Introduction to lithospheric geodynamic modelling

The marker-in-cell technique

David Whipp and Lars Kaislaniemi
Department of Geosciences and Geography, Univ. Helsinki

Goals of this lecture

- Introduce the **marker-in-cell technique** as a partial solution to numerical diffusion

- Provide some examples of how various values are calculated for the **markers**

Numerical diffusion in advection problems

- As we saw yesterday, the advection equation seems simple, but suffers from problems with numerical diffusion
- Consider the advection of a rock mass with density ρ

$$\frac{\partial \rho}{\partial t} = -v_x \left(\frac{\partial \rho}{\partial x} \right)$$

- The upwind finite-difference solution to this advection equation is

$$\rho_n^i = \rho_n^{i-1} - v_x \Delta t \frac{\rho_n^{i-1} - \rho_{n-1}^{i-1}}{\Delta x}$$

Numerical diffusion in advection problems

- As we saw yesterday, the advection equation seems simple, but suffers from problems with numerical diffusion
- Consider the advection of a rock mass with density ρ

$$\frac{\partial \rho}{\partial t} = -v_x \left(\frac{\partial \rho}{\partial x} \right)$$

- The upwind finite-difference solution to this advection equation is

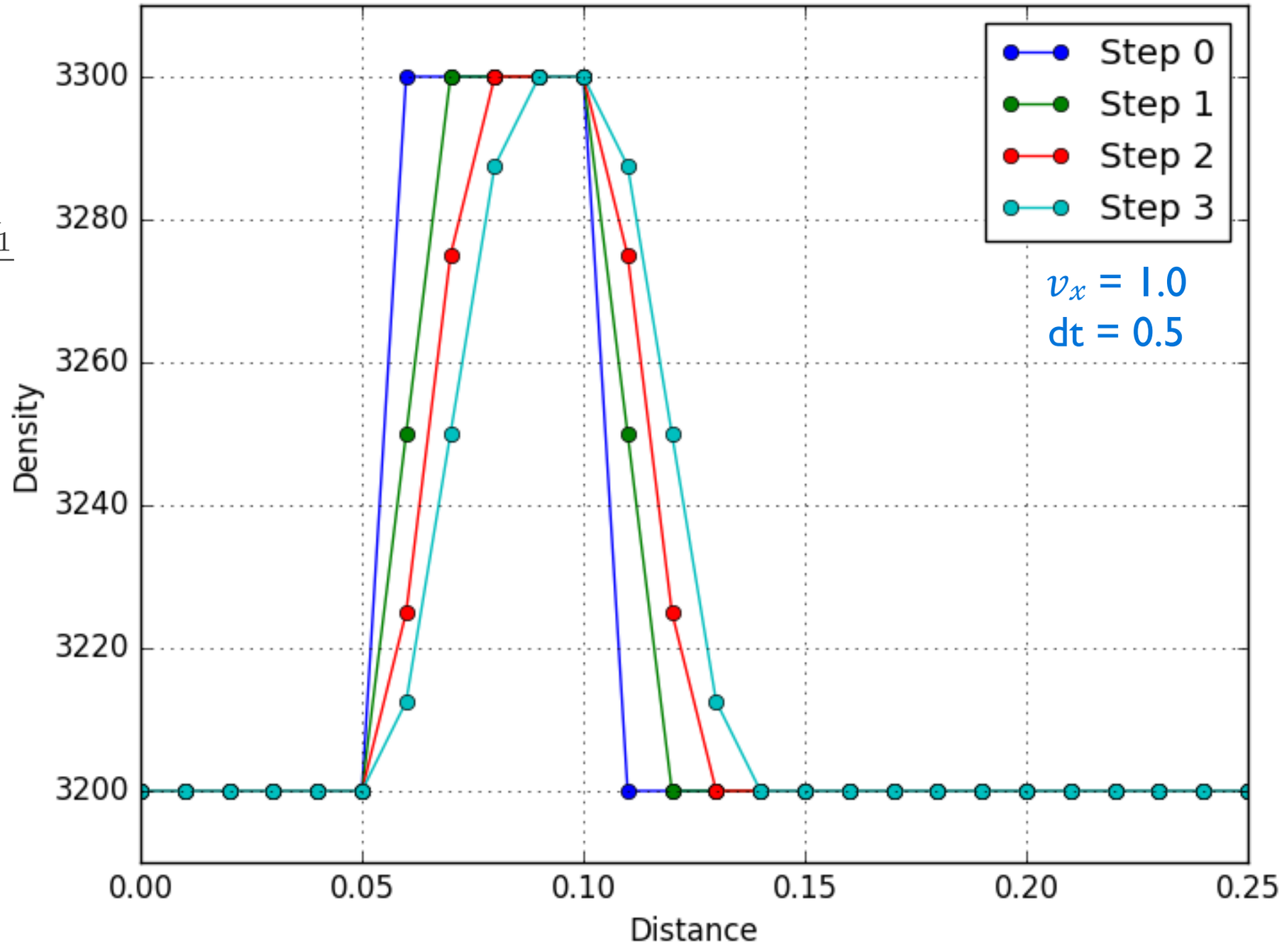
$$\rho_n^i = \rho_n^{i-1} - v_x \Delta t \frac{\rho_n^{i-1} - \rho_{n-1}^{i-1}}{\Delta x}$$

Numerical diffusion in advection problems

Upwind FD

$$\rho_n^i = \rho_n^{i-1} - v_x \Delta t \frac{\rho_n^{i-1} - \rho_{n-1}^{i-1}}{\Delta x}$$

The trouble with numerical diffusion is immediately visible



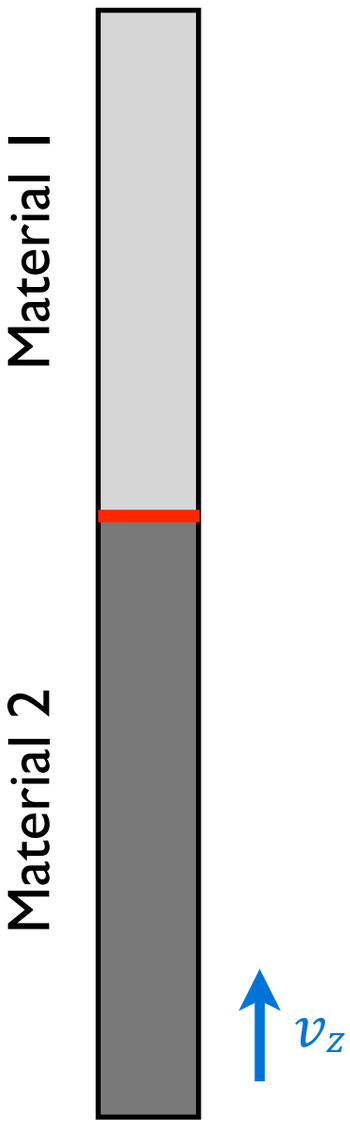
Numerical diffusion in advection problems

- In some cases we can handle a bit of diffusion, but in others diffusion is a problem
  - A bit of diffusion of heat may not be a major problem, but diffusion of rock types can be a serious problem
  - For example, it would not be acceptable for defined weak layer in a model, such as a evaporites in the stratigraphy of rocks in a fold-and-thrust belt, to diffuse into the surrounding, stronger bedrock
  - In this case, the weak layer that may be the principal detachment surface disappears and the fold-and-thrust belt may behave entirely differently

