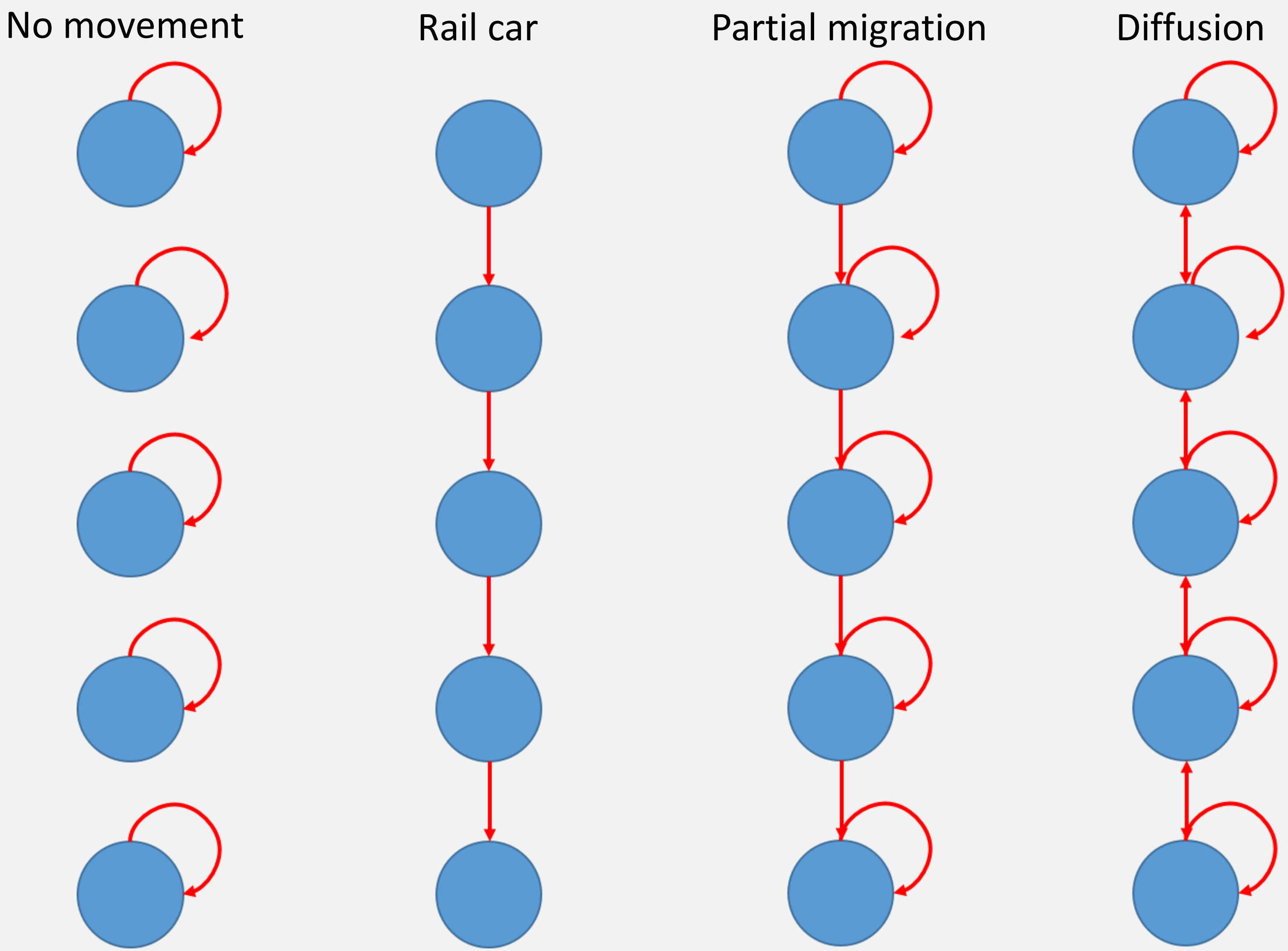
Lecture 9: Models with movement

May 31, 2016

Different forms of movement

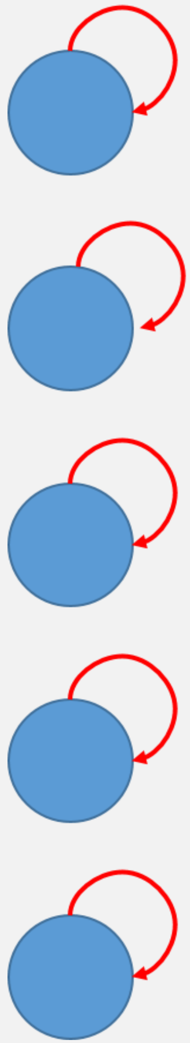


Different forms of movement

$$\frac{d}{dt}\mathbf{b} = \mathbf{M}\mathbf{b}$$

No movement

$$\mathbf{M} = \begin{bmatrix} \mathbf{1} & & & & \\ & \mathbf{1} & & & \\ & & \mathbf{1} & & \\ & & & \mathbf{1} & \\ & & & & \mathbf{1} \end{bmatrix}$$

