Sharp Interface Capturing Methods

Frederic G. Gibou

Department of Mechanical Engineering
Department of Computer Science
Department of Mathematics

University of California, Santa Barbara

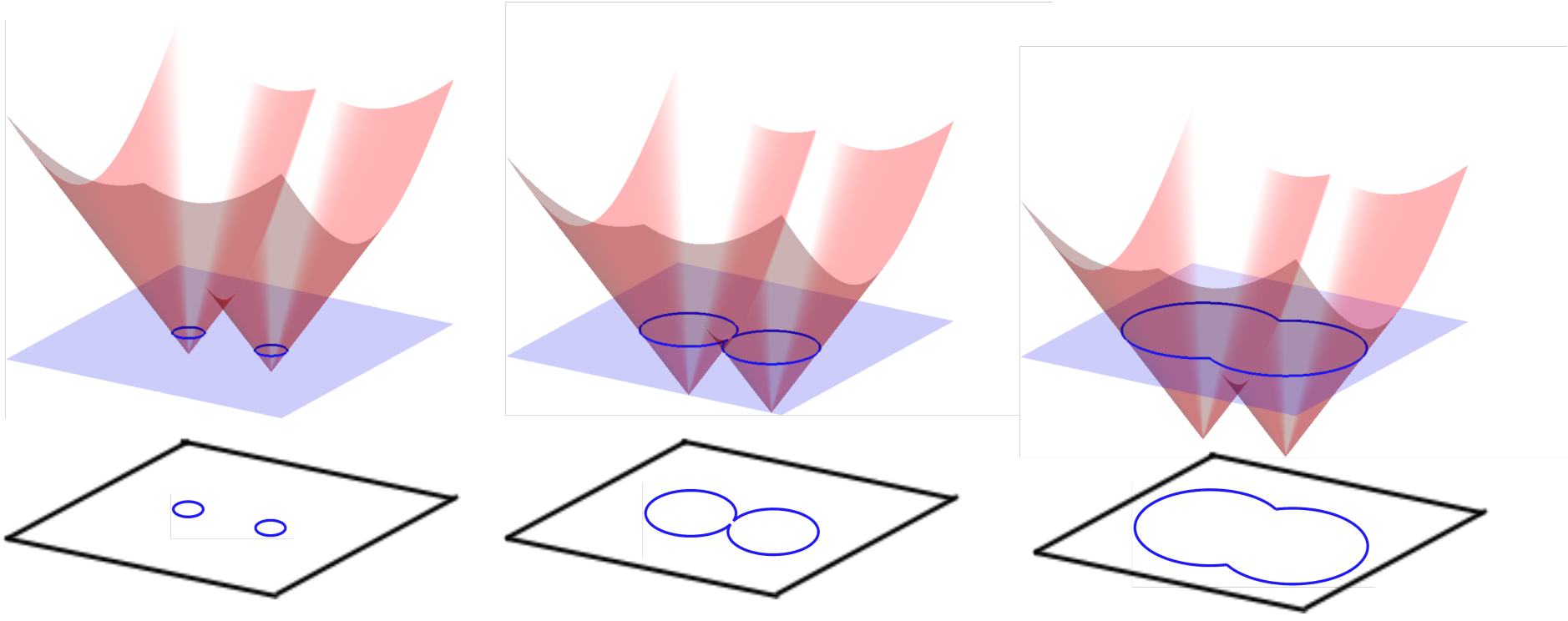
Introduction: challenges for free boundary problems

- Keep track of a moving front
- Efficiently capture small scales
- Impose boundary conditions at the front

DENDRITIC GROWTH

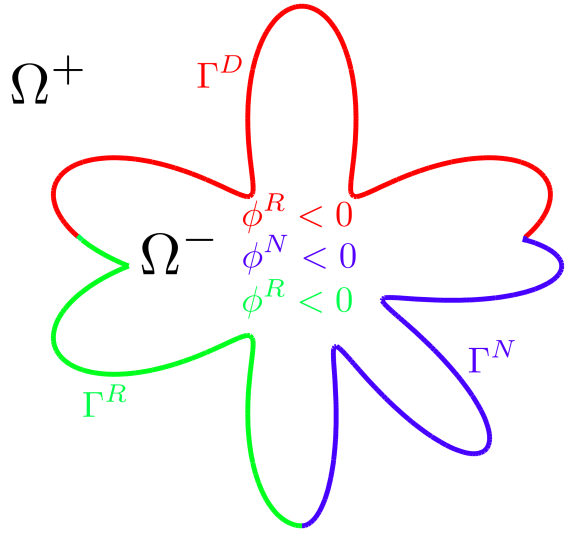
Challenges for free boundary problems

- Keep track of a moving front --> Level-Set Representation

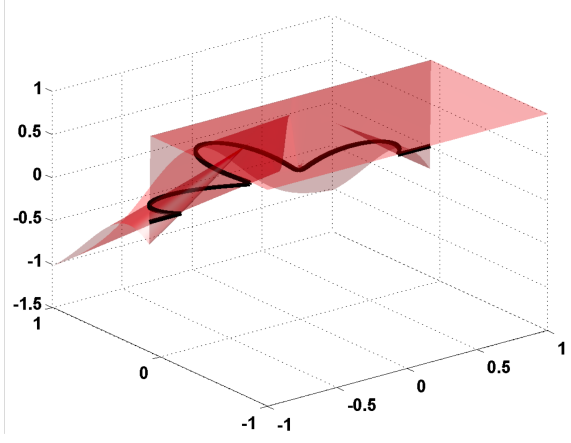


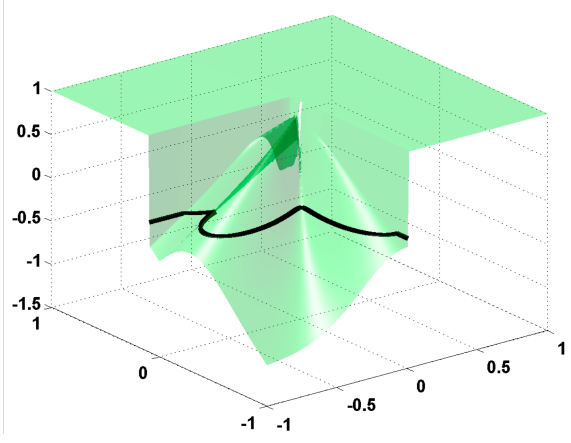
➢ Challenges for free boundary problems

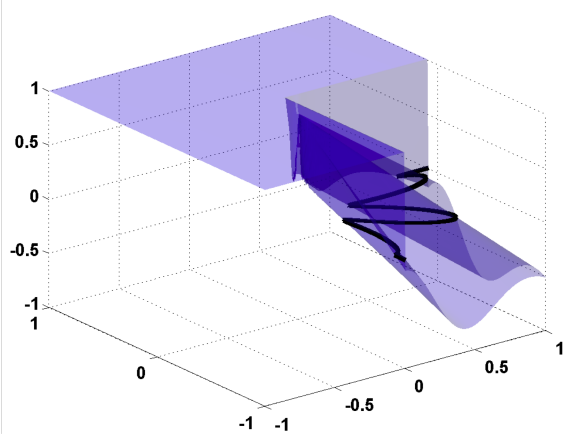
➤ Impose boundary conditions at the front --> Sharp Approach



$$\Delta u = F \qquad \mathbf{x} \in \Omega^-,$$
$$u = G \qquad \text{on } \phi^D = 0,$$
$$\frac{\partial u}{\partial n} = K \qquad \text{on } \phi^N = 0,$$
$$\frac{\partial u}{\partial n} + \alpha u = M \qquad \text{on } \phi^R = 0,$$

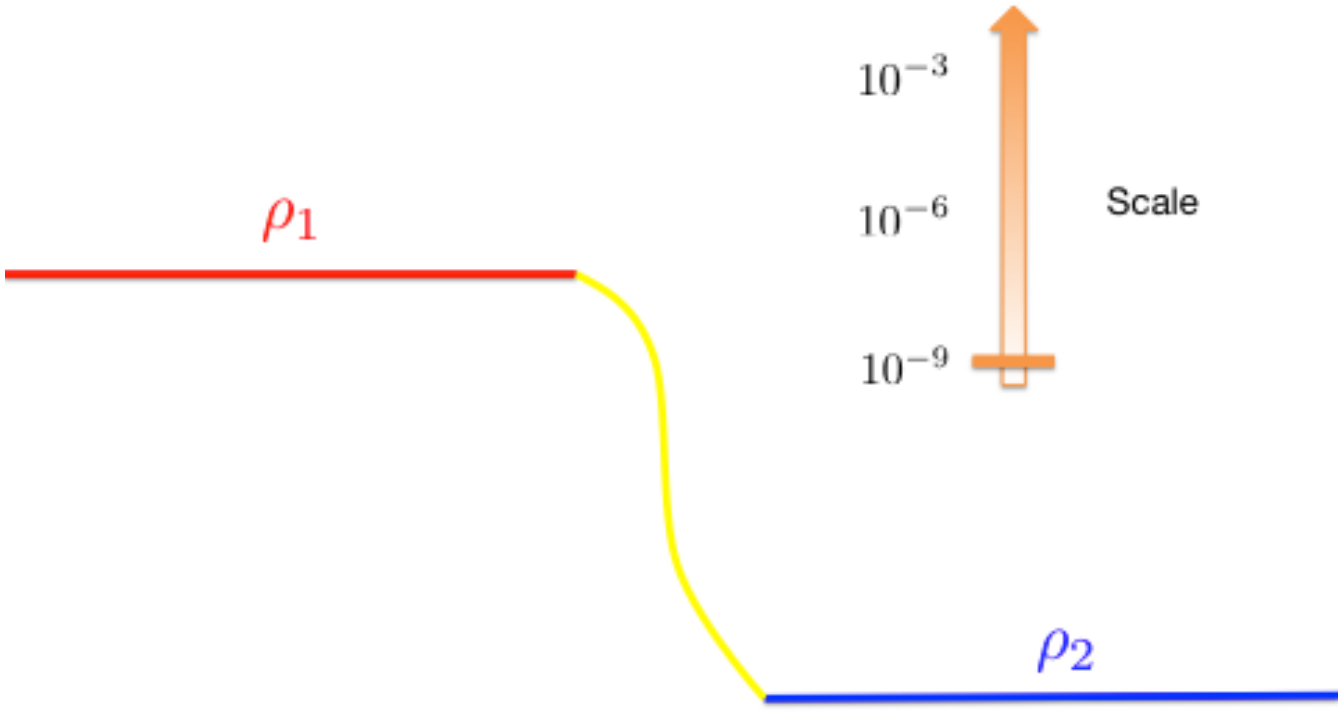






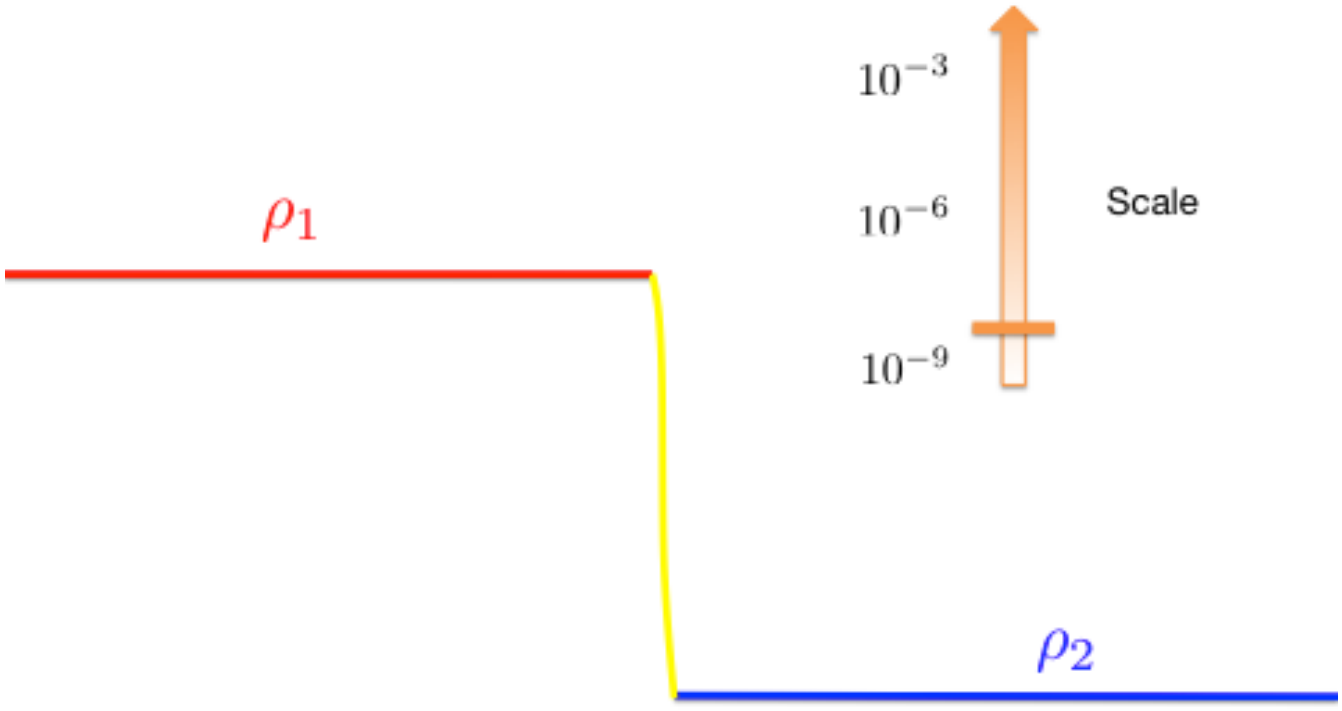
➢ Challenges for free boundary problems

- ➤ Impose boundary conditions at the front --> Sharp Approach
- ➤ Interested in macroscale simulations



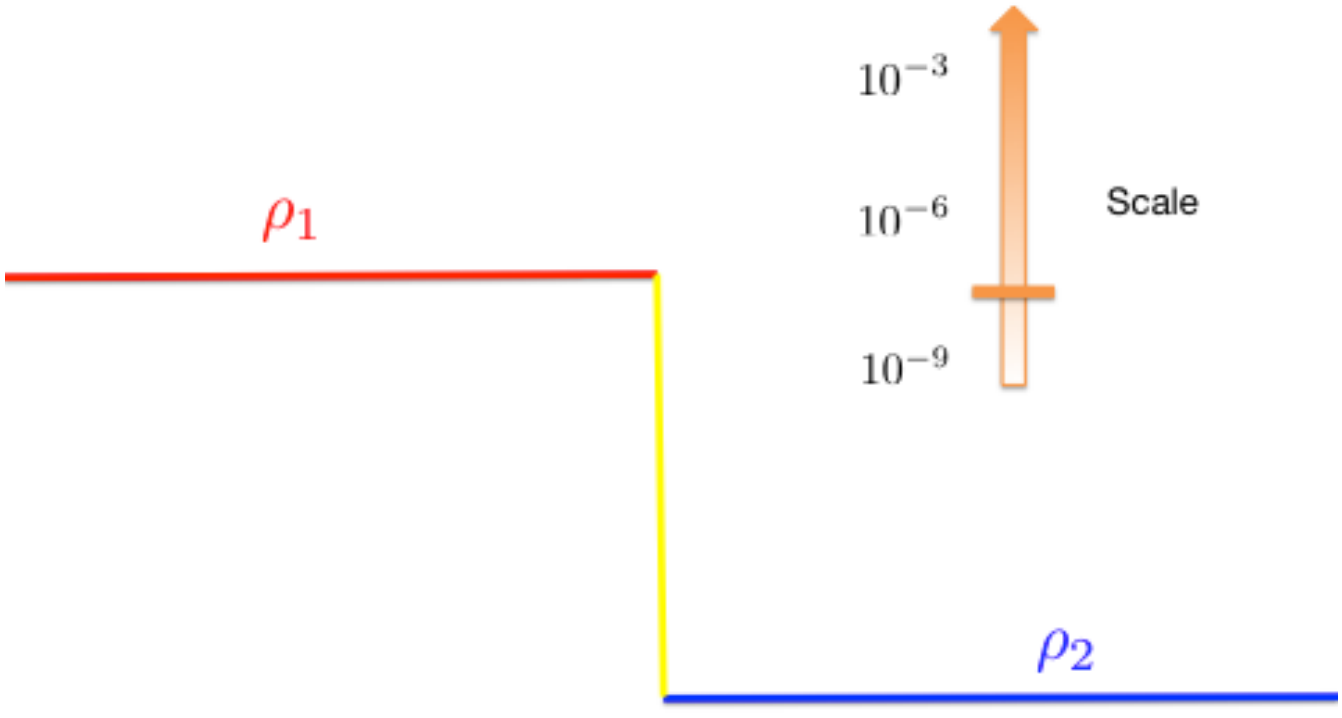
➢ Challenges for free boundary problems

- ➤ Impose boundary conditions at the front --> Sharp Approach
- ➤ Interested in macroscale simulations



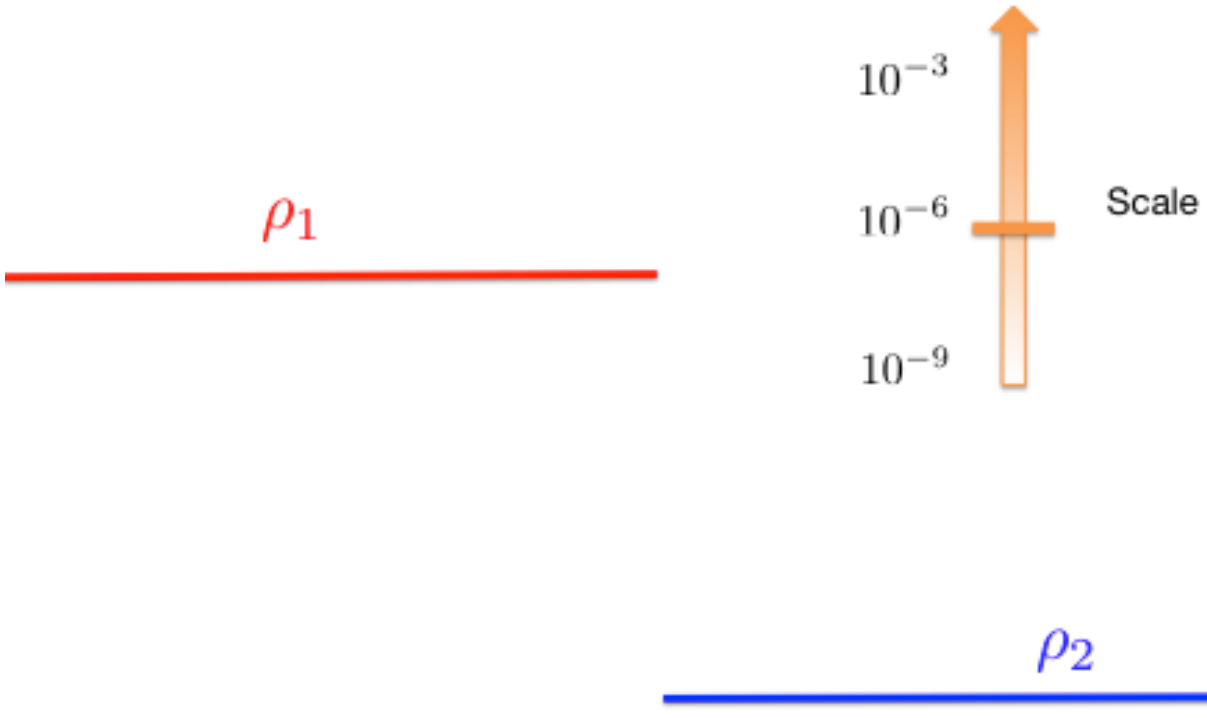
➢ Challenges for free boundary problems

- ➤ Impose boundary conditions at the front --> Sharp Approach
- ➤ Interested in macroscale simulations



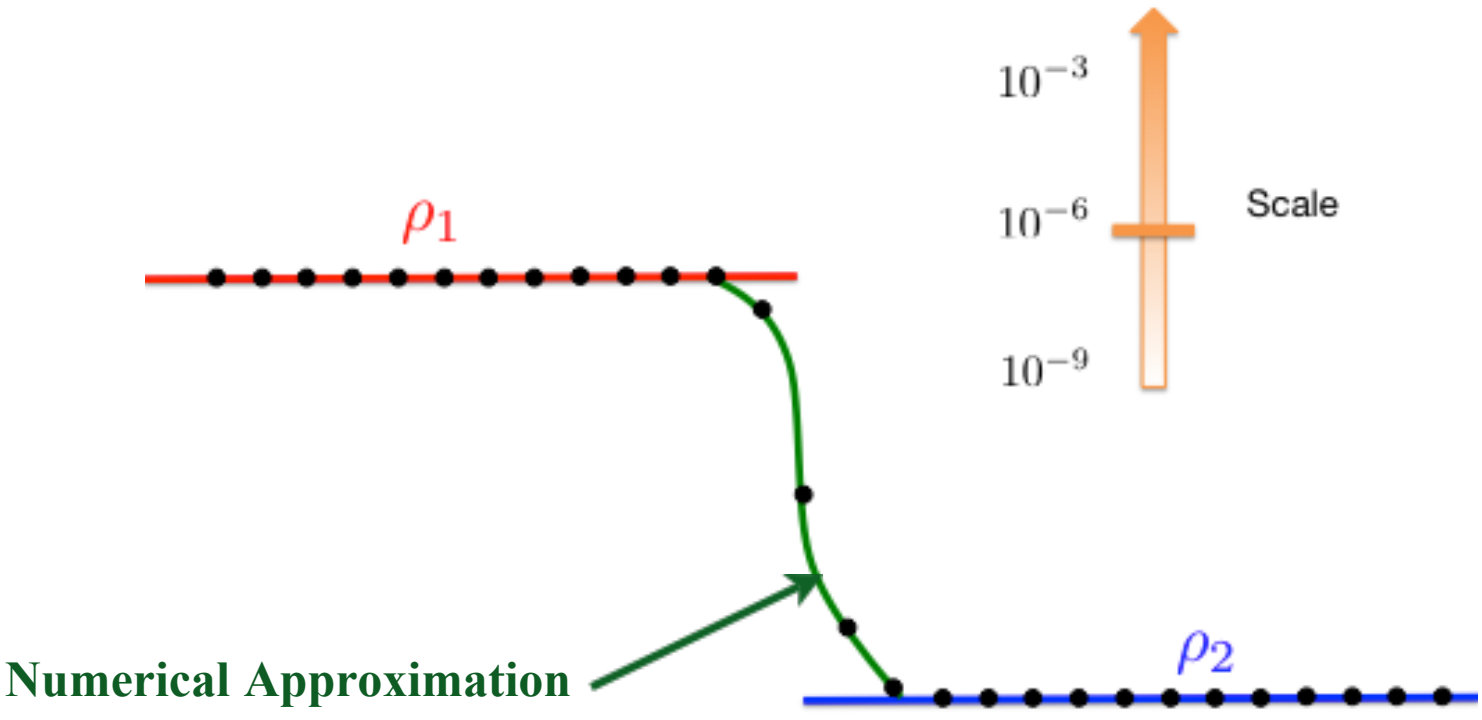
➢ Challenges for free boundary problems

- ➤ Impose boundary conditions at the front --> Sharp Approach
- ➤ Interested in macroscale simulations



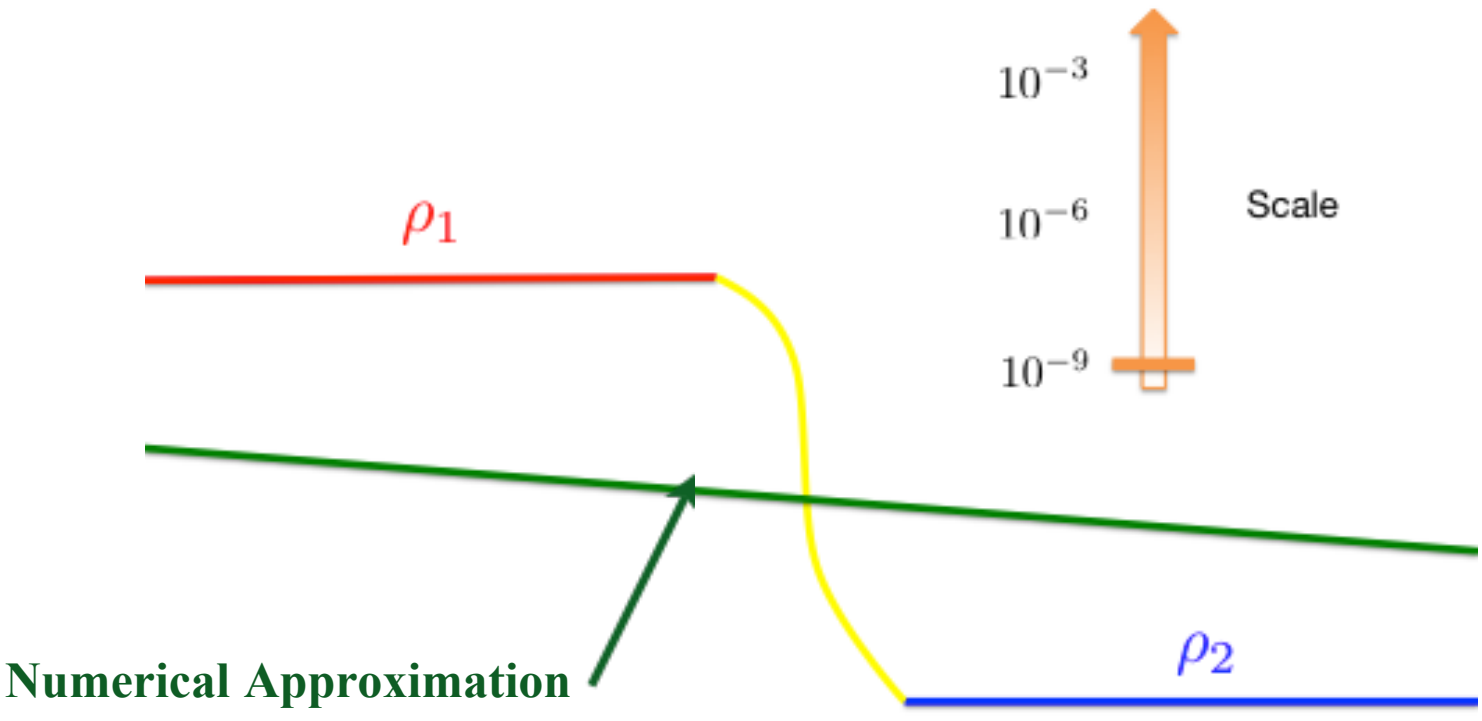
➢ Challenges for free boundary problems

- ➤ Impose boundary conditions at the front --> Sharp Approach
- ➤ Interested in macroscale simulations



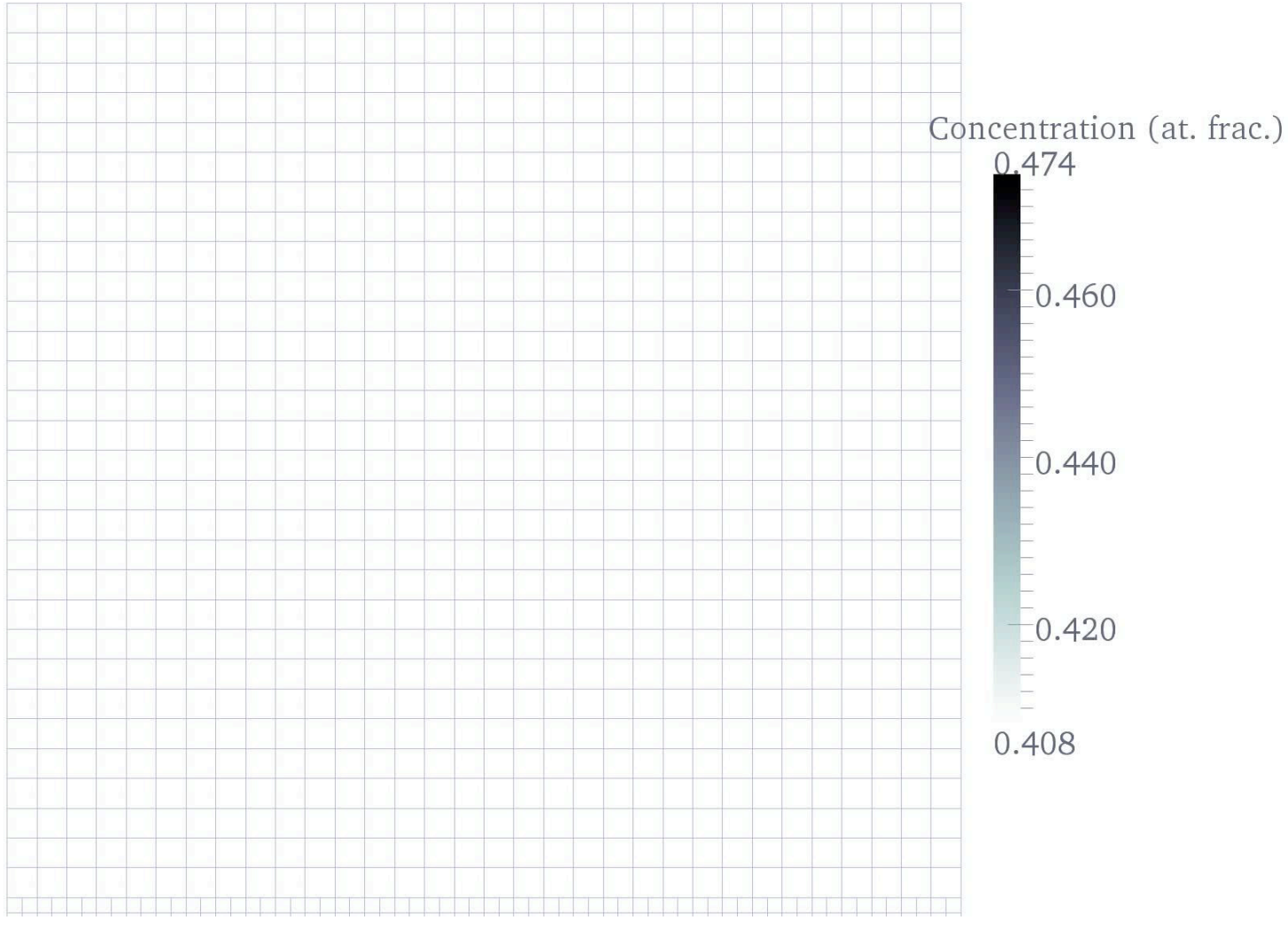
➢ Challenges for free boundary problems

- ➤ Impose boundary conditions at the front --> Sharp Approach
- ➤ Interested in macroscale simulations