Reference frames: Eulerian versus Lagrangian

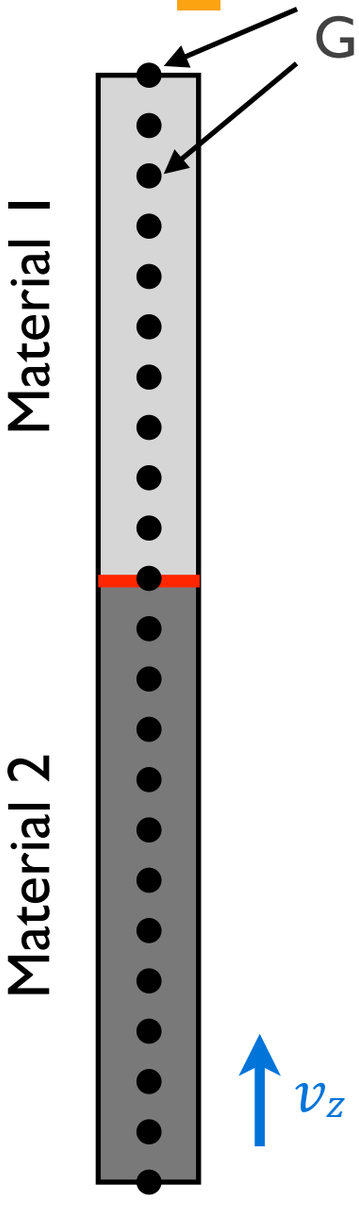
- We haven't said anything about it, but up to this point we have been performing our calculations on an Eulerian grid of points
  - **Eulerian points** are fixed in space
    - For example, we have calculated changes in temperature, but the locations of our grid points x_i have not changed
- **Lagrangian points** are free to move with time, following the velocity at which material is being advected, for example
- You can think of these as two different reference frames:
  - The Eulerian reference frame is external, the Lagrangian reference frame is internal

A simple example



- Consider a 1D heat transfer model simulating exhumation of rock as a result of erosion
  - The model has two materials and vertical advection of rock

A simple example

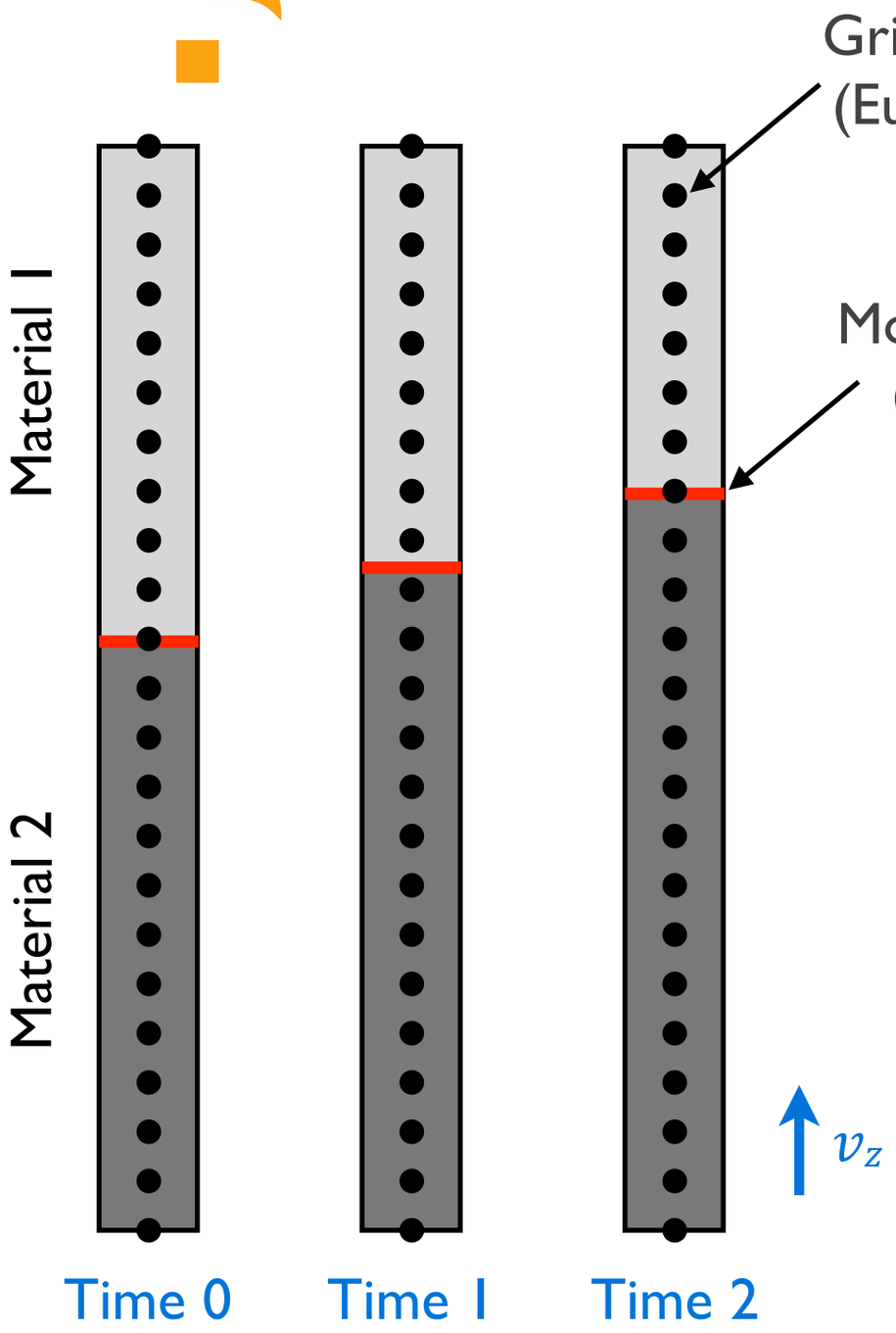


- Previously, we were able to calculate temperatures including the thermal effects of advection

$$T_n^i = \left(\left(\alpha_n \frac{T_{n+1}^{i-1} - T_n^{i-1}}{\Delta x^2} - \alpha_{n-1} \frac{T_n^{i-1} - T_{n-1}^{i-1}}{\Delta x^2} + H_n \right) \Big/ \rho_n C_{p,n} - v_x \frac{T_{n+1}^{i-1} - T_{n-1}^{i-1}}{2\Delta x} \right) \Delta t + T_n^{i-1}$$

- This is OK, but as we know, erosion of rock and the resulting transport toward the surface should also advected the rocks (and their properties) toward the surface, not just temperature

