Teaching numerical modeling with research problems

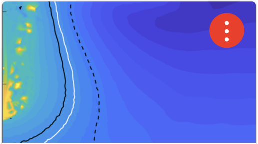
Marc A. Hesse

Department of Geological Sciences
Oden Institute for Computational Science and Engineering
Center for Planetary Systems Habitability
The University of Texas at Austin

October 17, 2021

GEO 325M Numerical Modeling for Geoscientists

Cryovolcanism in Occator crater, Ceres



Numerical modeling
GEO 325J
Spring 2018

Hesse & Castillo-Rogez (2018)

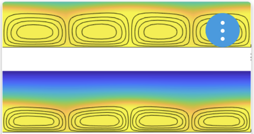
Raymond et al. (2020)

Oxidant transport by brine on Europa

Numerical modeling
GEO 325M/398M
Spring 2019

Hesse et al. (2021)

T-dependent ice shell convection on Europa

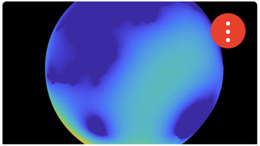


Numerical Modeling
GEO 325M/GEO 398M
Spring 2020

Carnahan et al. (2021)

Wolfenbarger et al. (2021)

Groundwater on Mars response to impacts

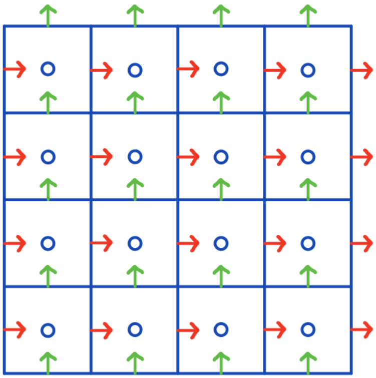


Numerical modeling
GEO 325M/398M
Spring 2021

manuscript in the works ...

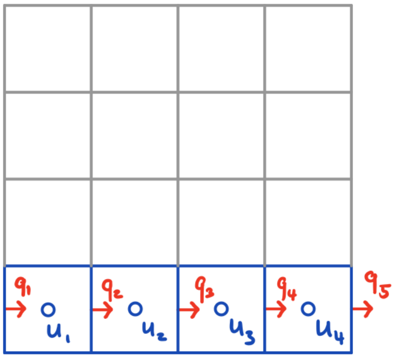
Research project provides motivation and and guidance for course.

