coupling schemes

As the burnup equation is an ODE, the coupling schemes are derived from ODE numerical methods, such as:

- Explicit Euler (Beginning-of-step constant flux approximation)
- Predictor-Corrector
- Mid-point method

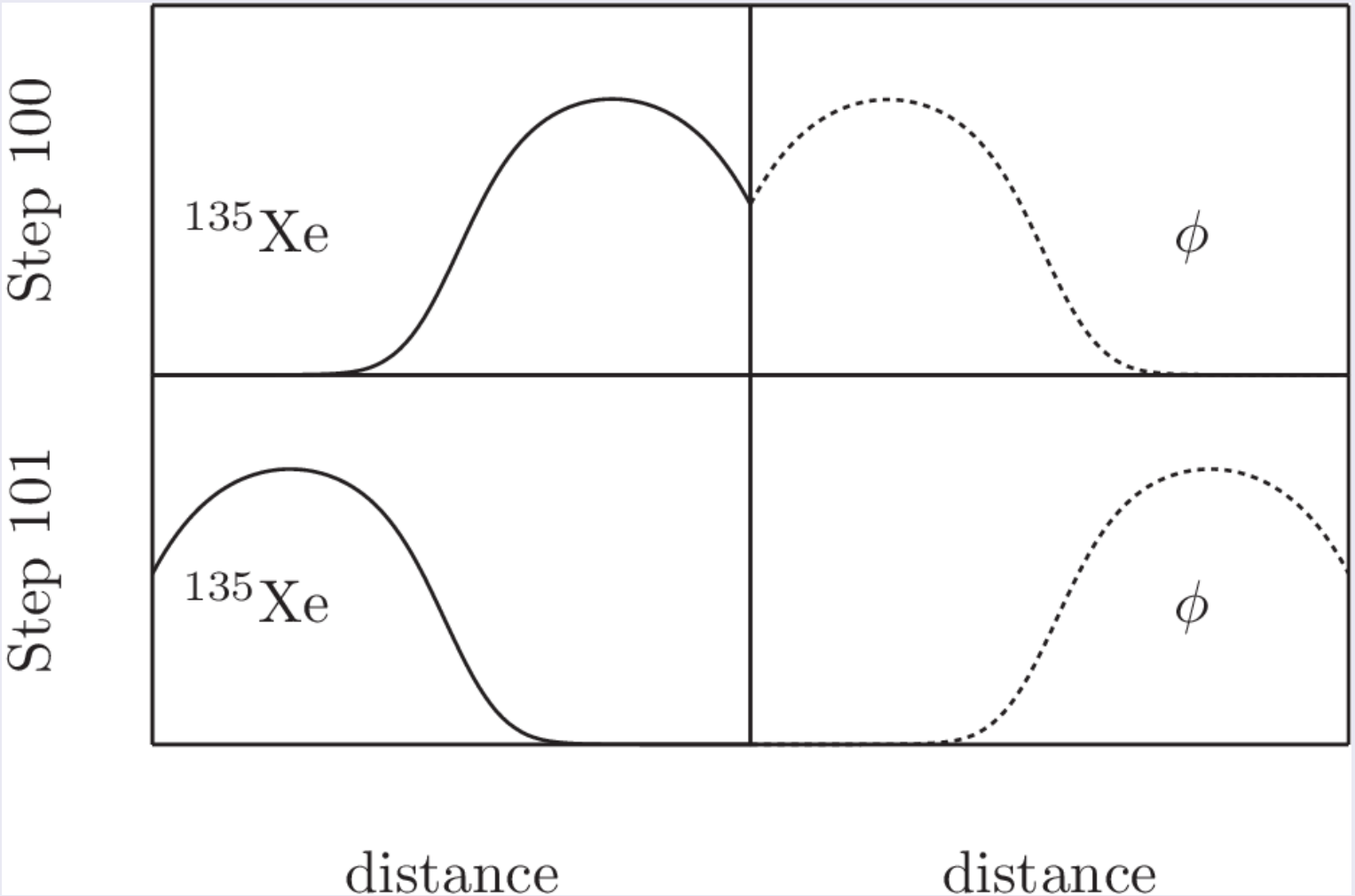
Explicit Euler

The beginning-of-step constant flux approximation coupling scheme (MCB, SERPENT)

1: **input:** $\vec{N}_0$
2: **for** $i \leftarrow 0, 1, \ldots$ **do**
3: $\quad \phi_i \leftarrow \phi_B(\vec{N}_i)$
4: $\quad \vec{N}_{i+1} \leftarrow \exp[\mathbb{M}(\phi_i)\Delta t_i]\vec{N}_i$
5: **end for**

