Monte Carlo Methods and Simulations in Nuclear Technology

Monte Carlo Burnup Calculations

Jan Dufek

2023

KTH Royal Institute of Technology

Contents

Governing Equations in MC Burnup Calculations

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

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

where

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

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

Formal solution of the burnup equation

The burnup equation has a formal solution

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

for ϕ fixed in time.

Governing equations

Criticality equation

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

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

$$\phi = \textbf{fundamental mode of } B$$

Governing equations

Fuel cycle calculations

Most commonly, MC burnup codes are used for simulations of **fuel cycles**. The purpose o fuel cycle simulation is to deplete the fuel with the **steady-state** power distribution (that changes in time).

Common coupling schemes in 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 (the beginning-of-step constant flux approximation)
- Predictor-Corrector
- Mid-point method (the middle-of-step constant flux approximation)

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. $\quad \phi_i \leftarrow$ **fundamental mode of** $B(\vec{N}_i)$
4. $\quad \vec{N}_{i+1} \leftarrow \exp[\mathbb{M}(\phi_i)\Delta t_i]\vec{N}_i$
5. end for

Stability of explicit Euler method

The explicit Euler is conditionally stable, i.e., it is stable for sufficiently short time steps. Numerical instability of this scheme is common in Monte Carlo burnup simulations with strong untreated feedbacks (mainly xenon-135) - the results may then contain large errors. See the examples on next slides.

Explicit Euler

Stability test using a simple num. model ($\Delta t = 7$d)

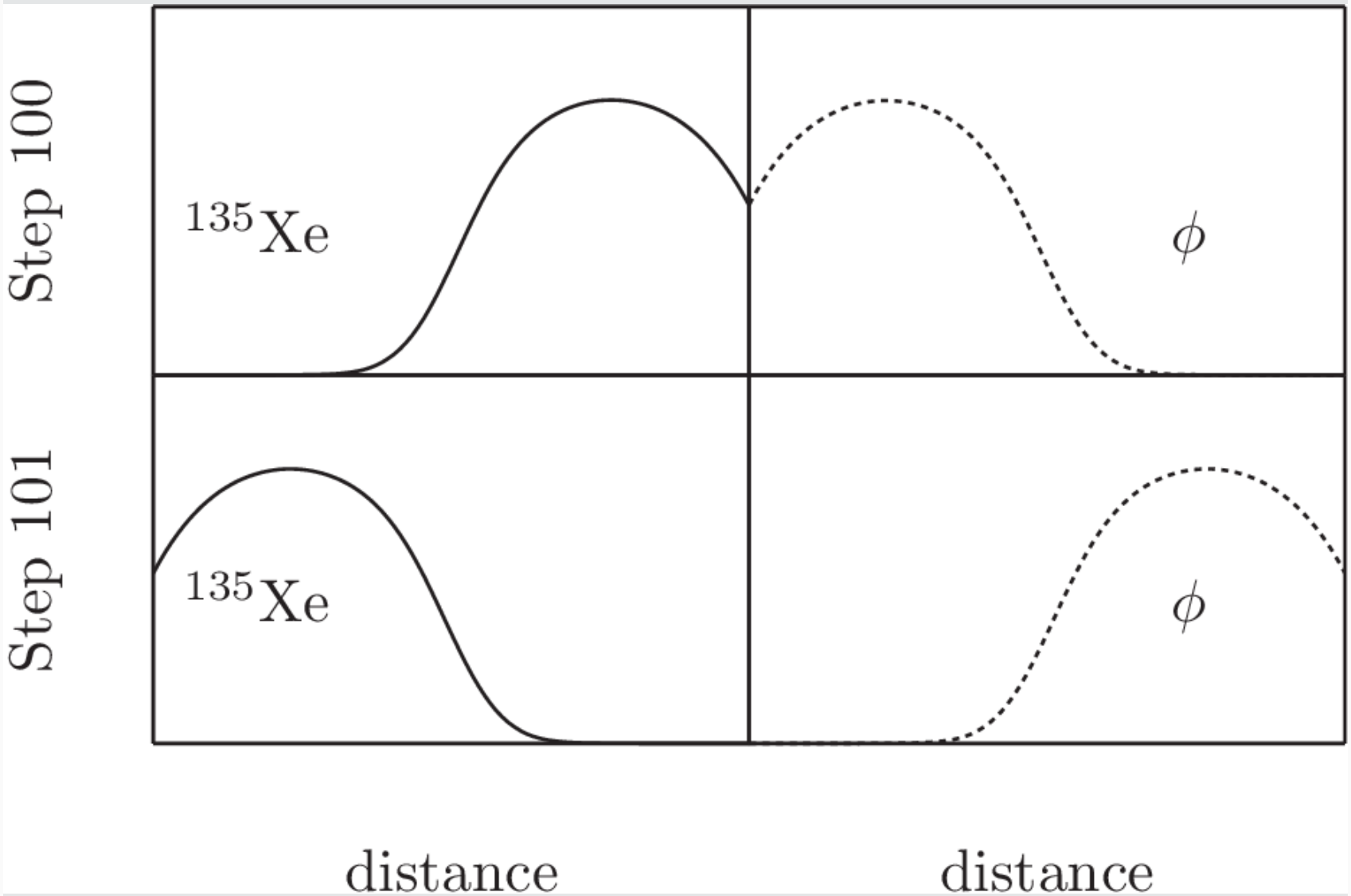


Figure 1: Slab with black BC. Deterministic solver.

