# Workshop: Magnets Problem

*Adám Brudzewsky*
*Richard Park*
*Rodrigo Girão Serrão*

## Workshop Overview

- Day 1: TotalEnergy
  a. Algorithms
  b. Writing general code
  c. Exercises
- Day 2: Simulate
  a. Code Review
  b. Performance Tuning
  c. Exercises

## TotalEnergy

$$E = -J \sum_{\langle ij \rangle} S_i S_j$$

$$\begin{array}{ccc} & N & \\ W & s & E \\ & S & \end{array}$$

$$s_E = -J \times s \times (N + E + S + W)$$

# TotalEnergy using Stencil

$$\begin{array}{cc} \begin{bmatrix} -1 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix} & \begin{array}{cc} -1 & 1 \\ -1 & -1 \\ 1 & -1 \end{array} \\ \begin{array}{ccc} -1 & 1 & 1 \\ 1 & -1 & -1 \end{array} & \begin{array}{cc} 1 & 1 \\ -1 & 1 \end{array} \end{array}$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} -1 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

## TotalEnergy using Stencil

$$\begin{array}{cc} \begin{bmatrix} -1 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix} & \begin{matrix} -1 & 1 \\ -1 & -1 \\ 1 & -1 \end{matrix} \\ \begin{matrix} -1 & 1 & 1 \\ 1 & -1 & -1 \end{matrix} & \begin{matrix} 1 & 1 \\ -1 & 1 \end{matrix} \end{array}$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} -1 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

## TotalEnergy using Shifting

$$\begin{array}{cc}
\left[\begin{array}{ccc} -1 & -1 & 1 \\ 1 & \textcircled{$-1$} & 1 \\ 1 & 1 & -1 \end{array}\right] & \begin{array}{cc} -1 & 1 \\ -1 & -1 \\ 1 & -1 \end{array} \\
\begin{array}{ccc} -1 & 1 & 1 \\ 1 & -1 & -1 \end{array} & \begin{array}{cc} 1 & 1 \\ -1 & 1 \end{array}
\end{array}$$

# TotalEnergy using Shifting

$$
\begin{array}{ccccc}
0 & -1 & -1 & 1 & -1 \\
0 & \boxed{1} & -1 & 1 & -1 \\
0 & 1 & 1 & -1 & 1 \\
0 & -1 & 1 & 1 & 1 \\
0 & 1 & -1 & -1 & -1
\end{array}
$$

## TotalEnergy using Shifting

$$\begin{array}{cc} \begin{bmatrix} -1 & -1 & 1 \\ \enclose{circle}{1} & \color{orange}{-1} & 1 \\ 1 & 1 & -1 \end{bmatrix} & \begin{array}{c} -1 \\ -1 \\ 1 \end{array} \\ \begin{array}{ccc} -1 & 1 & 1 \\ 1 & -1 & -1 \end{array} & \begin{array}{c} 1 \\ -1 \end{array} \end{array}$$

## TotalEnergy using Shifting

$$\begin{array}{cc}
\left[\begin{array}{ccc} -1 & -1 & 1 \\ 1 & \enclose{circle}{-1} & 1 \\ 1 & 1 & -1 \end{array}\right] & \begin{array}{cc} -1 & 1 \\ -1 & -1 \\ 1 & -1 \end{array} \\
\begin{array}{ccc} -1 & 1 & 1 \\ 1 & -1 & -1 \end{array} & \begin{array}{cc} 1 & 1 \\ -1 & 1 \end{array}
\end{array}$$

## TotalEnergy using Shifting

$$\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & 1 & 1 \end{array}$$

## TotalEnergy using Shifting

$$\begin{array}{cc} \begin{bmatrix} -1 & -1 & 1 \\ 1 & \enclose{circle}{-1} & 1 \\ 1 & 1 & -1 \end{bmatrix} & \begin{array}{cc} -1 & 1 \\ -1 & -1 \\ 1 & -1 \end{array} \\ \begin{array}{ccc} -1 & 1 & 1 \\ 1 & -1 & -1 \end{array} & \begin{array}{cc} 1 & 1 \\ -1 & 1 \end{array} \end{array}$$

## TotalEnergy using N-wise Reduce

$$\begin{matrix} -1 & -1 & 1 & -1 & 1 \\ 1 & \enclose{circle}{-1} & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 \end{matrix}$$

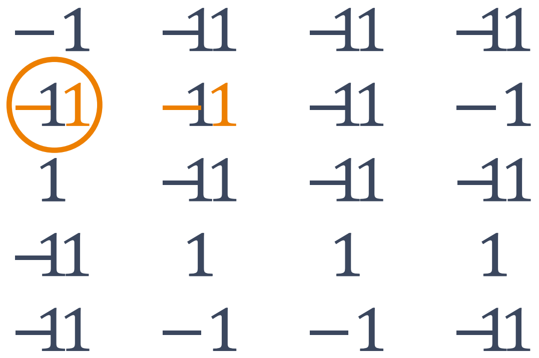
# TotalEnergy using N-wise Reduce

$$\begin{array}{ccccc}
-1 & -1 & 1 & -1 & 1 \\
\multicolumn{2}{c}{\boxed{1 \;\; \times -1}} & 1 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 1 & 1 & 1 \\
1 & -1 & -1 & -1 & 1
\end{array}$$