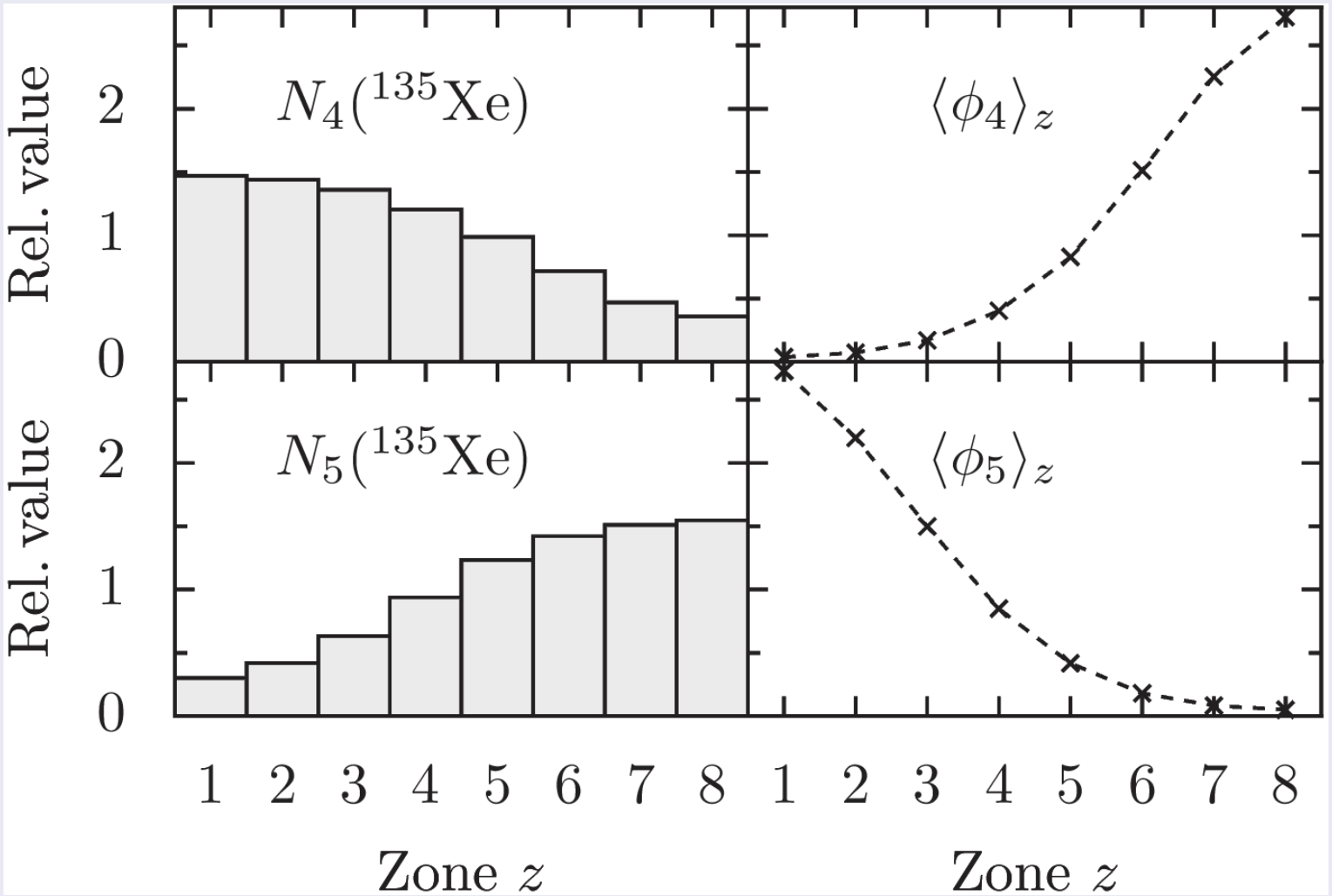
Explicit Euler

Results: stability test using a slab model ($\Delta t = 7$d)



Explicit Euler

MCB calculation of fuel rod (with axial reflectors) ($\Delta t = 3$d)



Predictor-Corrector

The predictor-corrector scheme (MCODE, SERPENT)