➢ Challenges for free boundary problems

- ➤ Efficiently capture small scales --> Adaptive Mesh Refinement

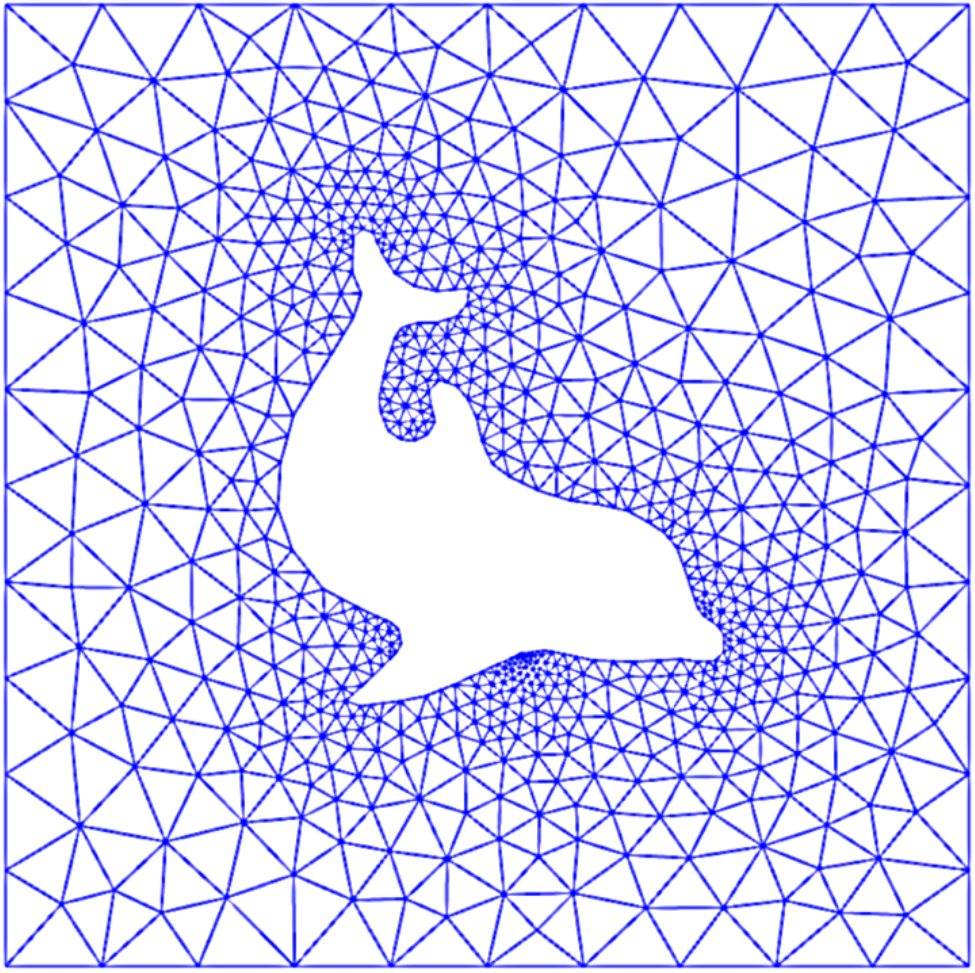


PART I:

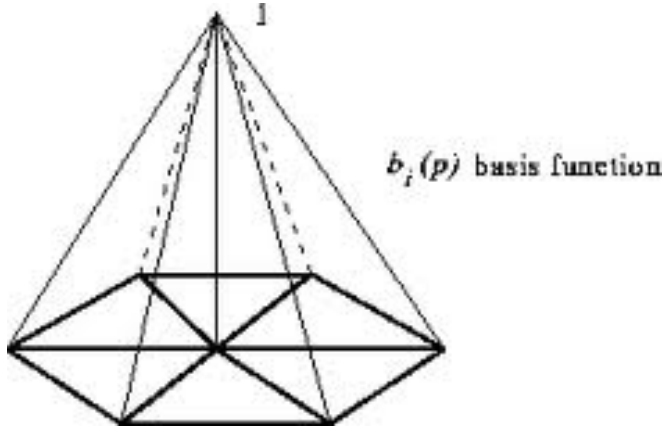
Adaptive Quadtree/Octree Grids

➢ PDE solvers on Quadtree/Octree Cartesian grids

➤ FEM: Many pros - mesh generation challenging



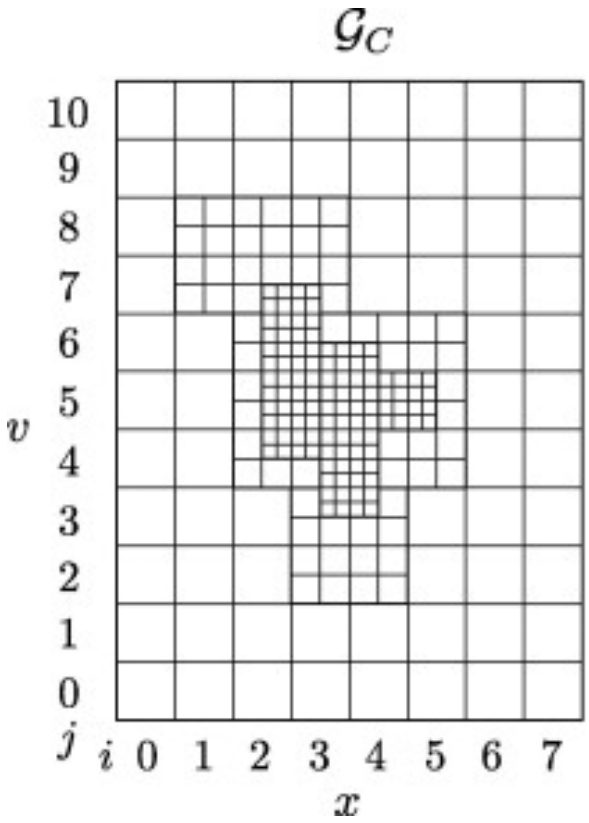
- Test functions, IBP, …
- Approximate the space
- Impose BC easily
- Leads to SPD

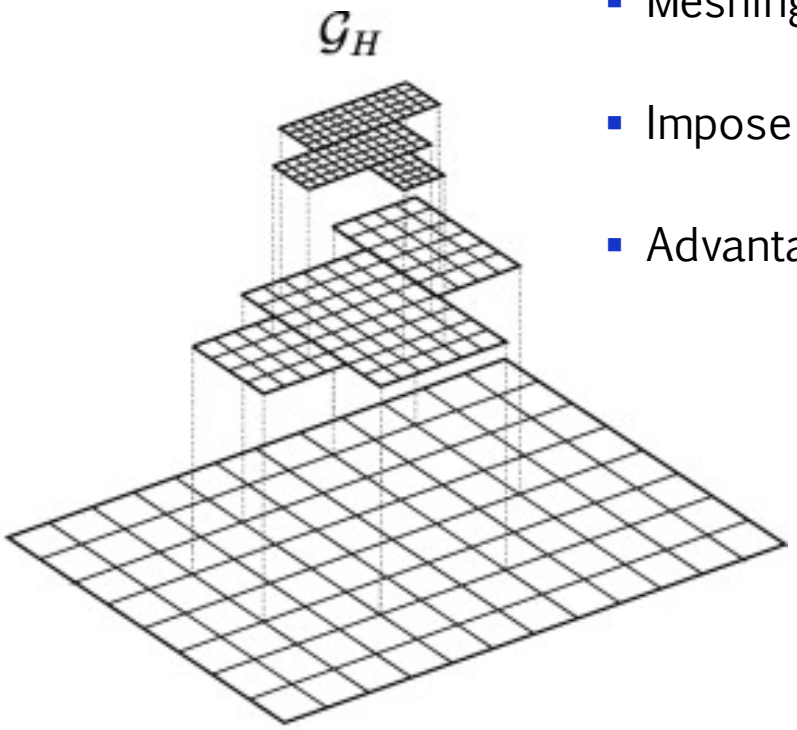


➢ PDE solvers on Quadtree/Octree Cartesian grids

➤ AMR: Many pros - block structures consume CPU and memory

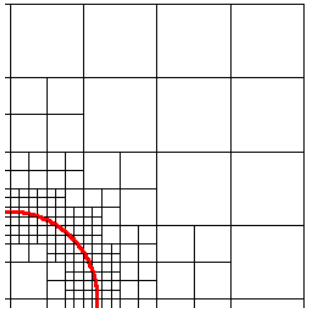
- Approximate the operator
- Meshing straightforward
- Impose BC harder
- Advantages in parallelization

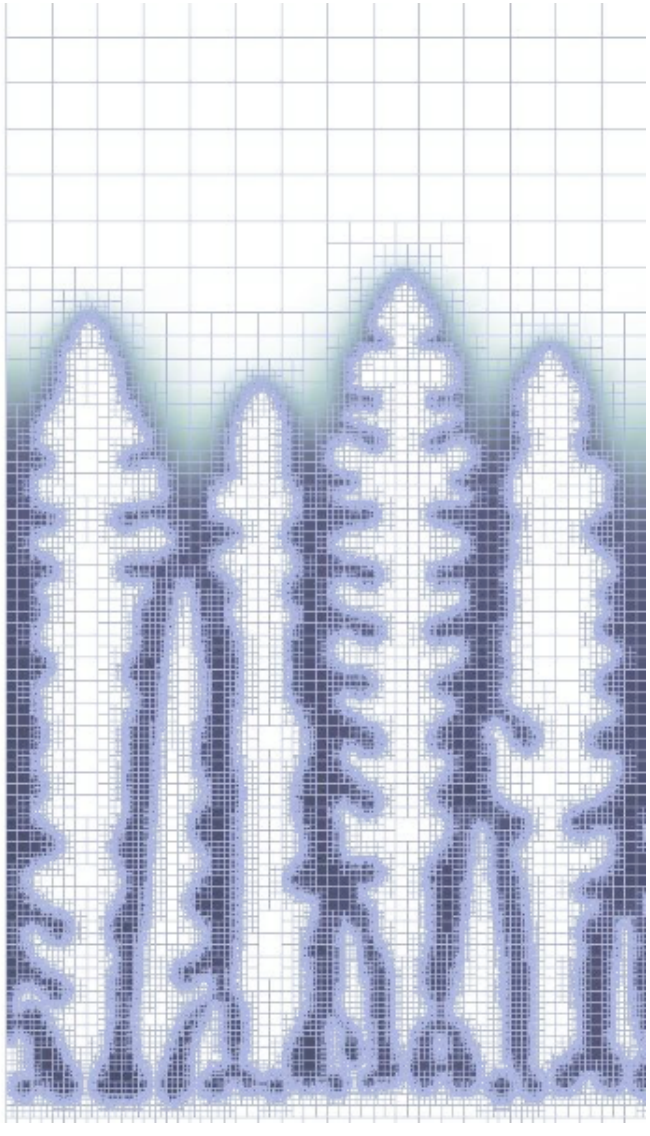




➢ PDE solvers on Quadtree/Octree Cartesian grids

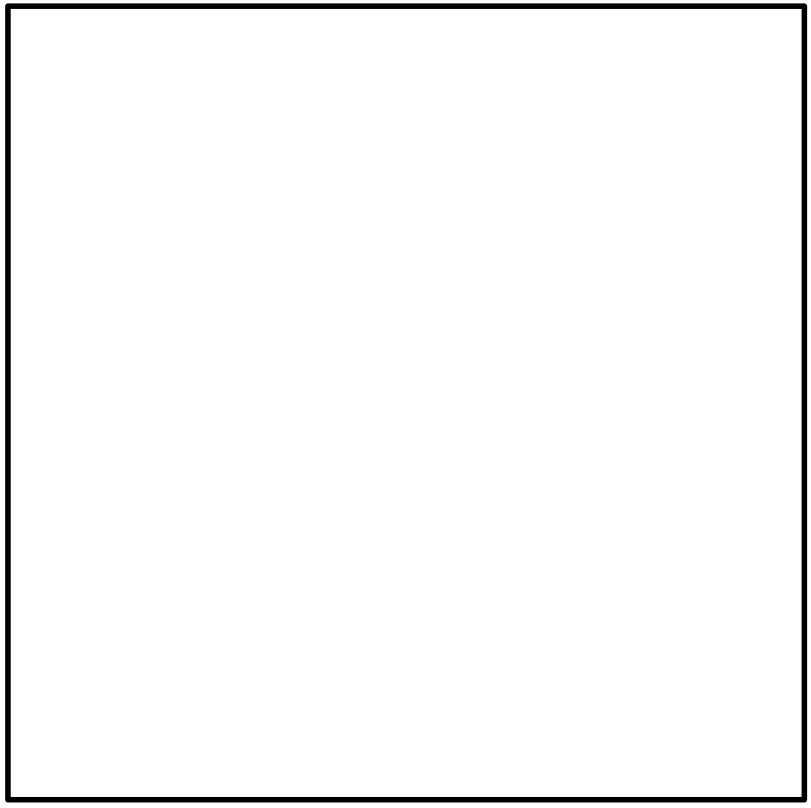
➤ Tree grids: Rather recent - versatile
- reduce number of nodes

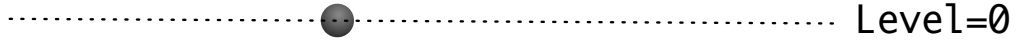




➢ PDE solvers on Quadtree/Octree Cartesian grids

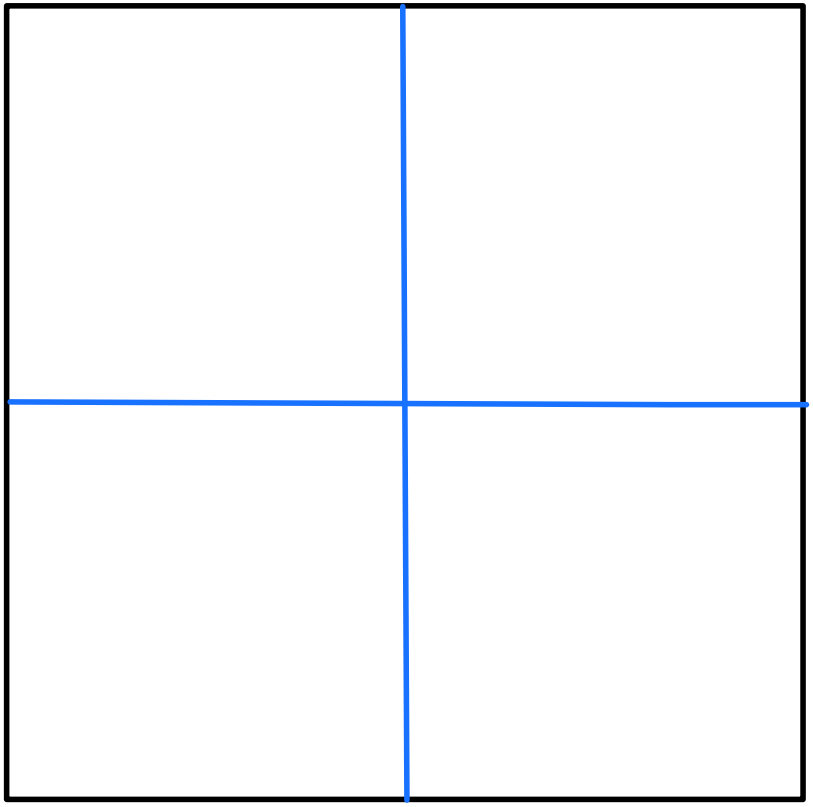
➤ Tree grids: construction

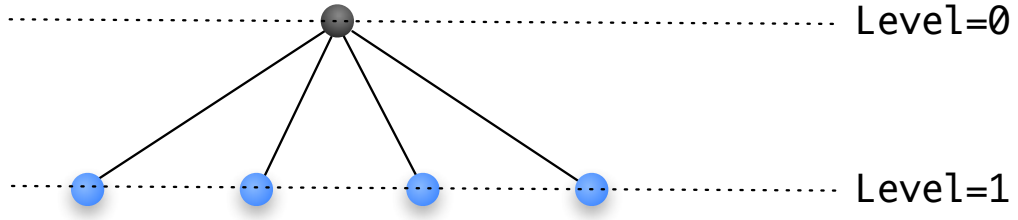




➢ PDE solvers on Quadtree/Octree Cartesian grids

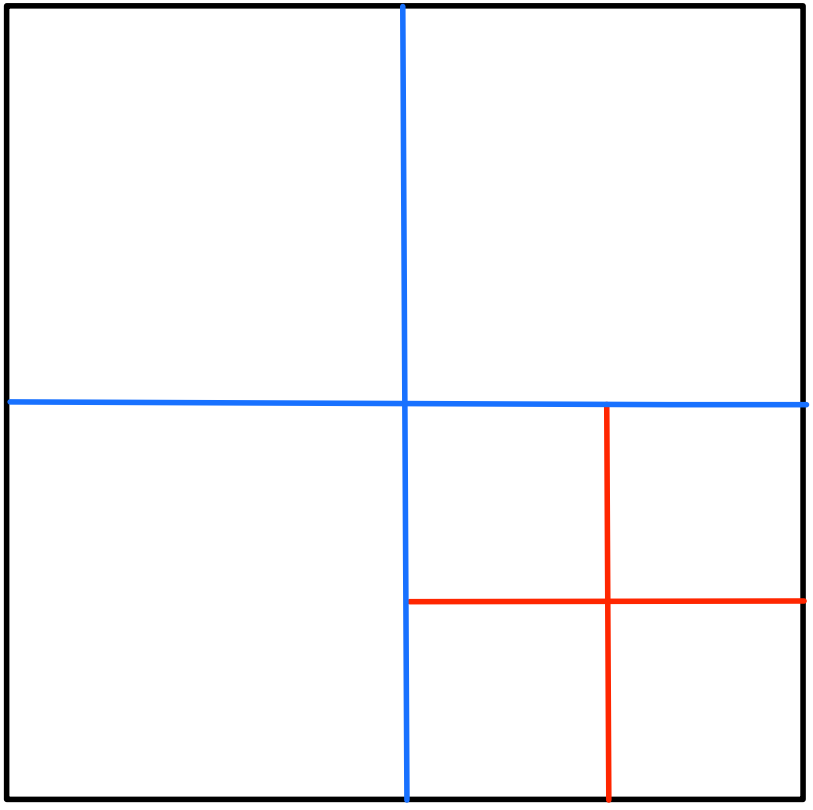
➤ Tree grids: construction

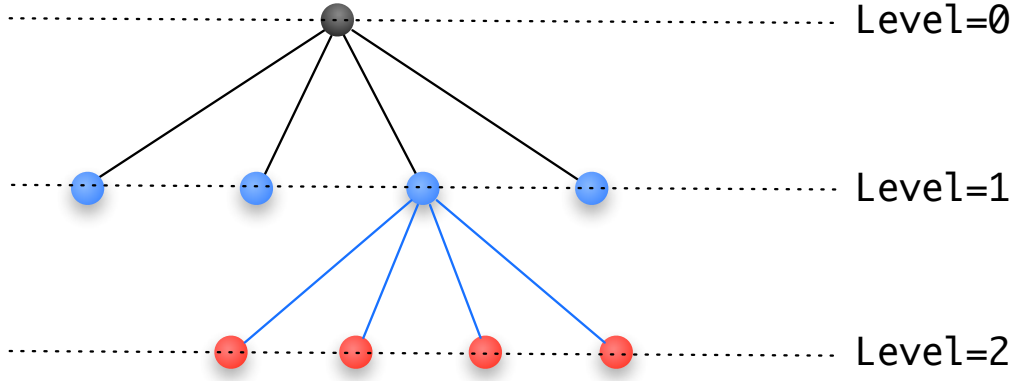




➢ PDE solvers on Quadtree/Octree Cartesian grids

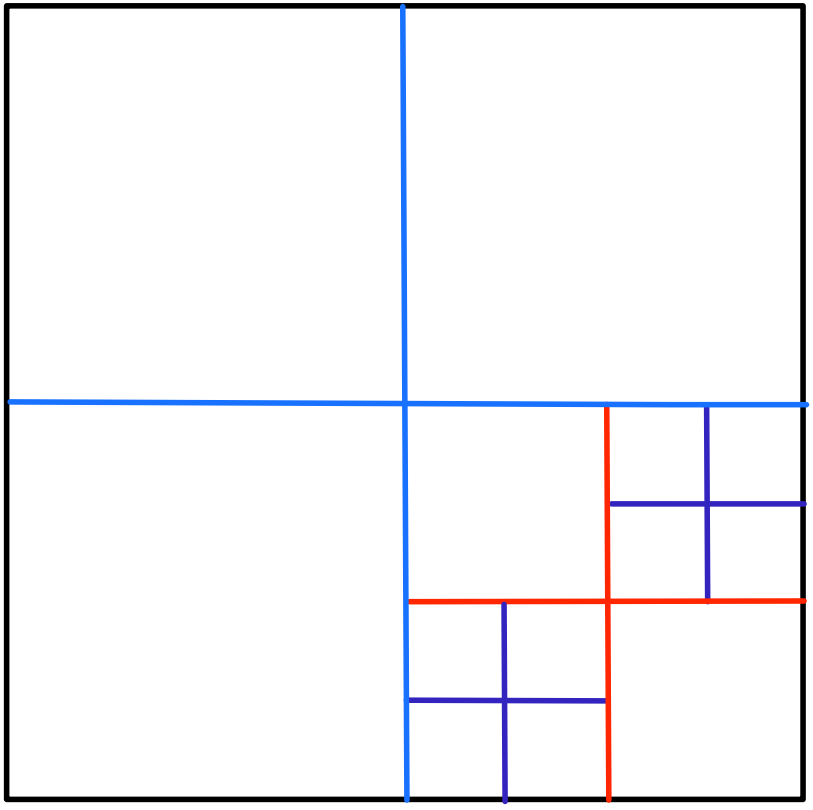
➤ Tree grids: construction

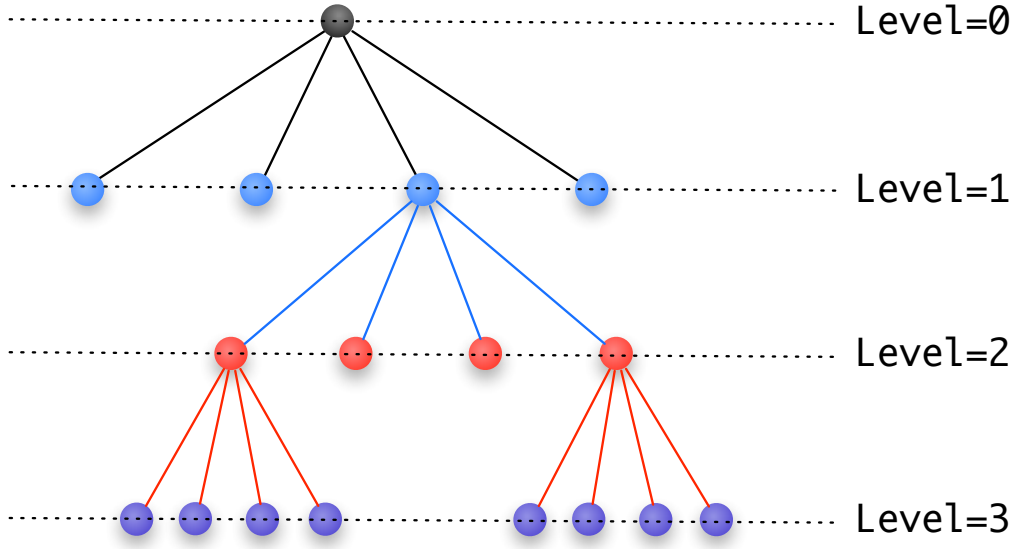




➢ PDE solvers on Quadtree/Octree Cartesian grids

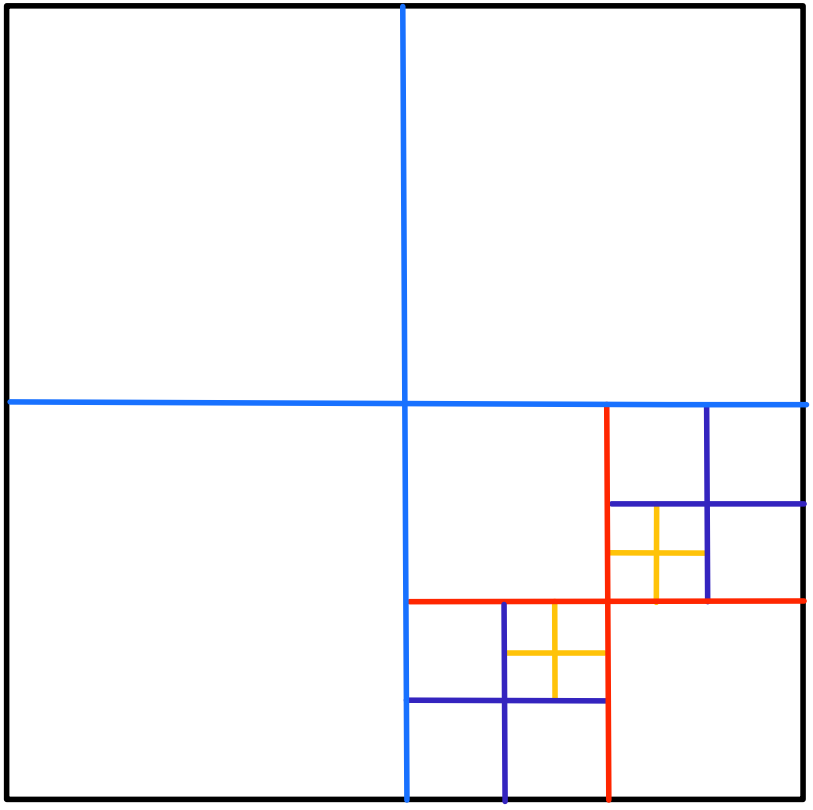
➤ Tree grids: construction

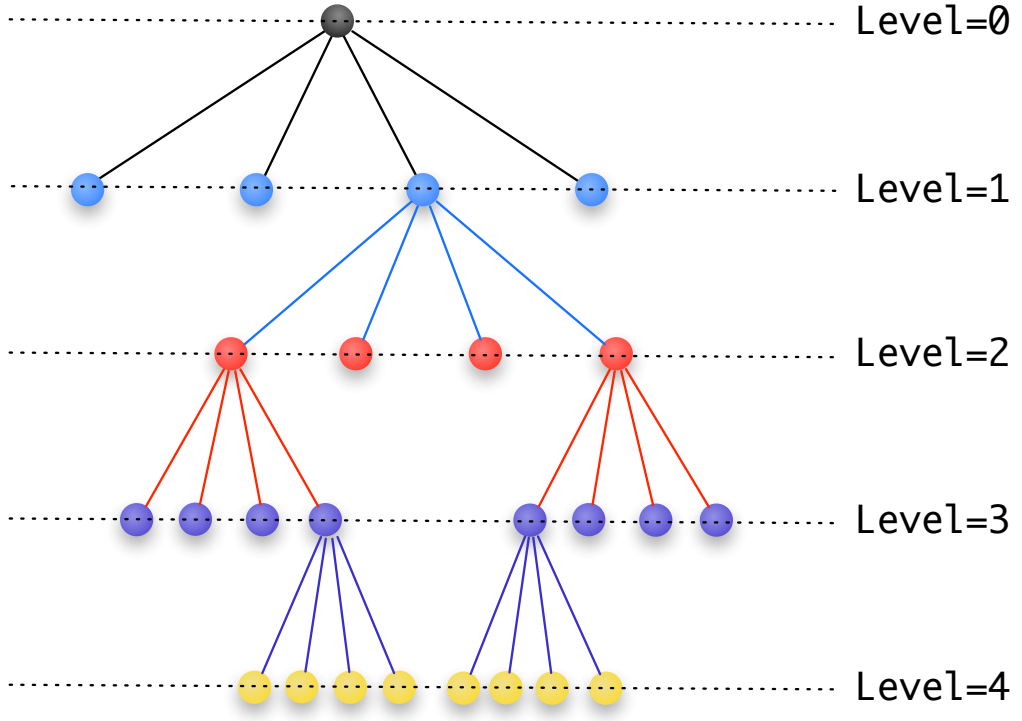




➢ PDE solvers on Quadtree/Octree Cartesian grids

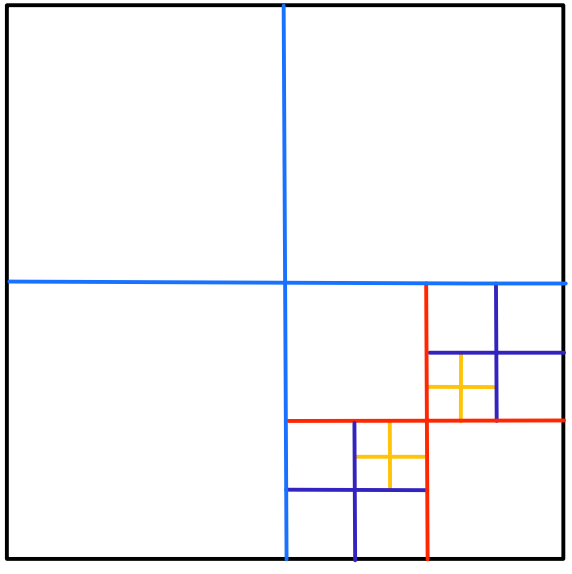
➤ Tree grids: construction

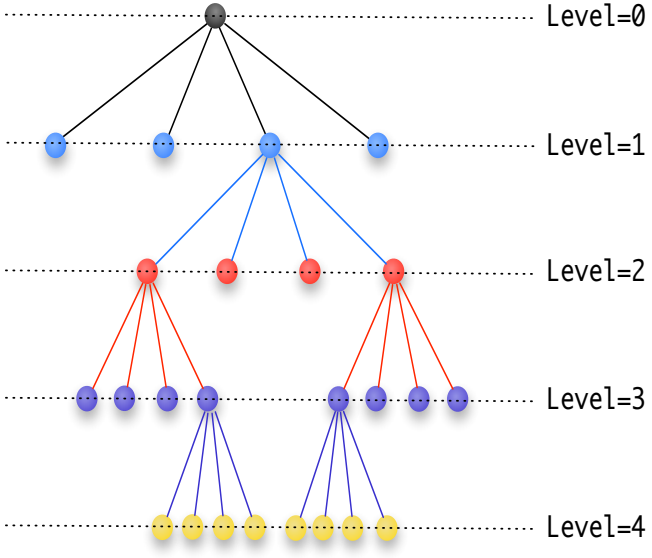




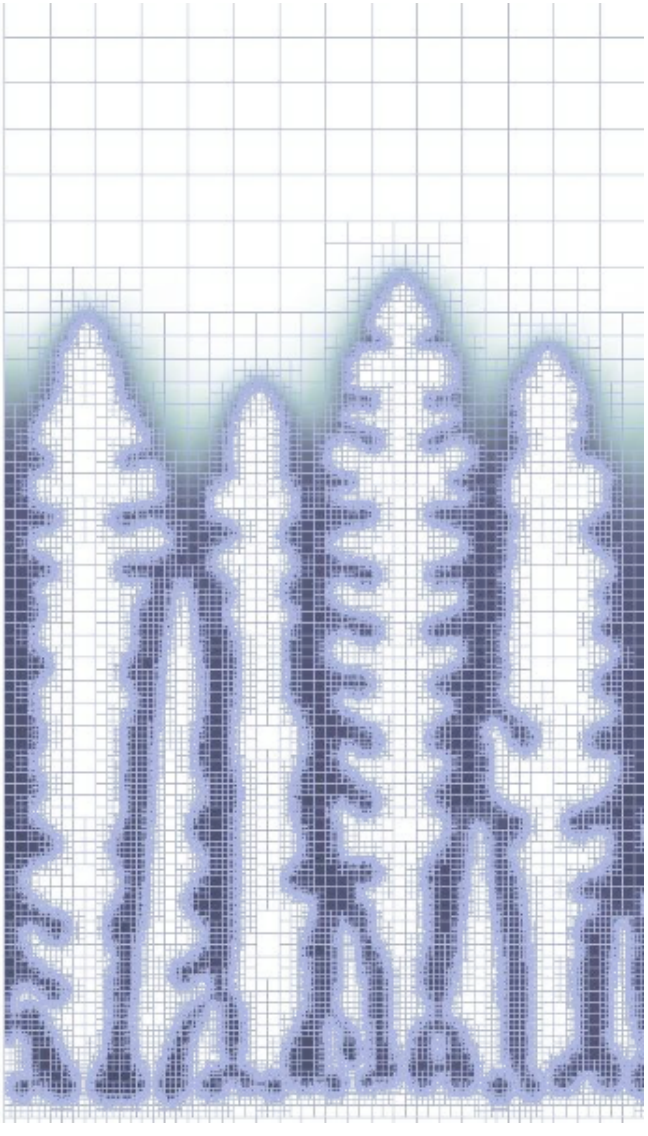
➢ PDE solvers on Quadtree/Octree Cartesian grids