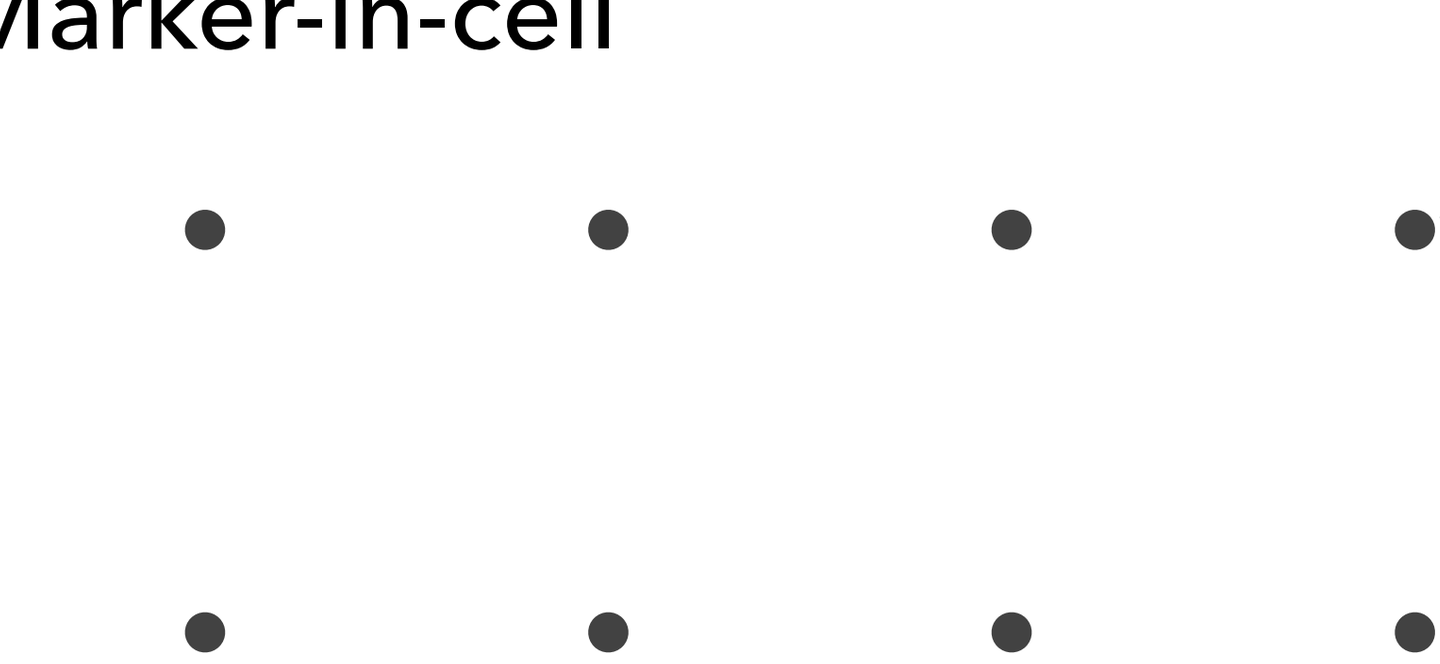
A simple example

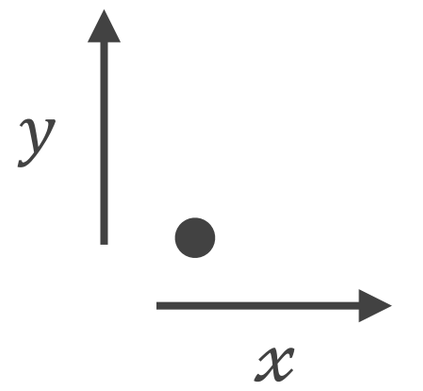


- One solution to this problem would be to track and store the location of the interface between the two rock types and use that to determine the properties of the grid points

- This is a simple way to track material properties in 1D, but tracking this kind of interface becomes more challenging in 2D or 3D, so alternative techniques are used

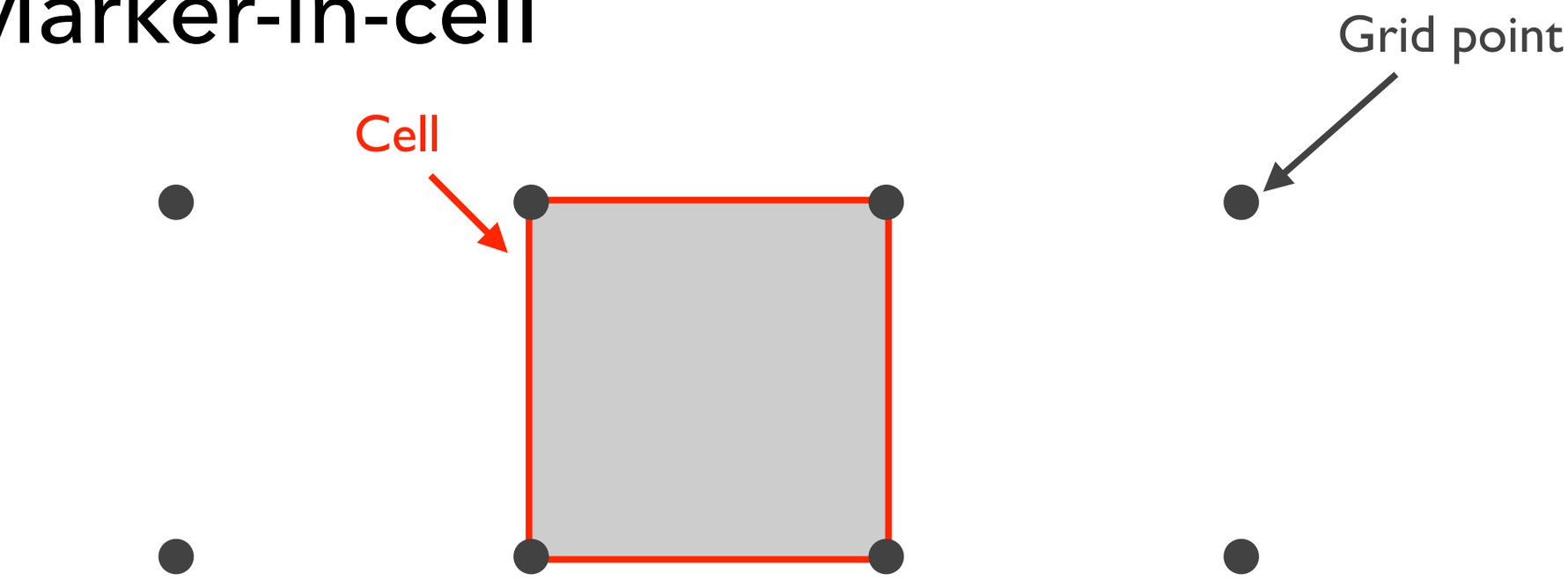
Marker-in-cell

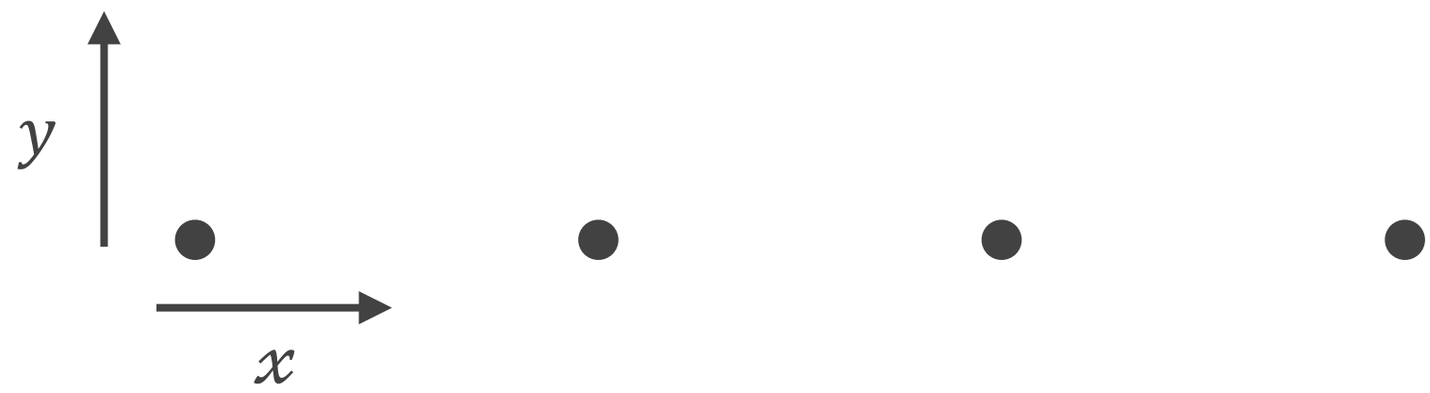




- For 2D and 3D geodynamic models it is common to use **markers** (or particles or tracers) for tracking things like material properties