MCB calculation ($\Delta t = 3$d)

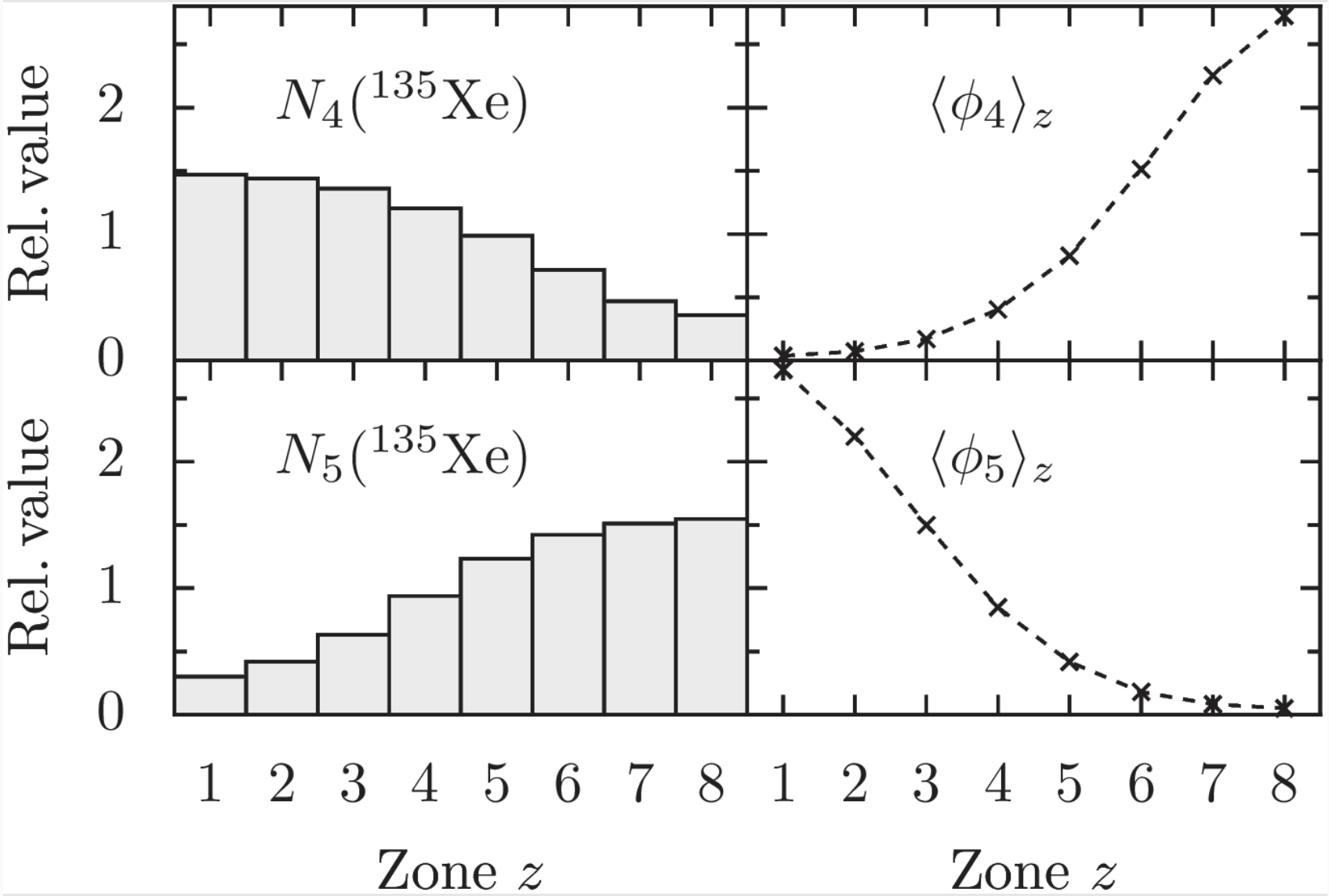


Figure 2: Slab with reflective BC. MC solver.

Predictor-Corrector

The predictor-corrector scheme (MCODE, SERPENT)

1. **input:** $\vec{N}_0$
2. for $i \leftarrow 0, 1, \ldots$ do
3. $\quad \phi_i \leftarrow$ **fundamental mode of** $B(\vec{N}_i)$
4. $\quad \vec{N}_{i+1}^{(\mathrm{P})} \leftarrow \exp[\mathbb{M}(\phi_i)\Delta t_i]\vec{N}_i$
5. $\quad \phi_{i+1}^{(\mathrm{P})} \leftarrow$ **fundamental mode of** $B(\vec{N}_{i+1}^{(\mathrm{P})})$
6. $\quad \bar{\phi}_i^{(\mathrm{C})} \leftarrow (\phi_i + \phi_{i+1}^{(\mathrm{P})})/2$
7. $\quad \vec{N}_{i+1} \leftarrow \vec{N}_i \exp[\mathbb{M}(\bar{\phi}_i^{(\mathrm{C})})\Delta t_i]$
8. end for

Stability of predictor-corrector scheme

The predictor-corrector scheme is also conditionally stable, and similar instabilities as for EE scheme may develop for PC scheme as well. See the examples on next slides.

Stability test using a simple num. model ($\Delta t = 7$d)