Simplest working discretization



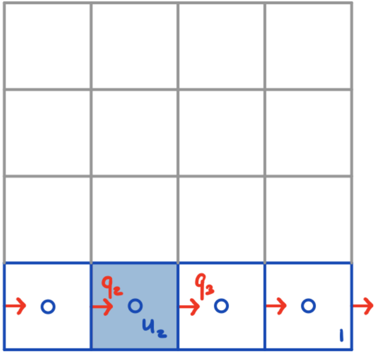
Staggered grid

Simplest working discretization



Potential in cell centers
Fluxes on cell faces

Simplest working discretization

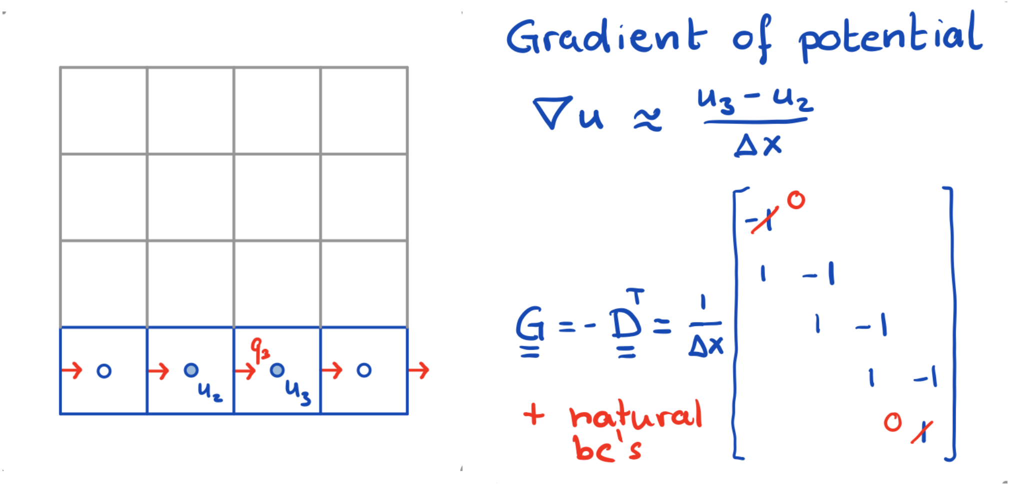


Divergence of flux

$$\nabla \cdot q \approx \frac{q_3 - q_2}{\Delta x}$$

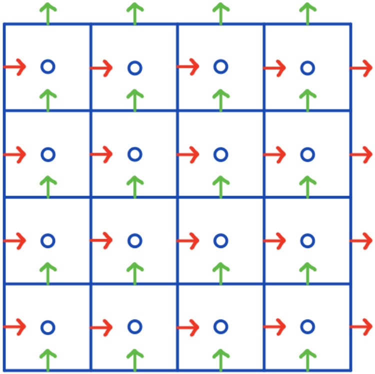
$$\underline{\underline{D}} = \frac{1}{\Delta x}\begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & -1 & 1 & \\ & & & -1 & 1 \end{bmatrix}$$

Simplest working discretization

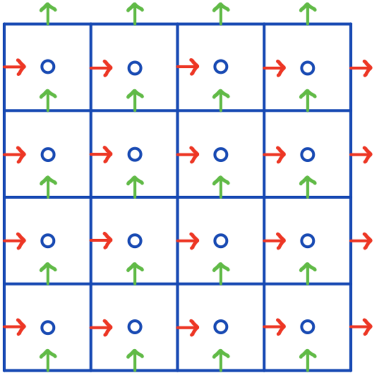


Programming concept: sparse matrices (**spdiags.m**)

Going 2D/3D with Tensor products



Going 2D/3D with Tensor products



2D discrete operators

$$\underline{\underline{D}}_2 = \begin{bmatrix} \underline{\underline{D}}_1 & & & \\ & \underline{\underline{D}}_1 & & \\ & & \underline{\underline{D}}_1 & \\ & & & \underline{\underline{D}}_1 \end{bmatrix}$$

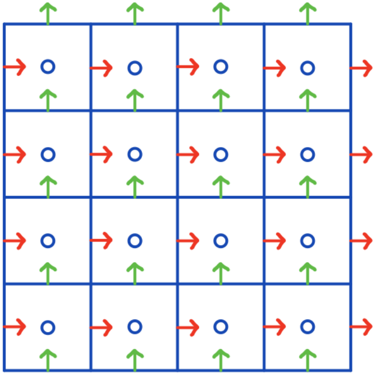
Build $\underline{\underline{D}}_2$ directly from $\underline{\underline{D}}_1$

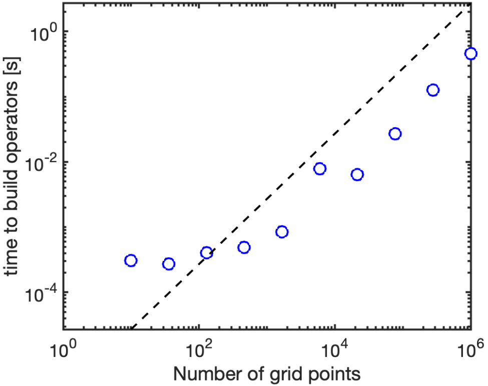
$$\underline{\underline{D}} = \underline{\underline{I}} \otimes \underline{\underline{D}} = \text{kron}(\underline{\underline{I}}, \underline{\underline{D}})$$