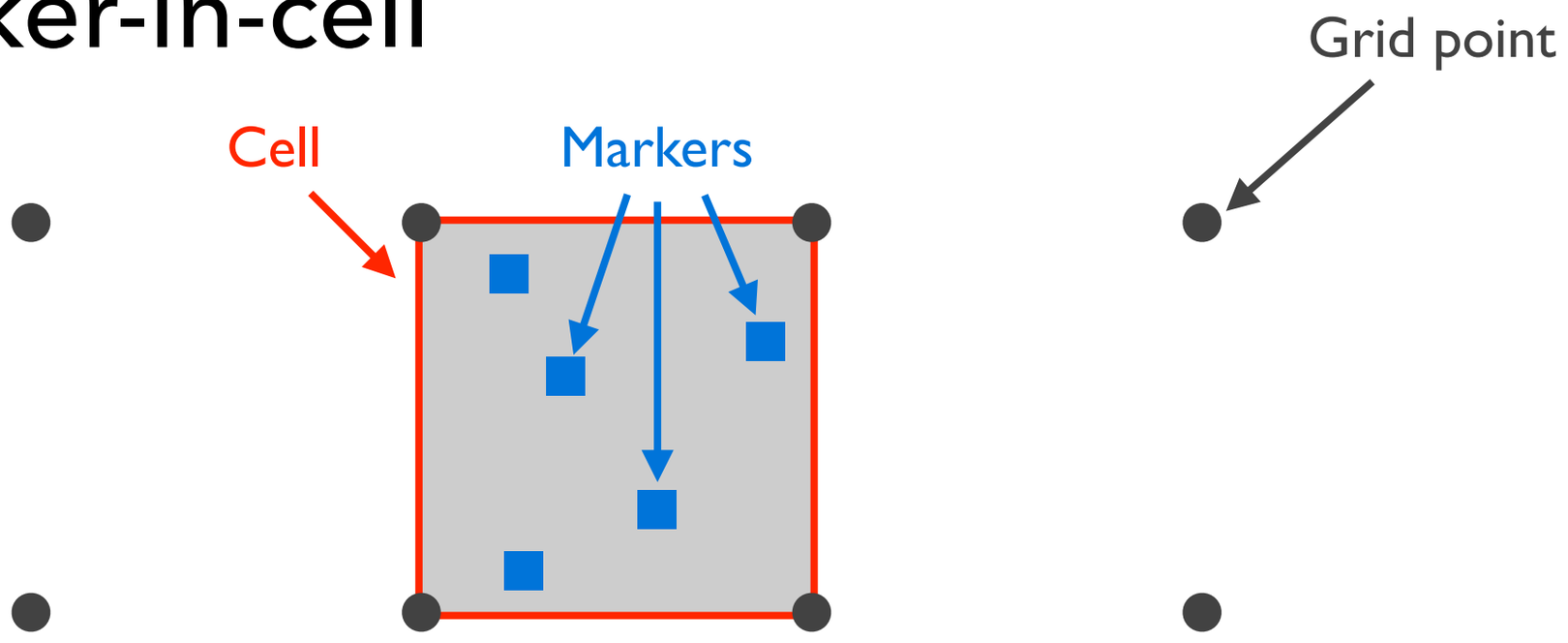
Marker-in-cell

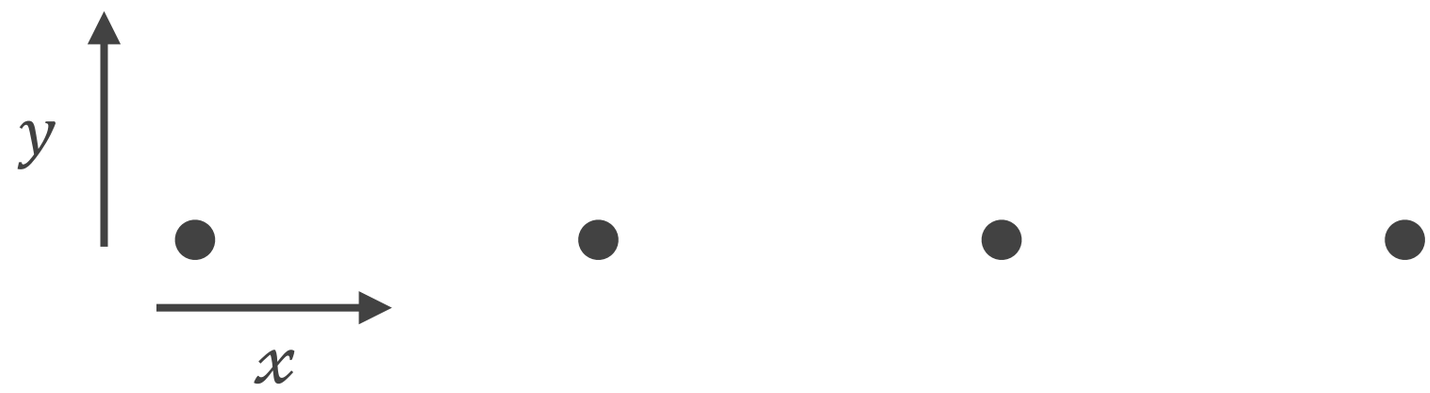




- For 2D and 3D geodynamic models it is common to use **markers** (or particles or tracers) for tracking things like material properties