➤ Tree grids: refine near the free boundary



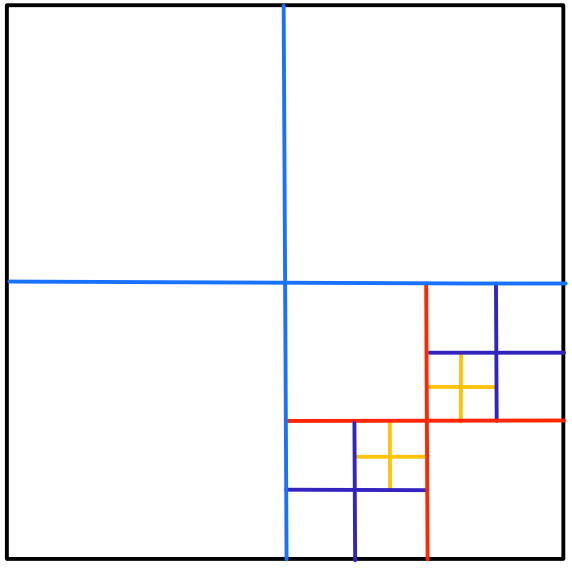


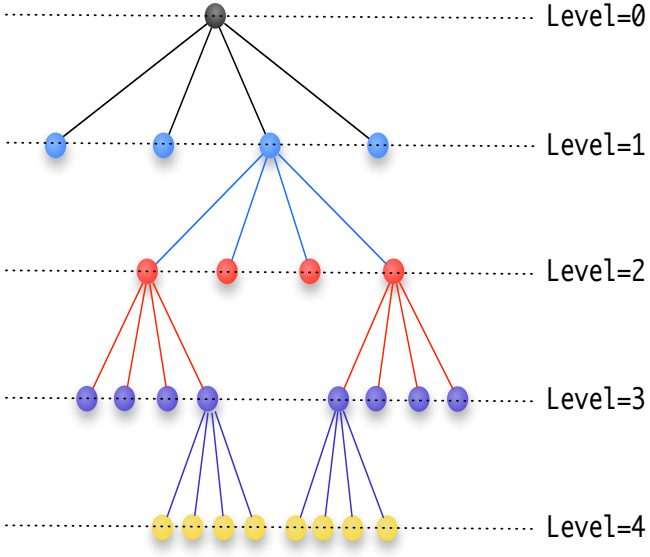
$$\min_{v \in \text{vertices}(C)} |\phi(v)| \leq \text{Lip}(\phi) \cdot \text{diag-size}(C)$$

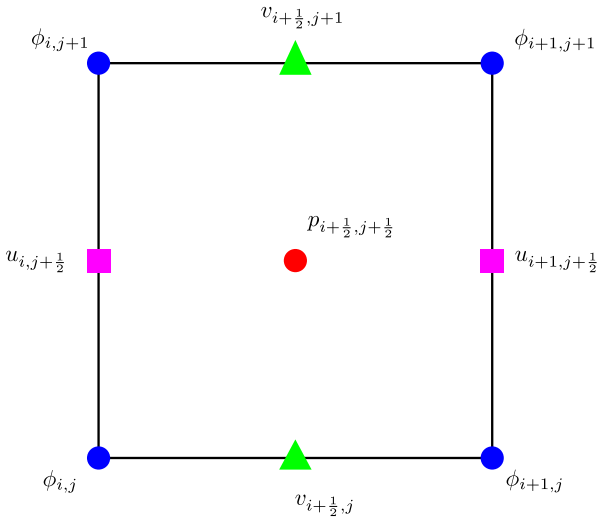


➢ PDE solvers on Quadtree/Octree Cartesian grids

➤ Cell structure: choice of data location

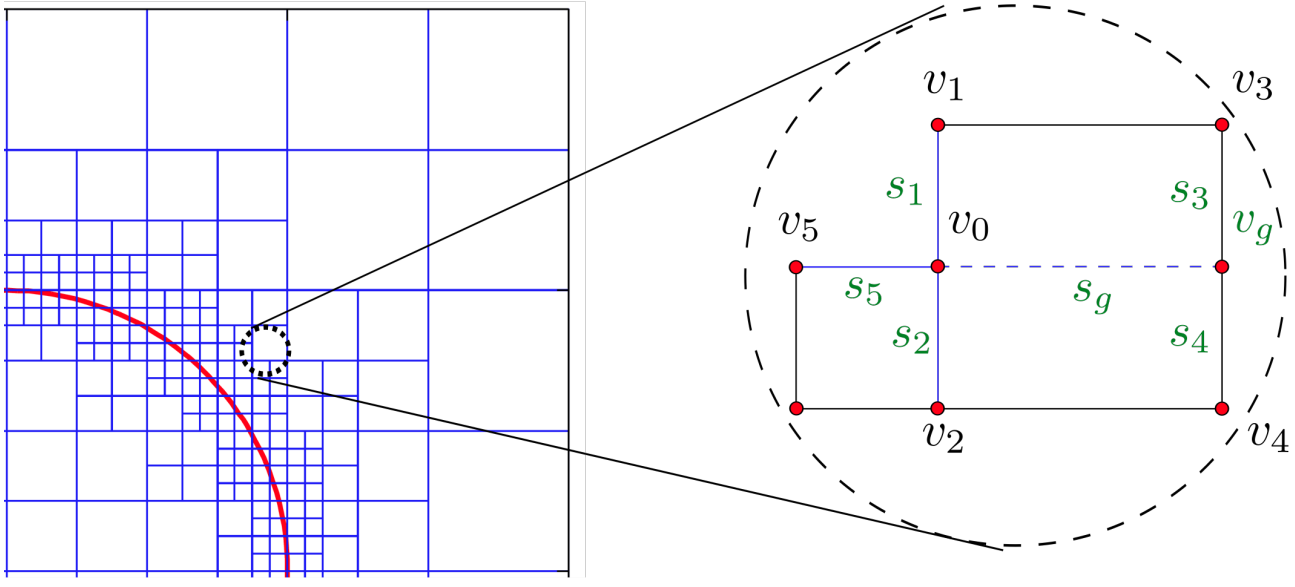






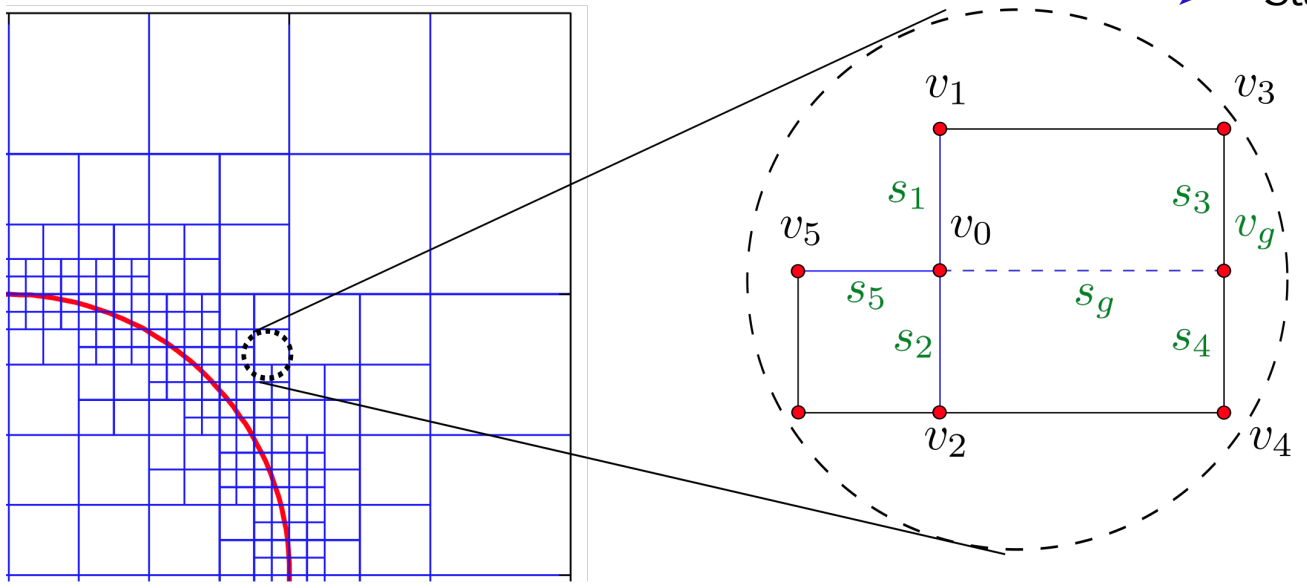
➢ Definition at T-junction nodes

- ➢ Existing approaches: Aftosmis, Losasso, ... : Cell based - FV.
- ➢ Present approach: Node based (pros for level-set methods [Strain])



$$\phi^G = \frac{s_4\phi_3 + s_3\phi_4}{s_3 + s_4}$$

➢ Definition at T-junction nodes

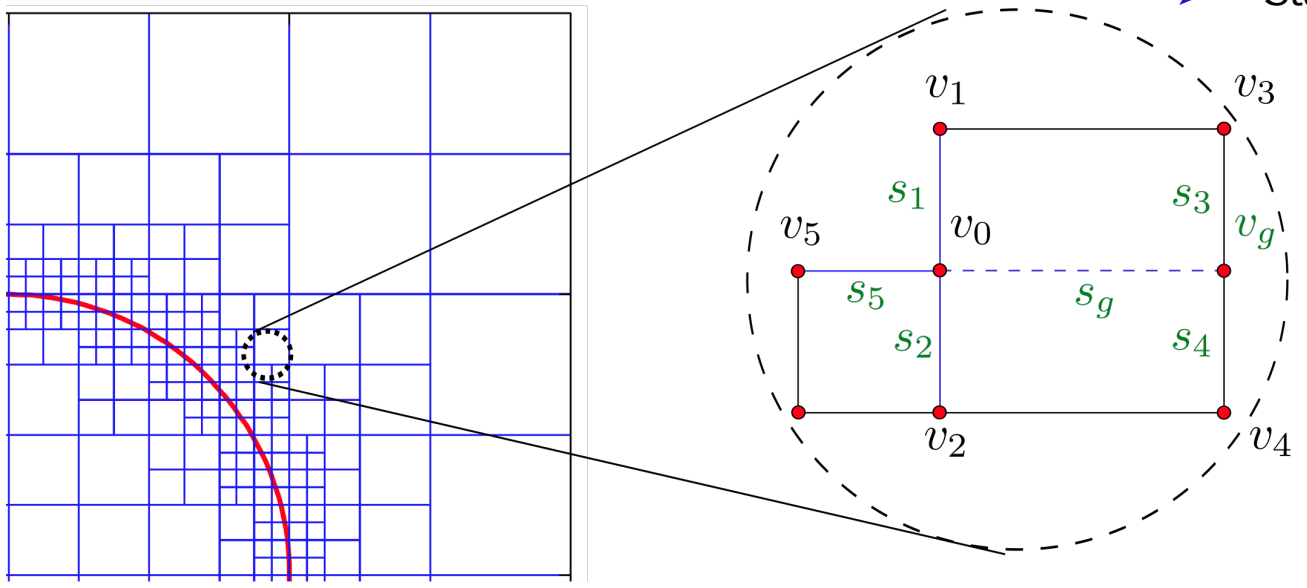


➤ Standard linear interpolation

$$\phi^G = \frac{s_4\phi_3 + s_3\phi_4}{s_3 + s_4}$$

$$\phi^G = \frac{s_4\phi_3 + s_3\phi_4}{s_3 + s_4} = \phi(v_g) + \frac{s_3 s_4}{2}\phi_{yy}(v_0) + O(\Delta x^3)$$

➢ Definition at T-junction nodes



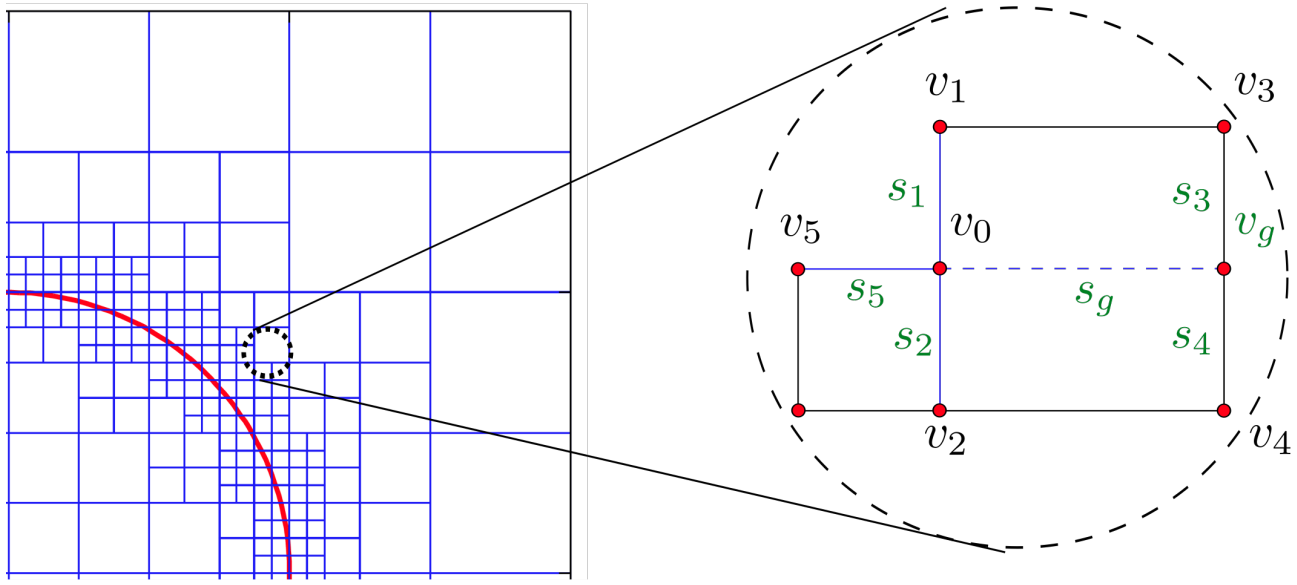
➤ Standard linear interpolation

$$\phi^G = \frac{s_4\phi_3 + s_3\phi_4}{s_3 + s_4}$$

$$\phi^G = \frac{s_4\phi_3 + s_3\phi_4}{s_3 + s_4} = \phi(v_g) + \frac{s_3 s_4}{2}\phi_{yy}(v_0) + O(\Delta x^3)$$

$$\frac{2}{s_1 + s_2}\left(\frac{\phi_1 - \phi_0}{s_1} - \frac{\phi_0 - \phi_2}{s_2}\right)$$

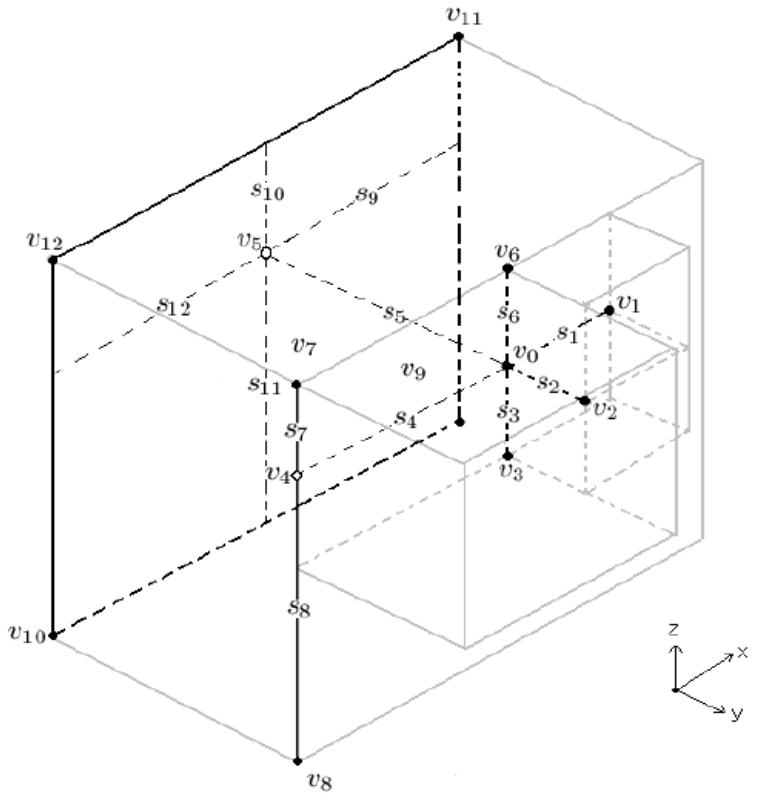
➢ Definition at T-junction nodes



$$\phi^G = \frac{s_4\phi_3 + s_3\phi_4}{s_3 + s_4} - \frac{s_3 s_4}{s_1 + s_2}\left(\frac{\phi_2 - \phi_0}{s_2} + \frac{\phi_1 - \phi_0}{s_1}\right)$$

- Use derivative in the transversal direction – always possible for node-based grids
- Third-order accurate interpolation
- Only uses 2 (adjacent) cells

➢ Definition at T-junction nodes



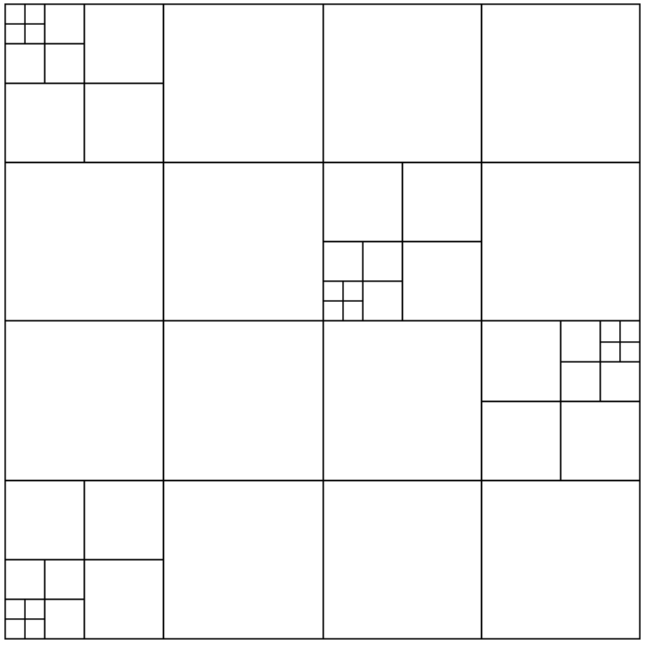
$$\phi_4^G = \frac{s_7\phi_8 + s_8\phi_7}{s_7 + s_8} - \frac{s_7 s_8}{s_3 + s_6}\left(\frac{\phi_3 - \phi_0}{s_3} + \frac{\phi_6 - \phi_0}{s_6}\right)$$

$$\phi_5^G = \frac{s_{11}s_{12}\phi_{11} + s_{11}s_9\phi_{12} + s_{10}s_{12}\phi_9 + s_9 s_{10}\phi_{10}}{(s_{10} + s_{11})(s_9 + s_{12})} - \frac{s_{10}s_{11}}{s_3 + s_6}\left(\frac{\phi_3 - \phi_0}{s_3} + \frac{\phi_6 - \phi_0}{s_6}\right) - \frac{s_9 s_{12}}{s_1 + s_4}\left(\frac{\phi_1 - \phi_0}{s_1} + \frac{\phi_4^G - \phi_0}{s_4}\right)$$

- Use derivative in the transversal direction - always possible for node-based grids
- Third-order accurate interpolation
- Only uses 3 (adjacent) cells

➢ Application to solving the Poisson equation

- ➤ Linear system non-symmetric
  - ➤ M-matrix, thus invertible
- ➤ Second-order accurate solutions in max norm
- ➤ Second-order accurate **gradient** in max norm



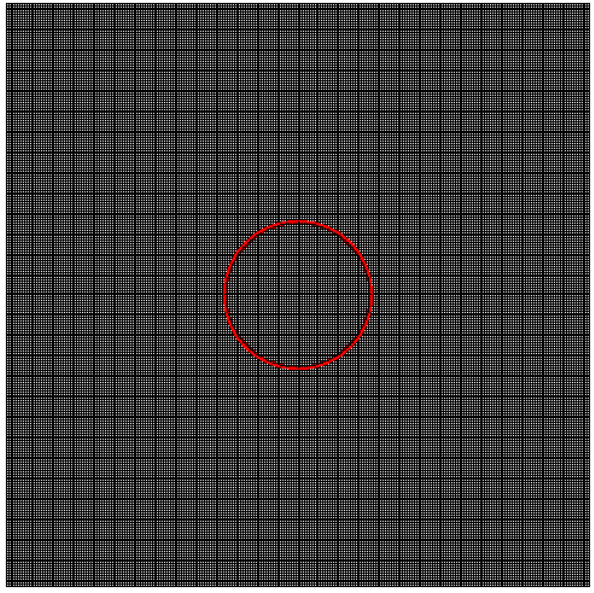
$$\Omega = [0,\pi]^2$$

$$u(x,y) = e^{-x-y}$$

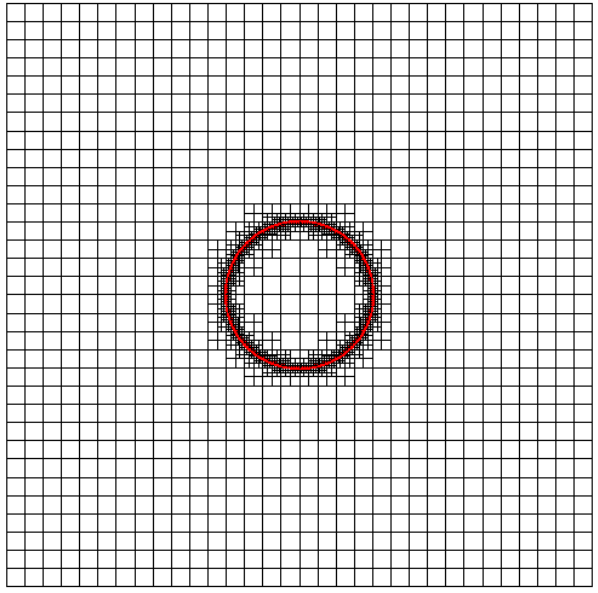
Effective Resolution	$\|\|u-u_h\|\|_\infty$	Order	$\|\|\nabla u-\nabla u_h\|\|_\infty$	order
32^2	7.04×10^{-3}	–	3.42×10^{-2}	–
64^2	1.74×10^{-3}	2.01	1.41×10^{-2}	1.28
128^2	3.97×10^{-4}	2.13	4.38×10^{-3}	1.68
256^2	9.33×10^{-5}	2.09	1.18×10^{-3}	1.89
512^2	2.26×10^{-5}	2.05	3.09×10^{-4}	1.93

➢ Application to Stefan problems

- ➤ Diffusion dominated phenomena
- ➤ Used in modeling solidification processes, epitaxial growth...
- ➤ Parabolic problem



$$V = -[D\nabla T]$$



$$\| u - u_h \| = 1.498\text{x}10^{-4}$$

$$\| u - u_h \| = 1.659\text{x}10^{-4}$$

2 to 3 orders of magnitude faster

UC SANTA BARBARA engineering

COMPUTATIONAL APPLIED SCIENCE LABORATORY

PART II:

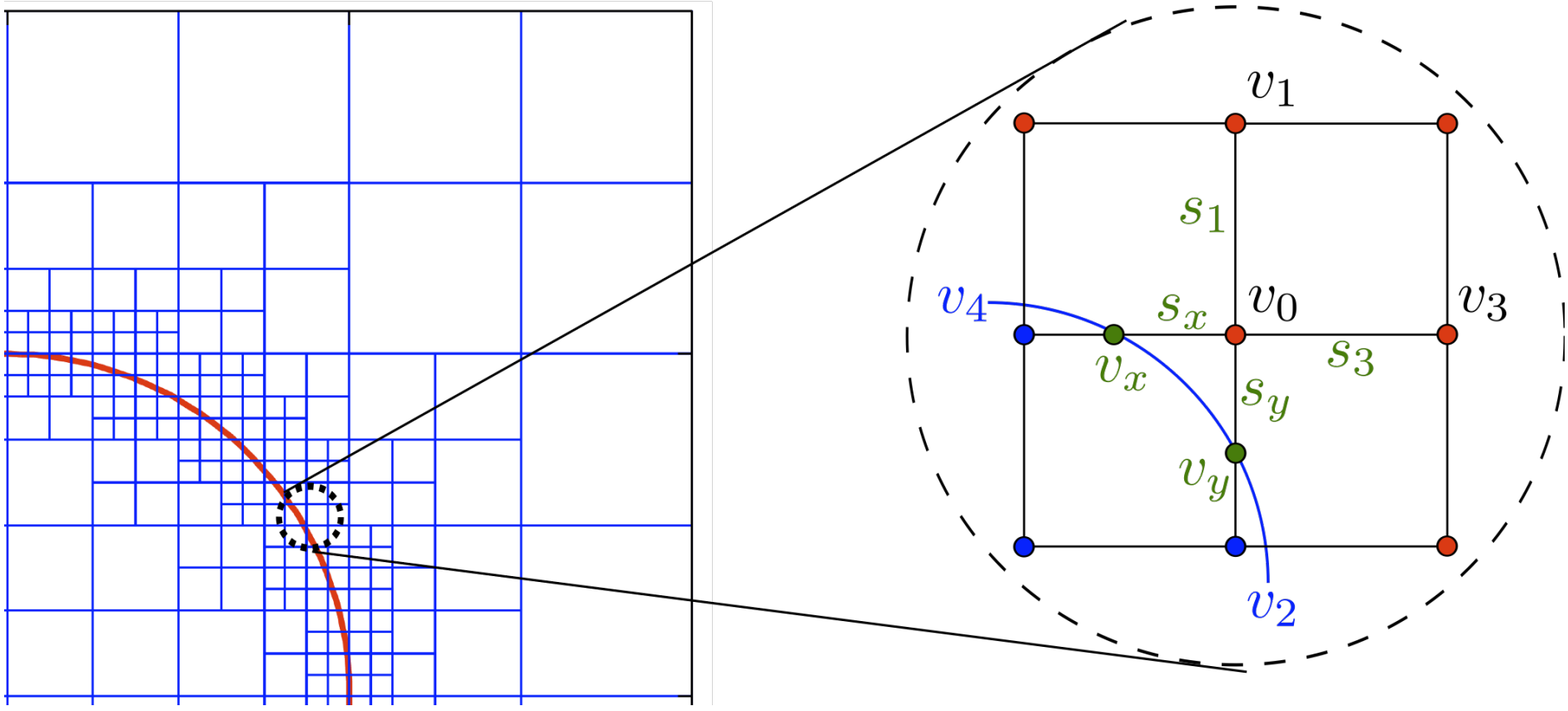
Mixed Boundary Conditions

At

Irregular and Free Boundaries

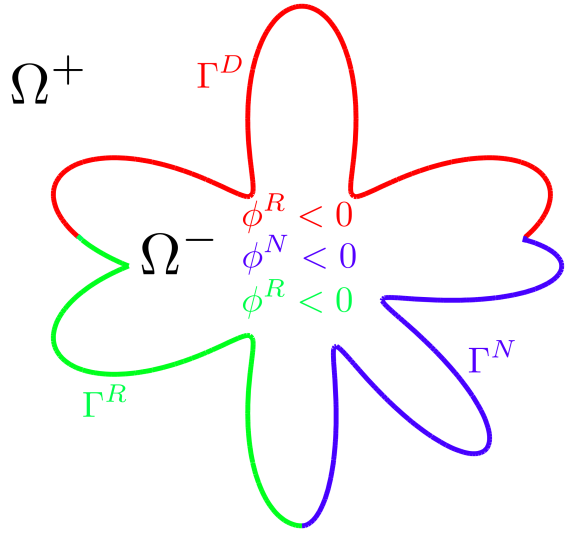
➢ Imposing Boundary Conditions

- ➤ Locally uniform near the interface

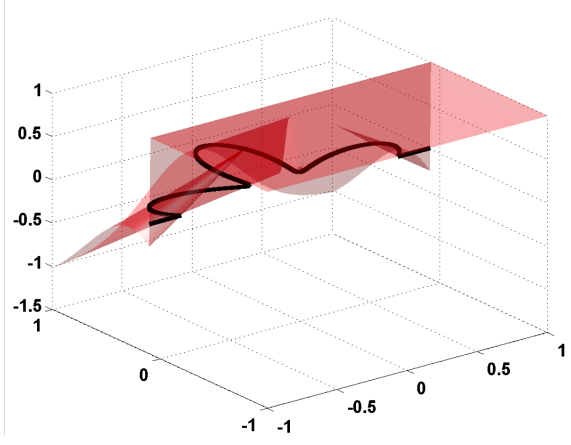


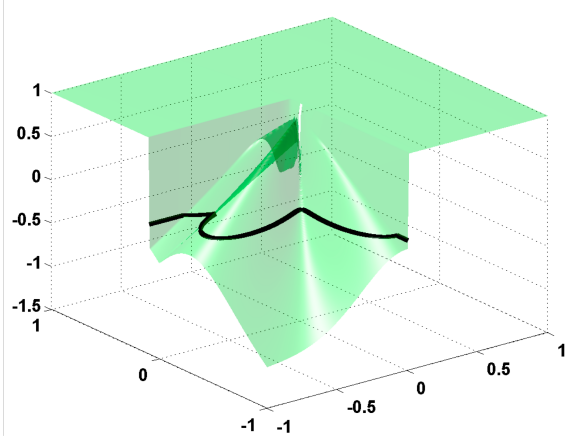
➢ Mixed boundary conditions: Dirichlet-Neumann-Robin

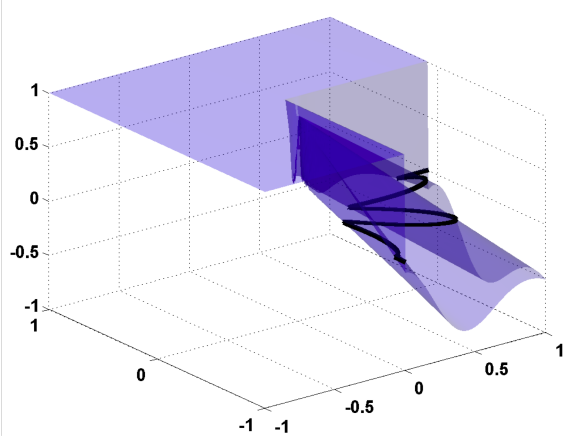
➤ Implicit representation of the domain – Applicable to moving boundaries