input: $\vec{N}_0$

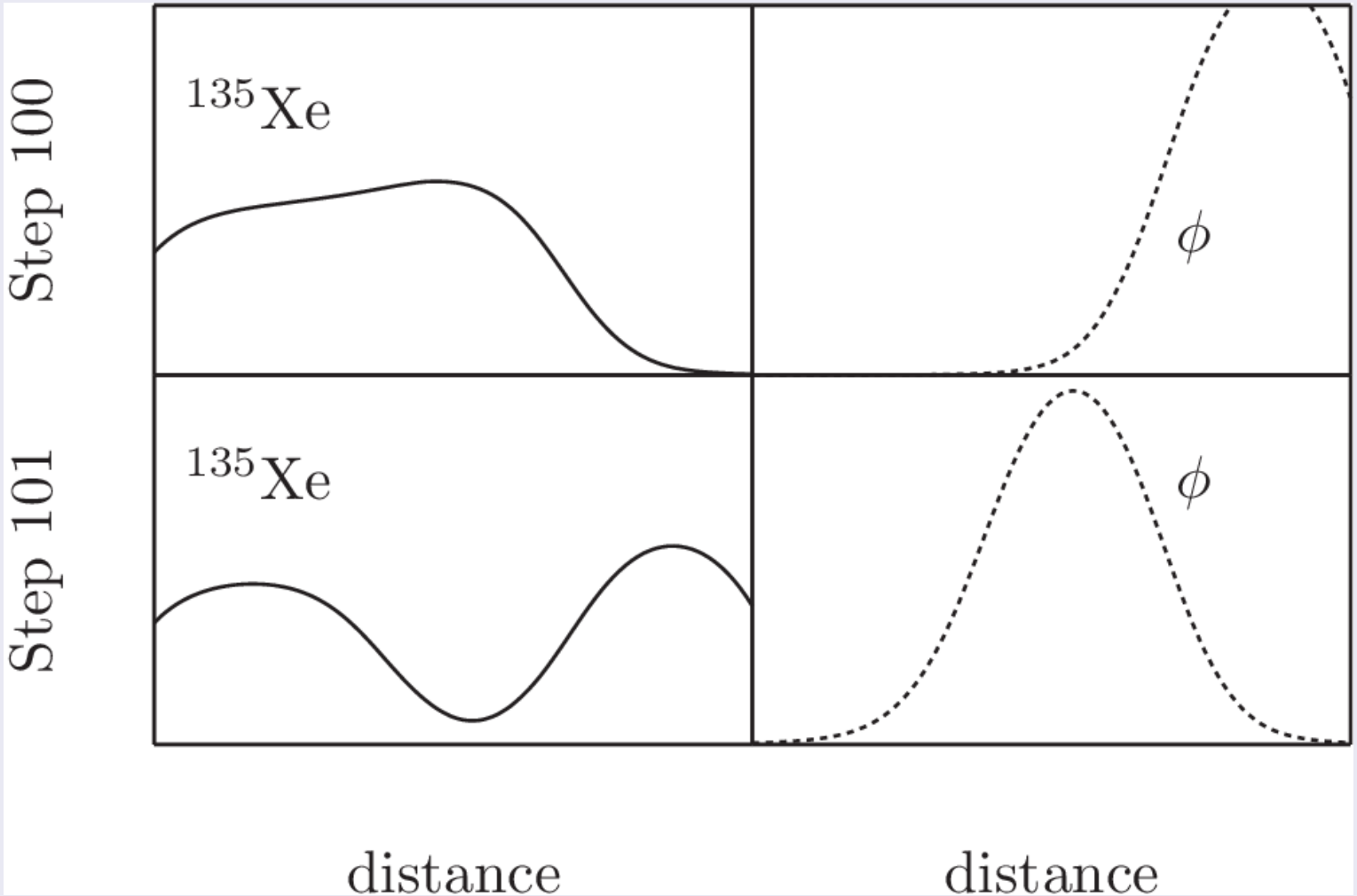
for $i \leftarrow 0, 1, \ldots$ **do**

$$\phi_i \leftarrow \phi_B(\vec{N}_i)$$
$$\vec{N}_{i+1}^{(\mathrm{P})} \leftarrow \exp[\mathbb{M}(\phi_i)\Delta t_i]\vec{N}_i$$
$$\phi_{i+1}^{(\mathrm{P})} \leftarrow \phi_B(\vec{N}_{i+1}^{(\mathrm{P})})$$
$$\bar{\phi}_i^{(\mathrm{C})} \leftarrow (\phi_i + \phi_{i+1}^{(\mathrm{P})})/2$$
$$\vec{N}_{i+1} \leftarrow \exp[\mathbb{M}(\bar{\phi}_i^{(\mathrm{C})})\Delta t_i]\vec{N}_i$$

