DYALOG

SimCorp

Workshop:
Magnets Problem – day 1

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

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

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

TotalEnergy using Stencil

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

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

TotalEnergy using Shifting

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

TotalEnergy using Shifting

$$\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} \begin{bmatrix} -1 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{matrix} -1 \\ -1 \\ 1 \\ 1 \\ -1 \end{matrix}$$

TotalEnergy using Shifting

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

TotalEnergy using Shifting

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

TotalEnergy using Shifting

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ \left[\begin{matrix} -1 \\ 1 \\ 1 \end{matrix}\right. & \begin{matrix} -1 \\ -1 \\ 1 \end{matrix} & \left.\begin{matrix} 1 \\ 1 \\ -1 \end{matrix}\right] & \begin{matrix} -1 \\ -1 \\ 1 \end{matrix} & \begin{matrix} 1 \\ -1 \\ -1 \end{matrix} \\ -1 & 1 & 1 & 1 & 1 \end{matrix}$$

TotalEnergy using Shifting

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

TotalEnergy using N-wise Reduce

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

TotalEnergy using N-wise Reduce

$$\begin{matrix} -1 & -1 & 1 & -1 & 1 \\ 1 & \times -1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 \end{matrix}$$

TotalEnergy using N-wise Reduce

$$\begin{matrix} -1 & -1 & 1 & -1 & 1 \\ 1 & \times -1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 \end{matrix}$$

TotalEnergy using N-wise Reduce

−1	−11	−11	−11
−11	−11	−11	−1
1	−11	−11	−11
−11	1	1	1
−11	−1	−1	−11

TotalEnergy using N-wise Reduce

$$
\begin{array}{cccc}
1 & -1 & -1 & -1 \\
-1 & -1 & -1 & 1 \\
1 & -1 & -1 & -1 \\
-1 & 1 & 1 & 1 \\
-1 & 1 & 1 & -1
\end{array}
$$