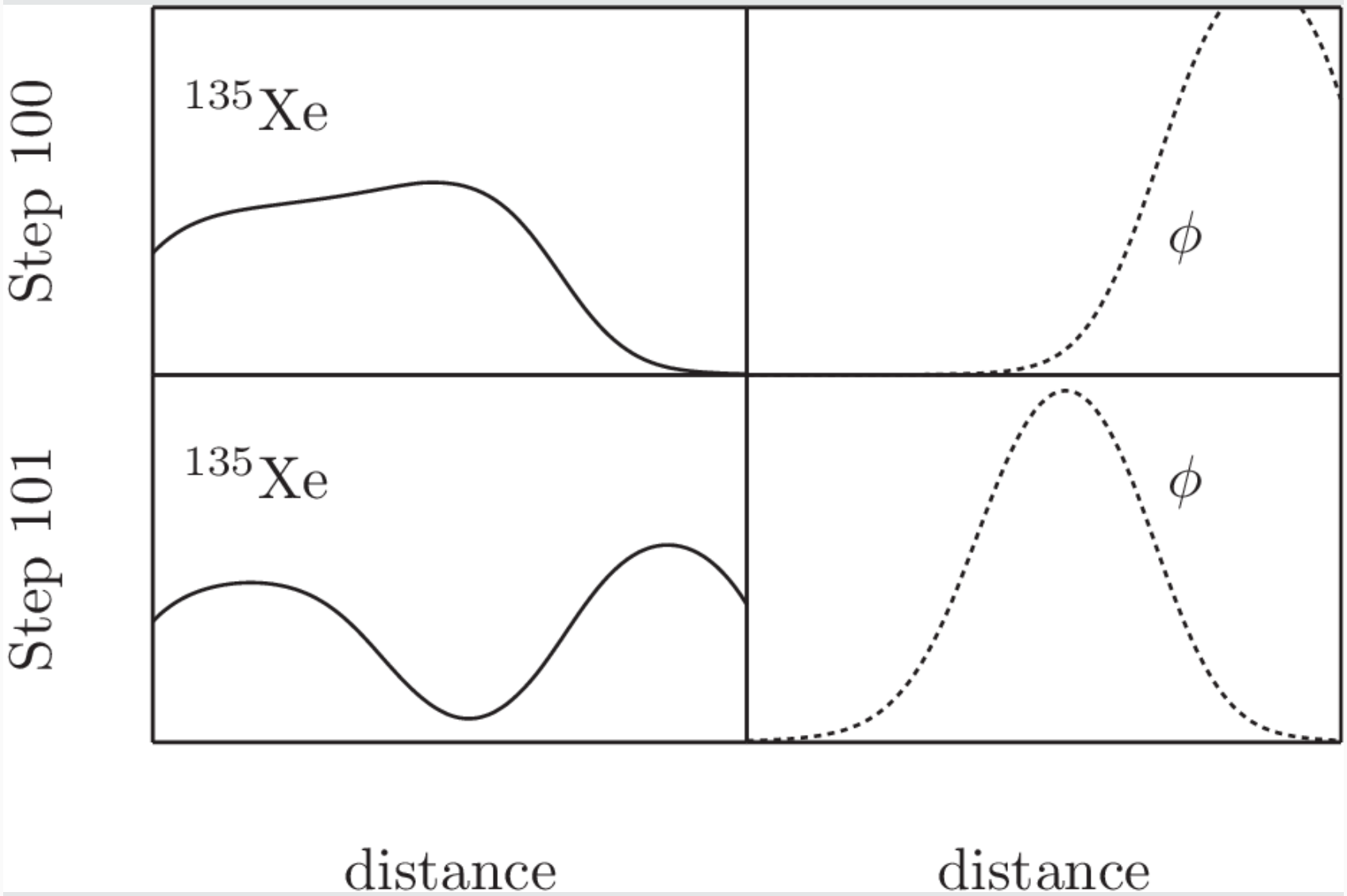
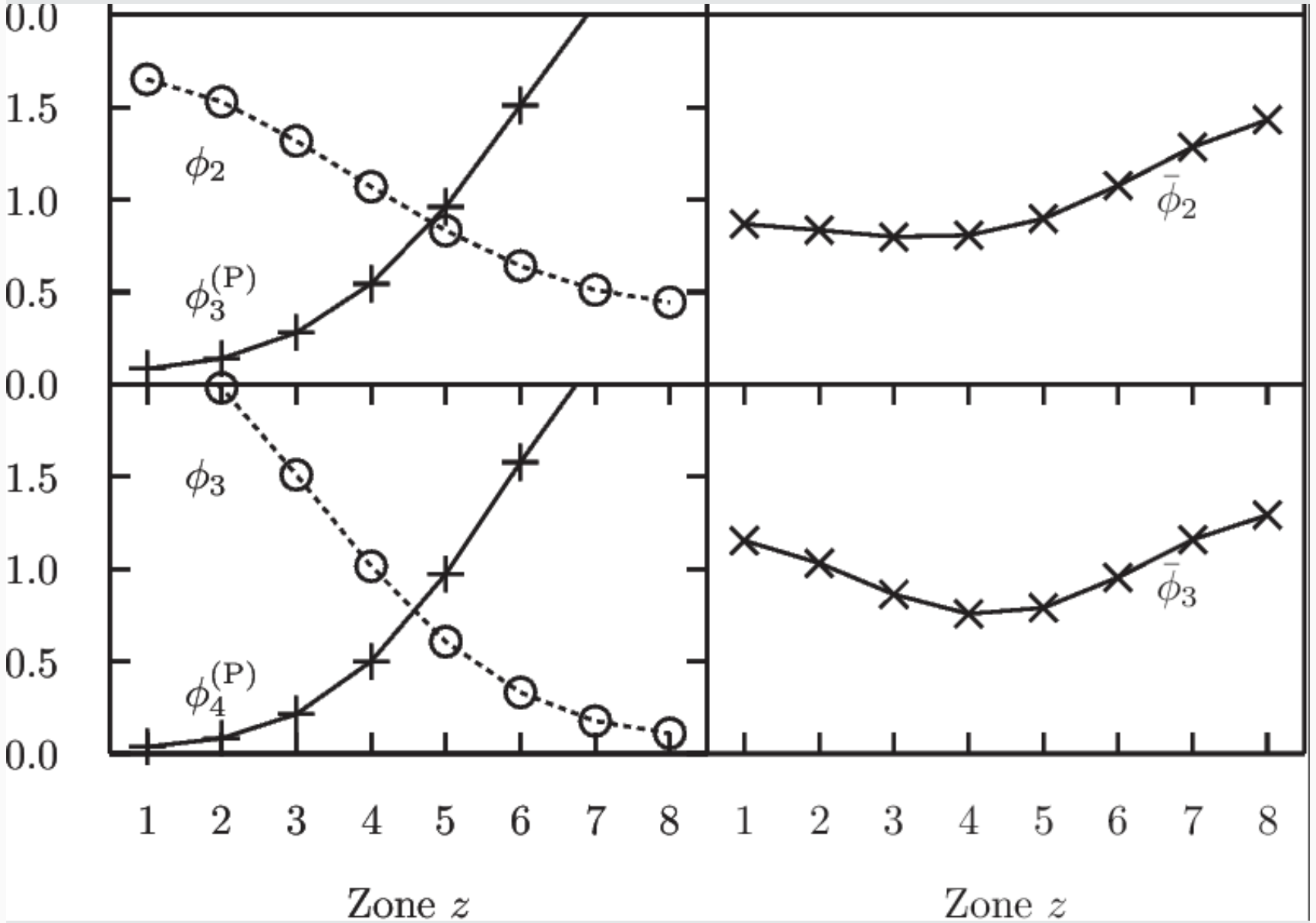