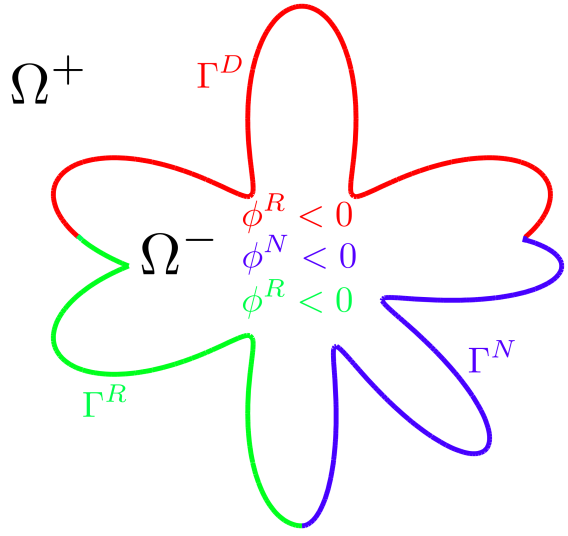
$$\Delta u = F \qquad \mathbf{x} \in \Omega^-,$$
$$u = G \qquad \text{on } \phi^D = 0,$$
$$\frac{\partial u}{\partial n} = K \qquad \text{on } \phi^N = 0,$$
$$\frac{\partial u}{\partial n} + \alpha u = M \qquad \text{on } \phi^R = 0,$$







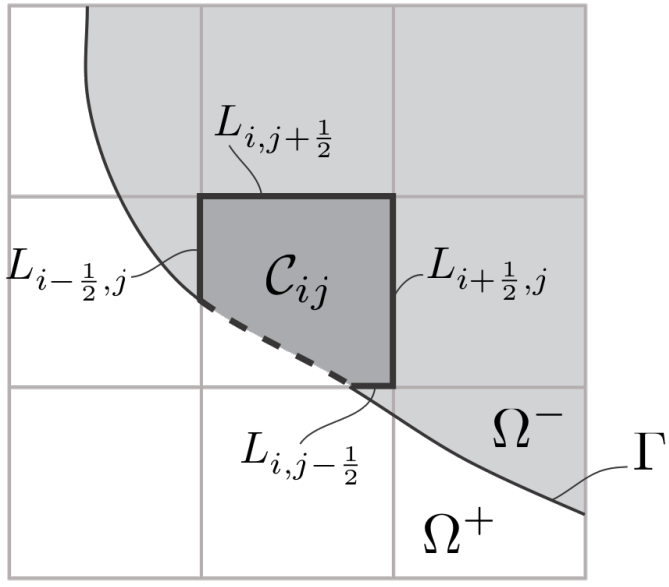
➢ Neumann and Robin boundary conditions



$$\Delta u = F, \quad \mathbf{x} \in \Omega^-,$$
$$\frac{\partial u}{\partial n} = K, \quad \mathbf{x} \in \Gamma^N,$$
$$\frac{\partial u}{\partial n} + \alpha u = M, \quad \mathbf{x} \in \Gamma^R.$$

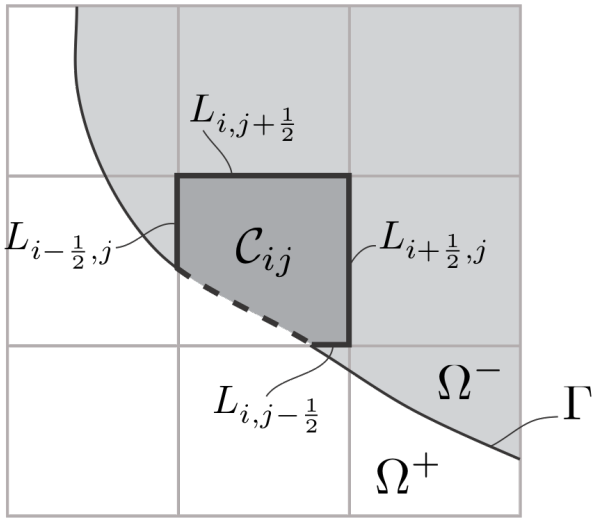
➤ Divergence theorem

$$\int_{C_{ij} \cap \Omega^-} \Delta u \; d\Omega = \int_{\partial (C_{ij} \cap \Omega^-)} n \cdot \nabla u \; d\Gamma,$$
$$= \int_{\partial C_{ij} \cap \Omega^-} n \cdot \nabla u \; d\Gamma + \int_{\Gamma \cap \Omega^-} n \cdot \nabla u \; d\Gamma$$

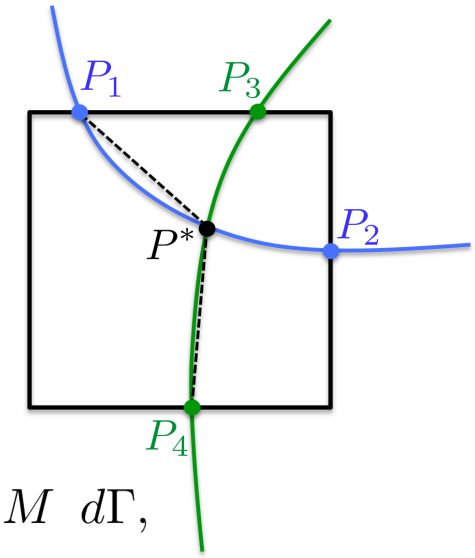


➢ Neumann and Robin boundary conditions

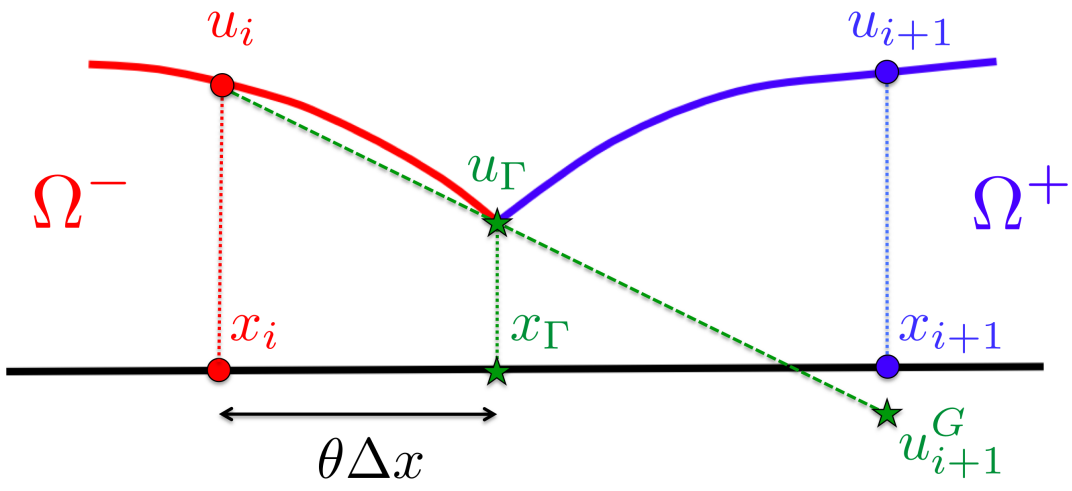
$$\Delta u = F, \quad \mathbf{x} \in \Omega^-,$$
$$\frac{\partial u}{\partial n} = K, \quad \mathbf{x} \in \Gamma^N,$$
$$\frac{\partial u}{\partial n} + \alpha u = M, \quad \mathbf{x} \in \Gamma^R.$$



$$\int_{\partial(C_{ij}\cap\Omega^-)} n\cdot\nabla u \; d\Gamma \simeq L_{i+\frac{1}{2},j}\frac{u_{i+1,j}-u_{i,j}}{\Delta x} - L_{i-\frac{1}{2},j}\frac{u_{i,j}-u_{i-1,j}}{\Delta x} + L_{i,j+\frac{1}{2}}\frac{u_{i,j+1}-u_{i,j}}{\Delta y} - L_{i,j-\frac{1}{2}}\frac{u_{i,j}-u_{i,j-1}}{\Delta y} - \alpha u_{i,j}\int_{C_{ij}\cap\Gamma^R} d\Gamma + \int_{C_{ij}\cap\Gamma^N} K \; d\Gamma + \int_{C_{ij}\cap\Gamma^R} M \; d\Gamma,$$



➢ Dirichlet boundary conditions



$$\int_{\partial(C_{ij}\cap\Omega^-)} n\cdot\nabla u \; d\Gamma \simeq L_{i+\frac{1}{2},j}\frac{u_{i+1,j}-u_{i,j}}{\Delta x} \quad \longleftarrow \quad L_{i+\frac{1}{2},j}\frac{u^G_{i+1}-u_{i,j}}{\theta\Delta x}$$

$$- L_{i-\frac{1}{2},j}\frac{u_{i,j}-u_{i-1,j}}{\Delta x}$$

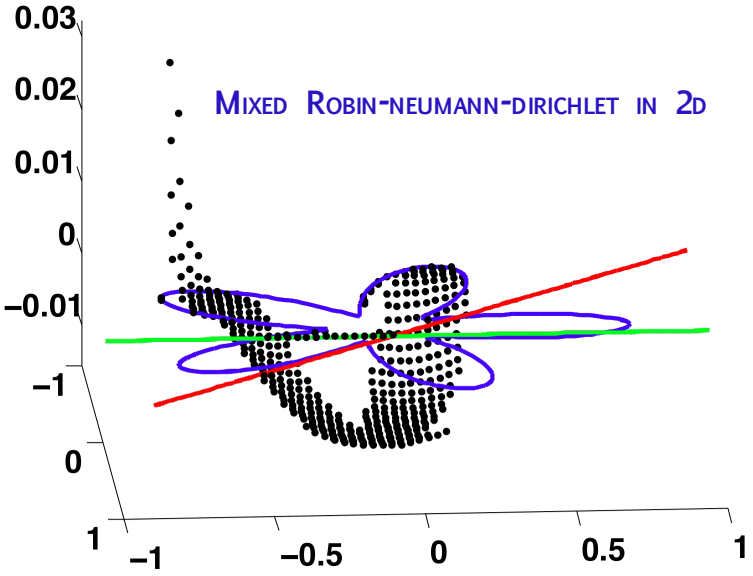
$$+ L_{i,j+\frac{1}{2}}\frac{u_{i,j+1}-u_{i,j}}{\Delta y}$$

$$- L_{i,j-\frac{1}{2}}\frac{u_{i,j}-u_{i,j-1}}{\Delta y}$$

$$- \alpha u_{i,j}\int_{C_{ij}\cap\Gamma^R} d\Gamma + \int_{C_{ij}\cap\Gamma^N} K \; d\Gamma + \int_{C_{ij}\cap\Gamma^R} M \; d\Gamma,$$

➢ Main features

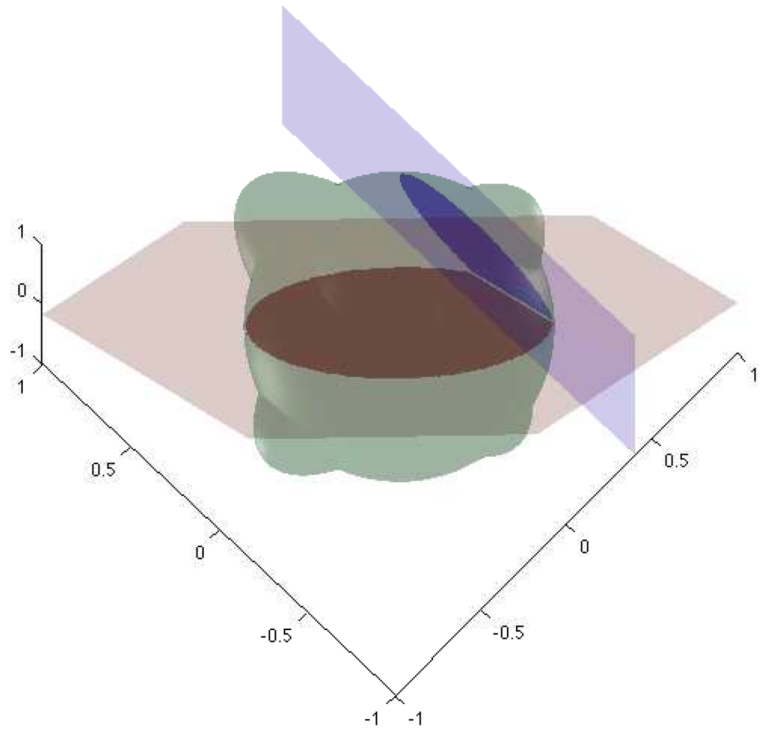
- ➤ SDP – Guarantees stability – Fast solvers
- ➤ 2^{nd} order accurate in the max-norm



MIXED ROBIN-NEUMANN-DIRICHLET IN 2D

Resolution	$\|\|u - u_h\|\|_\infty$	Order
32^2	1.67×10^{-3}	—
64^2	4.24×10^{-4}	1.98
128^2	1.18×10^{-4}	1.84
256^2	2.89×10^{-5}	2.03

MIXED ROBIN-NEUMANN-DIRICHLET IN 3D



Resolution	$\|\|u - u_h\|\|_\infty$	Order
16^3	5.74×10^{-2}	—
32^3	1.66×10^{-2}	1.79
64^3	4.42×10^{-3}	1.91
128^3	1.06×10^{-3}	2.06

UC SANTA BARBARA engineering

COMPUTATIONAL APPLIED SCIENCE LABORATORY

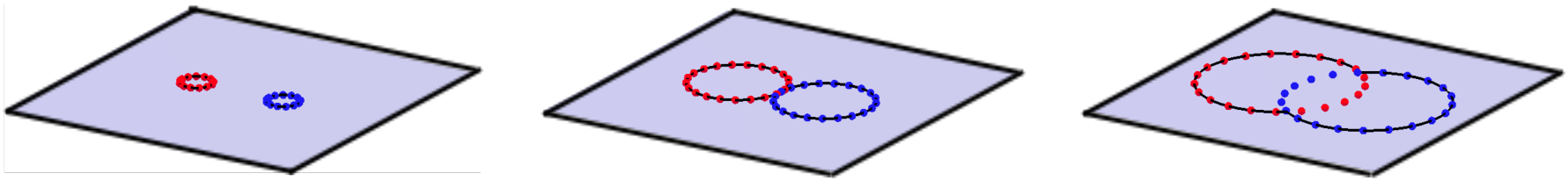
PART III:

LEVEL-SET TECHNOLOGY

CURVATURES

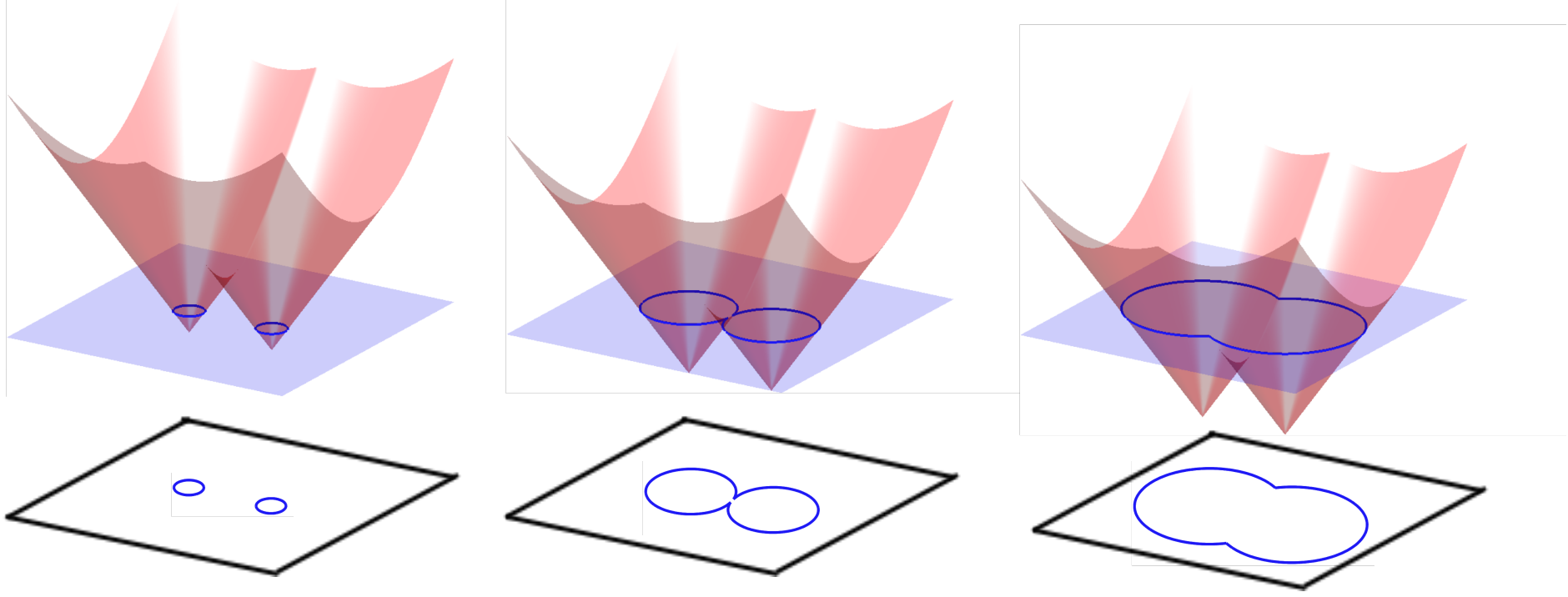
GRAIN GROWTH

➢ Front-Tracking (Explicit)



Accurate but changes in topology are difficult to handle

➢ Level-Set (Implicit)



Automatic changes in topology but less accurate

➢ Level-set evolution equation

$$\phi_t + V_n|\nabla\phi| = 0$$

➤ Method of lines

$$\frac{d\phi}{dt} + V_n \cdot H_G(\phi) = 0$$

➤ Godunov approximation of the Hamiltonian

$$H_G(\phi) = \begin{cases} \sqrt{\max\left(|(D_x^+\phi)^-|^2, |(D_x^-\phi)^+|^2\right) + \max\left(|(D_y^+\phi)^-|^2, |(D_y^-\phi)^+|^2\right)} & \text{if } V_n > 0 \\ \sqrt{\max\left(|(D_x^+\phi)^+|^2, |(D_x^-\phi)^-|^2\right) + \max\left(|(D_y^+\phi)^+|^2, |(D_y^-\phi)^-|^2\right)} & \text{otherwise} \end{cases}$$

➤ Time evolution: TVD-RK schemes

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} + V_n \cdot H_G(\phi^n) = 0$$

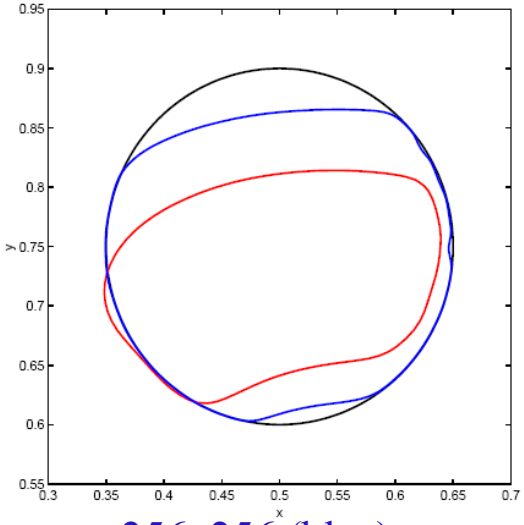
$$\frac{\phi^{n+2} - \phi^{n+1}}{\Delta t} + V_n \cdot H_G(\phi^{n+1}) = 0$$

$$\phi^{n+1} = \frac{\phi^n + \phi^{n+2}}{2}$$

➢ Application to the level-set method

➤ Loss of volume in level-set methods

T = 4

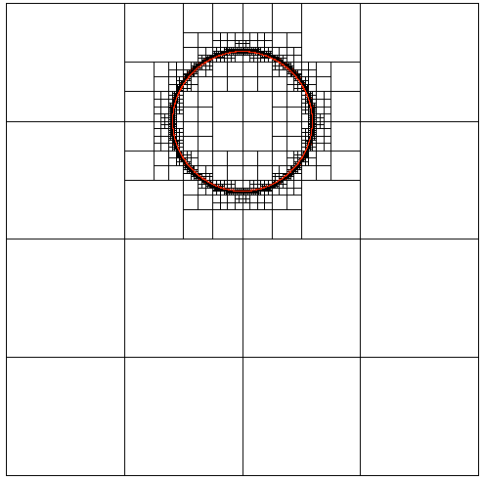


256x256 (blue)

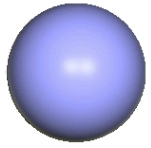
128x128 (red)

64x64 (green)

T = 6



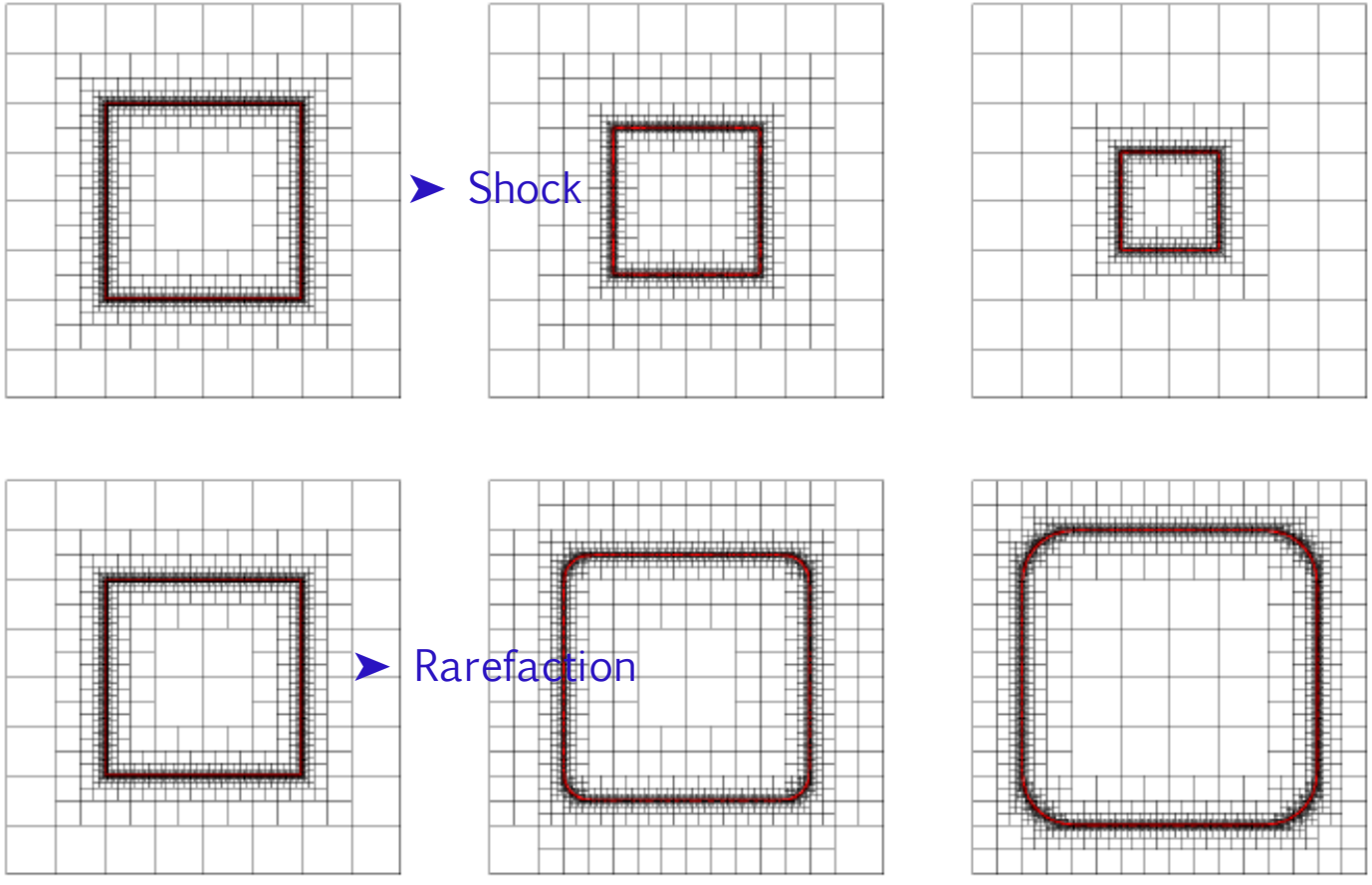
ADAPTIVE 2D



ADAPTIVE 3D

➤ Adaptivity balances volume conservation with CPU efficiency

➢ Nonlinear effects

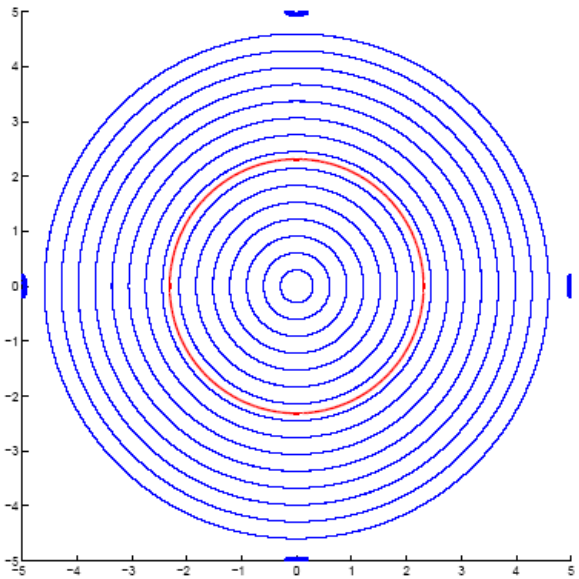


➢ Computation of curvatures

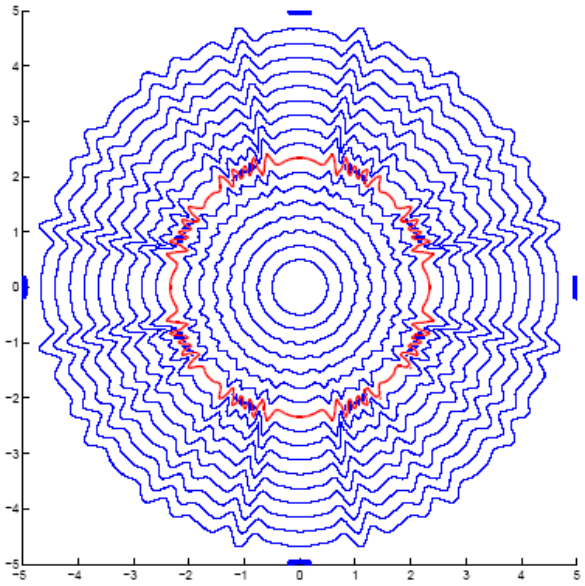
➤ Implicit representation

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|} \qquad \kappa = \nabla \cdot \mathbf{n}$$

➤ Computations are sensitive to noise



EXACT



NUMERICAL

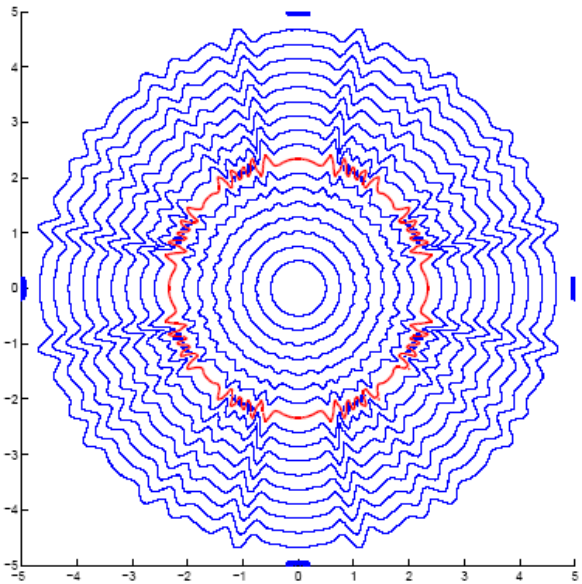
➢ Computation of curvatures

➤ Reinitialization equation

$$\phi_t + \mathrm{Sign}(\phi)|\nabla\phi| = 1$$

➤ HJ-WENO scheme --> Fifth order accurate

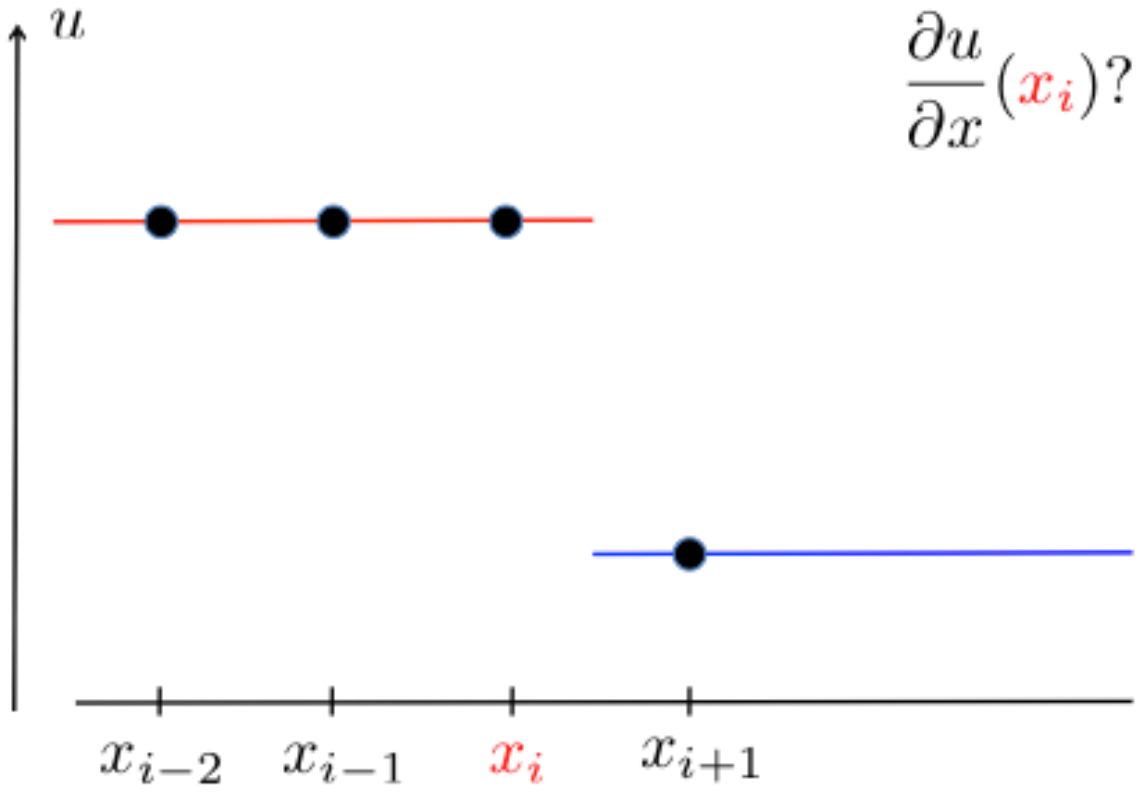
➤ Gibou-Fedkiw (JCP 2005) --> Show that it is second-order accurate