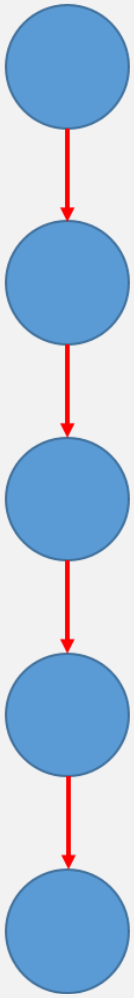


Different forms of movement

$$\frac{d}{dt}\mathbf{b} = \mathbf{Mb}$$

$$\mathbf{M} = \begin{bmatrix} & & & & \\ 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & 1 \end{bmatrix}$$

Rail car



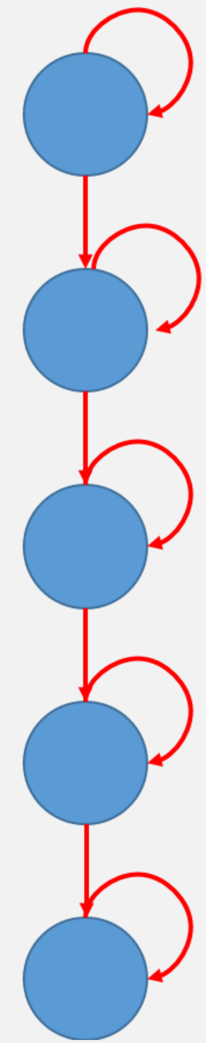
Different forms of movement

$$\frac{d}{dt}\mathbf{b} = \mathbf{M}\mathbf{b}$$

M =

1-p				
p	**1-p**			
	p	**1-p**		
		p	**1-p**	
			p	**1**

Partial migration



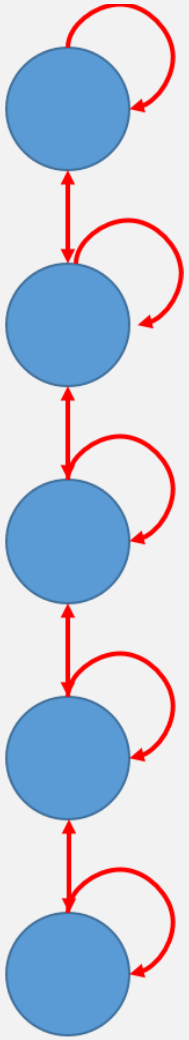
Different forms of movement

$$\frac{d}{dt}\mathbf{b} = \mathbf{Mb}$$

M =

1-p	**p**			
p	**1-2p**	**p**		
	p	**1-2p**	**p**	
		p	**1-2p**	**p**
			p	**1-p**

Diffusion



Review: Solving differential equations

- Say you know a rate:

$$\frac{d}{dt} b_t = \alpha b_t$$

- How do you solve for change after some time Δt?

$$b_{t+\Delta t} = b_t \times \exp(\alpha \Delta t)$$

- This is the definition of the exponential function

Review: simultaneous differential equations

- Say you know a rate:

$$\delta \mathbf{b}_t = \mathbf{A}\mathbf{b}_t \delta t$$

- How do you solve for change after some time Δt?
- Well, if A is diagonal:

$$\mathbf{A} = \begin{bmatrix} \alpha_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \alpha_n \end{bmatrix}$$

- Then its easy!

$$\mathbf{b}_{t+\Delta t} = \mathbf{B}\mathbf{b}_t$$

$$\mathbf{B} = \begin{bmatrix} \exp(\alpha_1 \Delta t) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \exp(\alpha_n \Delta t) \end{bmatrix}$$

Review: simultaneous differential equations

- Say you know a rate:

$$\delta \mathbf{b}_t = \mathbf{A}\mathbf{b}_t \delta t$$

- Well, if A is diagonalizable:

$$\mathbf{A} = \mathbf{U}\mathbf{A}^*\mathbf{U}^{-1}$$

where $\mathbf{A}^*$ is diagonal, then:

$$\mathbf{b}_{t+\Delta t} = \mathbf{U}\mathbf{B}\mathbf{U}^{-1}\mathbf{b}_t$$

$$\mathbf{B} = \begin{bmatrix} \exp(\alpha_1^* \Delta t) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \exp(\alpha_n^* \Delta t) \end{bmatrix}$$

Review: simultaneous differential equations

- Say you know a rate:

$$\delta \mathbf{b}_t = \mathbf{A}\mathbf{b}_t \delta t$$

- The a general solution is:

$$\mathbf{b}_{t+\Delta t} = \mathbf{B}\mathbf{b}_t$$
$$\mathbf{B} = \mathrm{expm}(\mathbf{A})$$

where $\mathrm{expm}()$ is the matrix exponential

Advection-diffusion:

- Advection – directed movement of individuals in a specific direction
- Diffusion – random movement of individuals away from their current location

- Possible to include if we can define a differential equation

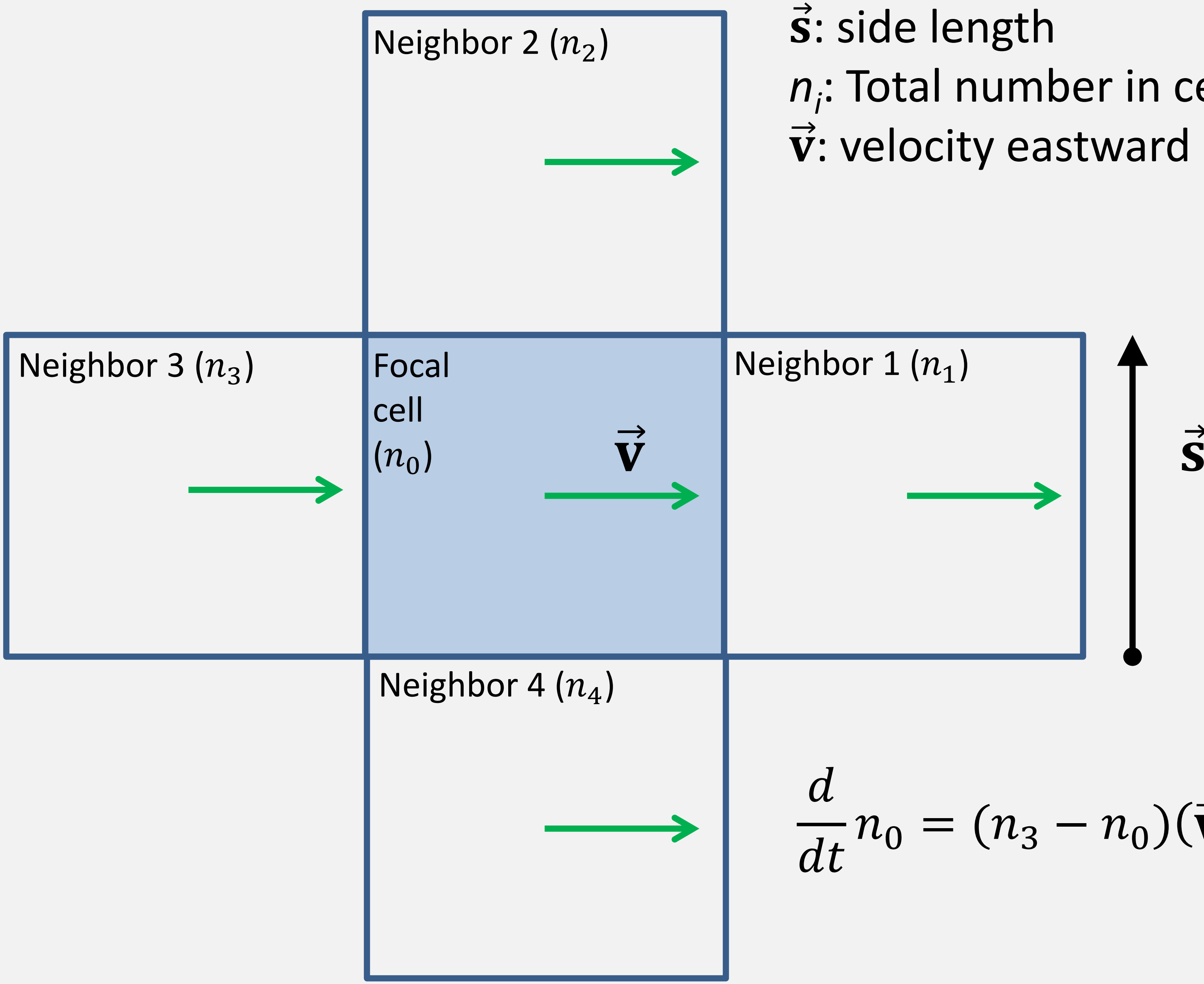
$$\delta \mathbf{b}_t = \mathbf{A}\mathbf{b}_t \delta t$$