Figure 3: Slab with black BC. Deterministic solver.

Predictor-Corrector

SERPENT calculation ($\Delta t = 30$ day)



Mid-point method

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

1. **input:** $\vec{N}_0$
2. for $i \leftarrow 0, 1, \ldots$ do
3. $\quad \phi_i \leftarrow$ **fundamental mode of** $B(\vec{N}_i)$
4. $\quad \vec{N}_{i+1/2} \leftarrow \vec{N}_i \exp[\mathbb{M}(\phi_i)\Delta t_i/2]$
5. $\quad \phi_{i+1/2} \leftarrow$ **fundamental mode of** $B(\vec{N}_{i+1/2})$
6. $\quad \vec{N}_{i+1} \leftarrow \vec{N}_i \exp[\mathbb{M}(\phi_{i+1/2})\Delta t_i]$
7. end for

Stability of the mid-point method

The mid-point method is also conditionally stable, and similar instabilities as for EE scheme may develop as well. See the examples on next slides.

Mid-point method

Stability test using a simple num. model ($\Delta t = 7$d)

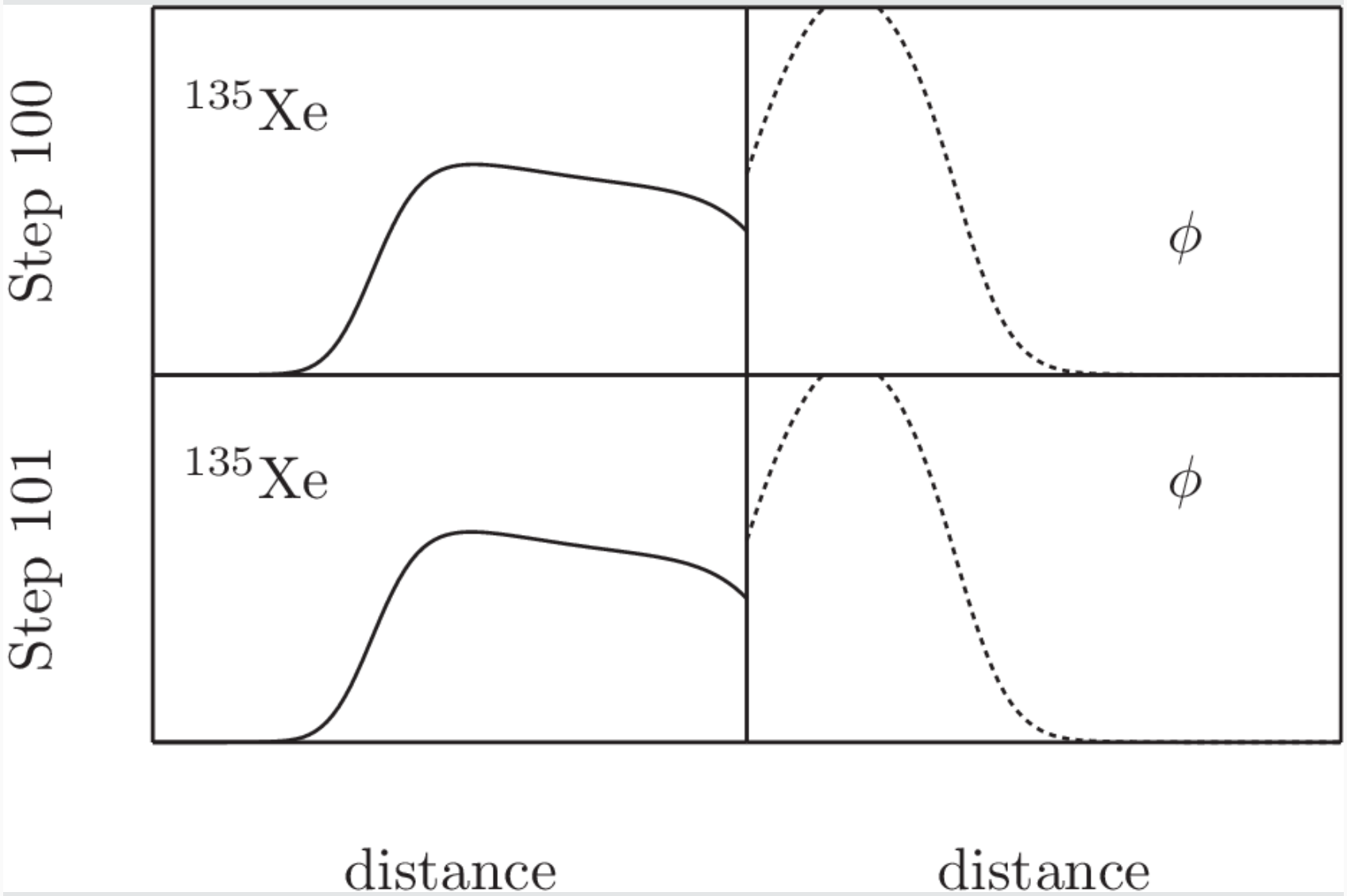


Figure 5: Slab with black BC. Deterministic solver.