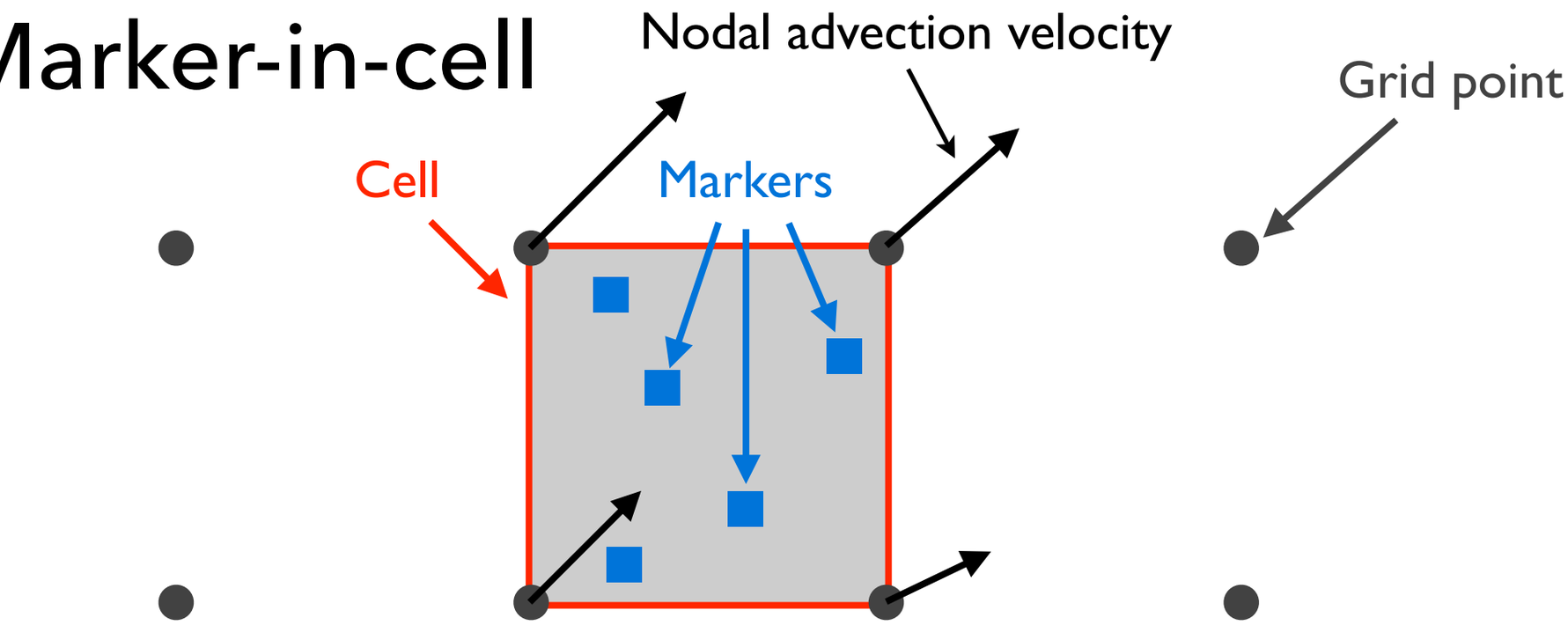
Marker-in-cell

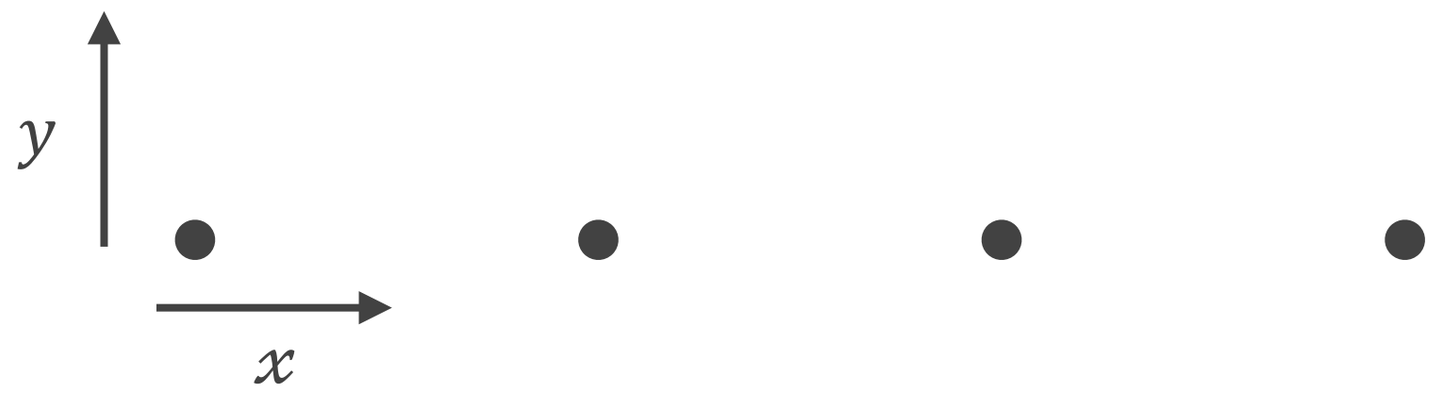




- For 2D and 3D geodynamic models it is common to use **markers** (or particles or tracers) for tracking things like material properties

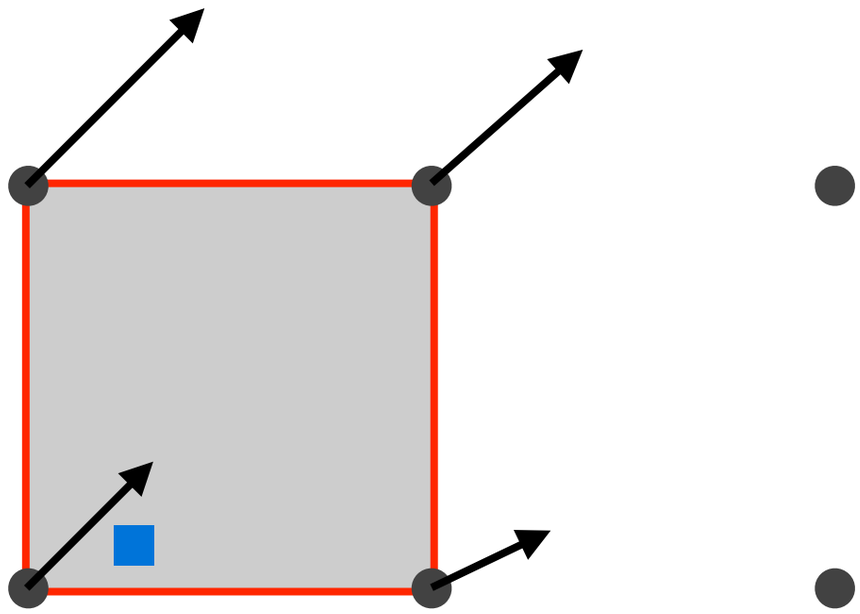
Marker-in-cell

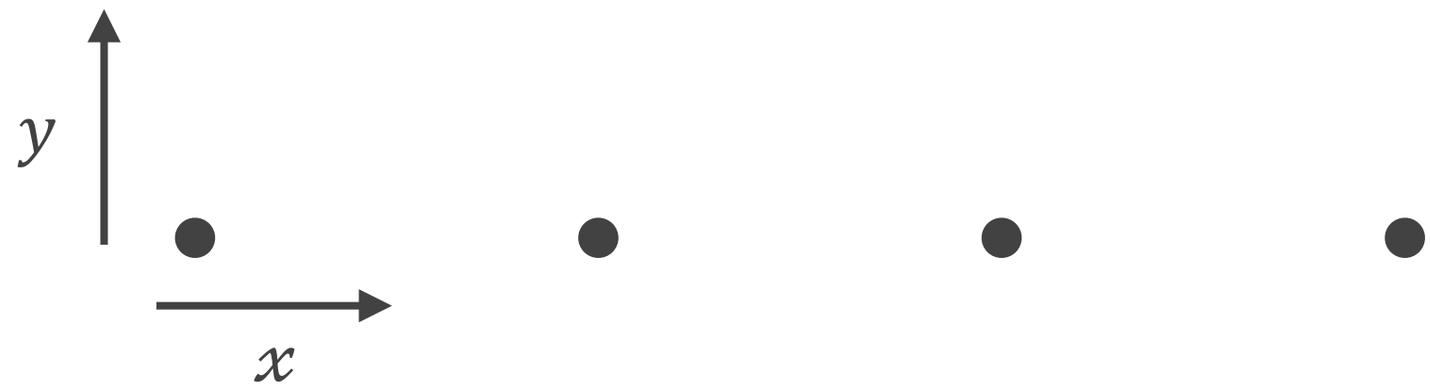




- For 2D and 3D geodynamic models it is common to use **markers** (or particles or tracers) for tracking things like material properties