$$\underline{\underline{G}} = -\underline{\underline{D}}^T$$

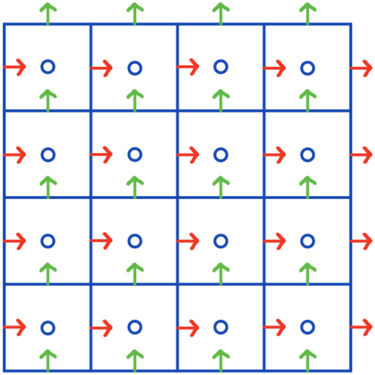
⇒ simple, fast, few mistakes

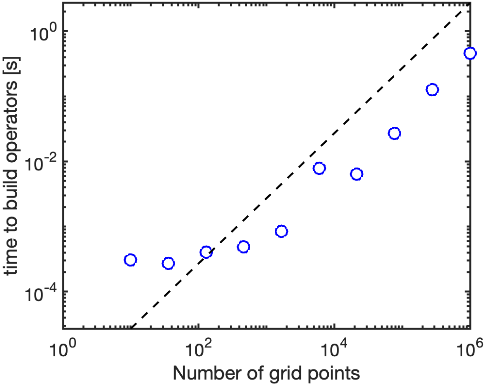
Going 2D/3D with Tensor products





Going 2D/3D with Tensor products





Programming concept: Tensor/Kronecker products (**kron.m**)

Modular building blocks

Discrete operators

$\underline{\underline{D}}$ = divergence

$\underline{\underline{G}}$ = gradient

$\underline{\underline{C}}$ = curl

$\underline{\underline{A}}(\underline{v})$ = advection

$\underline{\underline{I}}$ = identity

dimension: 1D, 2D, 3D

different geometries

Modular building blocks

Discrete operators

$\underline{\underline{D}}$ = divergence

$\underline{\underline{G}}$ = gradient

$\underline{\underline{C}}$ = curl

$\underline{\underline{A}}(\underline{v})$ = advection

$\underline{\underline{I}}$ = identity

dimension: 1D, 2D, 3D

different geometries

Poisson Equation

PDE: $-\nabla \cdot \nabla u = f$

Discrete: $-\underline{\underline{D}} * \underline{\underline{G}} * \underline{u} = \underline{f}$

Linear operator: $\underline{\underline{L}} = -\underline{\underline{D}} * \underline{\underline{G}}$

Solve: $\underline{u} = \underline{\underline{L}} \backslash \underline{f}$

Modular building blocks

Discrete operators

$\underline{\underline{D}}$ = divergence

$\underline{\underline{G}}$ = gradient

$\underline{\underline{C}}$ = curl

$\underline{\underline{A}}(\underline{v})$ = advection

$\underline{\underline{I}}$ = identity

dimension: 1D, 2D, 3D

different geometries

Advection-Diffusion Eqn

PDE: $\frac{\partial u}{\partial t} + \nabla \cdot [\underline{v} u - \nabla u] = f$

Discrete:

$$\underline{\underline{I}} \frac{\underline{u}^{n+1} - \underline{u}^{n}}{\Delta t} + \underline{\underline{D}} * (\underline{\underline{A}}(\underline{v}) - \underline{\underline{G}})\, \underline{u}^{n+1} = \underline{f}$$

Linear operator:

$$\underline{\underline{L}} = \underline{\underline{I}} + \Delta t\, \underline{\underline{D}} * (\underline{\underline{A}}(\underline{v}) - \underline{\underline{G}})$$

Solve: $\underline{u}^{n+1} = \underline{\underline{L}} \backslash (\Delta t\, \underline{f} + \underline{u}^{n})$

Modular building blocks

Discrete operators

$\underline{\underline{D}}$ = divergence

$\underline{\underline{G}}$ = gradient

$\underline{\underline{C}}$ = curl

$\underline{\underline{A}}(\underline{v})$ = advection

$\underline{\underline{I}}$ = identity

dimension: 1D, 2D, 3D

different geometries

Stokes Equation

PDE: $-\mu \nabla\times\nabla\times \underline{v} - \nabla p = f$

$$\nabla\cdot\underline{v} = 0$$

Discrete:

$$-\mu\, \underline{\underline{C}}^T * \underline{\underline{C}} * \underline{v} - \underline{\underline{G}}\, \underline{p} = \underline{f}$$