Mid-point method

MCNPX calculation ($\Delta t = 4$ d)

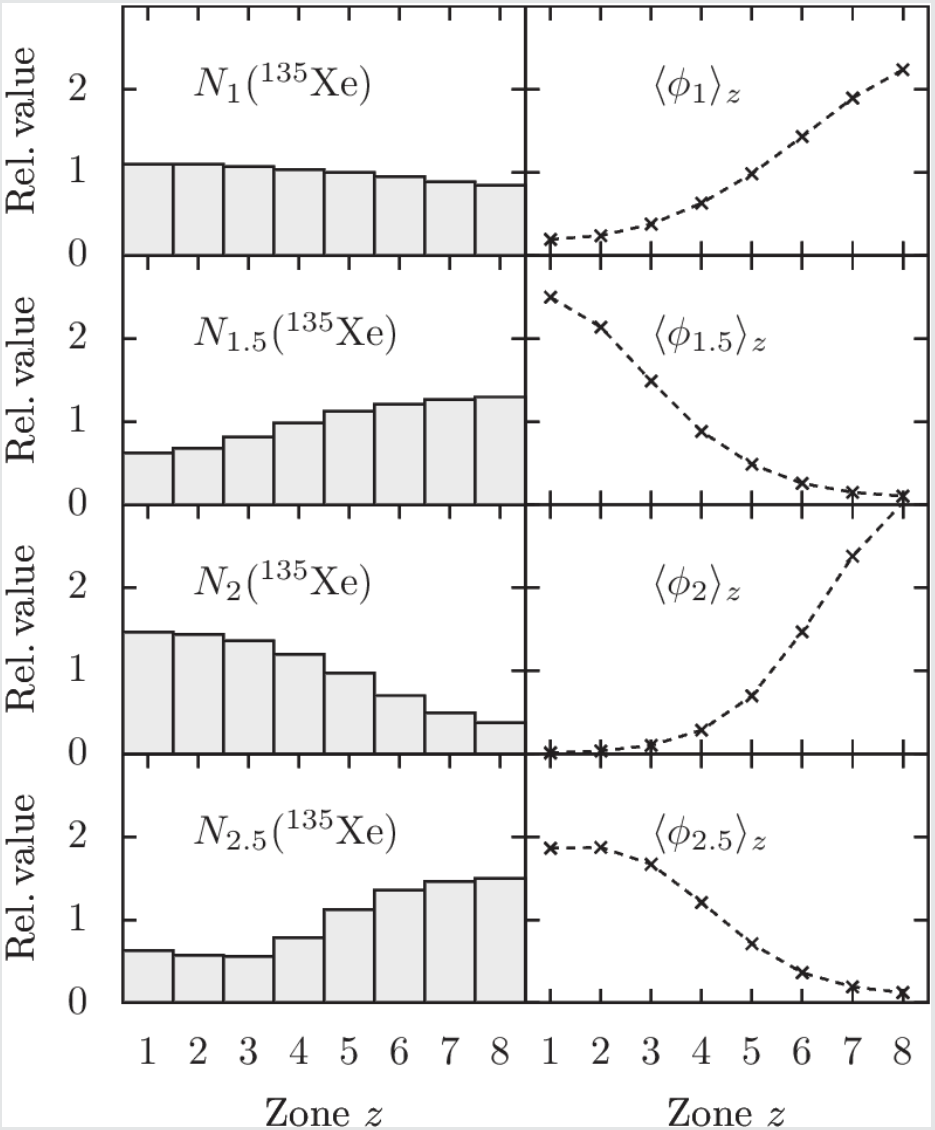


Figure 6: Slab with reflective BC. MC solver.

Stochastic Implicit Euler based coupling scheme for MC burnup calculations

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

SIE-based coupling scheme

1. **input:** $\vec{N}_0$
2. $\phi_0 \leftarrow$ **fundamental mode of** $B(\vec{N}_0)$
3. for $i \leftarrow 0, 1, \ldots$ do
4. $\quad \vec{N}_{i+1}^{(0)} \leftarrow \exp[\mathbb{M}(\phi_i)\Delta t]\vec{N}_i$
5. $\quad$ for $n \leftarrow 1, 2, \ldots$ do
6. $\quad\quad \phi_{i+1}^{(n)} \leftarrow$ **fundamental mode of** $B(\vec{N}_{i+1}^{(n-1)})$
7. $\quad\quad \bar{\phi}_{i+1}^{(n)} \leftarrow \sum_{j=1}^{n} \phi_{i+1}^{(j)}/n$
8. $\quad\quad \vec{N}_{i+1}^{(n)} \leftarrow \vec{N}_i \exp[\mathbb{M}(\bar{\phi}_{i+1}^{(n)})\Delta t]$
9. $\quad$ end for
10. $\quad \vec{N}_{i+1} \leftarrow \vec{N}_{i+1}^{(c)}$
11. $\quad \phi_{i+1} \leftarrow \bar{\phi}_{i+1}^{(c)}$
12. end for

Stability of SIE

The SIE method is unconditionally stable (stable for any time step), nevertheless only when a sufficient number of inner iterations is set up. See examples on next slides.