  - Where $\mathbf{A}$ is the net effect of advection and diffusion

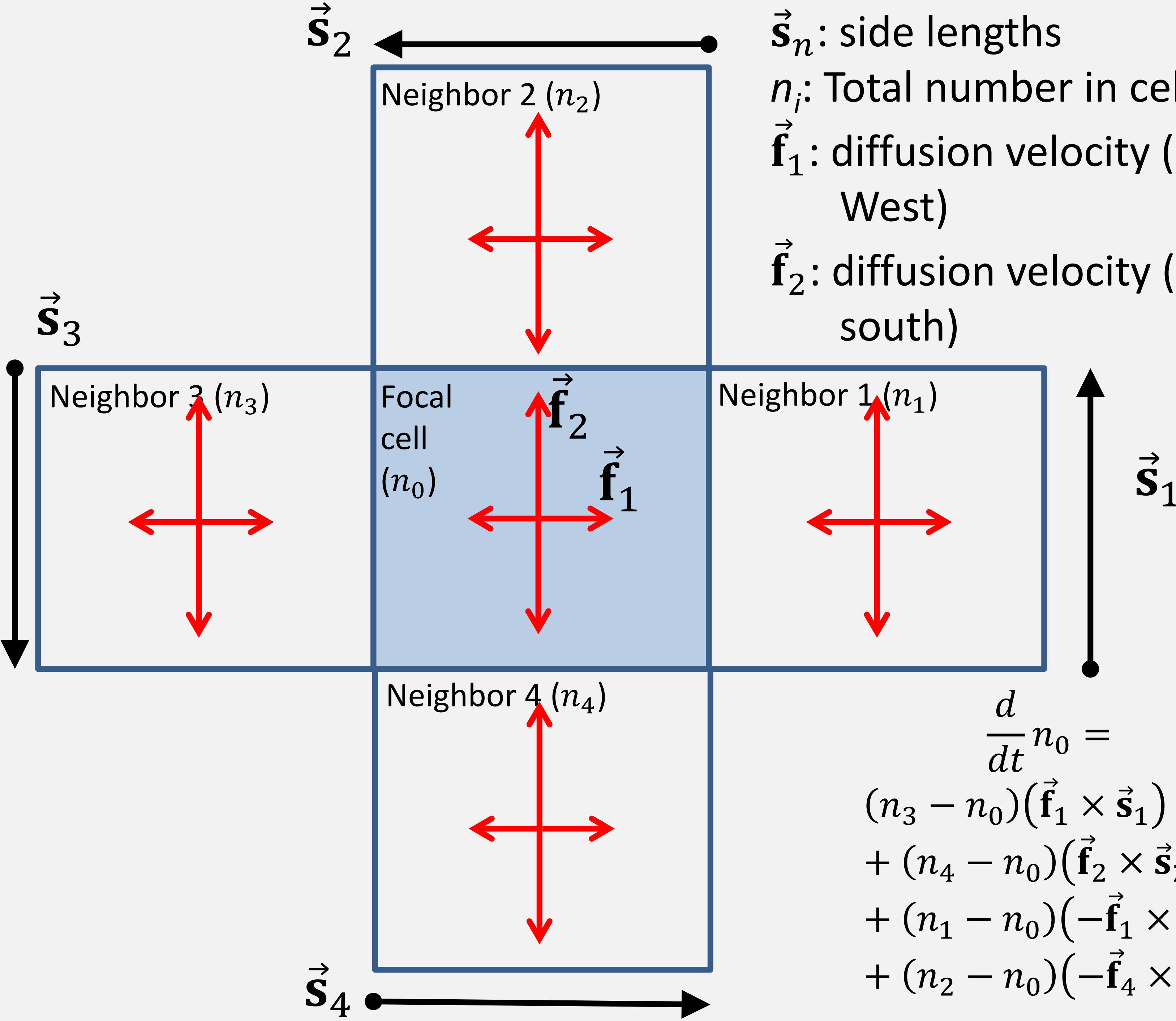
$\vec{\mathbf{s}}$: side length

n_i: Total number in cell i

$\vec{\mathbf{v}}$: velocity eastward



$$\frac{d}{dt} n_0 = (n_3 - n_0)(\vec{\mathbf{v}} \times \vec{\mathbf{s}})$$



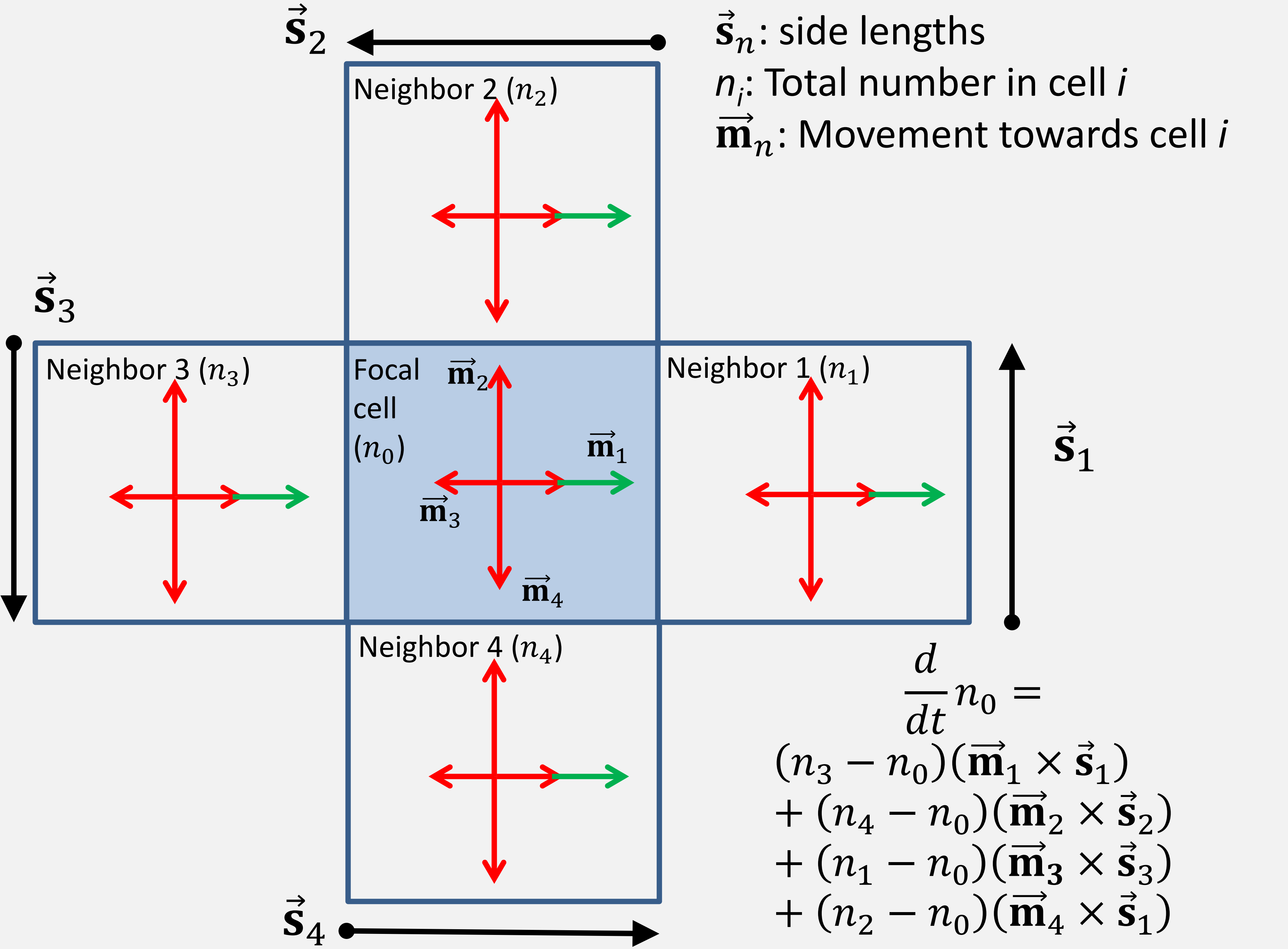
$\vec{\mathbf{s}}_n$: side lengths

n_i: Total number in cell i

$\vec{\mathbf{f}}_1$: diffusion velocity (East-West)

$\vec{\mathbf{f}}_2$: diffusion velocity (North-south)

$$\frac{d}{dt}n_0 = (n_3 - n_0)\left(\vec{\mathbf{f}}_1 \times \vec{\mathbf{s}}_1\right) + (n_4 - n_0)\left(\vec{\mathbf{f}}_2 \times \vec{\mathbf{s}}_2\right) + (n_1 - n_0)\left(-\vec{\mathbf{f}}_1 \times \vec{\mathbf{s}}_3\right) + (n_2 - n_0)\left(-\vec{\mathbf{f}}_4 \times \vec{\mathbf{s}}_1\right)$$



$\vec{\mathbf{s}}_n$: side lengths
n_i: Total number in cell i
$\vec{\mathbf{m}}_n$: Movement towards cell i
$\vec{\mathbf{s}}_2$
Neighbor 2 (n_2)
$\vec{\mathbf{s}}_3$
Neighbor 3 (n_3)
Focal cell (n_0)
$\vec{\mathbf{m}}_2$
$\vec{\mathbf{m}}_1$
$\vec{\mathbf{m}}_3$
$\vec{\mathbf{m}}_4$
Neighbor 1 (n_1)
$\vec{\mathbf{s}}_1$
Neighbor 4 (n_4)
$\vec{\mathbf{s}}_4$
$\frac{d}{dt}n_0 = (n_3 - n_0)(\vec{\mathbf{m}}_1 \times \vec{\mathbf{s}}_1) + (n_4 - n_0)(\vec{\mathbf{m}}_2 \times \vec{\mathbf{s}}_2) + (n_1 - n_0)(\vec{\mathbf{m}}_3 \times \vec{\mathbf{s}}_3) + (n_2 - n_0)(\vec{\mathbf{m}}_4 \times \vec{\mathbf{s}}_1)$