TotalEnergy using N-wise Reduce

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

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

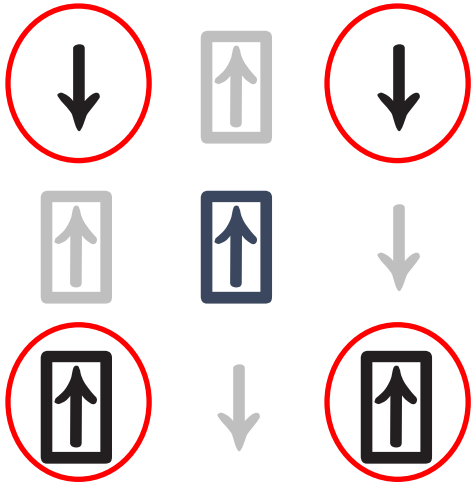
$$E = -J \sum_{ij} s_i s_j$$

External Field

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

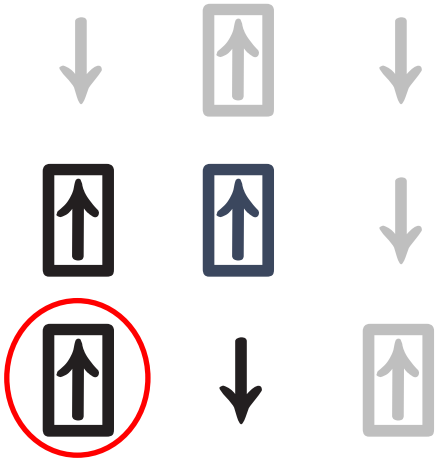
Change contribution from neighbours

- Corners also contribute



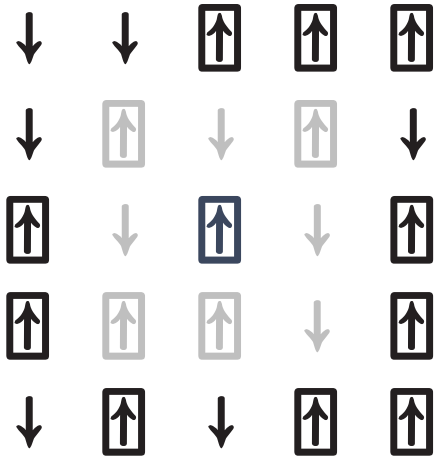
Change contribution from neighbours

- Anisotropic: southwest neighbours contribute more



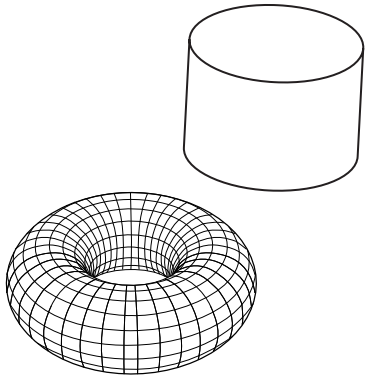
Change contribution from neighbours

- 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?