end for

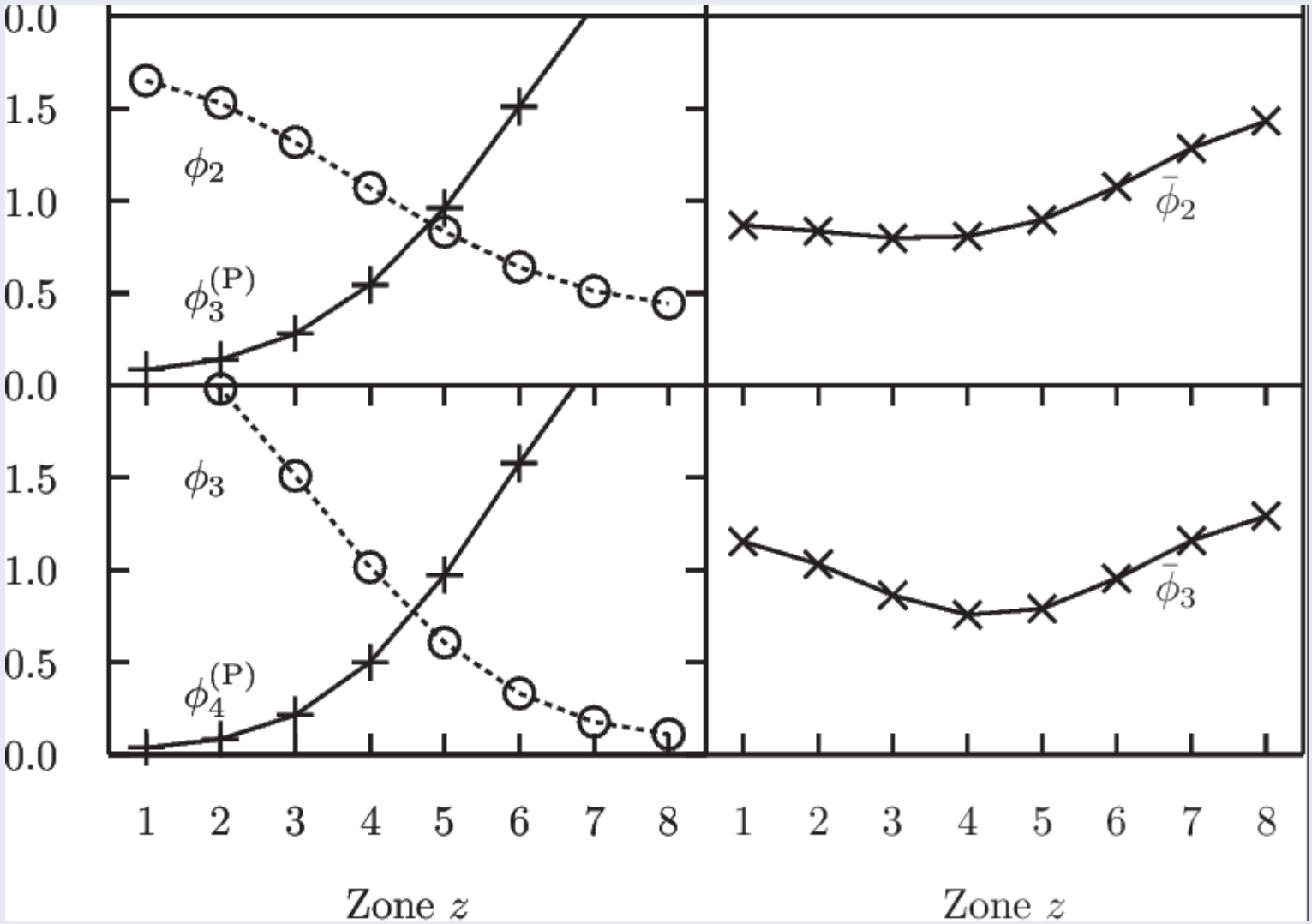
Predictor-Corrector

Stability test using a slab model ($\Delta t = 7$d)



Predictor-Corrector

SERPENT calculation, fuel rod with axial reflectors ($\Delta t = 30$ day)