NUMERICAL

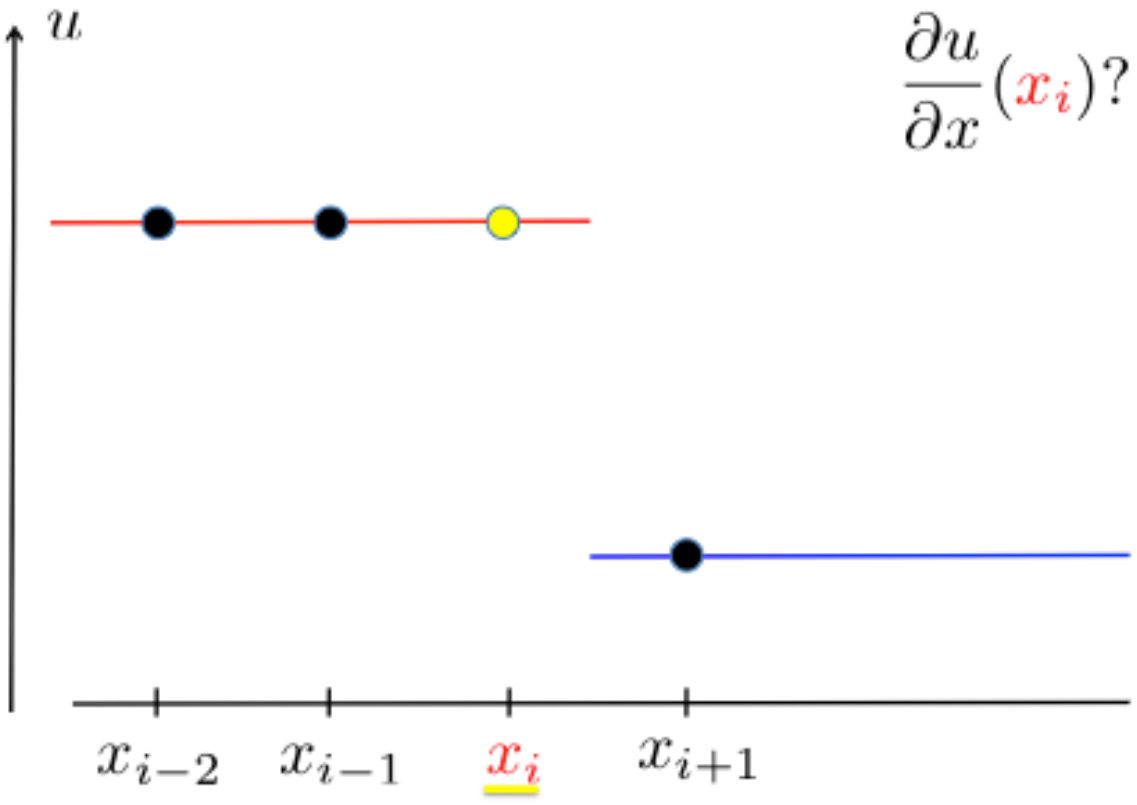
➢ Application to solving the Navier-Stokes equations

➤ Stability enforced at the discrete level



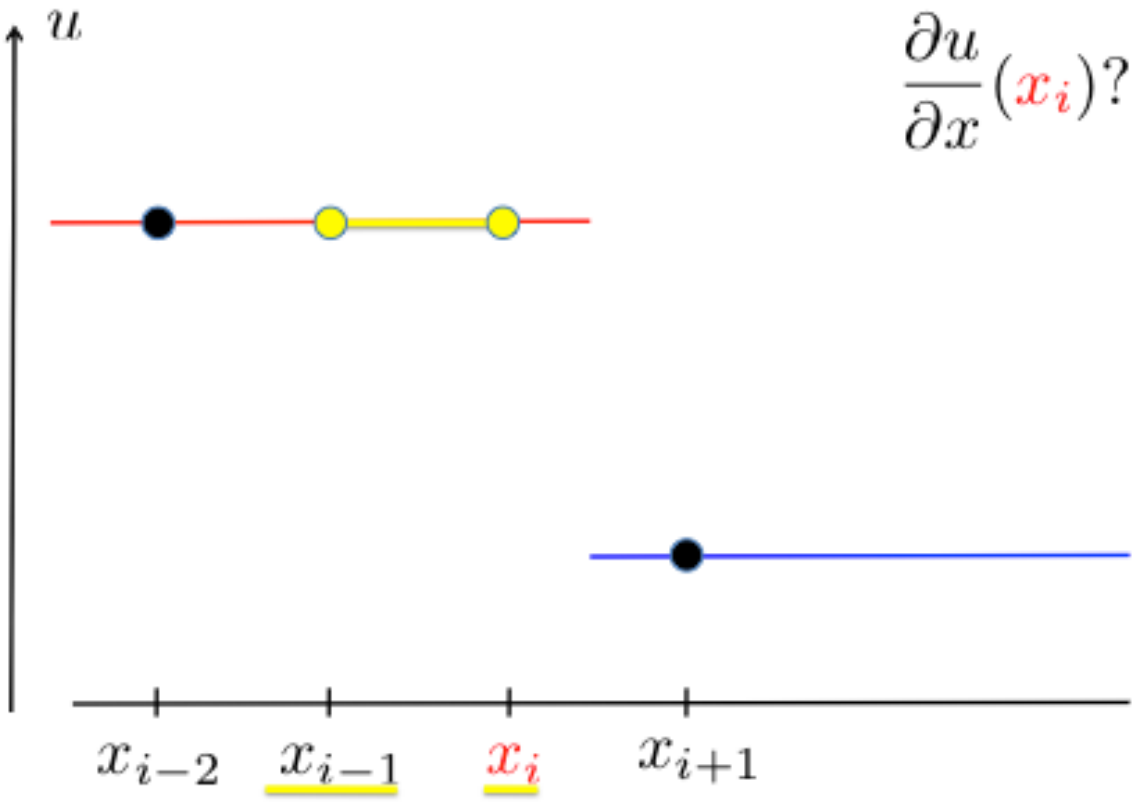
➢ Application to solving the Navier-Stokes equations

➤ Stability enforced at the discrete level



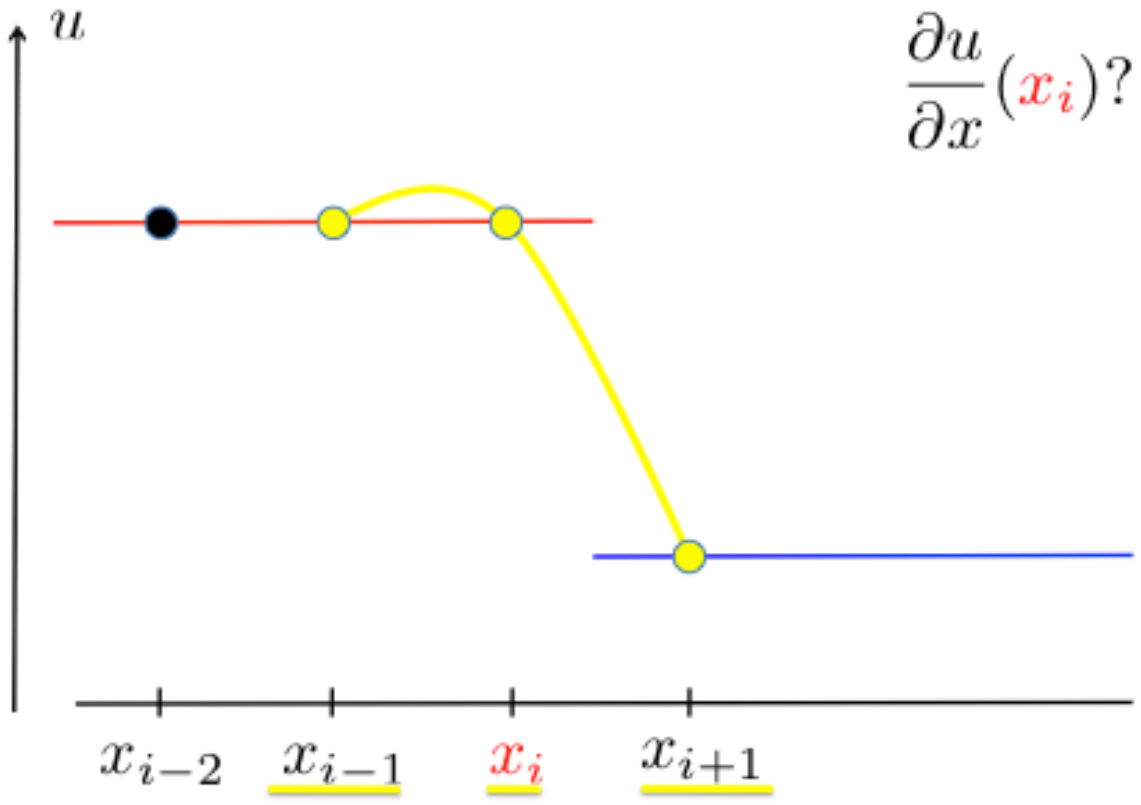
➢ Application to solving the Navier-Stokes equations

➤ Stability enforced at the discrete level



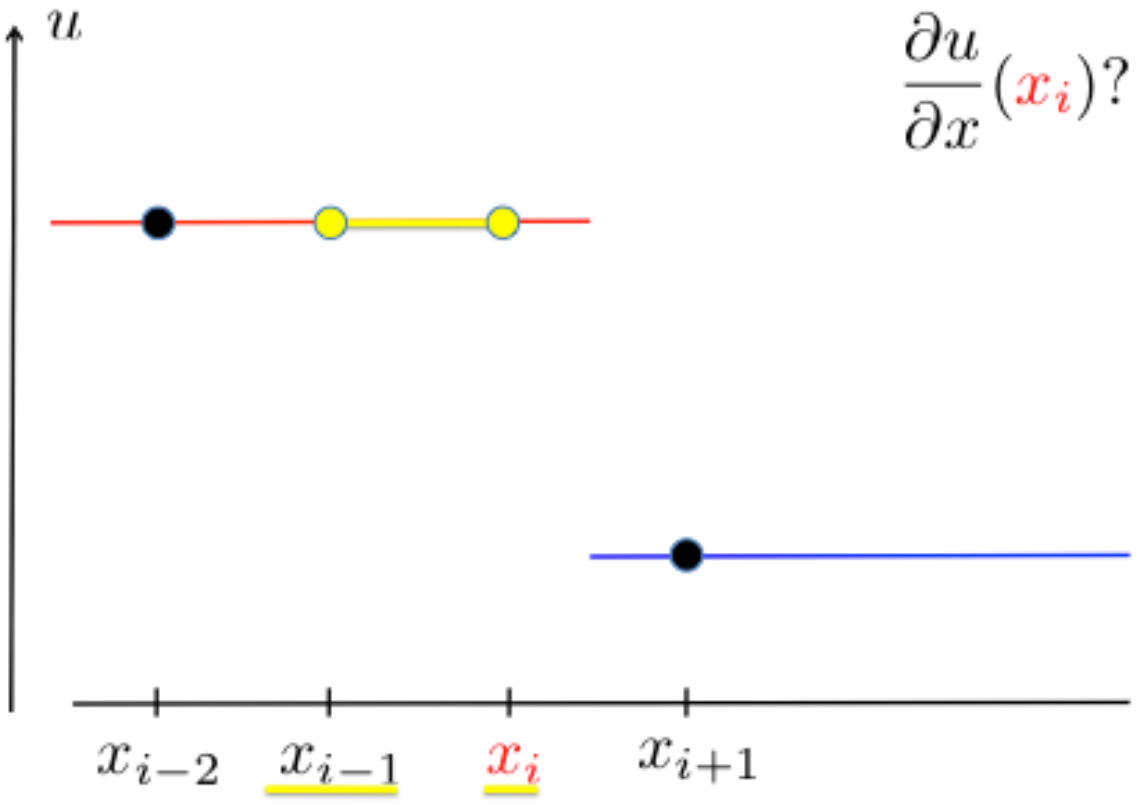
➢ Application to solving the Navier-Stokes equations

➤ Stability enforced at the discrete level



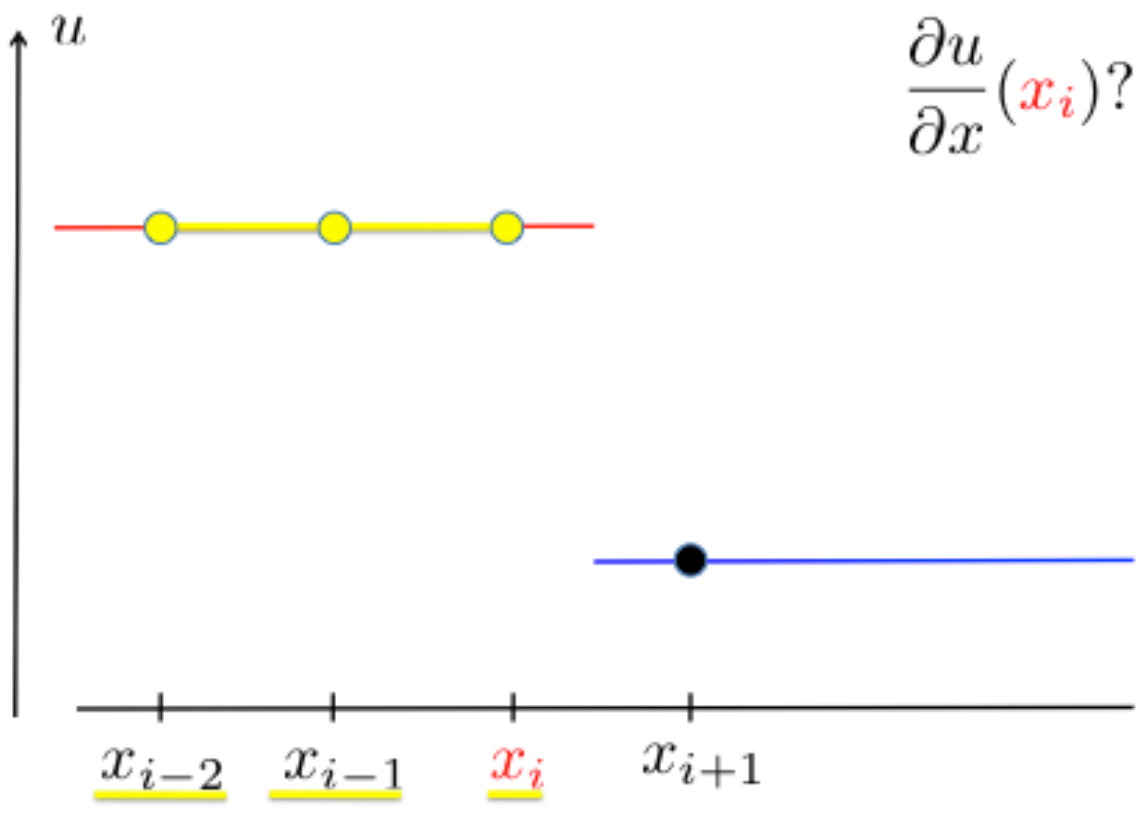
➢ Application to solving the Navier-Stokes equations

➤ Stability enforced at the discrete level



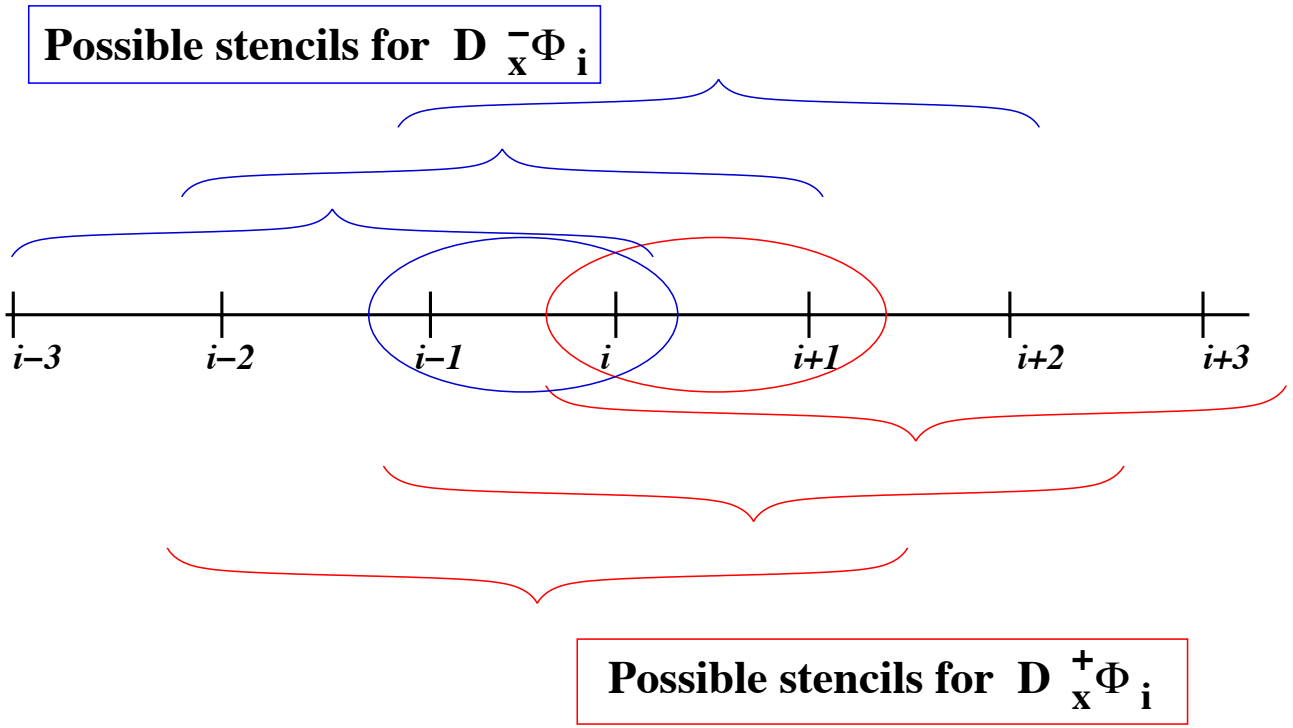
➢ Application to solving the Navier-Stokes equations

➤ Stability enforced at the discrete level



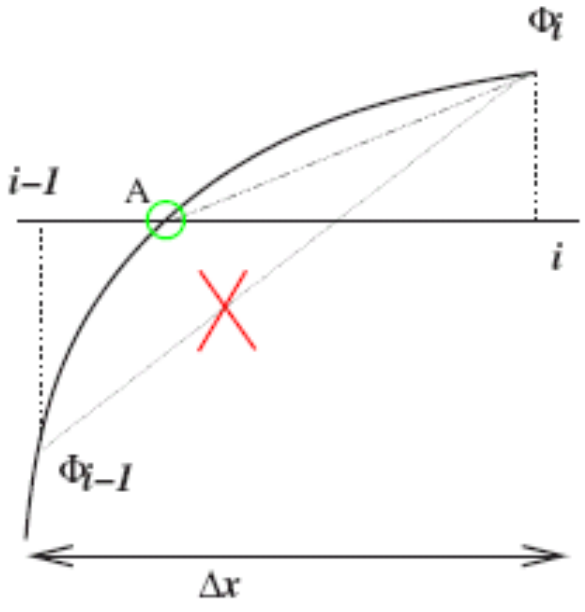
➢ Application to solving the Navier-Stokes equations

➤ Stability enforced at the discrete level

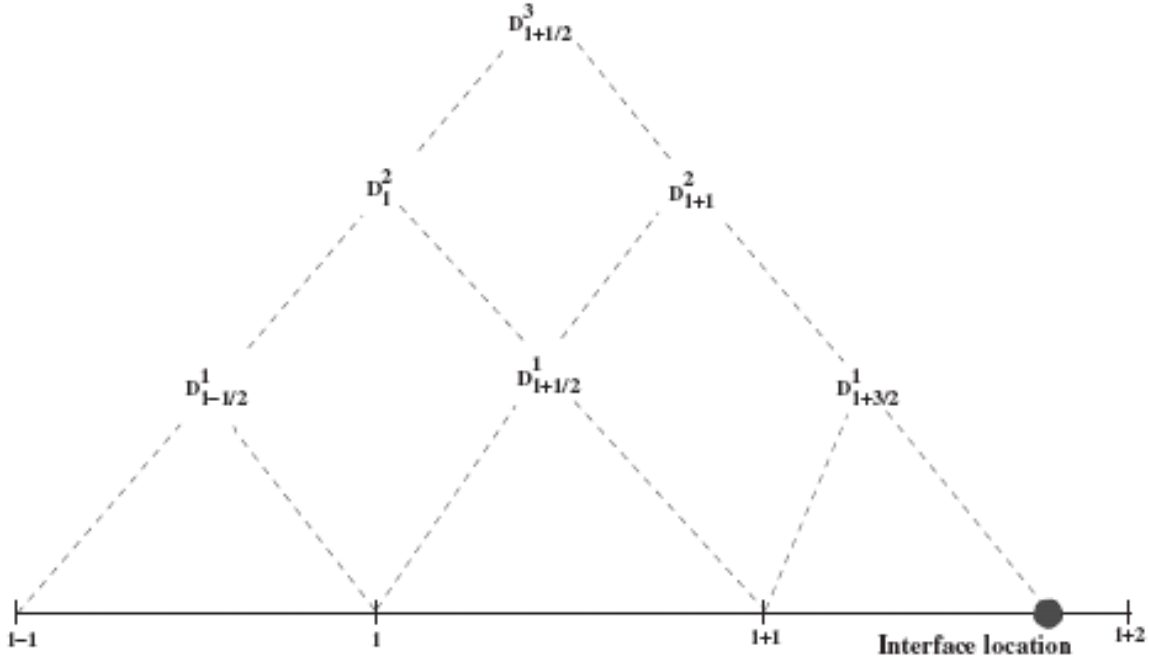


➢ Computation of curvatures

- ➤ Rarefaction wave solution (DuChene-Gibou - JSC 2009)
- ➤ Fourth-order accurate reinitialization equation
- ➤ Second-order accurate curvatures



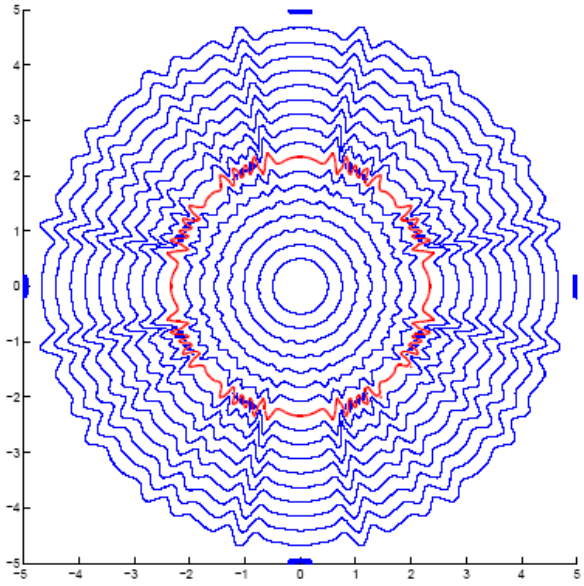
RAREFACTION WAVE



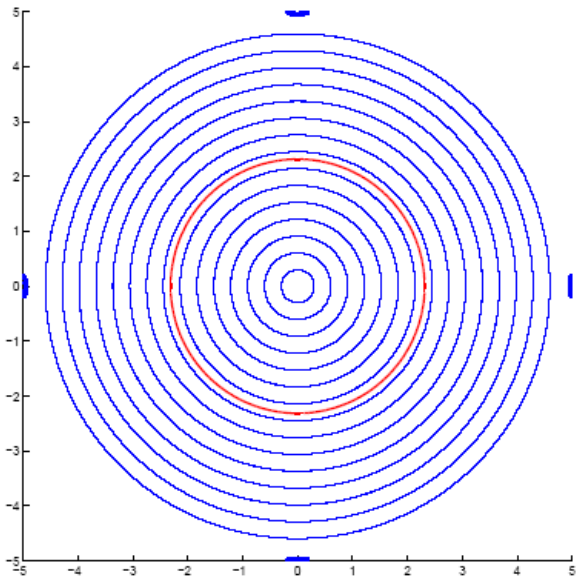
HJ-ENO CONSTRUCTIONS

➢ Computation of curvatures

- ➤ DuChene Gibou (JSC 2009)
- ➤ Fourth-order accurate reinitialization equation
- ➤ Second-order accurate curvatures



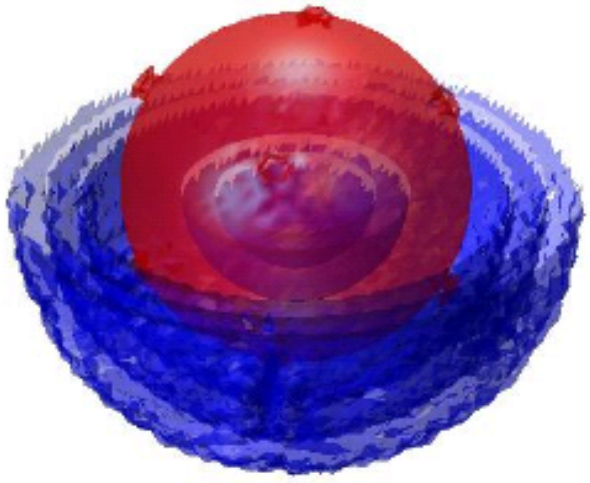
STANDARD LEVEL-SET METHOD



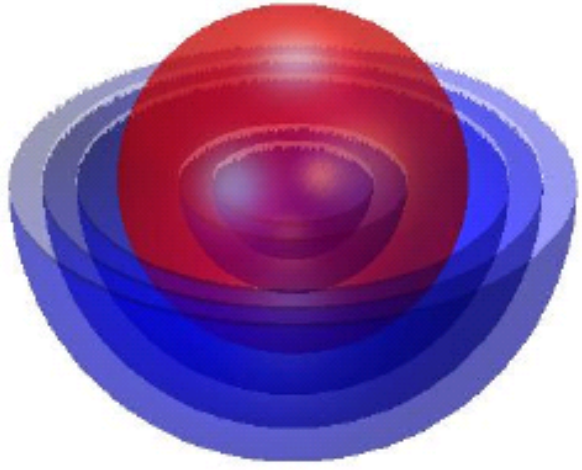
DUCHENE-GIBOU

➢ Computation of curvatures

- ➤ DuChene Gibou (JSC 2009)
- ➤ Fourth-order accurate reinitialization equation
- ➤ Second-order accurate curvatures



STANDARD LEVEL-SET METHOD

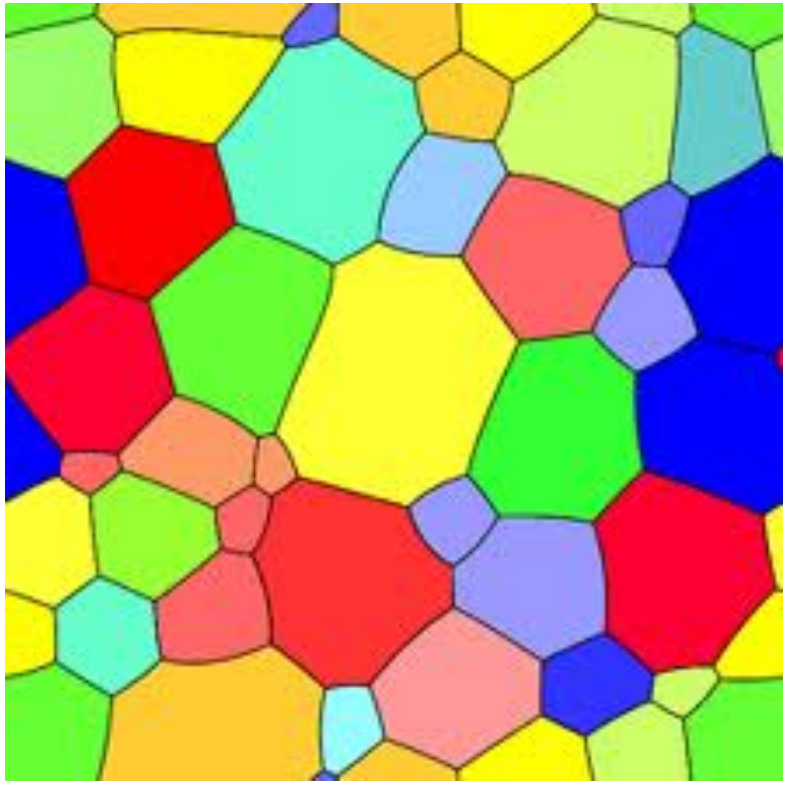


DUCHENE-GIBOU

Grain Growth

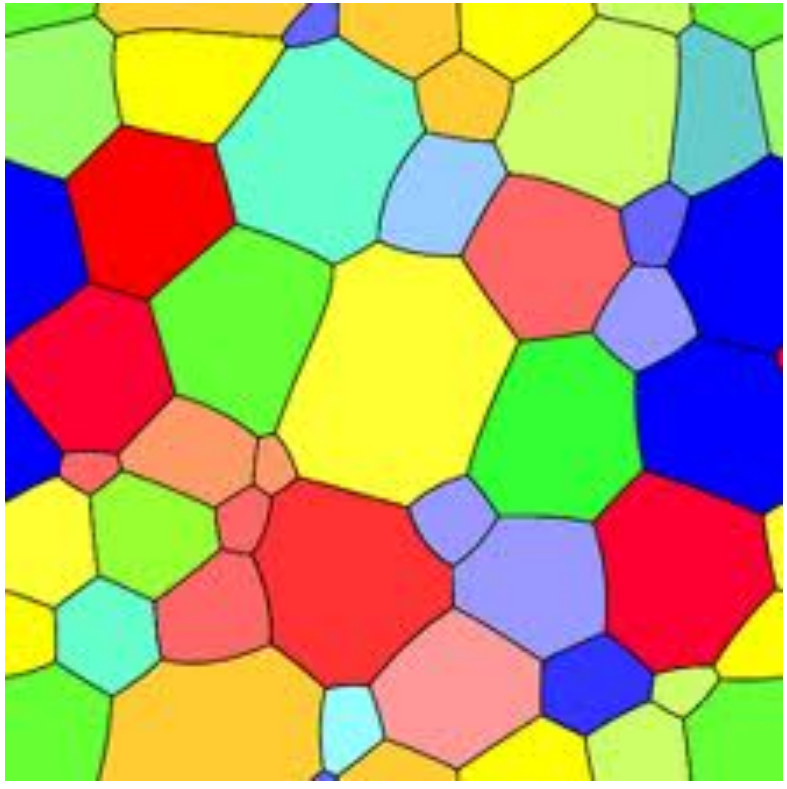
- Occurs during heat treatment of polycrystalline materials
- The grain boundaries evolve to reduce the total free energy
- Minimize the total area
- Triple junctions
- Driving force ~ curvature

$$f = \mu\gamma(\theta)\kappa$$



Grain Growth

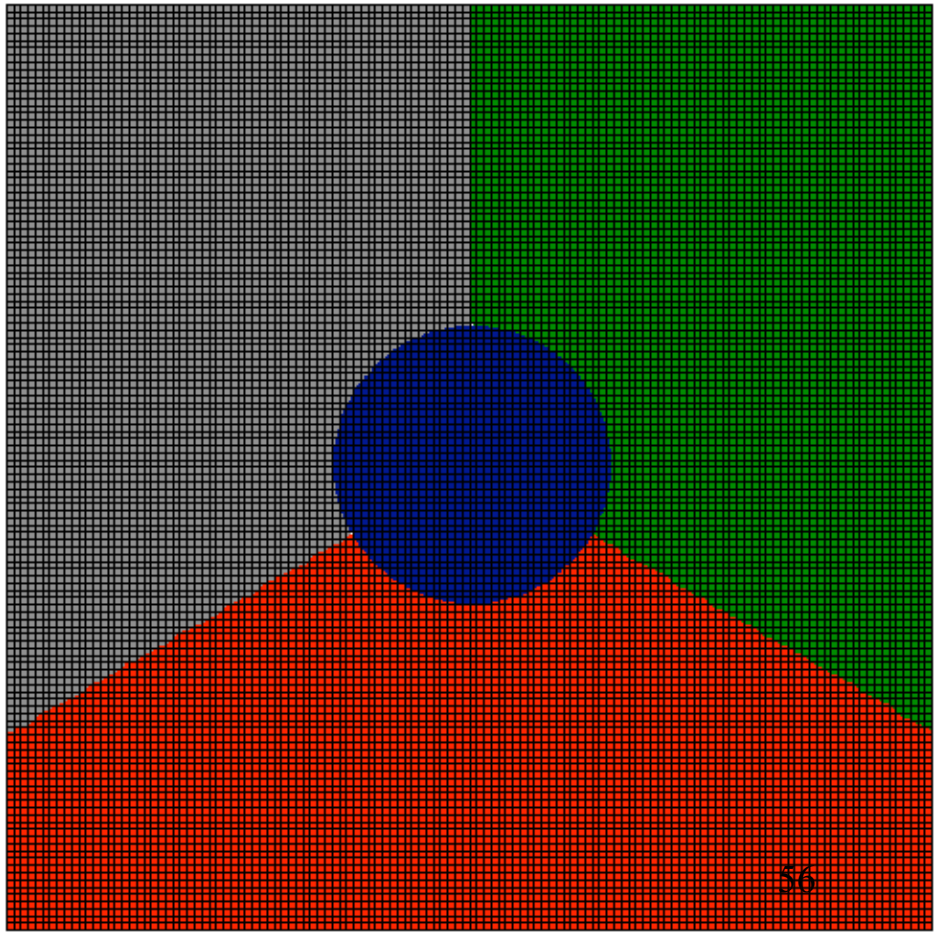
- Level-Set treatment possible
  - Uniform grids (local level-set?)
  - N level-sets --> N phases
- Ongoing effort
  - Adaptive grids
  - $O(10)$ level-sets --> $O(10^6)$ phases