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

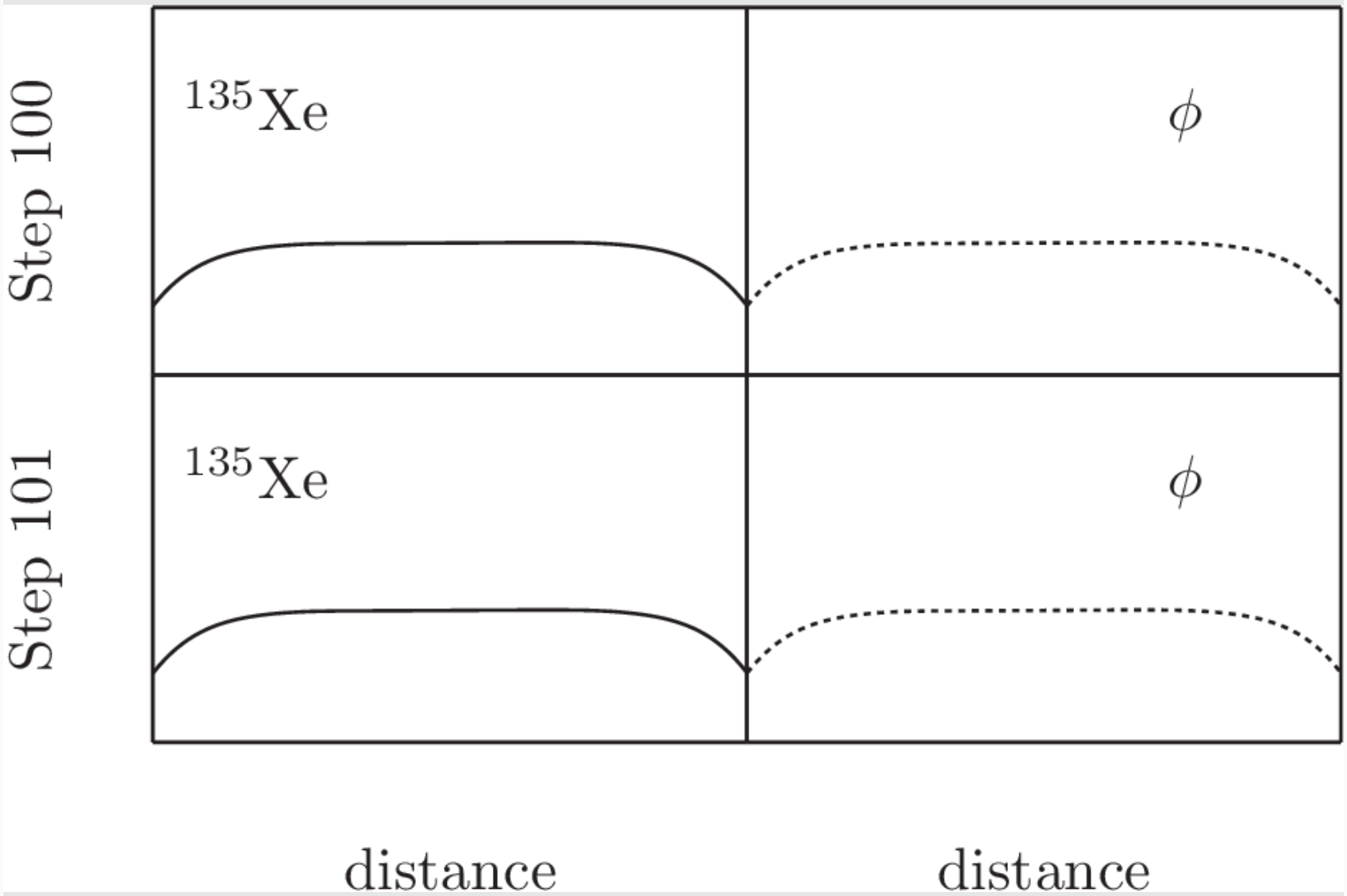


Figure 7: Slab with black BC. Deterministic solver.