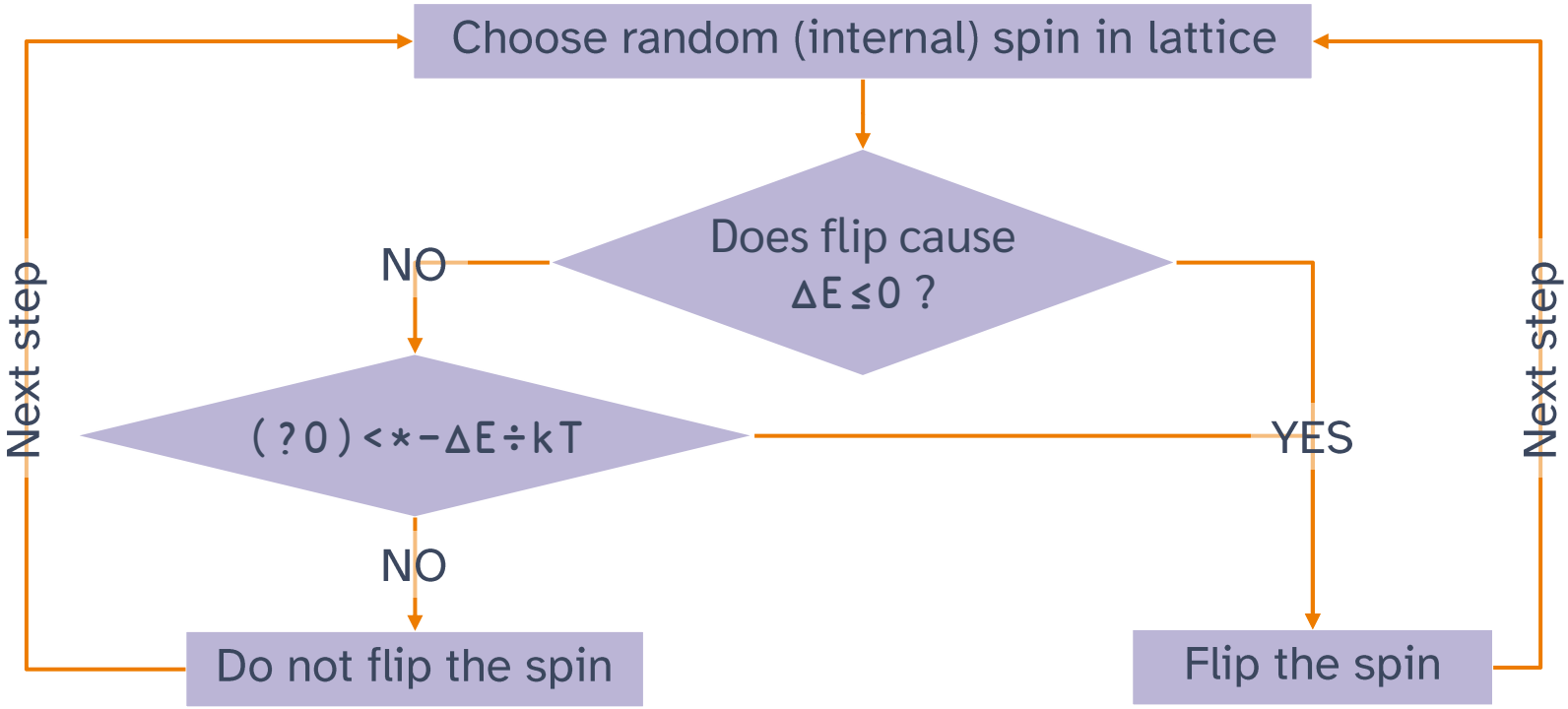
DYALOG

SimCorp

Workshop:
Magnets Problem – day 2

Adám Brudzewsky
Richard Park
Rodrigo Girão Serrão

Simulate: The Metropolis Algorithm



Code Review

This code supposedly chooses a random spin to flip.

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

- Can you spot the mistake?

How to detect / prevent errors?

- Simple visualisation
- Logging
- Plotting

Code Review

This code supposedly chooses a random spin to flip.

```
random ← 1+2?¯2+≠lat
```

- Can you spot the mistake?

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+⍴⍵)⊢⍵}
ΔE ← (TotalEnergy RandomFlip lat) - TotalEnergy lat
:If DoFlip ΔE
        lat ← RandomFlip lat
:EndIf
```

Break

Performance

Performance

]SpaceNeeded

Performance

Performance