Grain Growth - 2D

- 2 level-set functions represent 4 grains
- Exterior grains cause interior grains to shrink

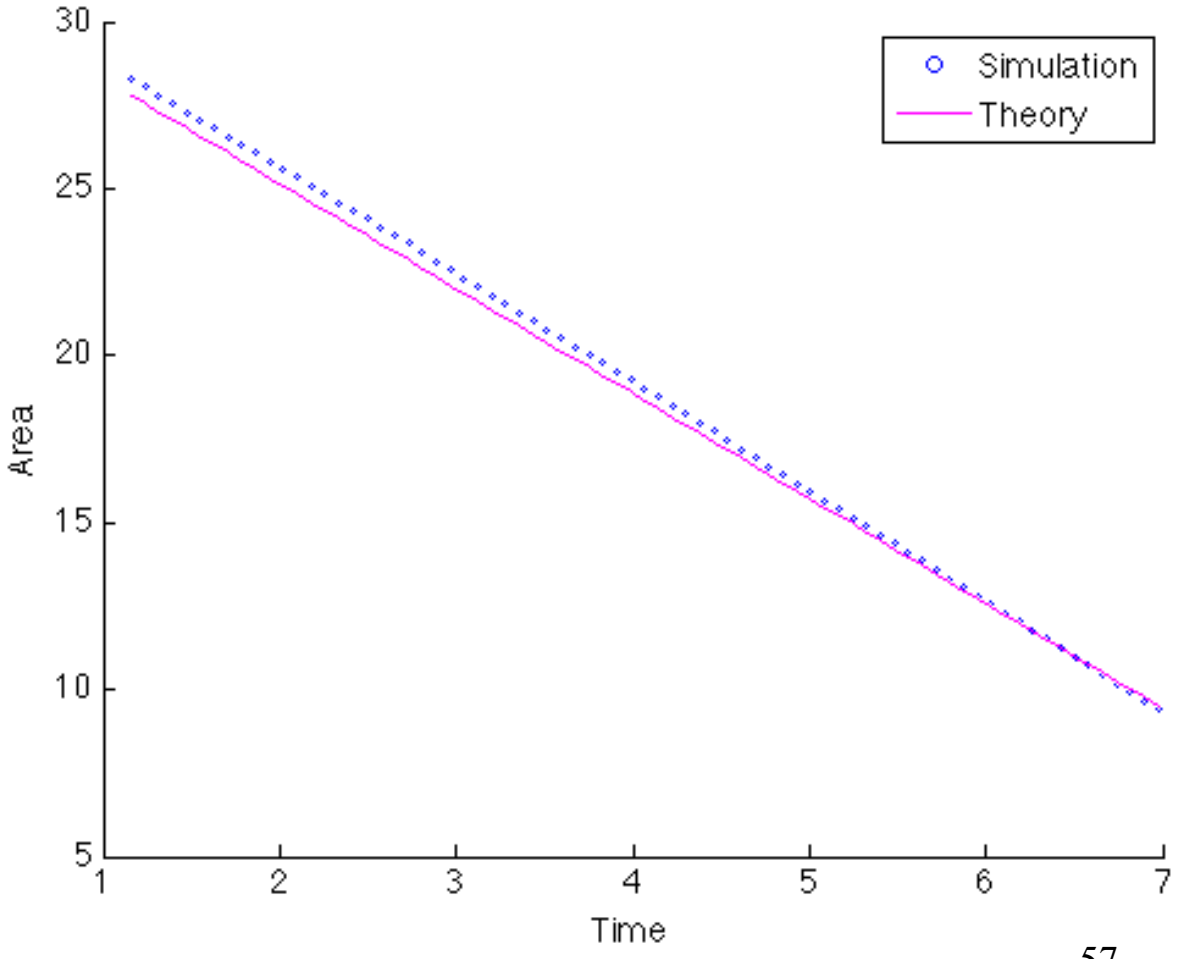
$$\frac{dA}{dt} = 2\pi\gamma\left(\frac{3}{6} - 1\right)$$



➢ Grain Growth - 2D

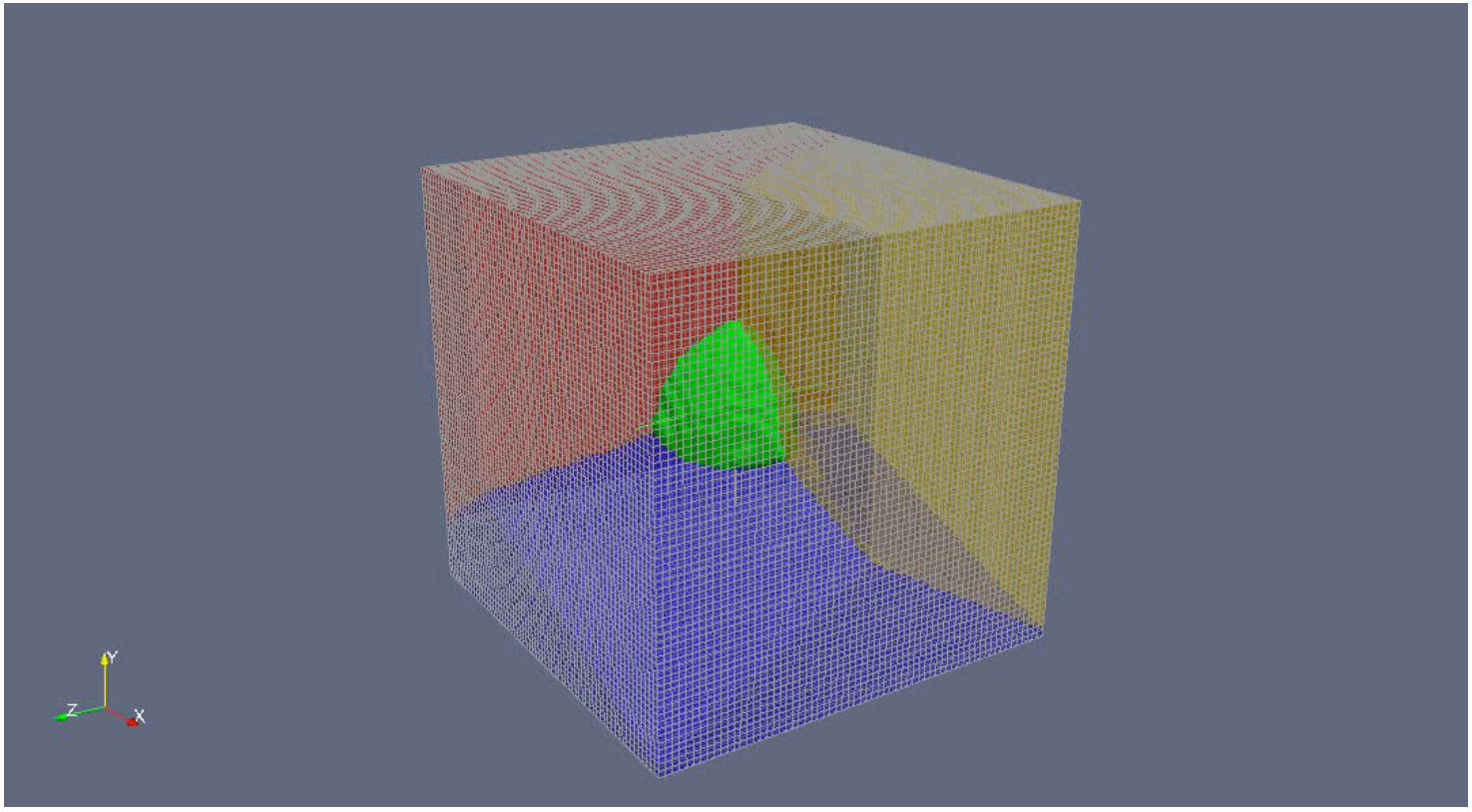
- ➤ Comparison to Von-Neumann/Mullins law
- ➤ Error in slope ~ 3.5%
- ➤ Max level 6

$$\frac{dA}{dt} = 2\pi\gamma\left(\frac{3}{6} - 1\right)$$



➢ Grain Growth - 3D

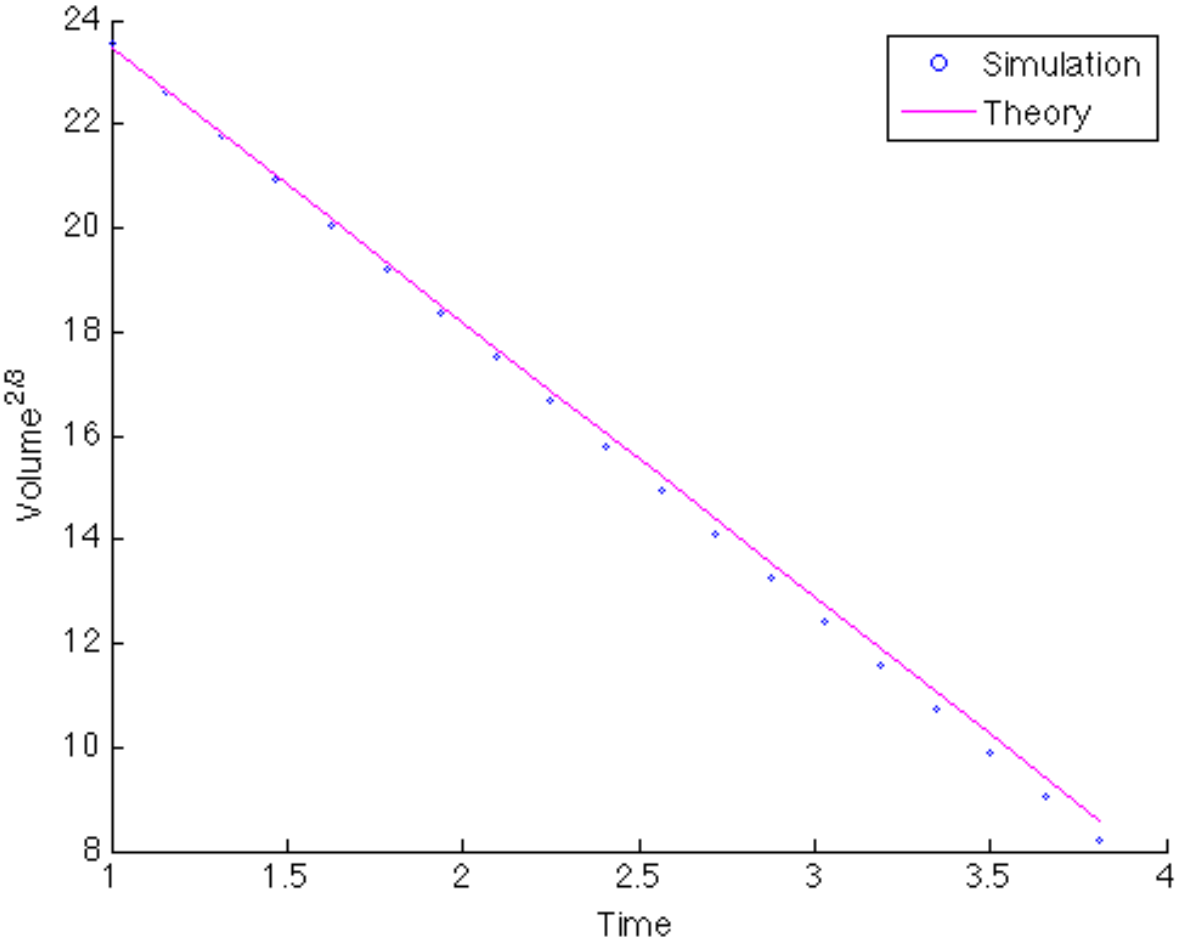
- ➤ 3 level-set functions represent 5 grains
- ➤ One grain cut away in video so inner grain is visible
- ➤ Reuleaux tetrahedron satisfies $\frac{d}{dt}V^{2/3} \approx 5.2885$



➢ Grain Growth - 3D

- ➤ Comparison to Theory
- ➤ Error in slope ~ 2.9%
- ➤ Max level 6

$$\frac{d}{dt}V^{2/3} \approx 5.2885$$



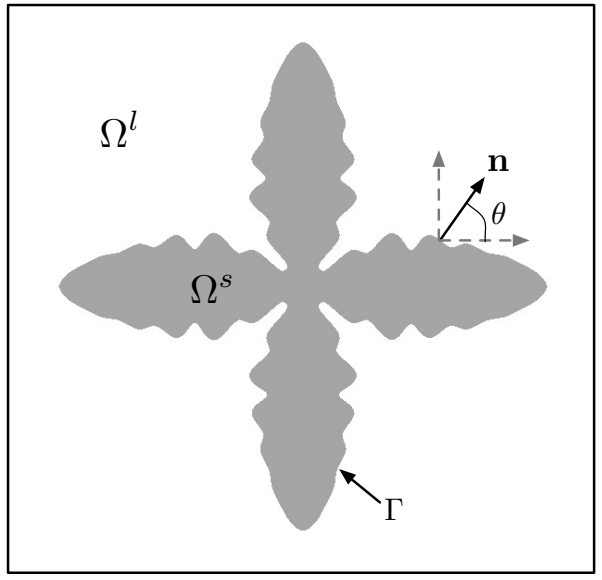
PART IV:

Application to Solidification of Binary Alloys

➢ Application to the Solidification of Binary Alloys

➤ Diffusion equations:

$$\begin{cases} \partial T/\partial t = \lambda \Delta T & \forall \boldsymbol{x} \in \Omega, \\ \partial C^l/\partial t = D^l \Delta C^l & \forall \boldsymbol{x} \in \Omega^l, \\ \partial C^s/\partial t = D^s \Delta C^s & \forall \boldsymbol{x} \in \Omega^s, \end{cases}$$



➤ Diffusion in the solid phase can be neglected $(D^s << D^l)$

➤ Convection effects are neglected here

Interface Conditions

- Continuity of the temperature field

$$[T]_{|\Gamma} = 0$$

- Heat flux balance

$$V_n = \frac{k}{L}\left[\frac{\partial T}{\partial n}\right]_{\Gamma} = \frac{k}{L}\left(\frac{\partial T^s}{\partial n} - \frac{\partial T^l}{\partial n}\right)$$

- Gibbs-Thompson relations

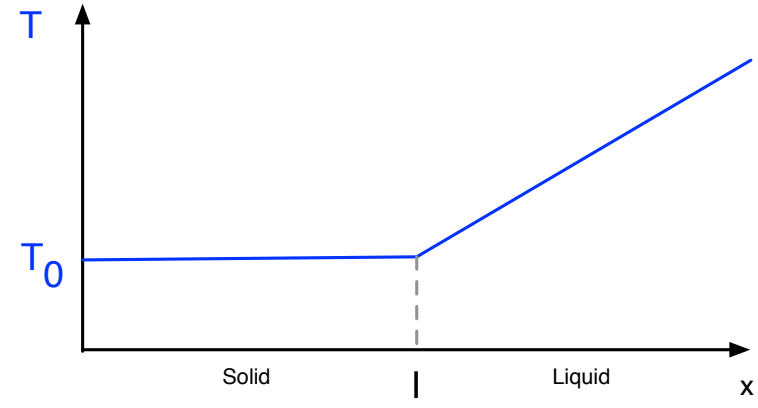
$$T_{|\Gamma} = T_m + m_L C^l + \epsilon_c(\theta)\kappa + \epsilon_v V_n$$

- Solute rejection equation

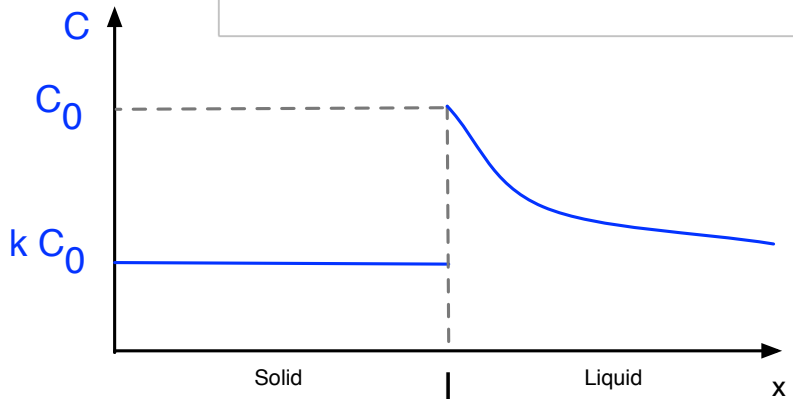
$$V_n(C^s - C^l) = D^s \frac{\partial C^s}{\partial n} - D^l \frac{\partial C^l}{\partial n}.$$

- Jump in the concentration

$$C^s = k_p C^l$$



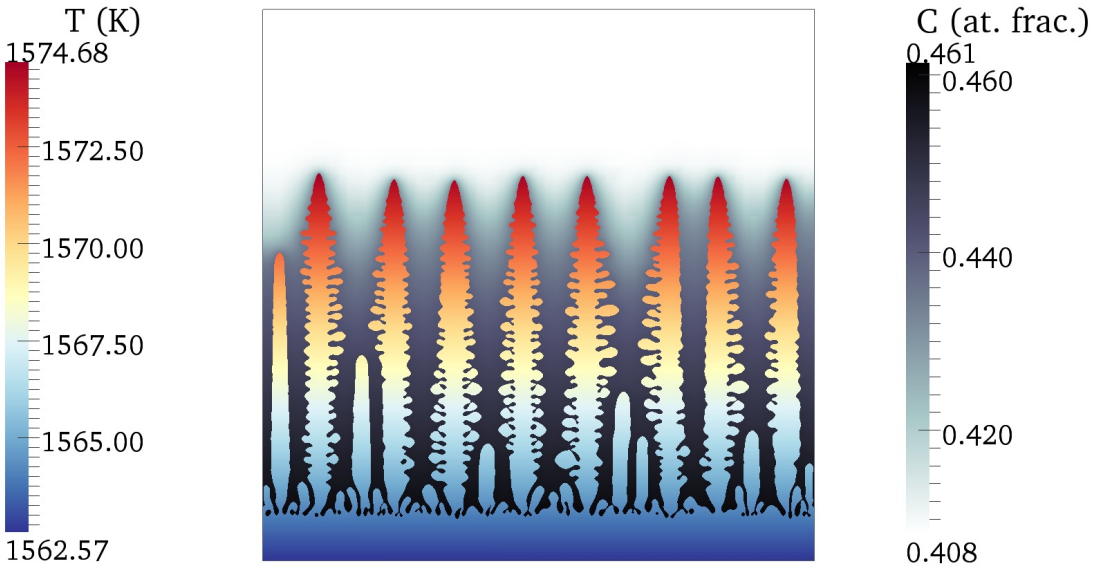
TEMPERATURE PROFILE NEAR INTERFACE

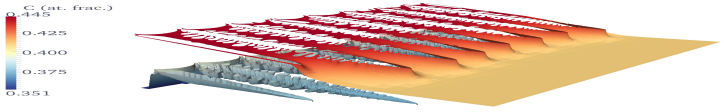


CONCENTRATION PROFILE NEAR INTERFACE

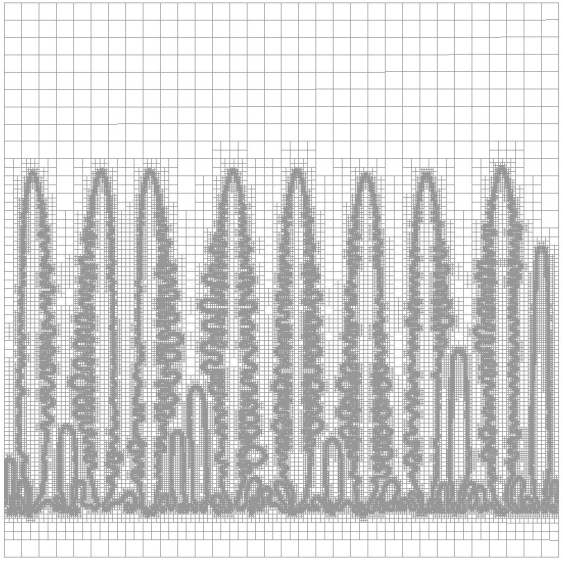
➢ Application to the Solidification of Binary Alloys

- Ni-Cu
- Ignore Convective Effects





SHARP CONCENTRATION PROFILE



- Solving for the temperature field:
  - Problem:

$$\begin{aligned} \frac{\partial T}{\partial t} &= \lambda \Delta T & x \in \Omega \\ [T] &= 0 & x \in \Gamma \\ \left[\frac{\partial T}{\partial n}\right] &= \frac{L V_N}{k} & x \in \Gamma \end{aligned}$$

  - Time Discretization:

$$\frac{T^{n+1} - T^n}{\Delta t} = \lambda \frac{\Delta T^{n+1} + \Delta T^n}{2}$$

    - Special treatment is needed close to the interface

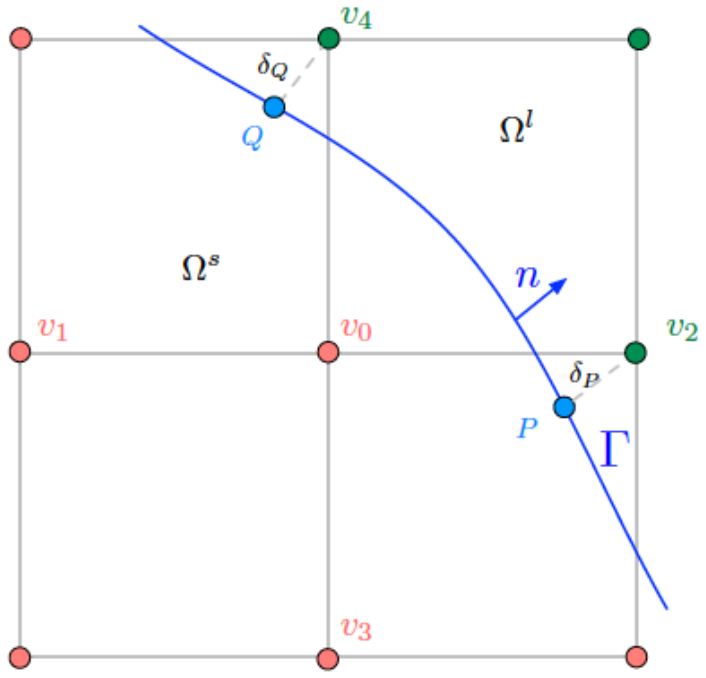
- Solving for the temperature field:
  - Imposing the jump conditions:

$$\Delta T_0 = \frac{T_2^s - 2T_0 + T_1}{\Delta x^2} + \frac{T_4^s - 2T_0 + T_3}{\Delta y^2} + O\left(h^2\right)$$

$$T_2^s = T_2 + [T]_P + \delta_P \left[\frac{\partial T}{\partial n}\right]_P + O\left(h^2\right)$$

$$T_4^s = T_4 + [T]_Q + \delta_Q \left[\frac{\partial T}{\partial n}\right]_Q + O\left(h^2\right)$$

$$\Delta T_0 = \widetilde{\Delta T_0} + \frac{\delta_P}{\Delta x^2}\left[\frac{\partial T}{\partial n}\right]_P + \frac{\delta_Q}{\Delta y^2}\left[\frac{\partial T}{\partial n}\right]_Q + O\left(h^2\right)$$



- Solving for the concentration in the liquid phase:
  - Time discretization:

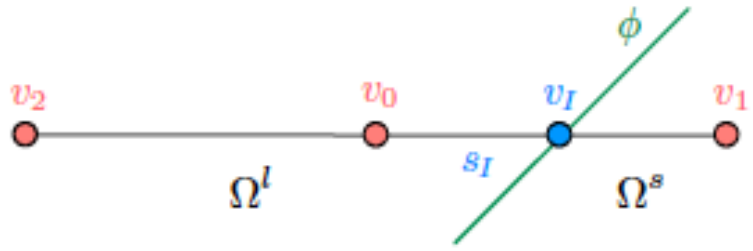
$$\frac{C_L^{n+1} - C_L^n}{\Delta t} = D_L \frac{\Delta {C_L}^{n+1} + \Delta {C_L}^n}{2} \qquad x \in \Omega^l$$

  - Dirichlet boundary condition:

$$C_L = \frac{T - T_m}{m_L} - \frac{\epsilon_c(\theta)}{m_L}\kappa - \frac{\epsilon_v}{m_L} V_N \qquad x \in \Gamma$$

    - Close to the interface:

$$C_{xx}(v_0) = \left(\frac{C_I - C_0}{s_I} - \frac{C_0 - C_2}{s_2} \right) \frac{2}{s_I + s_2}$$



  - Similar for the concentration in the solid phase

- Computing the normal velocity from the concentration fields:
  - Solute rejection equation

$$V_n(C^s - C^l) = D^s \frac{\partial C^s}{\partial n} - D^l \frac{\partial C^l}{\partial n}.$$

  - 4 steps calculation:

1. The concentration fields are extended over the interface
2. The normal gradient are computed
3. Compute velocity from the solute rejection equation
4. Extrapolate the velocity from the interface to the entire domain

  - Time step Definition:

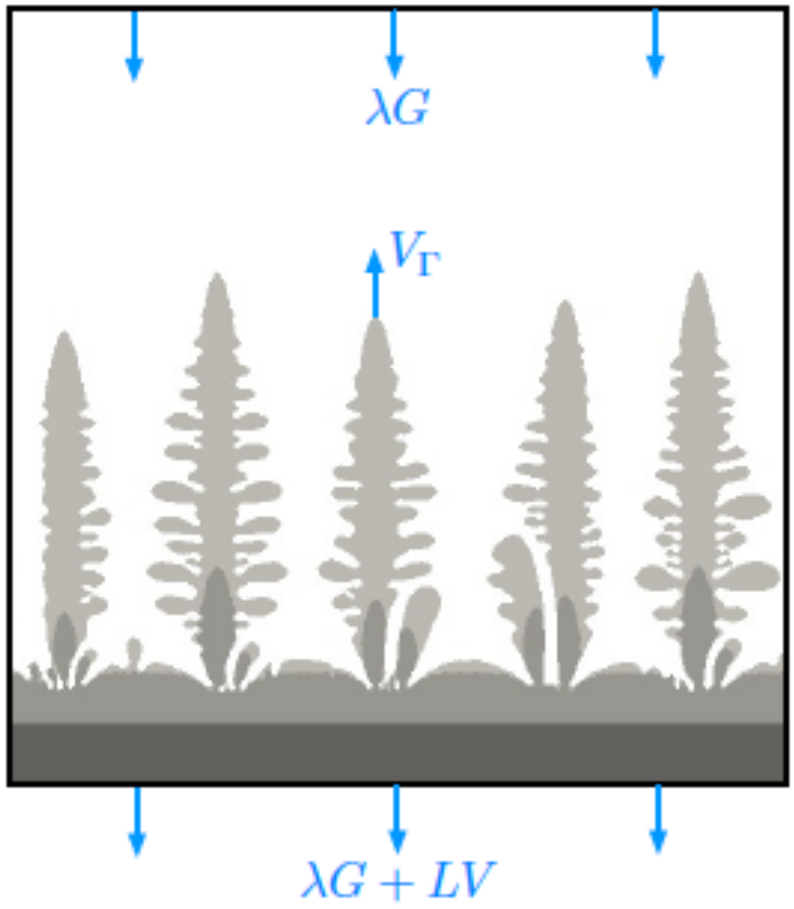
$$\Delta t = \min\left(\alpha \frac{\Delta x}{V_{\max}}, \frac{1}{\max\left(|\kappa V_n|\right)}\right)$$

Recap:

- Sharp: interface conditions are imposed at the interface at every time step
- No drastic time step restriction
- Tree-grid techniques lead to efficient computations
- Linear system are solved using a Multigrid method
- The algorithm scales with the number of nodes
- Parallelized using OpenMP
- On a level 10 (max res=1 024) it takes a few hours on an iMac (8cores)

➢ Validation:

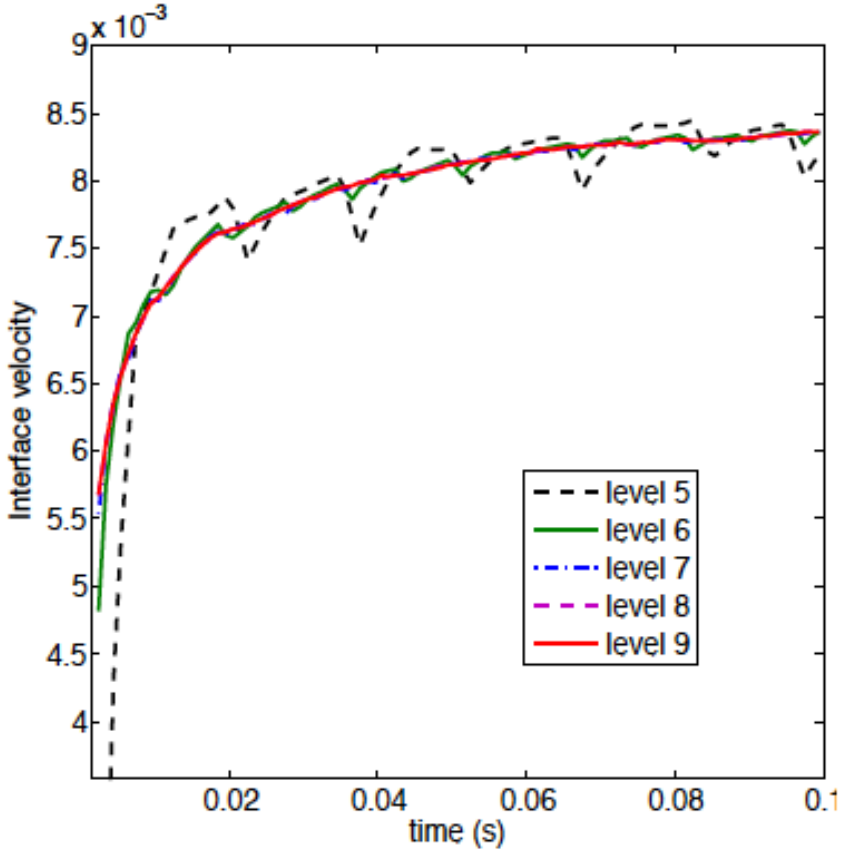
- ➤ Solidification of an Ni-Cu alloy



- Stable planar interface:
  - Choose cooling parameters so that theoretically the interface is stable