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

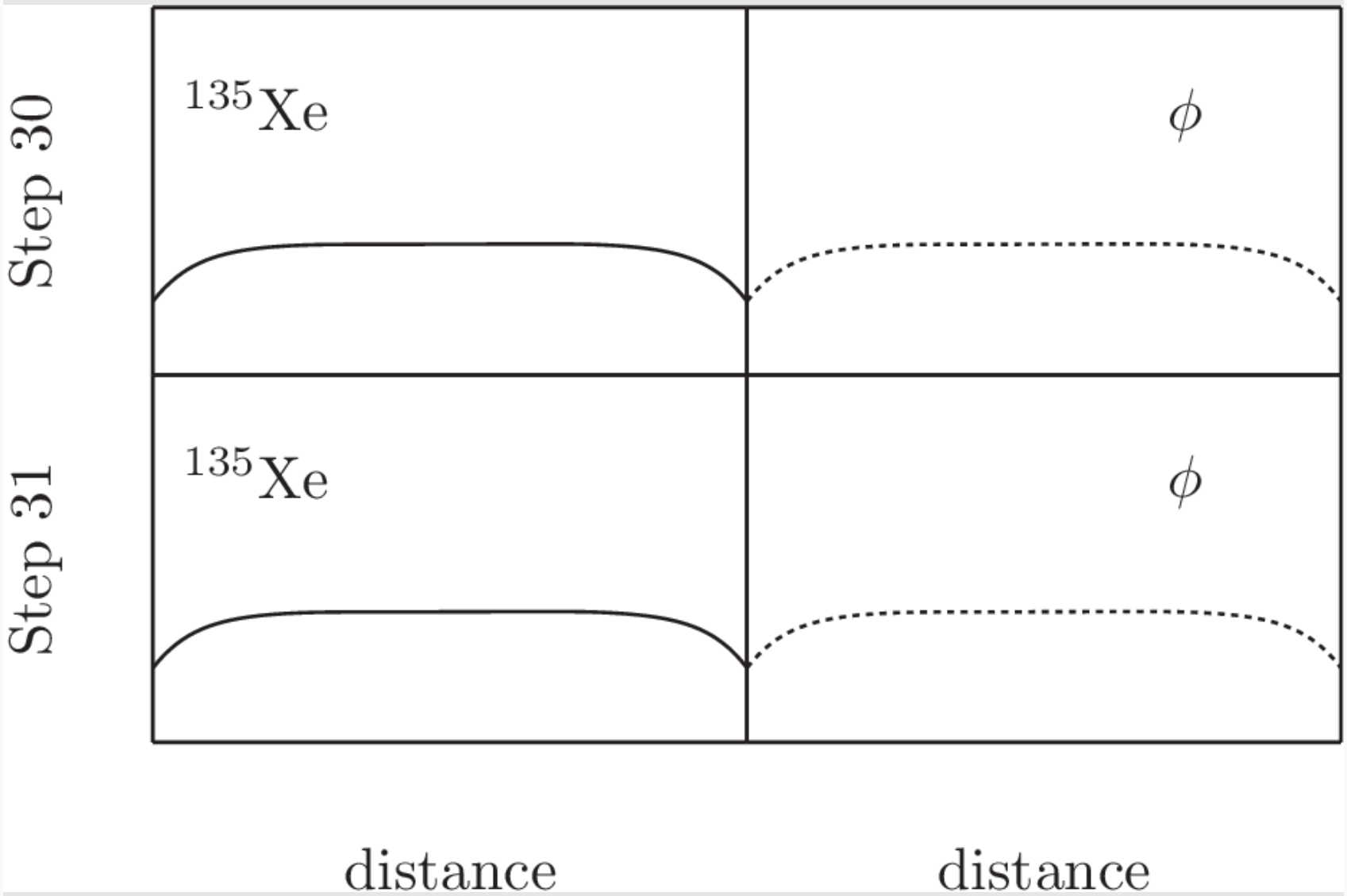


Figure 8: Slab with black BC. Deterministic solver.

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

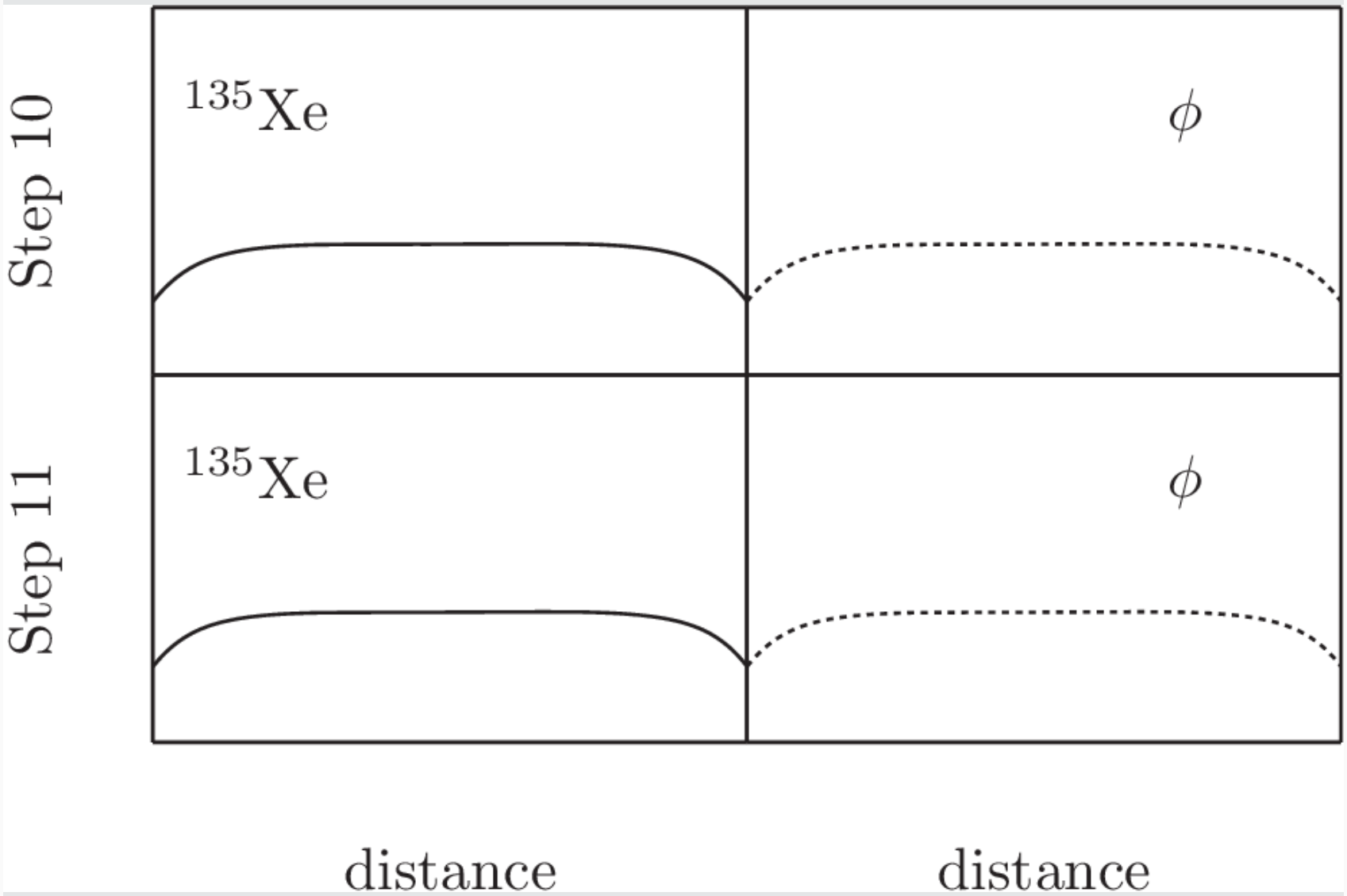


Figure 9: Slab with black BC. Deterministic solver.