Movement math

- $\mathbf{M}$ is a movement matrix

$$\frac{\partial}{dt}\mathbf{d}_t = (\mathbf{u}\nabla + \nabla \cdot \boldsymbol{\Sigma}\nabla)\mathbf{d}_t$$

- So

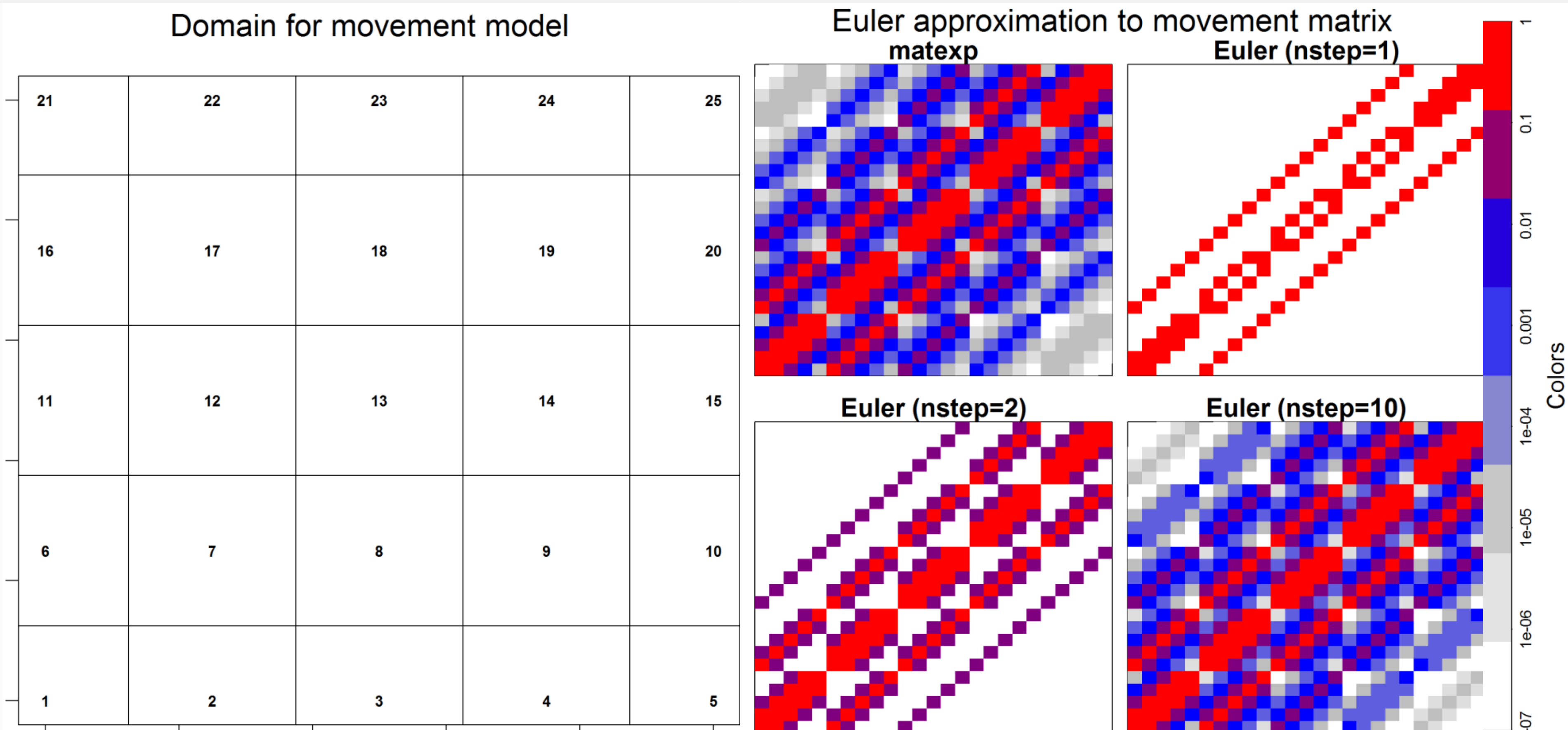
$$\mathbf{M} = \mathrm{matexp}(\mathbf{u}\nabla + \nabla \cdot \boldsymbol{\Sigma}\nabla)$$

- And using the Euler approximation with Δt steps:

$$\mathbf{M} \approx \left(\mathbf{I} + \frac{\mathbf{u}\boldsymbol{\nabla} + \boldsymbol{\nabla} \cdot \boldsymbol{\Sigma}\boldsymbol{\nabla}}{n_{\Delta t}}\right)^{n_{\Delta t}}$$

Movement math

- Euler approximation is sparse!
  - Sparseness / accuracy scales with number of steps