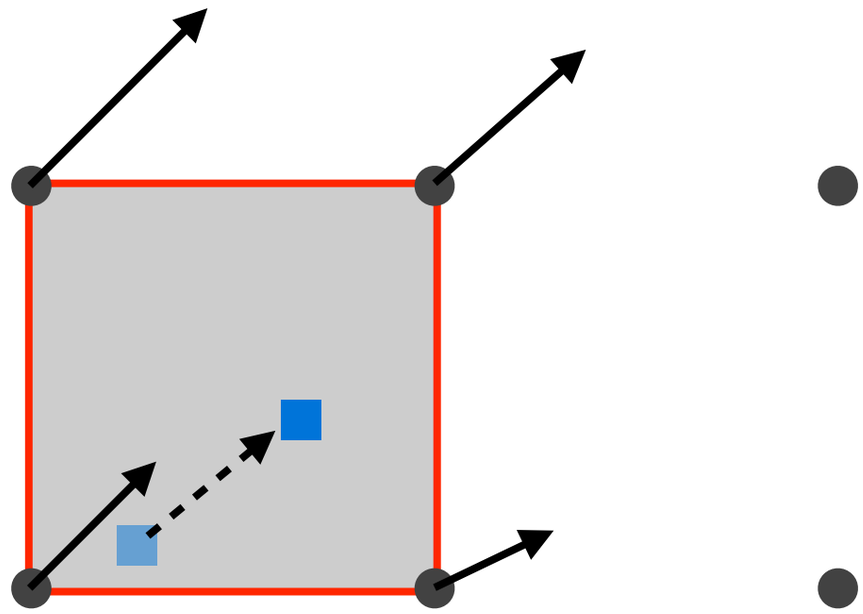
Marker advection

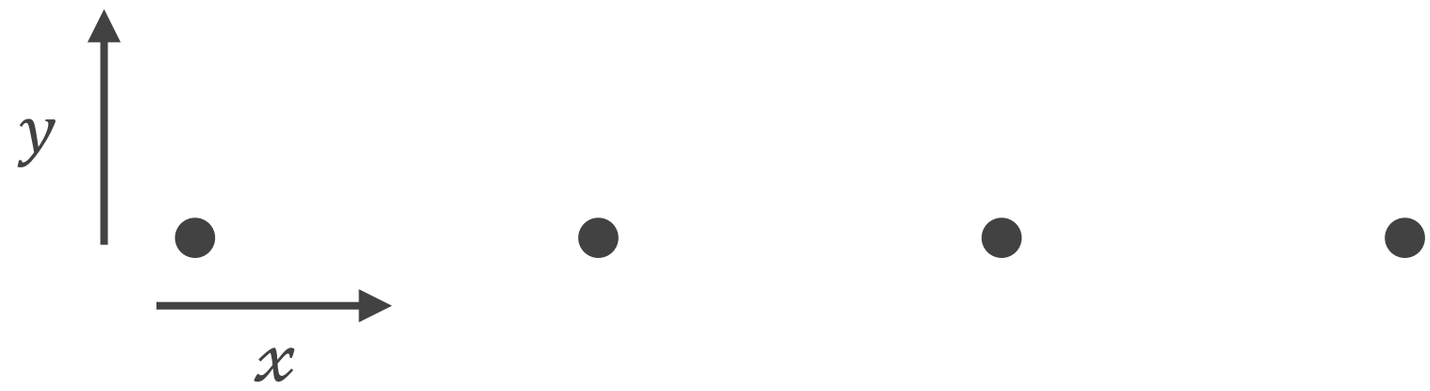




- Let's now consider a single marker in the model
  - Its position will change with time based on the velocities of material at the four surrounding nodes (the cell)

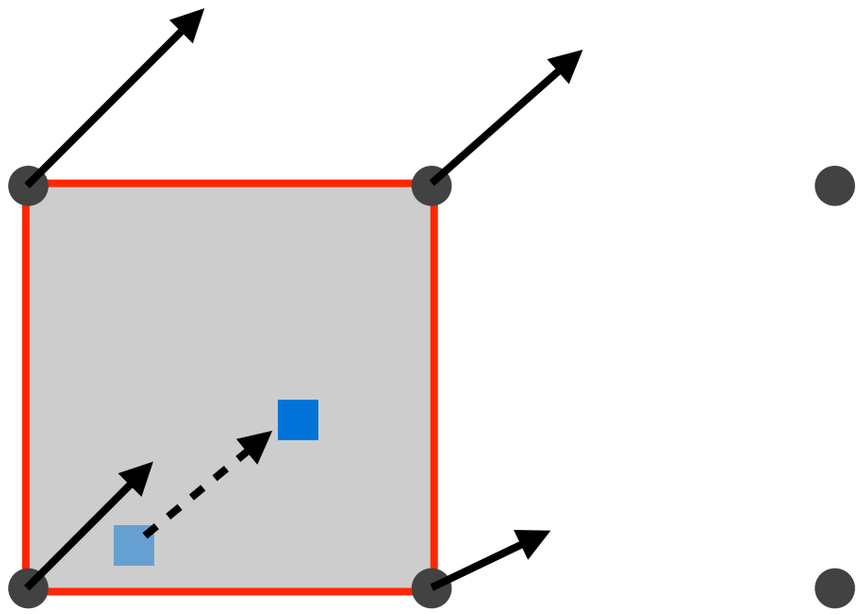
Marker advection

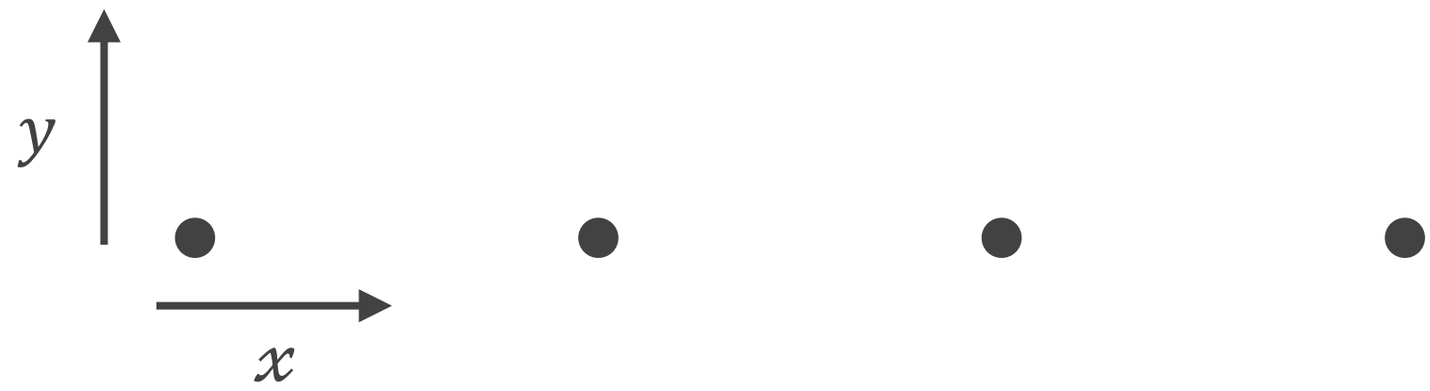




- The particle velocity is calculated using a bilinear interpolation of the nodal velocities
  - Obviously, this avoids numerical diffusion of the marker data