```
]SpaceNeeded
```


```
]RunTime -c

'cmpx'□cy'dfns'
cmpx '…' '…'
```


Performance

- Run-time is usually more important than memory usage
- Explore various algorithms
- Try differently scaled input sizes
- Compare parts of the solution to construct the best combo
  example: youtu.be/El0_RB4TTPA

Performant `Simulate`

- Control iterations with:
  - `:While counter`
  - `:For ... :In ⍳max_iter`
  - `{...}⍣max_iter`

Performant `Simulate`

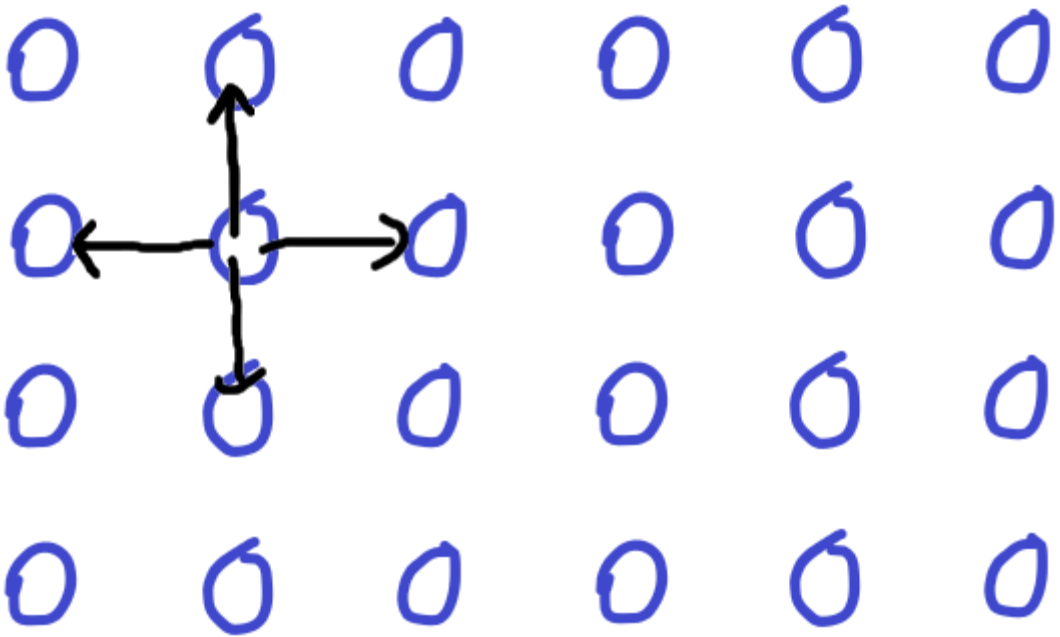
- Determine new spin with:
  - Mathematical computation `(¯1*...)`
  - Data-driven conditional `(...⊃¯1 1)`

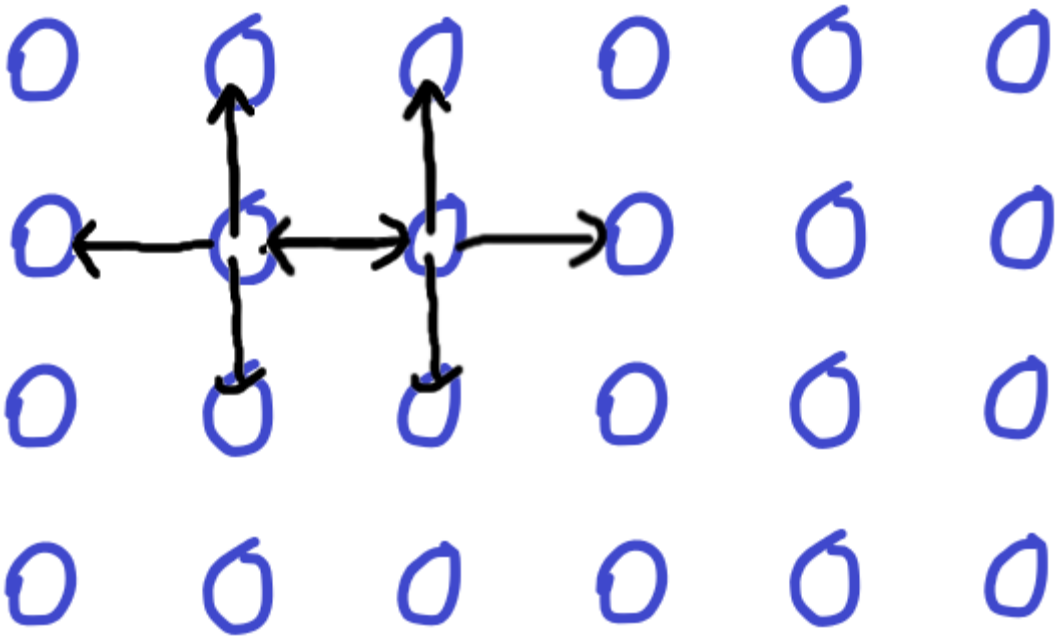
Performant `Simulate`

- Assign new spin with:
  - `new_spin@position`