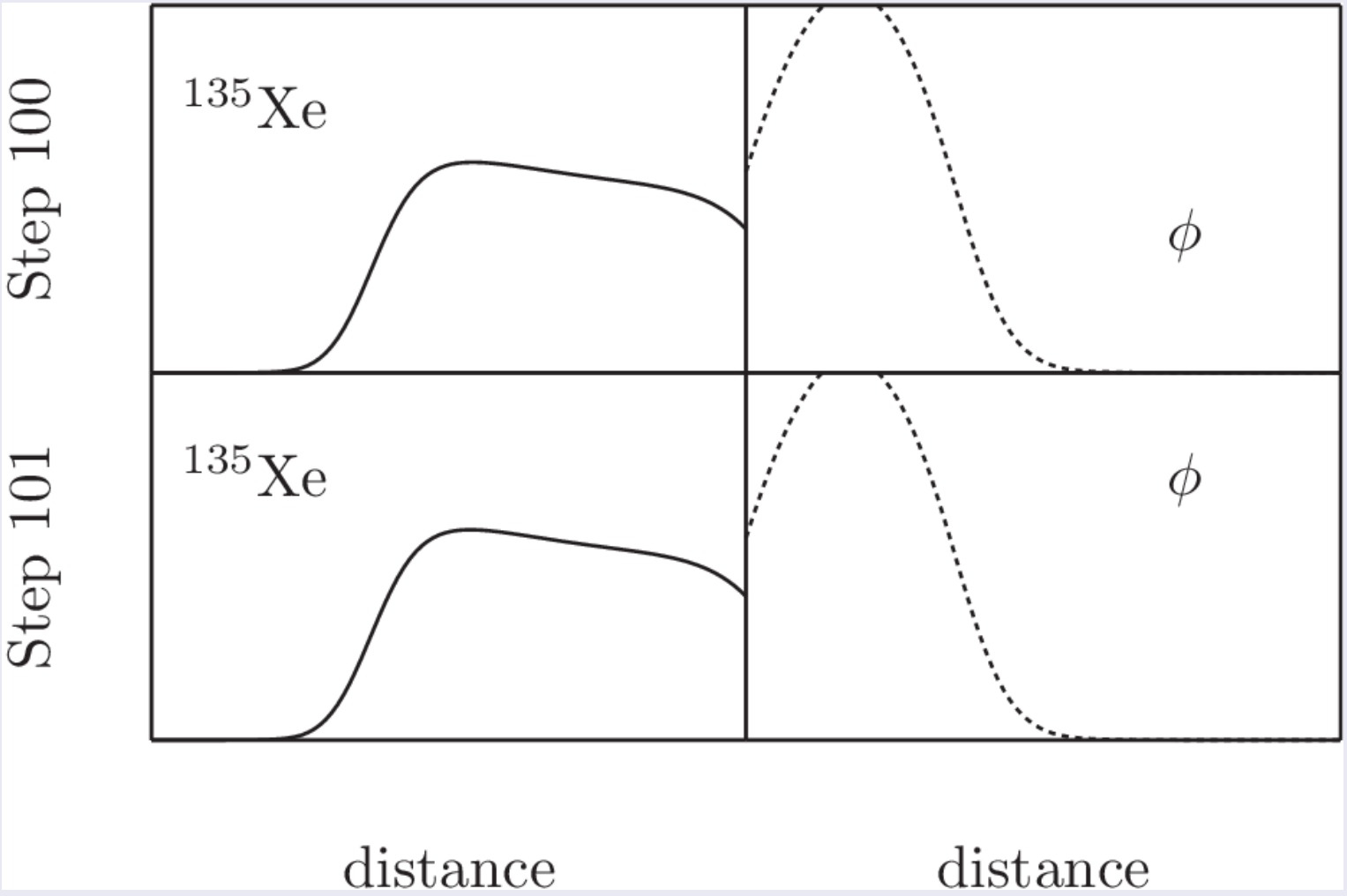
Mid-point method

The middle-of-step constant flux approximation (MONTEBURNS, MCNPX)

input: $\vec{N}_0$
for $i \leftarrow 0, 1, \ldots$ **do**
    $\phi_i \leftarrow \phi_B(\vec{N}_i)$
    $\vec{N}_{i+1/2} \leftarrow \exp[\mathbb{M}(\phi_i)\Delta t_i/2]\vec{N}_i$
    $\phi_{i+1/2} \leftarrow \phi_B(\vec{N}_{i+1/2})$
    $\vec{N}_{i+1} \leftarrow \exp[\mathbb{M}(\phi_{i+1/2})\Delta t_i]\vec{N}_i$
end for

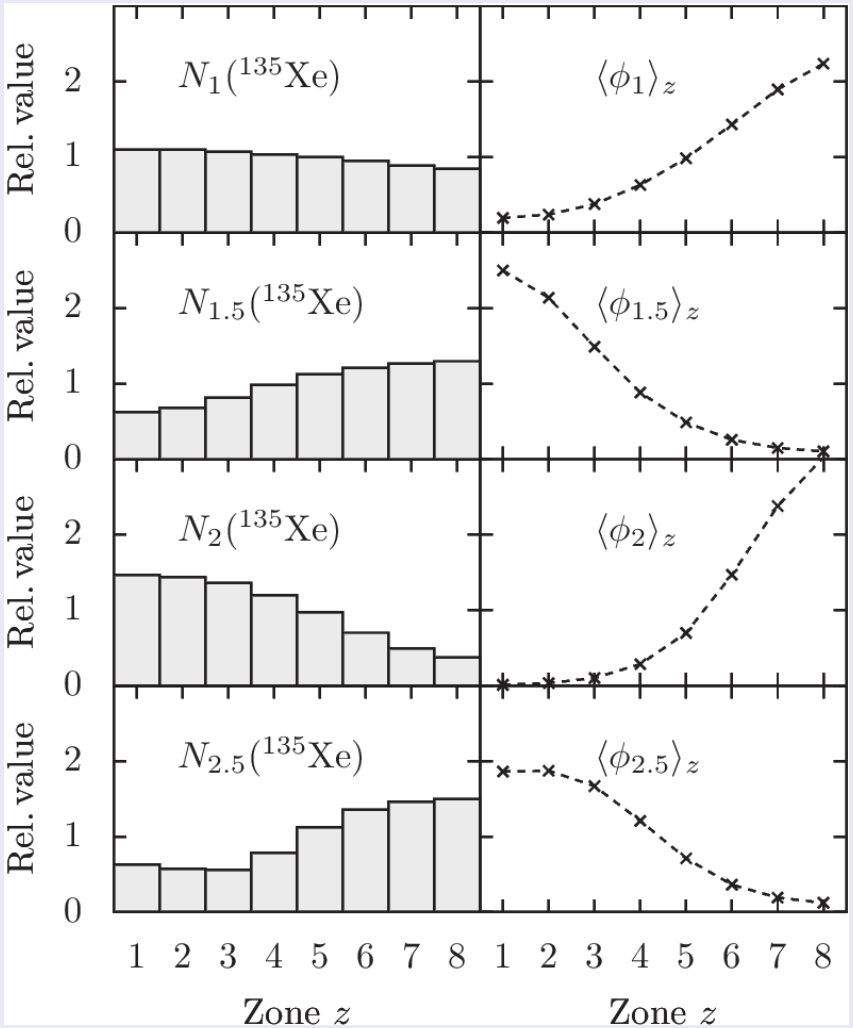
Mid-point method

Stability test using a slab model ($\Delta t = 7$d)



Mid-point method

MCNPX calculation, fuel rod with axial reflectors ($\Delta t = 4$ d)



Stochastic Implicit Euler (SIE) based coupling scheme for MC burnup calculations

Derivation of the SIE-based coupling scheme

Notes

- All previously shown schemes are based on so-called conditionally stable ODE methods.
- We choose to employ a stable method - the simplest one is the implicit Euler.
- In terms of MC burnup calculations, the implicit Euler method depletes the fuel with the end-of-step neutron flux over the whole time step.
- The problem now is to get the end-of-step flux.