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

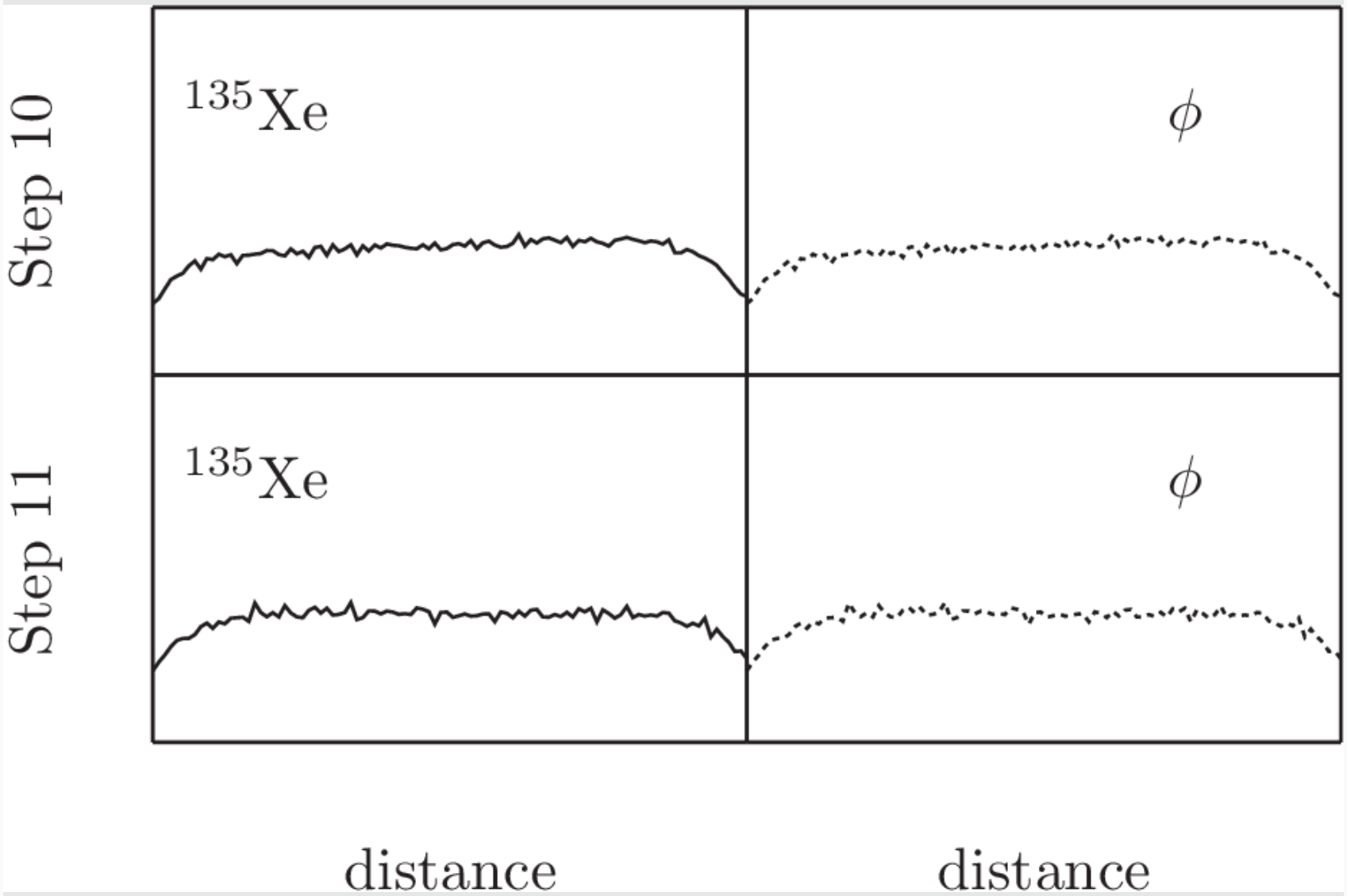


Figure 10: Slab with black BC. Deterministic solver + random noise.

Factors affecting the numerical stability

Factors affecting the numerical stability

Num. stability is affected by:

- **Number of fuel materials set for depletion:** The more fuel materials are defined the easier it is for xenon-powered oscillations to develop.
- **Strength of modelled feedbacks:** The xenon feedback is weak in fast reactors -> no numerical instabilities can develop there.
- **Size of the system (dominance ratio):** The closer the dominance ratio is to one the more difficult is to ensure numerical stability of burnup simulations.
- **Time step size:** The larger the time steps -> the less stable the burnup simulations.
- **Coupling scheme:** A stable coupling scheme can ensure stability.
- **Special treatment of xenon:** Some codes can compute equilibrium xenon concentration during criticality calculations. Activate this feature since it will ensure stability under almost any conditions.

Factors affecting the numerical stability

Special case

Note that in the case that you actually want, for some reason, to simulate the real xenon oscillation then you must set up very short time steps (e.g. several hours) and you must deactivate any special treatment of xenon (deactivate the equilibrium xenon).

Optimal efficiency of MC burnup simulations

- The efficiency of MC burnup simulations is affected by the choice of free parameters, such as the time step length, the number of cycles per criticality calculation and the number of neutrons per criticality cycle.
- Latest research suggests that the optimal burnup process occurs when the burnup solver cost is approximately equal to the MC solver cost in the MC burnup simulation.