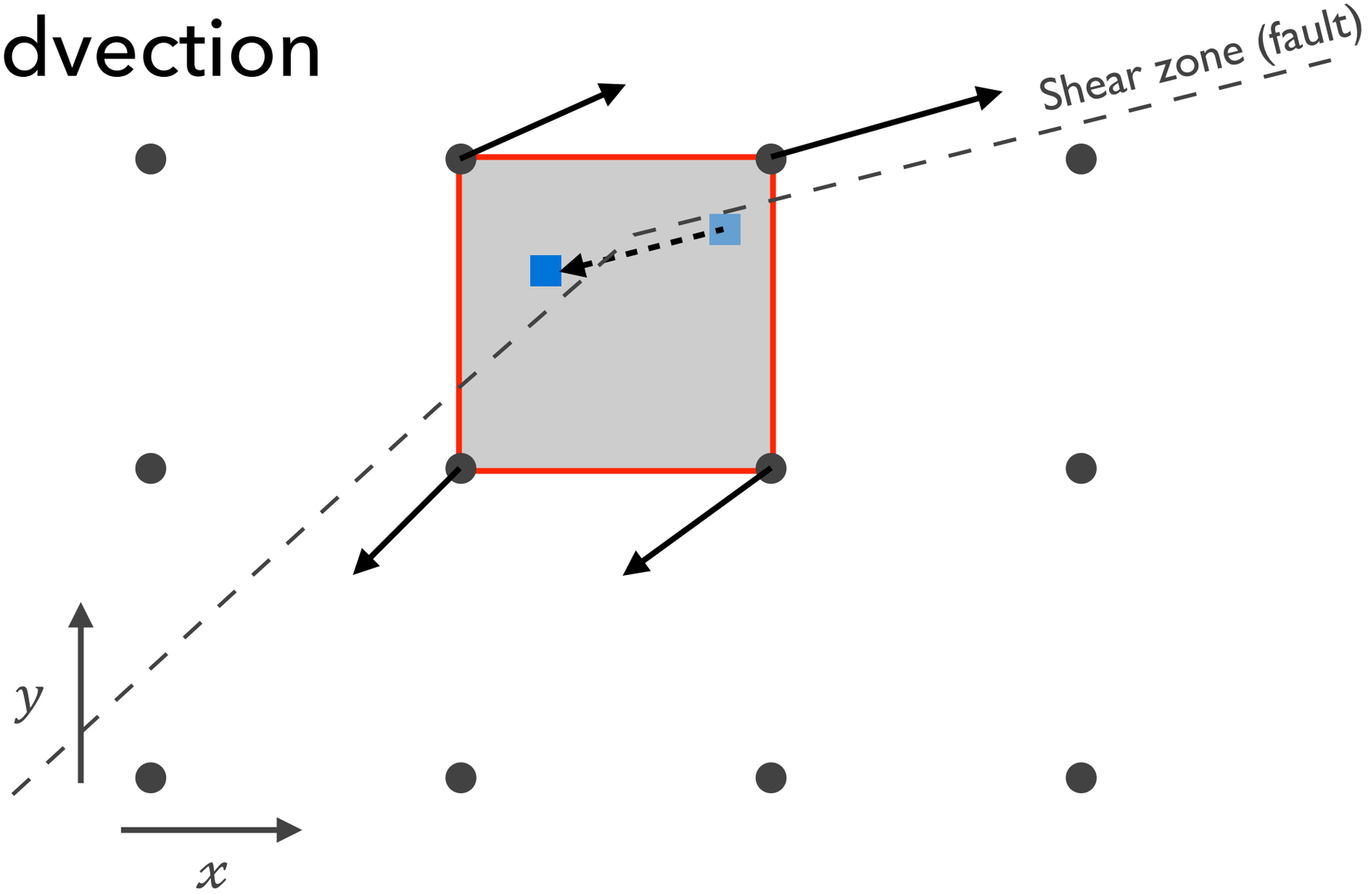
Marker advection





- It is important to note, though, that the velocity varies across a given cell, and error can be introduced by advecting a particle for an entire time step using the velocity at its origin

Marker advection



- Consider a particle in the footwall of a shear zone or fault
  - Here, because the initial velocity is used for the entire time step, material is transferred from the footwall to the hanging wall

Marker advection schemes

- The simple case from the previous slide is given below

$$x_A^i = x_A^{i-1} + v_{xA}\Delta t$$
$$y_A^i = y_A^{i-1} + v_{yA}\Delta t$$

where $x^i{}_A$ is the future x-coordinate of particle $x^{i\text{-}1}{}_A$ and v_x is the velocity in the x direction, and similarly for y

- This scheme is first-order accurate, and requires small time steps if there are large differences in velocity in the model

- An alternative is to consider a velocity that is not based solely on the velocity at the starting position of a marker, but one that considers more than a single point

Runge-Kutta advection

- The **Runge-Kutta advection** scheme uses 2-4 points to determine the velocity at which particles should be advected
- The *second-order* Runge-Kutta scheme is below

$$x_A^i = x_A^{i-1} + v_x^{\text{eff}} \Delta t$$

$$y_A^i = y_A^{i-1} + v_y^{\text{eff}} \Delta t$$

where

$$v_x^{\text{eff}} = v_{xB} \qquad x_B = x_A^i + v_{xA} \frac{\Delta t}{2}$$

$$v_y^{\text{eff}} = v_{yB} \qquad y_B = x_A^i + v_{yA} \frac{\Delta t}{2}$$

Properties on the markers

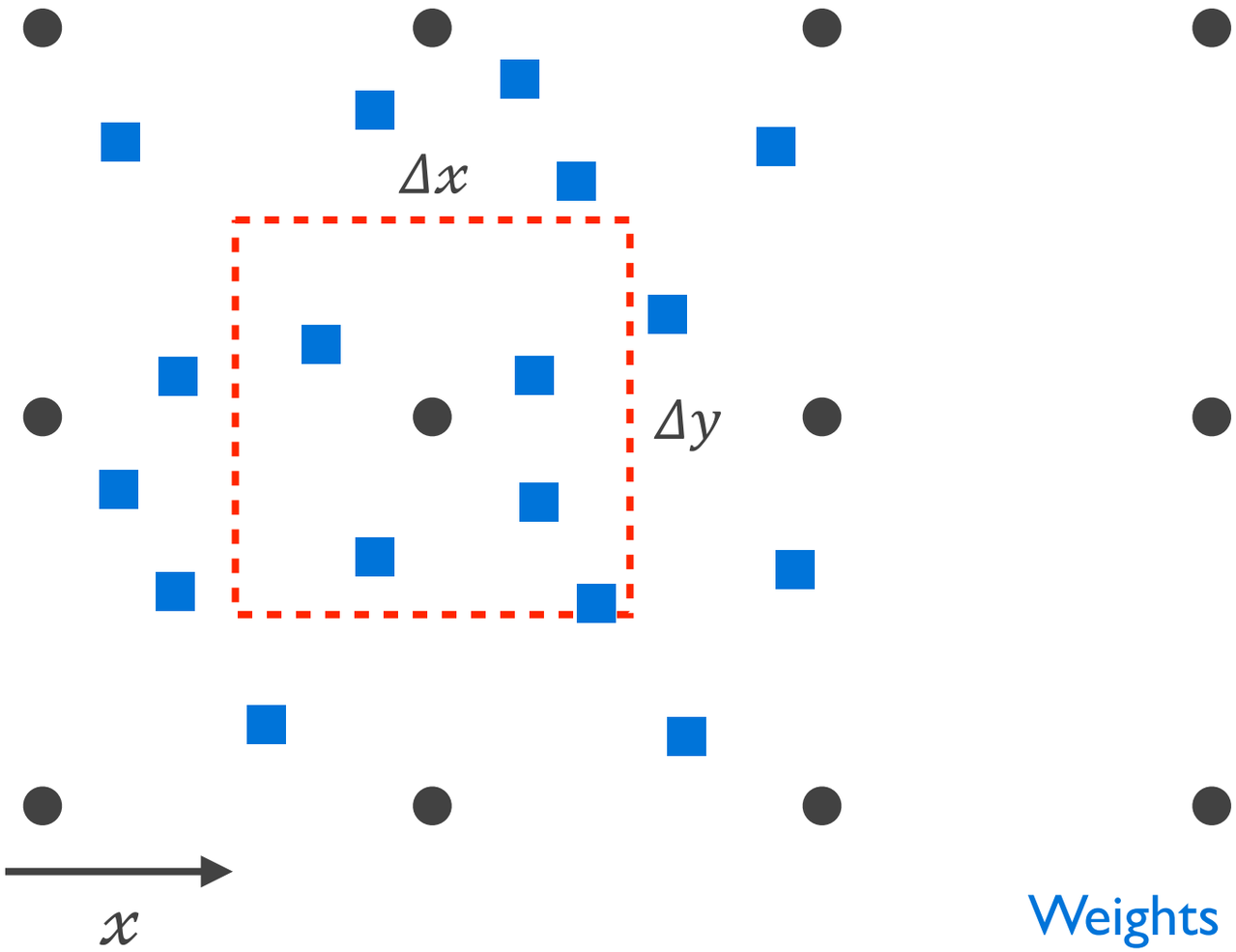
- It is quite common to use the markers to carry a variety of values that should not diffuse
  - Material properties (density, conductivity, viscosity, friction angle, etc.)
  - Fields with a history, such as total strain

- However, because the markers largely avoid typical numerical diffusion issues, they can be used to store nearly all of the information calculated in the numerical model (temperature, strain rates, etc.)