Derivation of the SIE-based coupling scheme

Getting the end-of-step flux

Let $\vec{N}_i$ and ϕ_i denote the nuclide field and neutron flux at the end of i^{th} time step, respectively. When $\vec{N}_{i-1}$ is depleted with the end-of-step flux ϕ_i over the i^{th} time step then $\vec{N}_i$ equals

$$\vec{N}_i = \exp[\mathbb{M}(\phi_i, T)(t_i - t_{i-1})]\vec{N}_{i-1}, \tag{8}$$

while ϕ_i is given by

$$\phi_i = \phi_B(\vec{N}_i). \tag{9}$$

Substituting N_i from Eq. (8) into Eq. (9) forms a non-linear equation for ϕ_i,

$$\phi_i = \phi_B\left(\vec{N}_{i-1}\exp[\mathbb{M}(\phi_i, T)(t_i - t_{i-1})]\right) \tag{10}$$

that can be efficiently solved by the relaxation scheme with the Robbins–Monro algorithm.

SIE-based coupling scheme for MC burnup calculations

SIE-based coupling scheme (with relaxation of flux)

input: $\vec{N}_0$
$\phi_0 \leftarrow$ **fundamental mode of** $B(\vec{N}_0)$
for $i \leftarrow 0, 1, \ldots$ **do**
    $\vec{N}_{i+1}^{(0)} \leftarrow \exp[\mathbb{M}(\phi_i)\Delta t]\vec{N}_i$
    for $n \leftarrow 1, 2, \ldots, c$ **do**
        $\phi_{i+1}^{(n)} \leftarrow \phi_B(\vec{N}_{i+1}^{(n-1)})$
        $\bar{\phi}_{i+1}^{(n)} \leftarrow \sum_{j=1}^{n} \phi_{i+1}^{(j)}/n$
        $\vec{N}_{i+1}^{(n)} \leftarrow \exp[\mathbb{M}(\bar{\phi}_{i+1}^{(n)})\Delta t]\vec{N}_i$
    end for
    $\vec{N}_{i+1} \leftarrow \vec{N}_{i+1}^{(c)}$
    $\phi_{i+1} \leftarrow \bar{\phi}_{i+1}^{(c)}$
end for

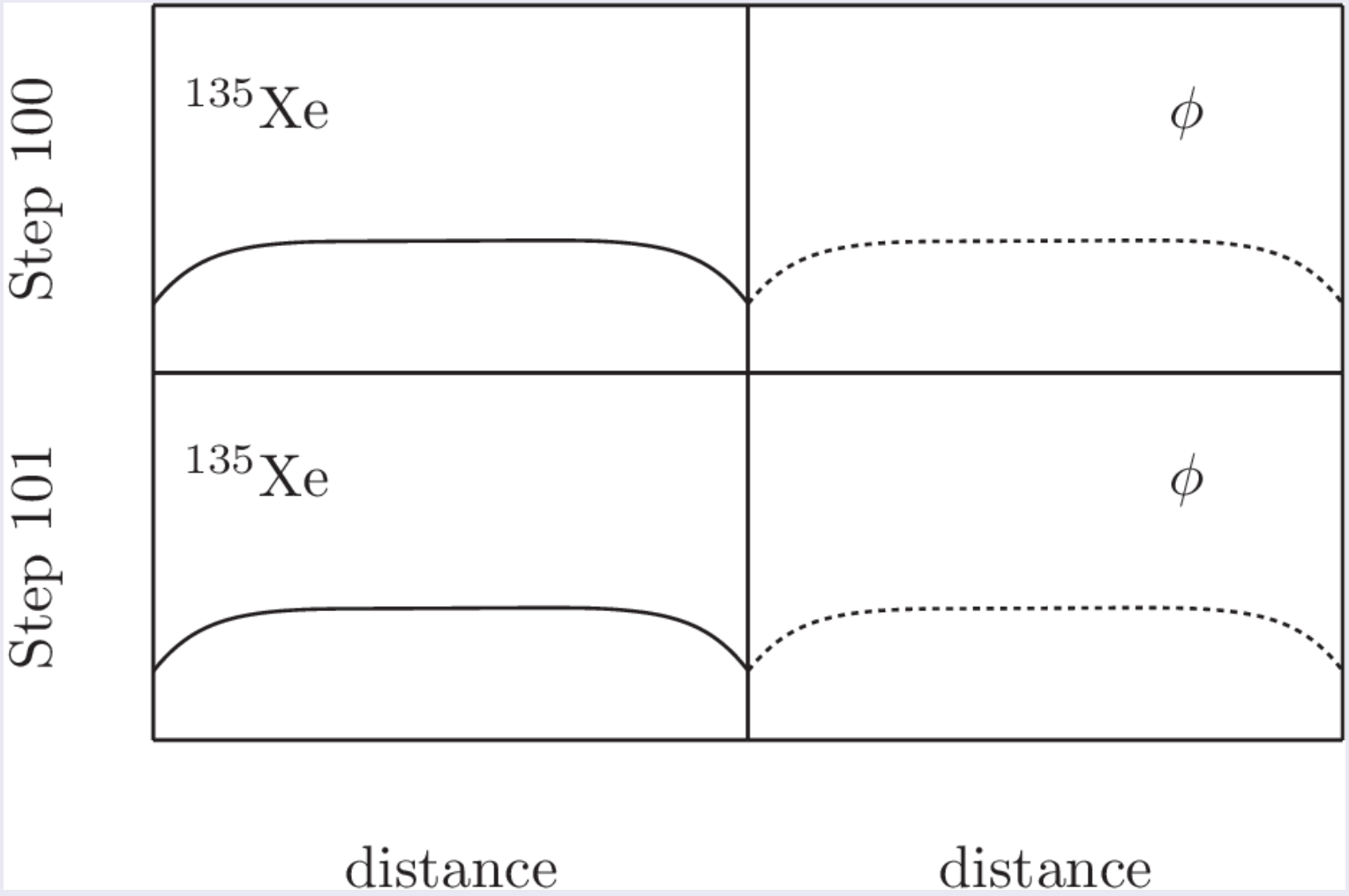
SIE-based coupling scheme for MC burnup calculations