$$\underline{\underline{D}} * \underline{v} = \underline{0}$$

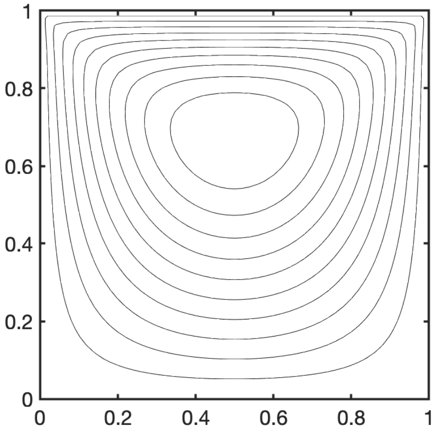
Linear operator:

$$\underline{\underline{L}} = \begin{bmatrix} -\mu \underline{\underline{C}}^T\underline{\underline{C}} & -\underline{\underline{G}} \\ \underline{\underline{D}} & \underline{\underline{0}} \end{bmatrix} \qquad \underline{u} = \begin{bmatrix} \underline{v} \\ \underline{p} \end{bmatrix}$$

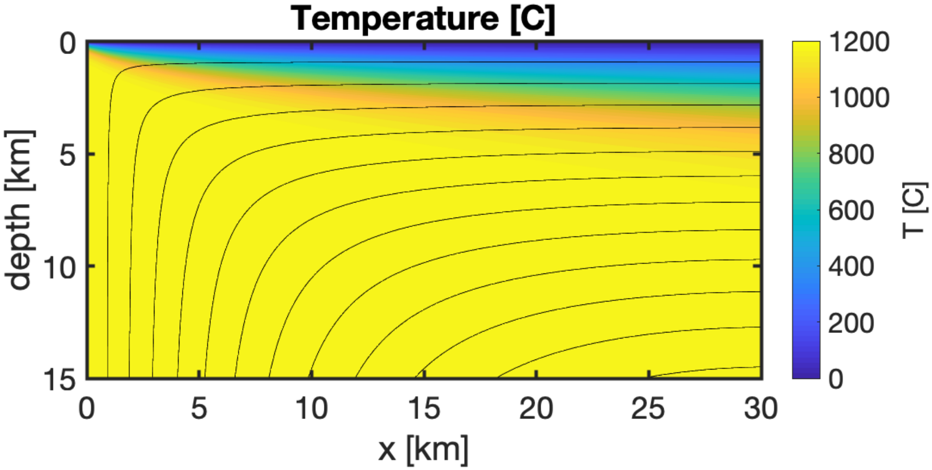
Allows us to solve a different problem every year!

Code example: Viscous Corner Flow

```
mu = 1;

%% Build staggered grids
Gridp.xmin = 0; Gridp.xmax = 1; Gridp.Nx = 50;
Gridp.ymin = 0; Gridp.ymax = 1; Gridp.Ny = 50;
Grid = build_stokes_grid(Gridp);

%% Build Stokes operators
[D,Edot,Dp,Gp,Z,I] = build_stokes_ops(Grid);
A = 2*mu*D*Edot; %
L = [A, -Gp;...
     Dp, Z];
fs = spalloc(Grid.N,1,0);

%% Boundary conditions
BC.dof_dir = [Grid.dof_ymax_vx(2:end-1);... % tangential velocity on the top
              Grid.dof_no_pene;...          % no penetration on all bnd's
              Grid.dof_pc];                 % pressure constraint

BC.g       = [ones(Grid.p.Nx-1,1);...       % tangential velocity on the top
              zeros(Grid.N_no_pene,1);...   % no penetration on all bnd's
              0];                           % pressure constraint
[B,N,fn] = build_bnd(BC,Grid,I);

%% Solve for Stokes flow
u = solve_lbvp(L,fs+fn,B,BC.g,N);
v = u(1:Grid.p.Nf); p = u(Grid.p.Nf+1:end);
PSI = comp_streamfun(v,Grid.p);
```



Homework example: Mid-ocean ridge



Undergraduate research

Amy De Luna (2018)

Cryovolcanism

Preston Durham (2019)

Fluids in salt

Jaxon Liebeck (2021)

Martian hydrology

Hope is to eventually have a undergrad authored paper using class tools.

Thank you for your attention.

Class website: https://mhesse.github.io/numerical_modeling/

Matlab Discrete Operator Toolbox:
https://github.com/mhesse/MatlabDiscreteOperatorToolbox