# TotalEnergy using N-wise Reduce

$$\begin{array}{ccccc}
-1 & -1 & 1 & -1 & 1 \\
\multicolumn{2}{c}{\boxed{1 \quad \times -1}} & 1 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 1 & 1 & 1 \\
1 & -1 & -1 & -1 & 1
\end{array}$$

## TotalEnergy using N-wise Reduce



## TotalEnergy using N-wise Reduce

$$\begin{matrix} 1 & -1 & -1 & -1 \\ \bigcirc\!\!\!\!\!\!-1 & -1 & -1 & 1 \\ 1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \end{matrix}$$

# TotalEnergy using N-wise Reduce

$$\begin{array}{ccccc}
0 & 1 & -1 & -1 & -1 \\
0 & \boxed{-1} & -1 & -1 & 1 \\
0 & 1 & -1 & -1 & -1 \\
0 & -1 & 1 & 1 & 1 \\
0 & -1 & 1 & 1 & -1
\end{array}$$

## Break

# Maintainability

- Others (and our future selves) can easily understand our code
  *Code is read much more often than it is written, so plan accordingly*

- It is easy to make changes to the behaviour

## Changing the Rules

- Change "constants"
  - Interaction constant
  - Temperature
- Add an external magnetic field

## Changing the Rules

- Which neighbours
    - Nearest neighbours
    - Anisotropic influence
    - Distant neighbours

## Changing the Rules

- World shape
  - Plane
  - Cylinder
  - Torus

## Interaction Constant

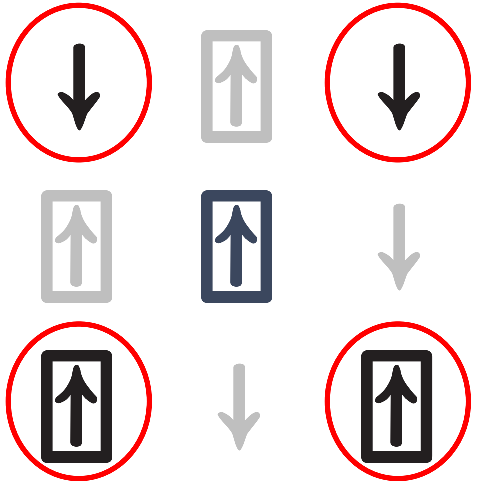
$$E = \bigcirc\!\!\!\!\!\!\!\!\!\!\;\; -J \sum_{ij} s_i s_j$$

## External Field

$$E = -J \sum_{ij} s_i s_j - h \sum_{j} s_j$$

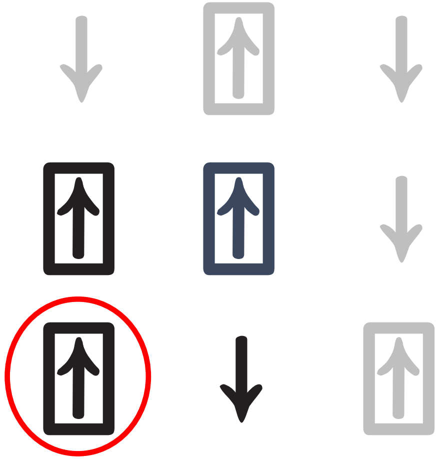
## Change contribution from neighbours

- Corners also contribute



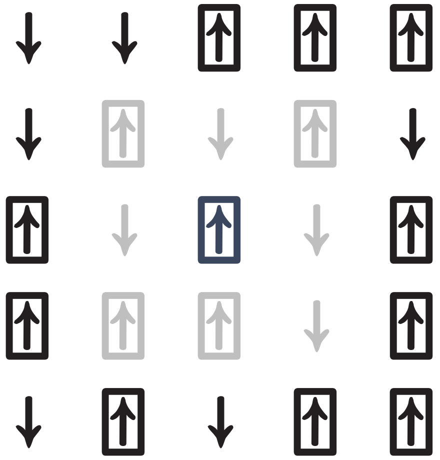
## Change contribution from neighbours

- Anisotropic: southwest neighbours contribute more



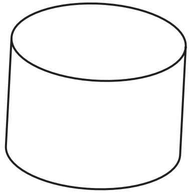
## Change contribution from neighbours

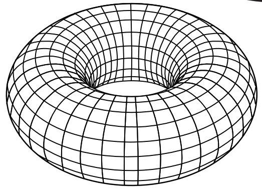
- More distant neighbours contribute more than nearby neighbours



# Change the World Shape

- Bounded plane
  From the problem description, we do not flip edge spins
- Cylinder: one edge wraps around
- Torus: all edges wrap around
- ***BONUS:*** Consider
  - Non-rectangular lattice
  - 3D (or higher?)





## Exercise

For each of the approaches we have looked at, modify your code to allow the system to be changed:

- Interaction constant
- Constant external field
- Modifiable neighbourhood

Which approaches do you find easy to understand? Which are easiest to change?