$V = 0.01cm/s \qquad G = 2.15 \times 10^4 K/m$

- Stable planar interface:
  - Convergence analysis:

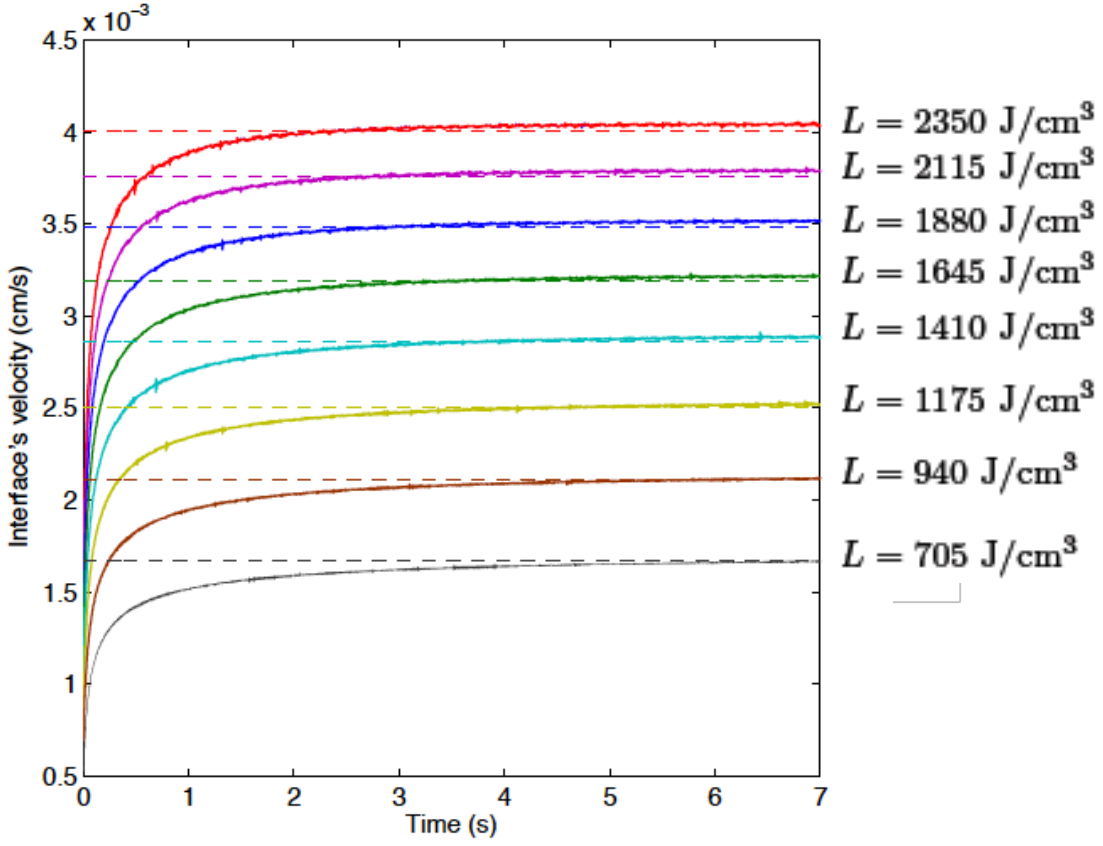


- Stable planar interface:
  - Convergence analysis:
    - Planar stable interface
    - $kG >> LV$
      - Energy balance

$$LV = LV_\Gamma + \frac{\rho c l}{L} G$$

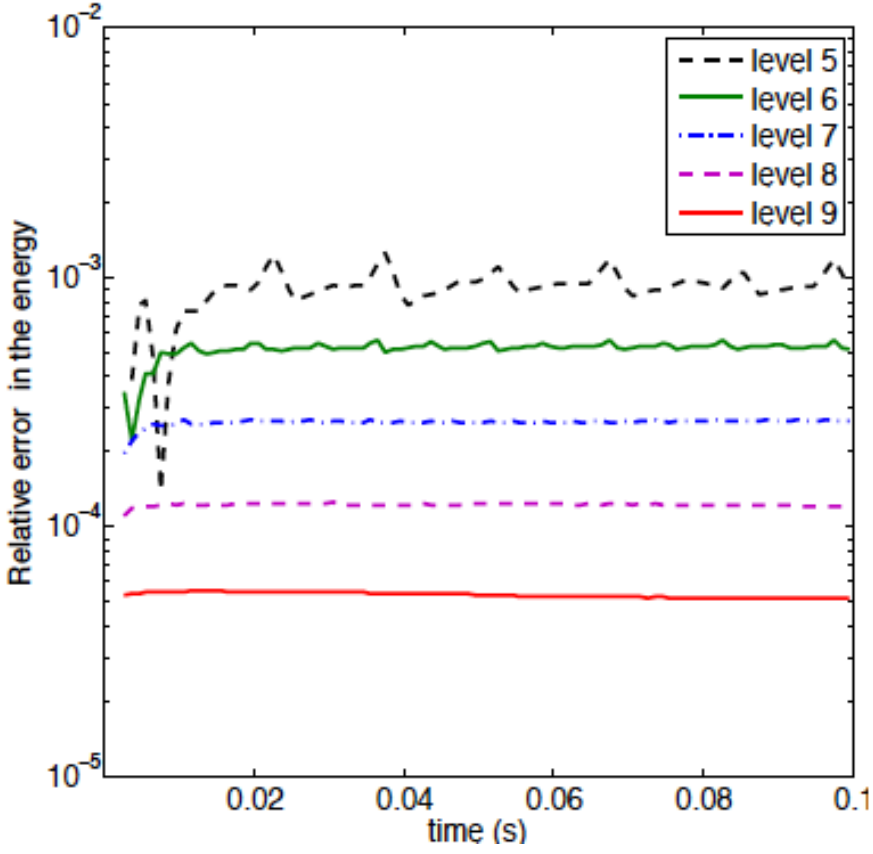
$$\downarrow$$

$$V_\Gamma = \frac{V}{1 + \frac{\rho c l G}{L}}$$

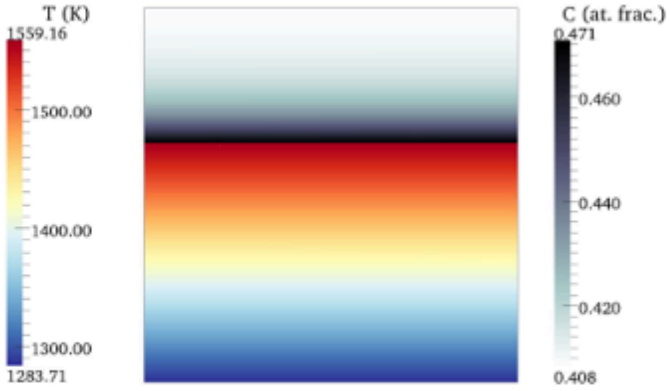


- Stable planar interface:
  - Accuracy analysis:
    - Total energy:

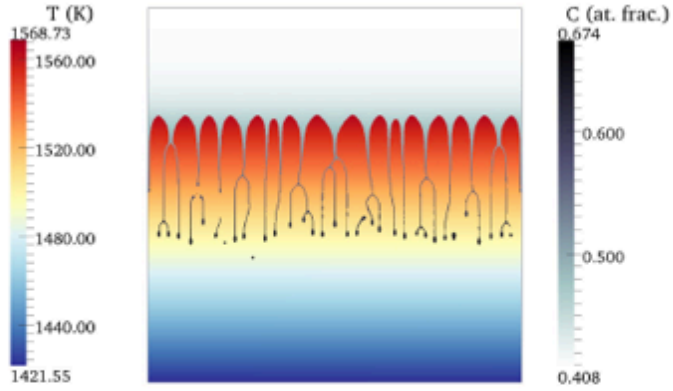
$$E(t) = LVht + E(0)$$



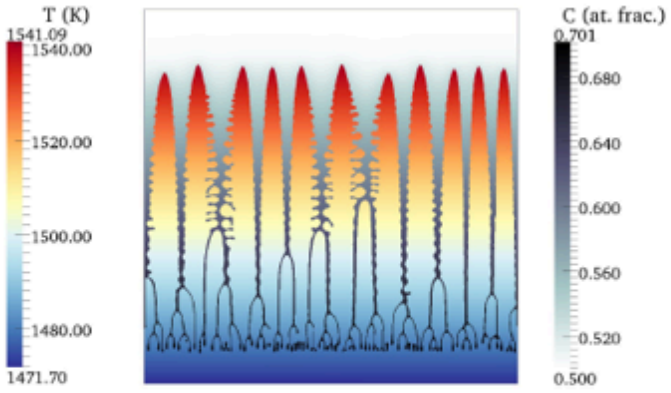
- Planar-Cellular-Dendritic transitions



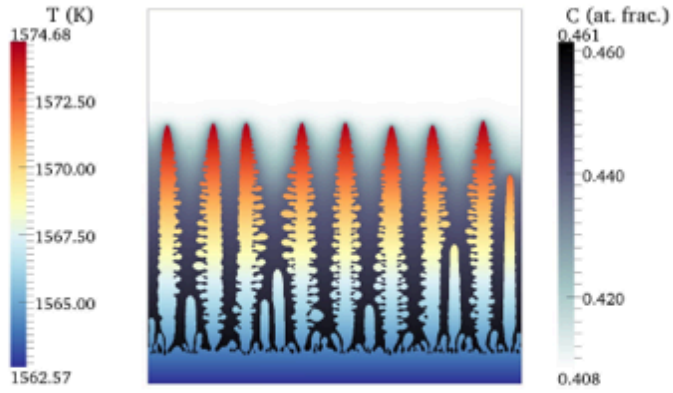
(a) $G = 215 \times 10^2$ K/cm, $l = .2$ mm



(b) $G = 100 \times 10^2$ K/cm, $l = .2$ mm



(c) $G = 20 \times 10^2$ K/cm, $l = .4$ mm



(d) $G = 4 \times 10^2$ K/cm, $l = .4$ mm

- Planar-Cellular transitions

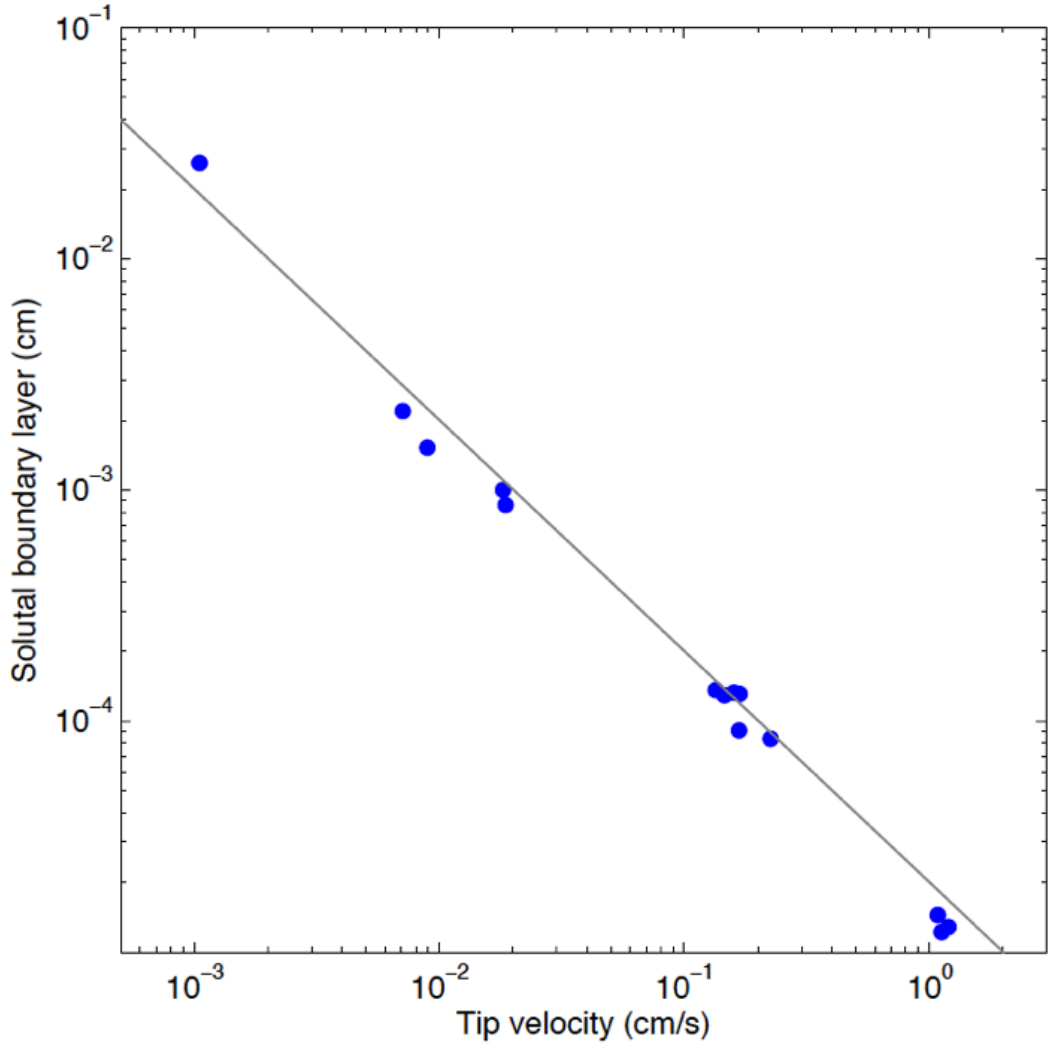
$V = 0.01 cm/s \qquad G = 1 \times 10^4 K/cm$

- Cellular-Dendritic transitions

$V = 0.01cm/s$ $G = 2 \times 10^3 K/cm$

- Solutal boundary layer:
  - Theoretical value:

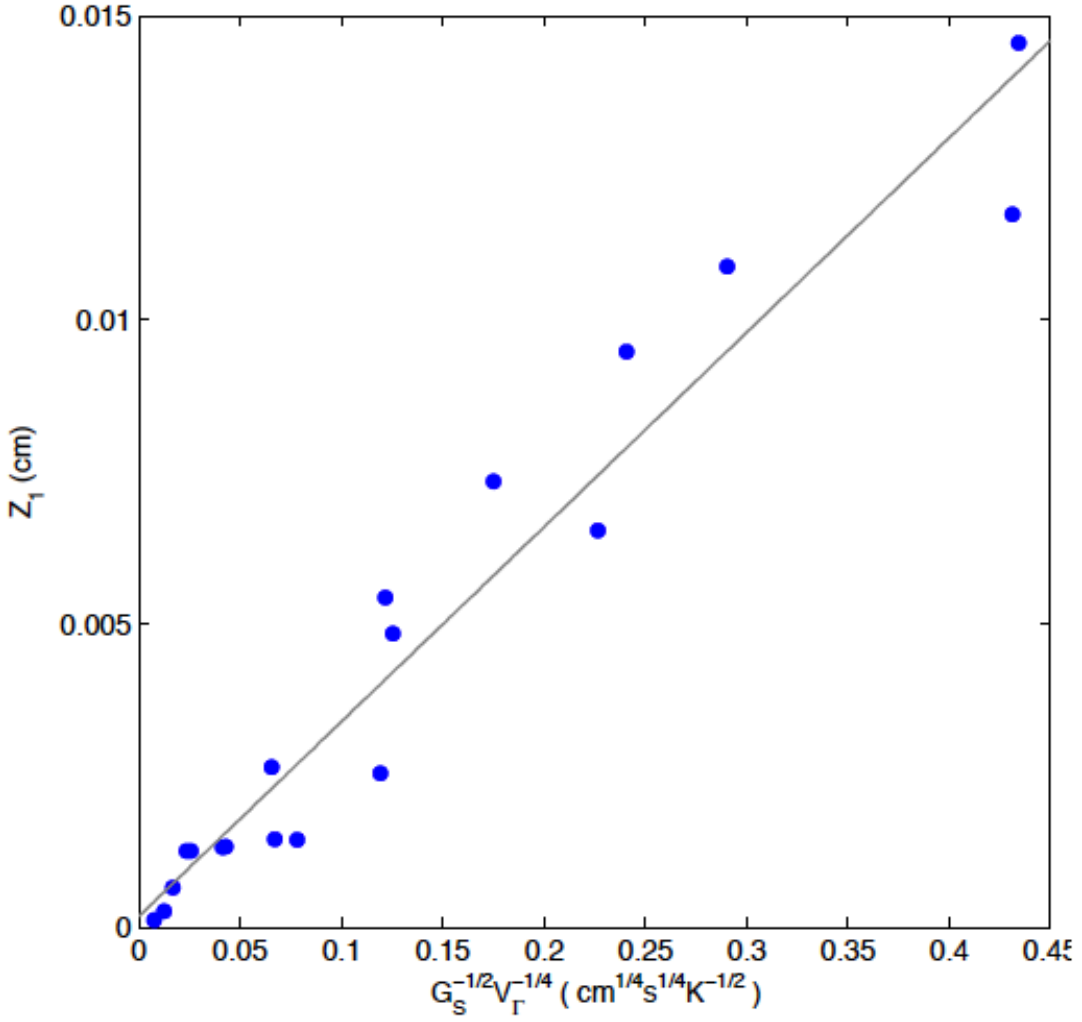
$$\delta_S = 2D^l / V_\Gamma$$



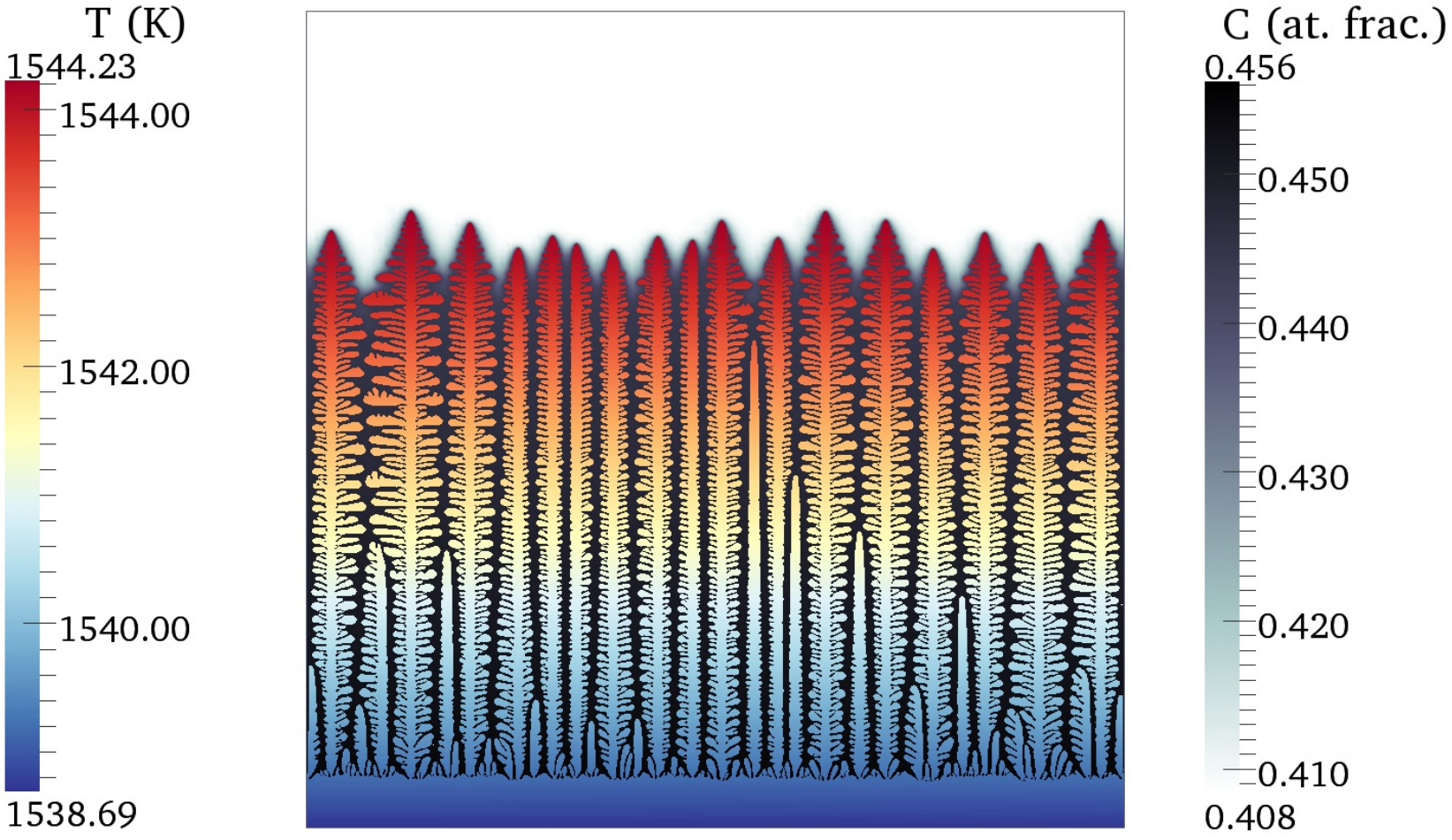
- Primary arm spacing:
  - Commonly accepted model:

$$Z_1 = A G_S^{-n} V_\Gamma^{-m}$$

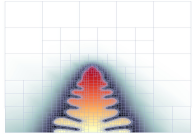
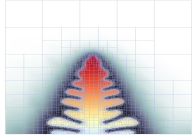
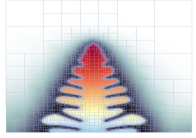
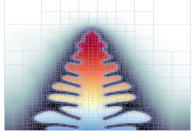
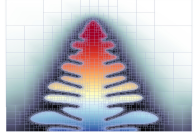
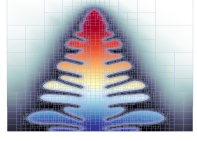
$$n = 1/2 \text{ and } m = 1/4$$



- Formation of the secondary arms
  - Max level 12 (4 096), V = 0.01 cm/s, G = 1 K/cm, l = 0.3 cm



- Formation of the secondary arms:

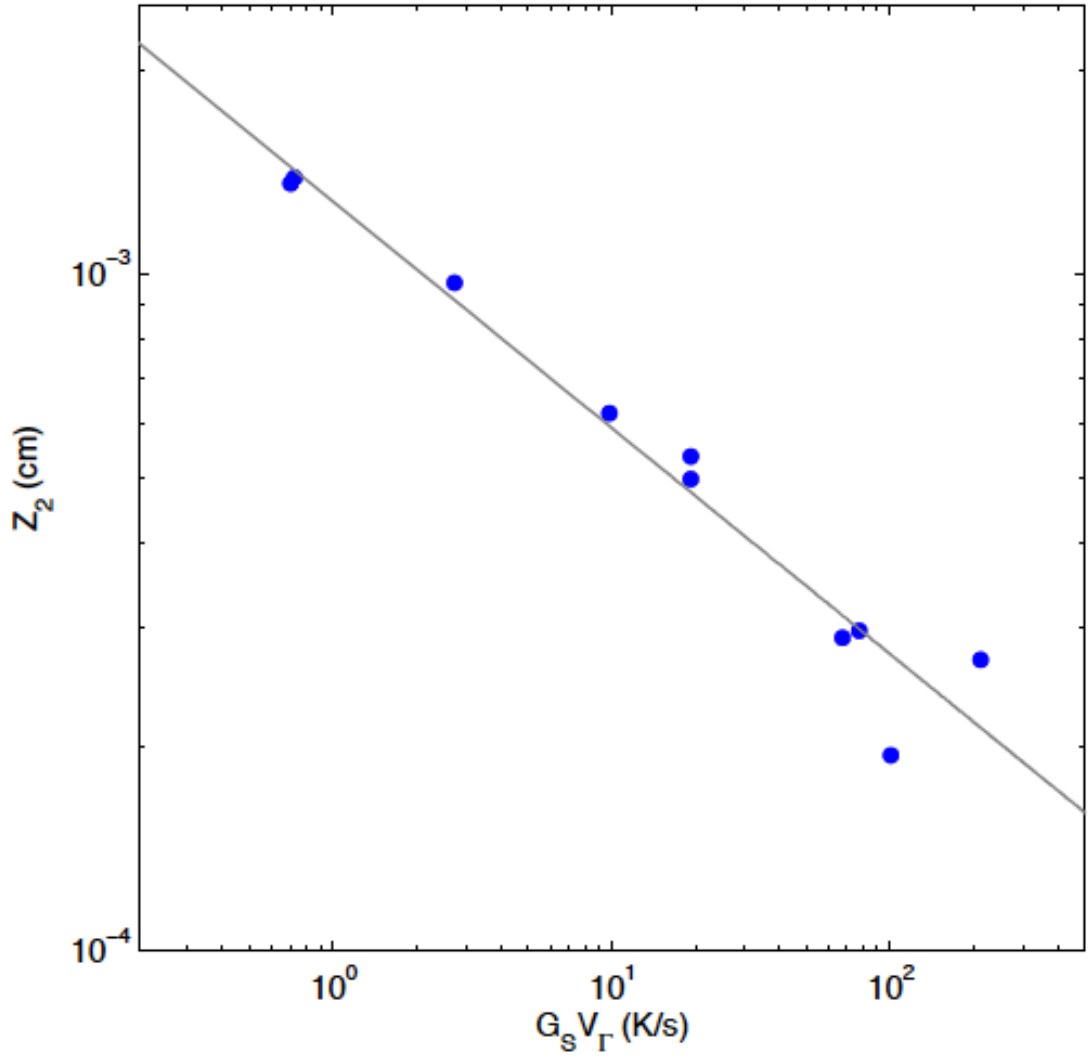
- Secondary arm spacing:
  - Commonly accepted model:

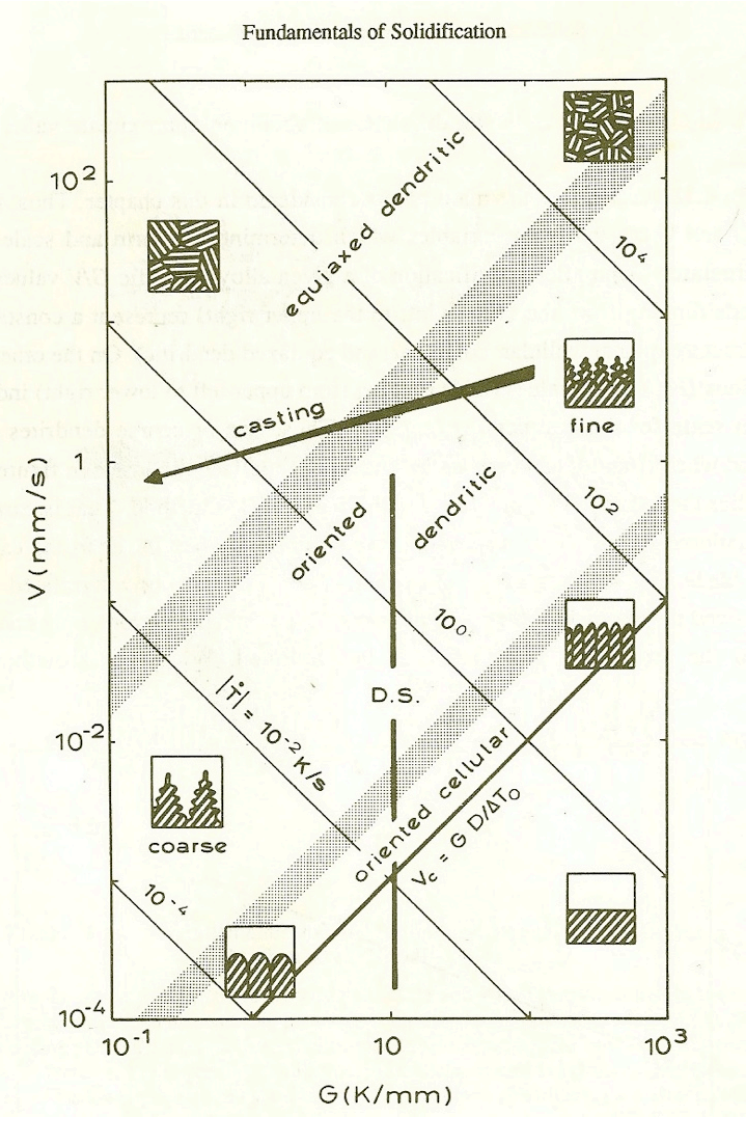
$$Z_2 = B\left(G_S V_\Gamma\right)^n$$

$$n = -\frac{1}{3}$$

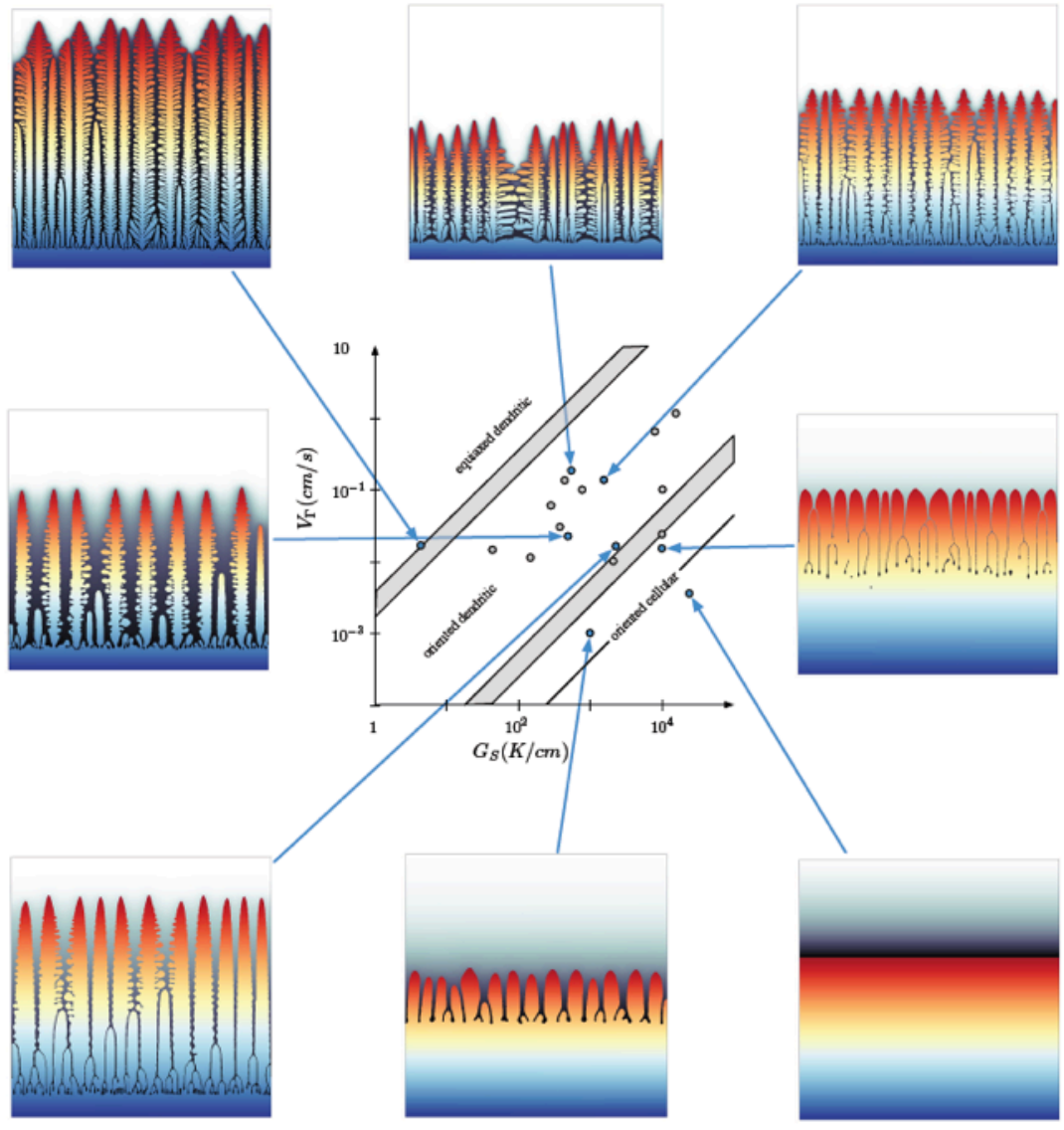
  - Numerically we found:

$$n = -0.334$$





FUNDAMENTALS OF SOLIDIFICATION, KURZ AND FISHER



NUMERICAL RESULTS

➢ Application to solving the Navier-Stokes equations

- Standard projection method (Chorin) based on the Hodge decomposition

$$\begin{aligned} \mathbf{U}^* &= \mathbf{U}^{n+1} + \nabla\psi \\ \nabla \cdot \mathbf{U}^{n+1} &= 0 \end{aligned} \qquad \mathbf{U}^{n+1} = P(\mathbf{U}^*)$$