Retrieving information from the markers

- Now, we have established the benefits of the marker-in-cell method for avoiding numerical diffusion, but it is not as if it an method that is free of error
- In order to transfer information between the markers and the grid points, we need some kind of function
- A **weighted difference averaging** approach is one simple option

Retrieving information from the markers



NOTE: i and j are now referring to grid points, not time

Nodal parameter

$$P_{i,j} = \frac{\sum_m P_m w_{m(i,j)}}{\sum_m w_{m(i,j)}}$$

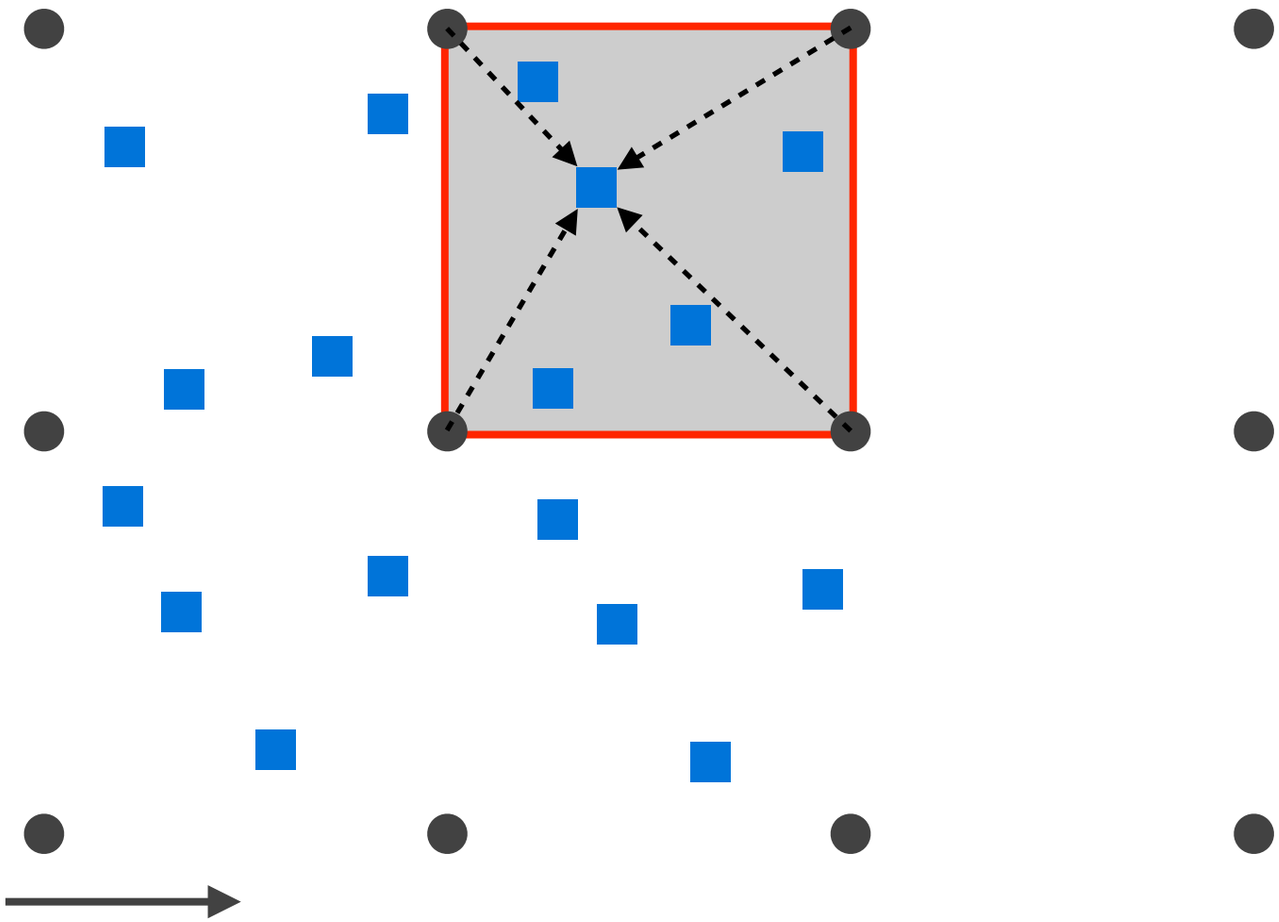
- $w_{m(n)}$ is the weight of the m-th marker for grid point (i,j)

Weights

$$w_{m(i,j)} = \left(1 - \frac{\Delta x_m}{\Delta x}\right) \times \left(1 - \frac{\Delta y_m}{\Delta y}\right)$$

- $w_{m(n)}$ is the weight of the m-th marker for grid point (i,j)

Giving information to the markers



NOTE: i and j are now referring to grid points, not time

$$P_m = P_{i,j}\left(1-\frac{\Delta x_m}{\Delta x}\right)\left(1-\frac{\Delta y_m}{\Delta y}\right) + P_{i,j+1}\frac{\Delta x_m}{\Delta x}\left(1-\frac{\Delta y_m}{\Delta y}\right)$$
$$+ P_{i+1,j}\left(1-\frac{\Delta x_m}{\Delta x}\right)\frac{\Delta y_m}{\Delta y} + P_{i+1,j+1}\frac{\Delta x_m}{\Delta x}\frac{\Delta y_m}{\Delta y}$$

Interpolation of values of parameter P from grid points i,j to marker m

Some words of caution

- Even when using the marker-in-cell technique there are some important considerations that can introduce error
  - First, there is a threshold number of markers that must exist in a cell in order to provide a reasonable resolution of the field data
  - A minimum of perhaps 5 markers would be OK in 2D, perhaps 9 in 3D
  - A larger number is better, but calculations with a large "cloud" of markers can get heavy

Some words of caution

- When the number of markers falls below the minimum needed, many models (including the one you will use) allow markers to be injected into the cells
  - How this is best done, and the how values are transferred to the new markers is not completely obvious
- This is a common issue for models where material is flowing into the model from the side and the markers are continually advected away from the side of the model
  - If you do not initialise the values on the markers carefully you can get unexpected results

Some words of caution

- Other concerns?

Our use of the marker-in-cell method

- In our code for 2D Stokes flow, you will not need to completely implement the marker-in-cell method, but define the positions of the markers at the start of the model
- This includes deciding how many markers should be placed in each cell
  - Lars recommends something like 40

Summary

- We have seen that the marker-in-cell technique, which uses Lagrangian markers, is a useful way to avoid numerical diffusion
- The Lagrangian markers are a fairly intuitive concept, but there can be some challenges in using markers, including how material is advected, how markers are injected, and how marker values are initialised

References

Gerya, T. (2010). *Introduction to numerical geodynamic modelling*. Cambridge University Press.