# Exercise: Neighbourhood

Consider:

- A static neighbourhood (similar to the problem description, Boolean)
- A function of position and/or distance relative the "this spin"
- How will you represent the neighbourhood influence?

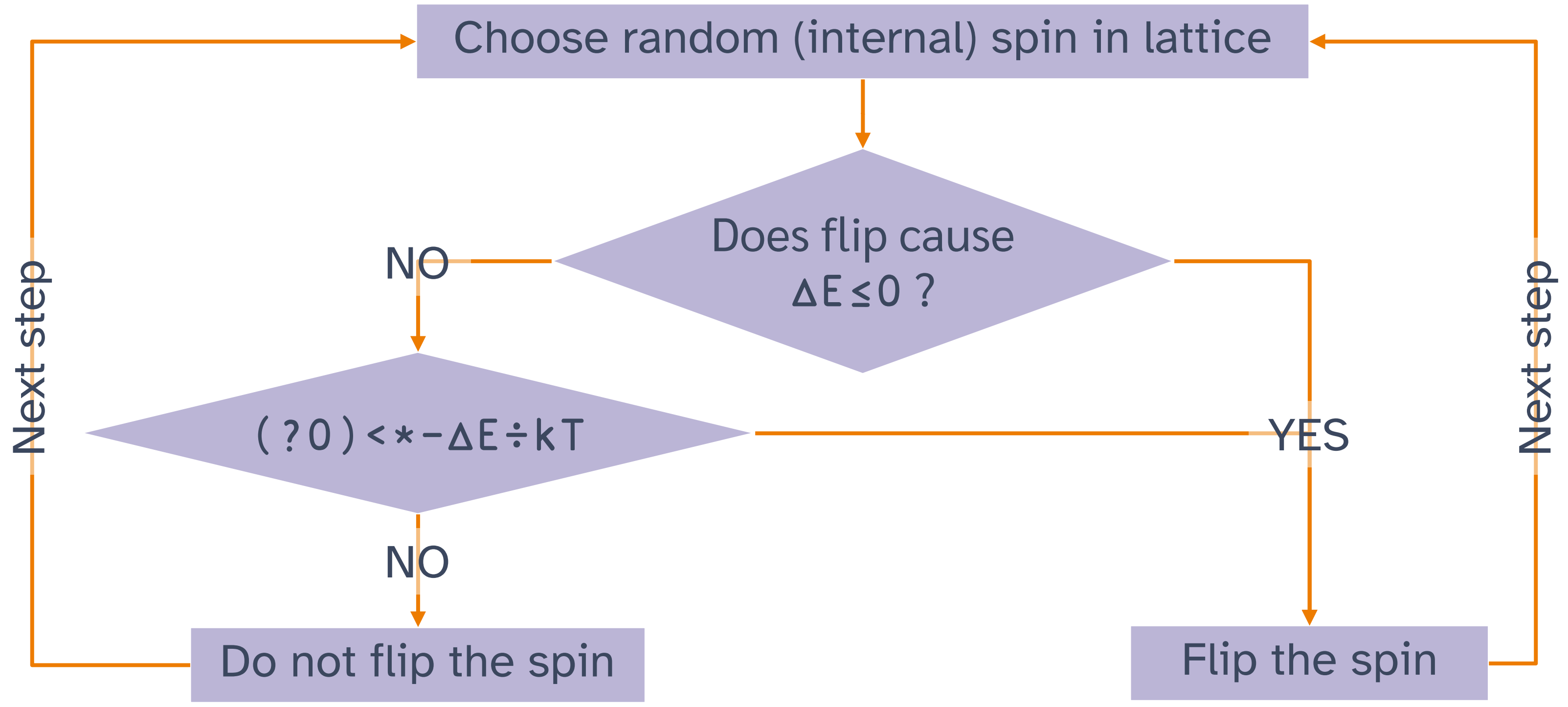
Try to write:

- Production quality code
- Sensible variable names
- Comments

## See you next week!

- Questions?

## Simulate: The Metropolis Algorithm



## Code Review

This code supposedly chooses a random spin to flip.

- Can you spot the mistake?

```
shape ← ⍴lat
random ← ?shape
random -← random=shape
random +← random=1
```

## Code Review

This code supposedly chooses a random spin to flip.

- Can you spot the mistake?

```
random ← 1+2?¯2+≢lat
```

## Code Review

This code supposedly chooses a random spin to flip, then does it or not, depending on DoFlip ΔE.

- Can you spot the mistake?

```
RandomFlip ← {-@(1+?¯2+ρω)⊢ω}
ΔE ← (TotalEnergy RandomFlip lat) - TotalEnergy lat
:If DoFlip ΔE
        lat ← RandomFlip lat
:EndIf
```

## Code Review: Bonus

This code supposedly chooses all random spins to flip, for the entire simulation, at once.

- Can you spot the mistake?

```
shape←⍴lat
all_random ← 1+shape⊥⍉?n 2⍴shape-2
```

## How to detect / prevent errors?

- Simple visualisation
- Logging
- Plotting