  - `grid[position] ← new_spin`

Performant `Simulate`

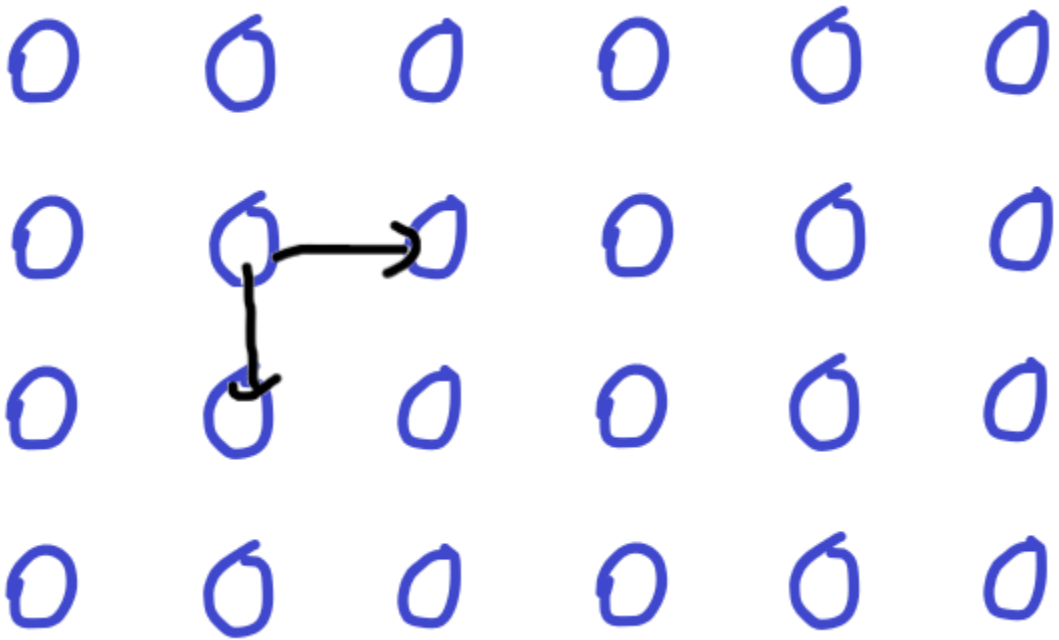
- Compute spin contributions:
  - NESW neighbours, then halve

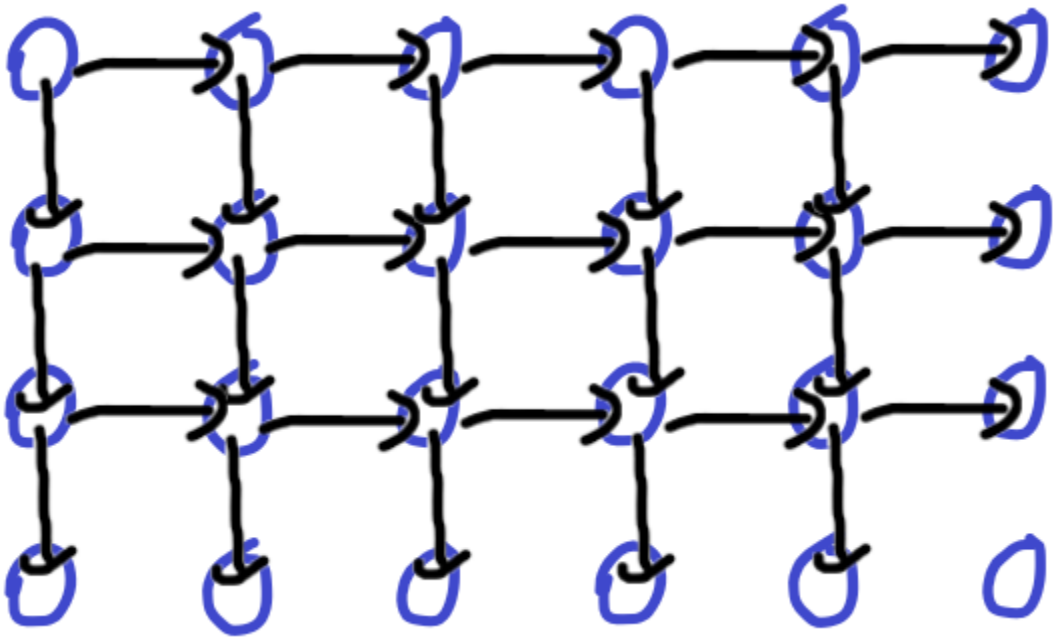




Performant `Simulate`

- Compute spin contributions:
  - NSEW neighbours, then halve
  - 2 neighbours, then tile



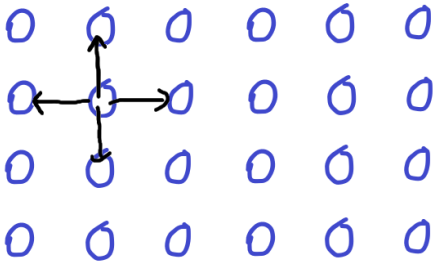


Performant `Simulate`

- Cache current total energy

Performant `Simulate`

- Change in energy:
  - Difference in total
  - Difference in the neighbourhood



DYALOG

SimCorp

We're here for you!

General support	support@dyalog.com
Forums	forums.dyalog.com
Chat room	apl.chat
Adám Brudzewsky	adam@dyalog.com
Richard Park	rpark@dyalog.com
Rodrigo Girão Serrão	rodrigo@dyalog.com