SIE-based coupling scheme (with relaxation of nuclide field)

input: $\vec{N}_0$
$\phi_0 \leftarrow \phi_B(\vec{N}_0)$
for $i \leftarrow 0, 1, \ldots$ **do**
  $\bar{\vec{N}}_{i+1}^{(0)} \leftarrow \vec{N}_i \exp[\mathbb{M}(\phi_i)\Delta t]$
  for $n \leftarrow 1, 2, \ldots, c$ **do**
    $\phi_{i+1}^{(n)} \leftarrow \phi_B\left(\bar{\vec{N}}_{i+1}^{(n-1)}\right)$
    $\vec{N}_{i+1}^{(n)} \leftarrow \exp[\mathbb{M}(\phi_{i+1}^{(n)})\Delta t]\vec{N}_i$
    $\bar{\vec{N}}_{i+1}^{(n)} \leftarrow \sum_{j=1}^{n} \vec{N}_{i+1}^{(j)}/n$
  end for
  $\vec{N}_{i+1} \leftarrow \bar{\vec{N}}_{i+1}^{(c)}$
  $\phi_{i+1} \leftarrow \sum_{j=1}^{c} \phi_{i+1}^{(j)}/c$
end for

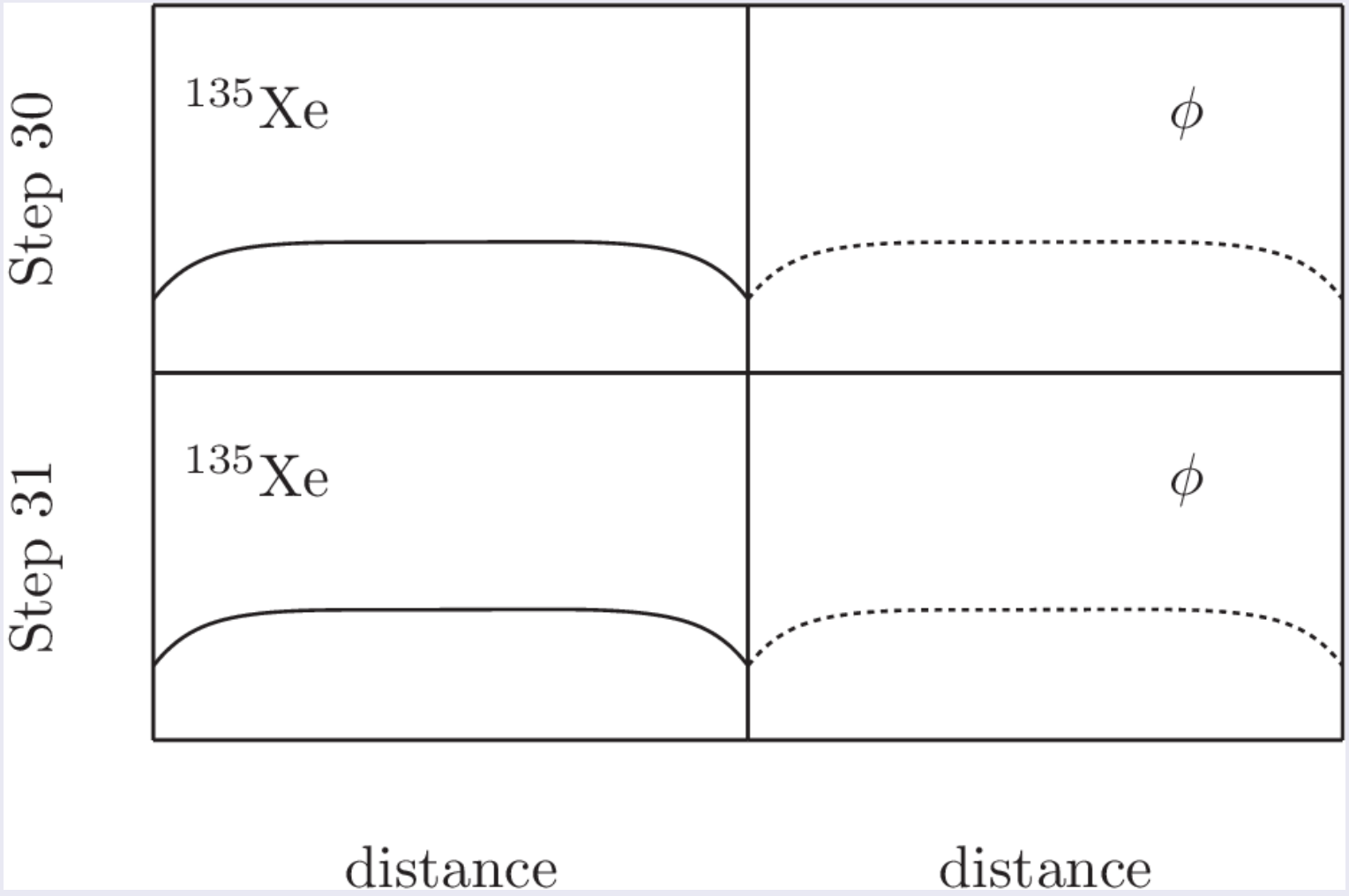
SIE-based coupling scheme for MC burnup calculations