Movement math

Spatial Gompertz with movement

$$\mathbf{d}_{t+1} = (\mathbf{M}\mathbf{d}_t) * e^{(\rho-1)\log(\mathbf{d}_t)+\boldsymbol{\omega}+\boldsymbol{\varepsilon}_t}$$

$$\boldsymbol{\omega} \sim \mathrm{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_{\omega})$$

$$\boldsymbol{\varepsilon}_t \sim \mathrm{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_{\varepsilon})$$

- If $\mathbf{M} = \mathbf{I}$, then:

$$\log(\mathbf{d}_{t+1}) = \rho\log(\mathbf{d}_t)+\boldsymbol{\omega} + \boldsymbol{\varepsilon}_t$$

  – … it reduces to the version with no movement

Movement math

Spatial Gompertz with movement

$$\mathbf{d}_{t+1} = (\mathbf{Md}_t) * e^{(\rho-1)\log(\mathbf{d}_t)+\boldsymbol{\omega}+\boldsymbol{\varepsilon}_t}$$

$$\boldsymbol{\omega} \sim \mathrm{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_{\omega})$$

$$\boldsymbol{\varepsilon}_{\boldsymbol{t}} \sim \mathrm{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}})$$

- Can re-parameterize using sparse matrices

$$\widehat{\boldsymbol{d}}_{\boldsymbol{t+1}} = (\mathbf{M}\boldsymbol{d}_{\boldsymbol{t}}) * e^{(\rho-1)\log(\boldsymbol{d}_{\boldsymbol{t}})+\boldsymbol{\omega}}$$

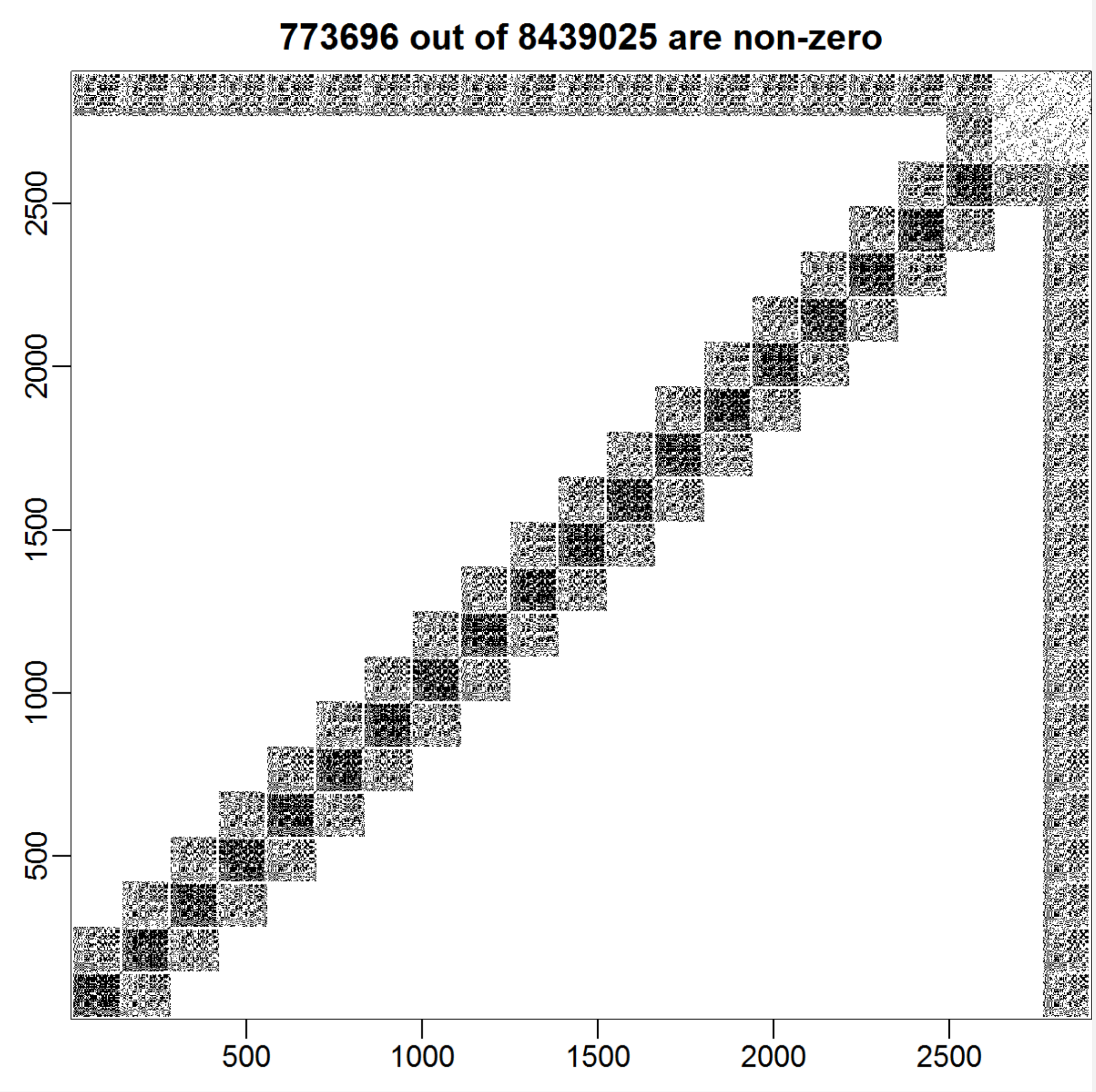
$$\boldsymbol{\omega} \sim \boldsymbol{GRF}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\omega}})$$

$$\log(\boldsymbol{d}_{\boldsymbol{t}}) \sim \boldsymbol{GRF}(\log(\widehat{\boldsymbol{d}}_{\boldsymbol{t}}), \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}})$$

Movement math

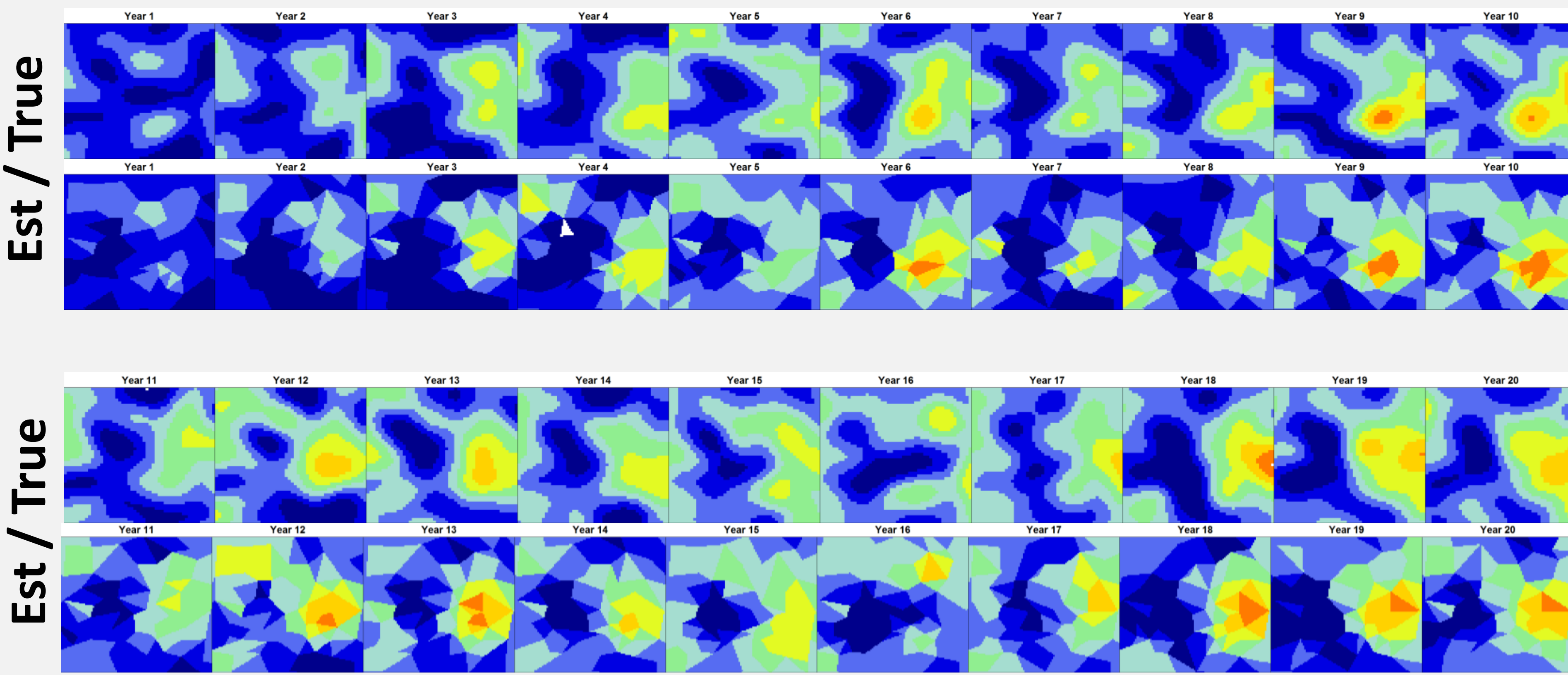
Sparse matrix

- Easier computation
- Less computer memory
- Faster speed



Movement math

- Highly separable computation
  - 200 triangles + 20 years = 5 min. on laptop
  - Seems to recovery dynamics and density



Movement math

- Movement can be estimated using count data
  - Euler approximation is sufficient for local advection-diffusion