Stability test using a slab model ($\Delta t = 7d$, 10 inner steps)



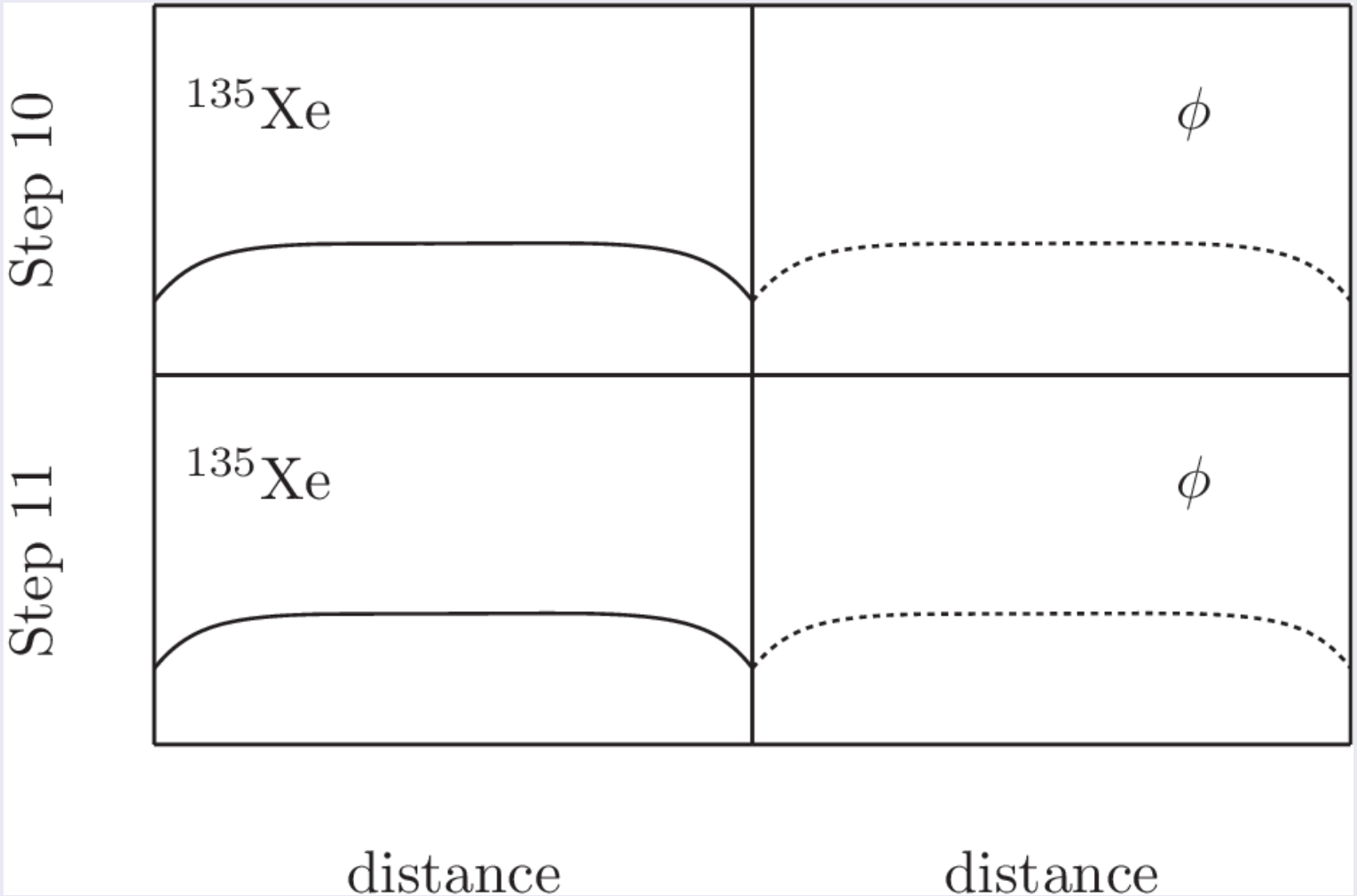
SIE-based coupling scheme for MC burnup calculations

Stability test using a slab model ($\Delta t = 30d$, 10 inner steps)



SIE-based coupling scheme for MC burnup calculations

Stability test using a slab model ($\Delta t = 90d$, 10 inner steps)



SIE-based coupling scheme for MC burnup calculations

Stability test using a slab MC model ($\Delta t = 90d$, 10 inner steps)