  - Decomposition at the discrete level

$$U^* = U + G\phi,$$
$$DU = 0.$$

  - Decomposition at the discrete level

$$DG\phi = DU^*$$

  - Oscillation triggered by even-odd decoupling

$$L\widetilde{\phi} = DU^*$$
$$\mathbf{U}^{n+1} = P_{\text{appr}}(U^*) = U^* - G\widetilde{\phi}$$

➢ Application to solving the Navier-Stokes equations

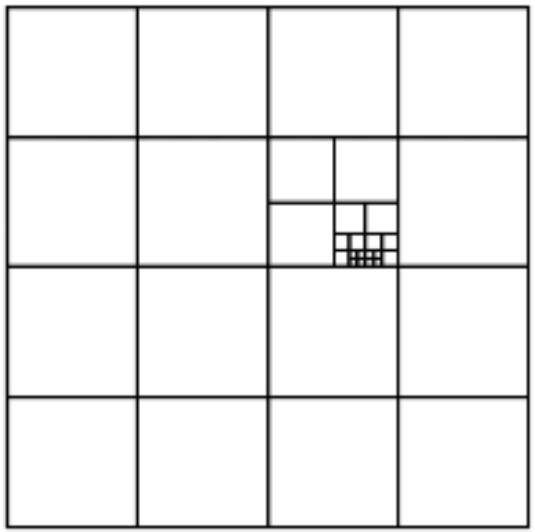
$$\mathbf{U}^{n+1} = P_{\text{appr}}(\mathbf{U}^*) = \mathbf{U}^* - \mathbf{G}\psi$$

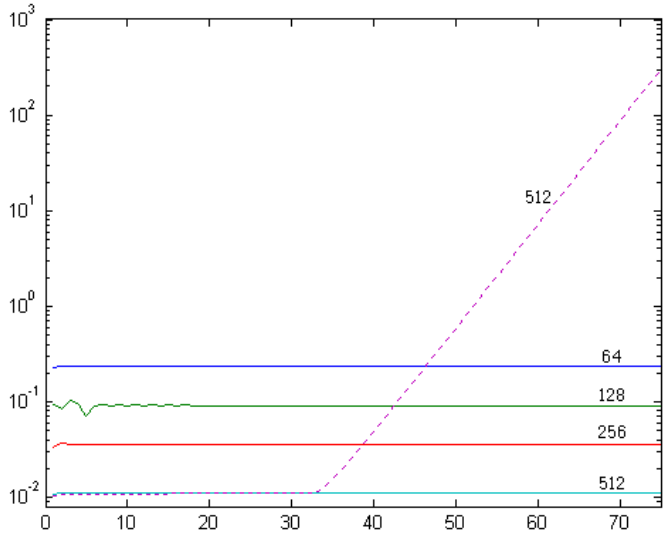
➤ Stability enforced at the discrete level

$G = -D^T$ The relation $G = -D^T$ is not discretely satisfied for our adaptive grids

$||P(\mathbf{U}^*)||_2 \leq ||\mathbf{U}^*||_2$ ➤ stability $P(\mathbf{U}^*) \perp \nabla\psi$

$$\mathbf{U}^{n+1} = P_{\text{orth}}(\mathbf{U}^*) = \mathbf{U}^* - \frac{\mathbf{U}^* \cdot \mathbf{G}\psi}{\mathbf{G}\psi \cdot \mathbf{G}\psi}\mathbf{G}\psi$$

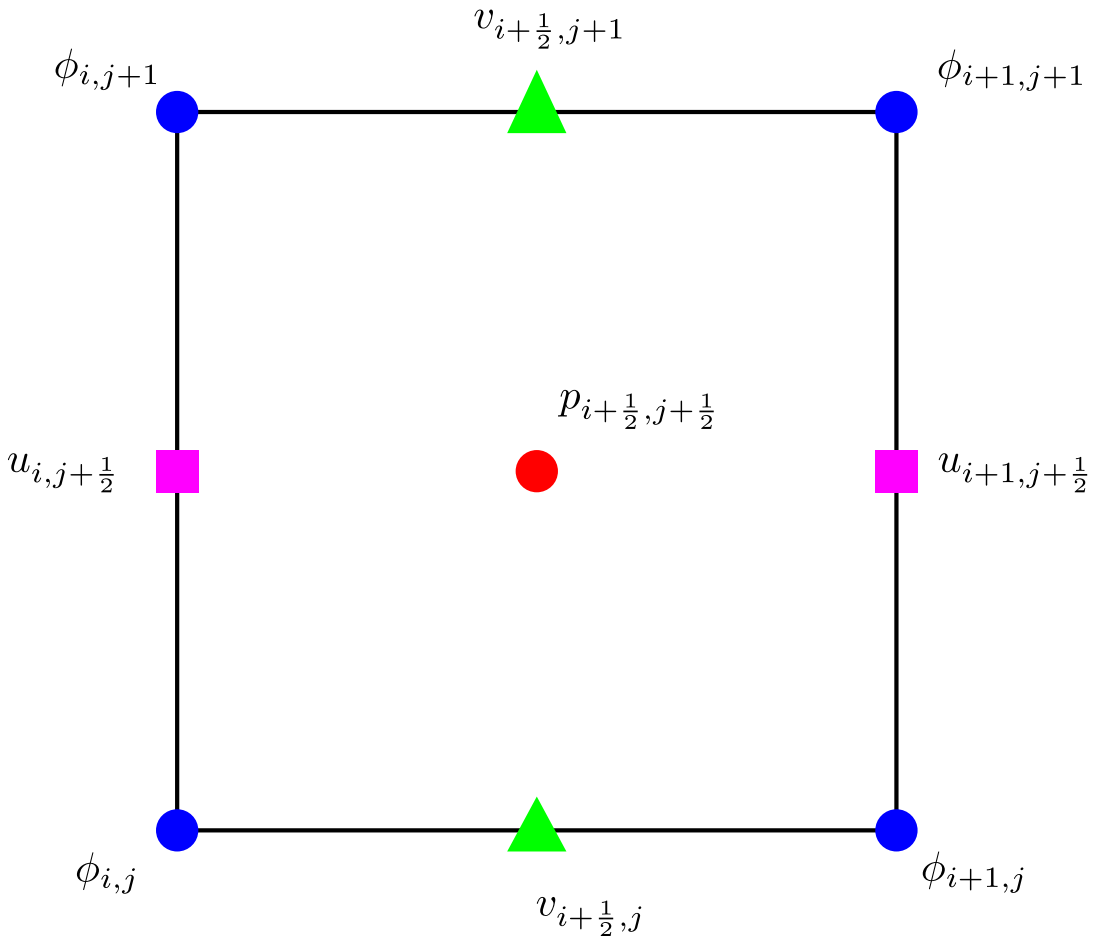




$||\mathbf{U}^{n+1} - \mathbf{U}_{\text{exact}}||_\infty$ after repeated projection

➢ Projection method – stability and accuracy

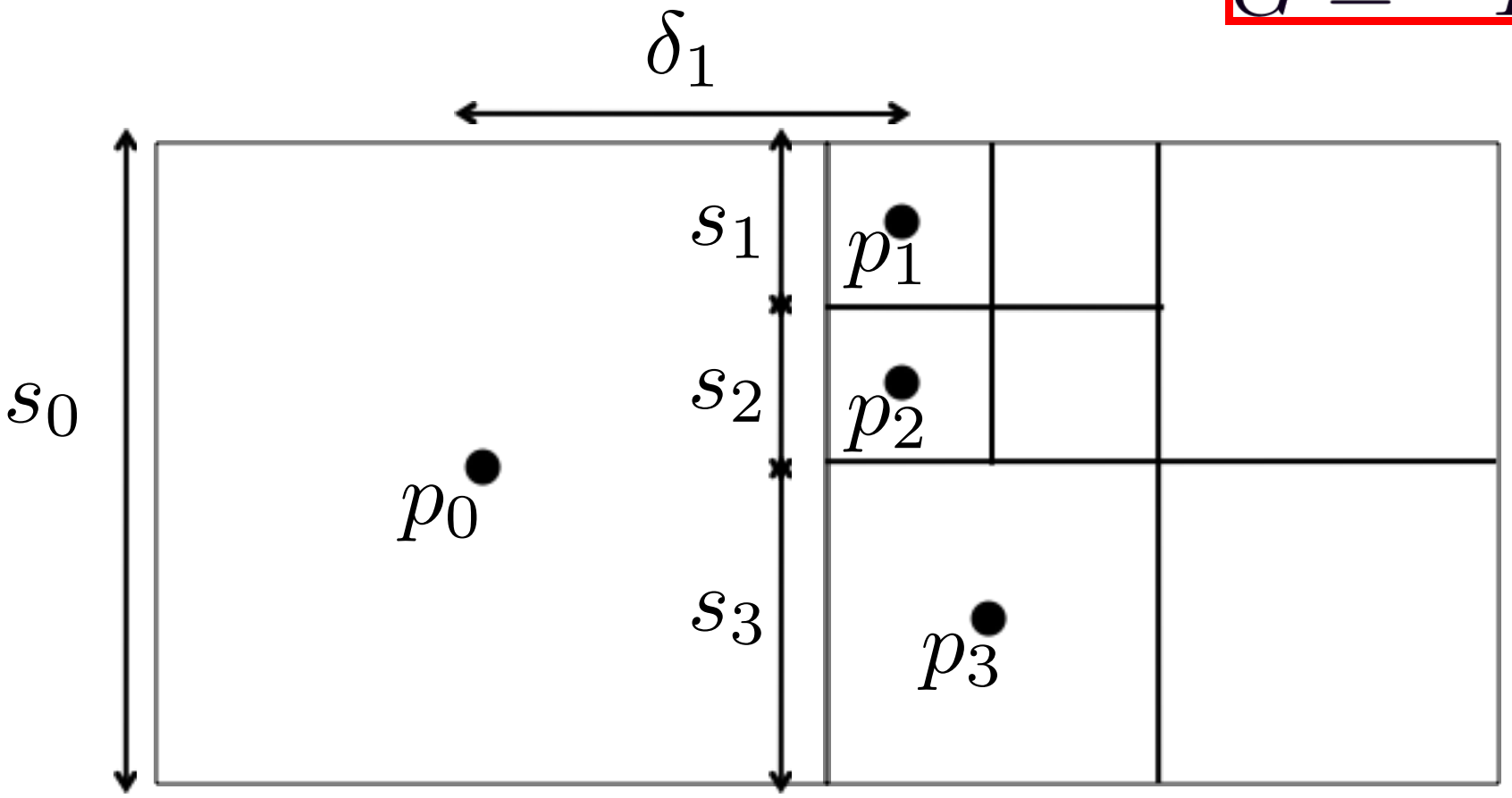
➤ MAC sampling provides stability on uniform grids



$$G = -D^T$$

➢ Cell-based – Poisson solver

$$G = -D^T$$



$$\Delta = \sum \frac{s_i \delta_i}{s_0}$$

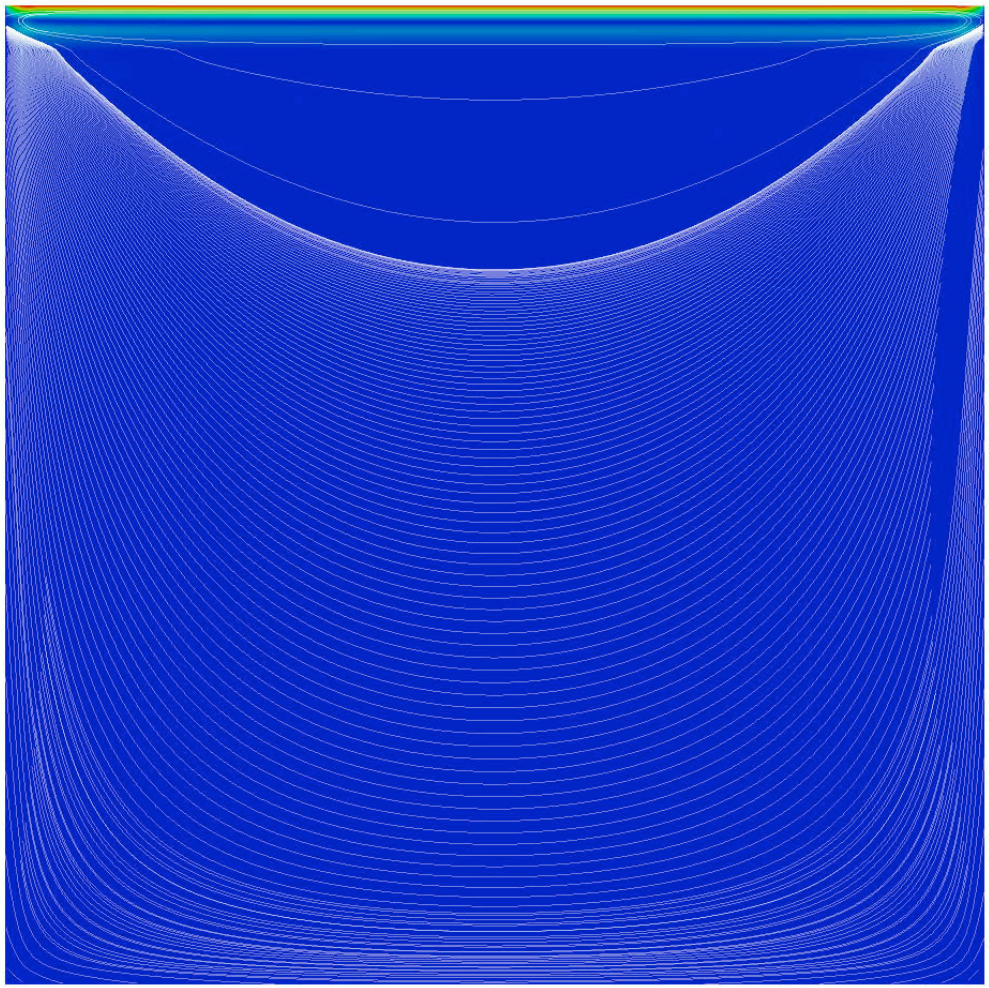
$$\nabla p \approx \sum \frac{s_i (p_i - p_0)}{s_0 \Delta}$$

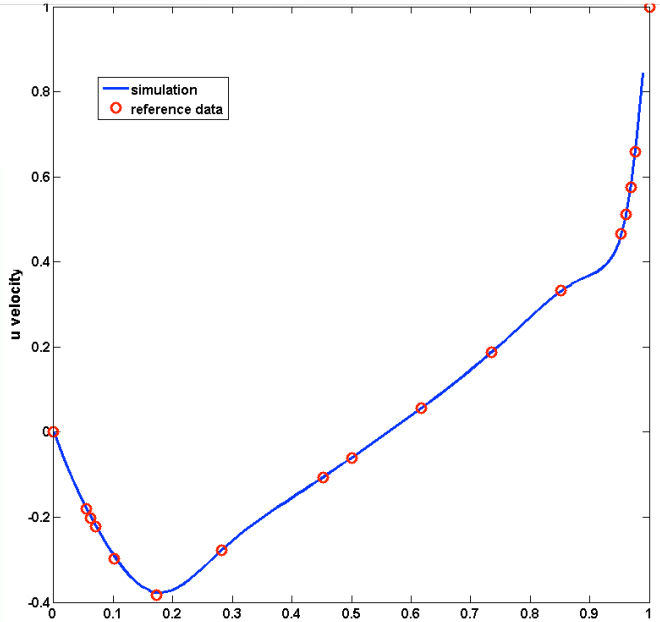
UC SANTA BARBARA engineering

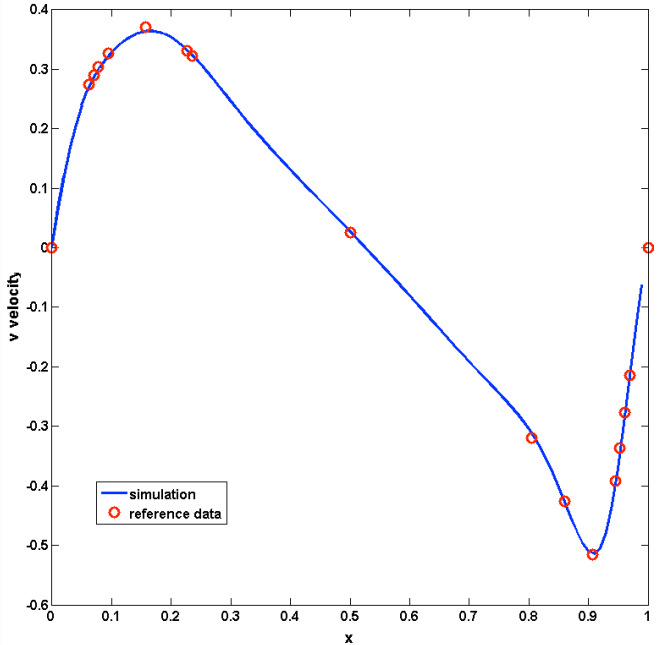
COMPUTATIONAL APPLIED SCIENCE LABORATORY

➢ Hybrid Fluid Solver on Quadtree

➤ Driven Cavity







PART V:

Efficient Parallel Computation of Eikonal Equation

Eikonal equation: a core equation in the level-set technology

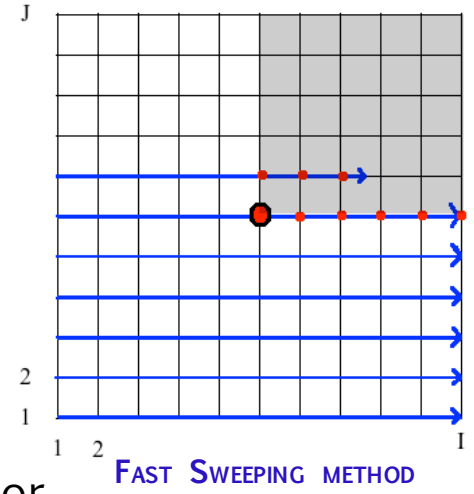
- Used to "reinitialize" the level-set function

$$|\nabla \phi| = 1$$

- Repeated at each time step

Existing approaches

- TVD schemes – Most accurate but computationally expensive
- Fast Marching Method – $O(n \log n)$
- Fast Sweeping Method – $O(n)$



FAST SWEEPING METHOD

Parallel FSM – Zhao *et al.*

- Straightforward – Assign each quadrant to different a processor
- Plateau for # processors > 2^d in R^d
- The // implementation requires more iterations than the serial one

➢ The FSM uses a Godunov upwind differencing scheme on the interior nodes

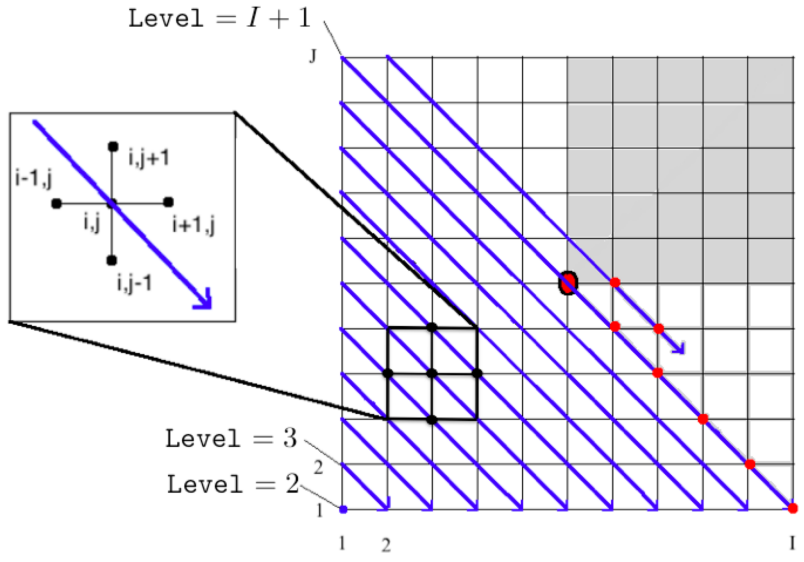
$$[(u_{i,j}^{new} - u_{xmin})^+]^2 + [(u_{i,j}^{new} - u_{ymin})^+]^2 = f_{i,j}^2 h^2$$

$$u_{xmin} = \min(u_{i-1,j}, u_{i+1,j}) \qquad \text{and} \qquad (x)^+ = \begin{cases} x & x > 0, \\ 0 & x \leq 0. \end{cases}$$

➢ Our approach: different sweeping ordering (Cuthill-McKee)

`Level=i+j`

- ➤ Nodes on a level are updated simultaneously
- ➤ Performance does not plateau

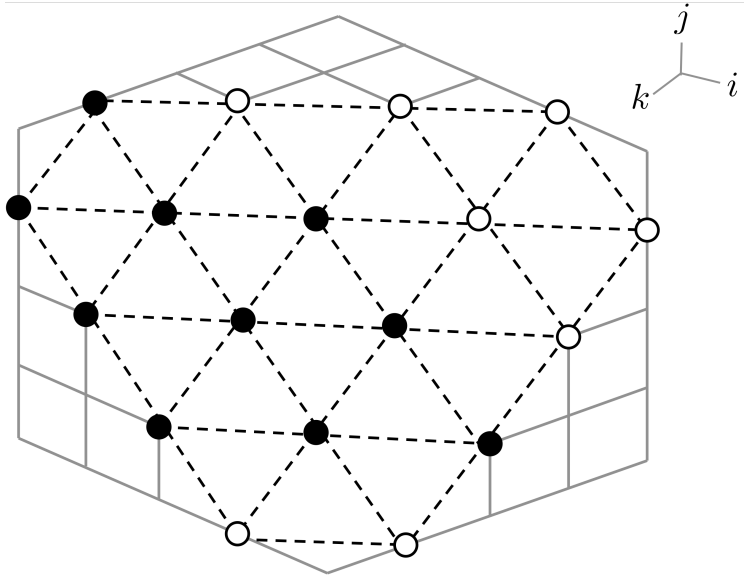


PRESENT PARALLEL IMPLEMENTATION

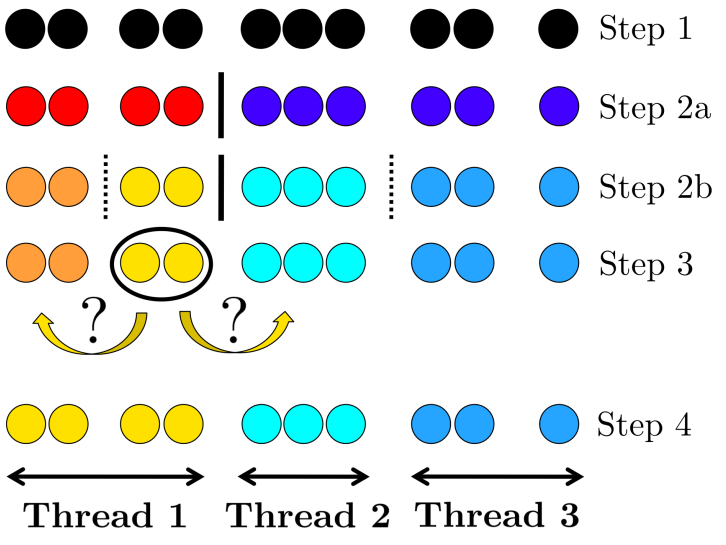
Similar idea in 3D

- Sweeping ordering in 3D

  `Level=i+j+k`



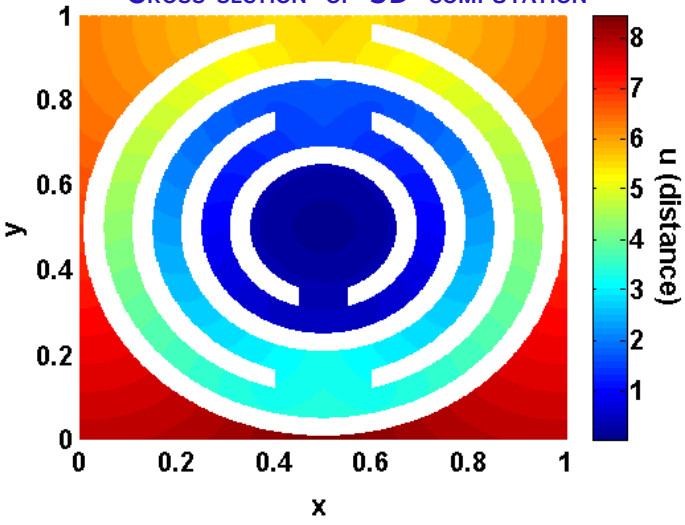
- Load balancing strategy
  - Divide and merge



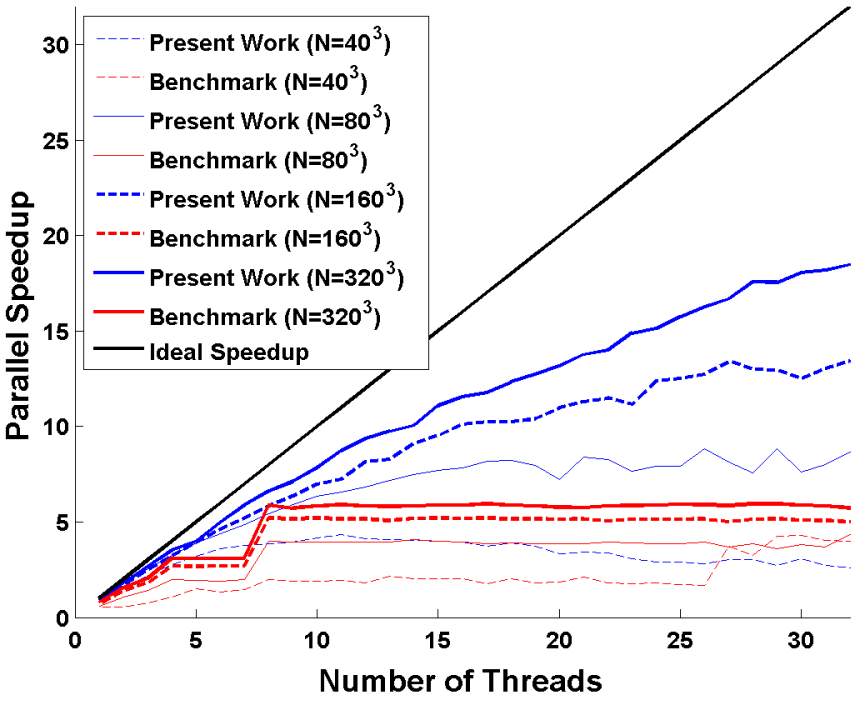
Efficiency

- Distance function from a center source point in 3D with four concentric spherical obstacles.

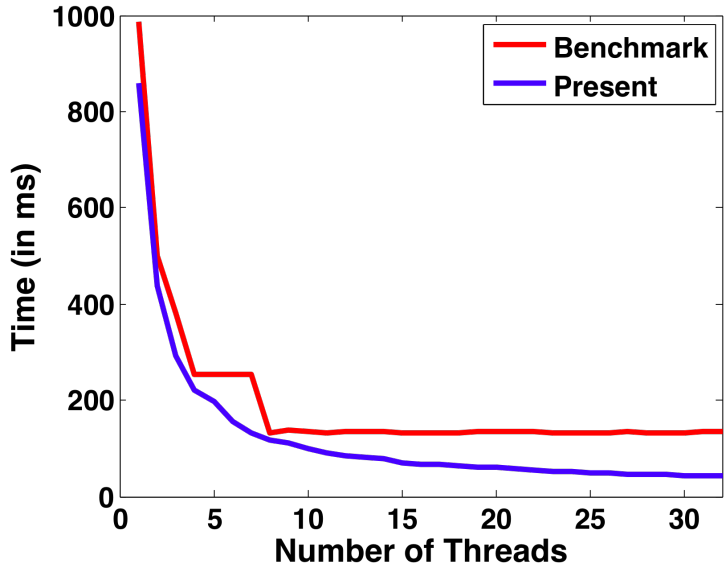
CROSS-SECTION OF 3D COMPUTATION



PARALLEL SPEEDUP

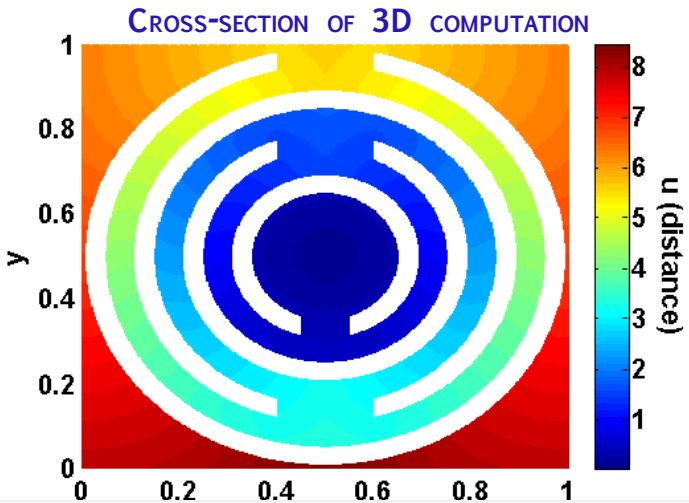


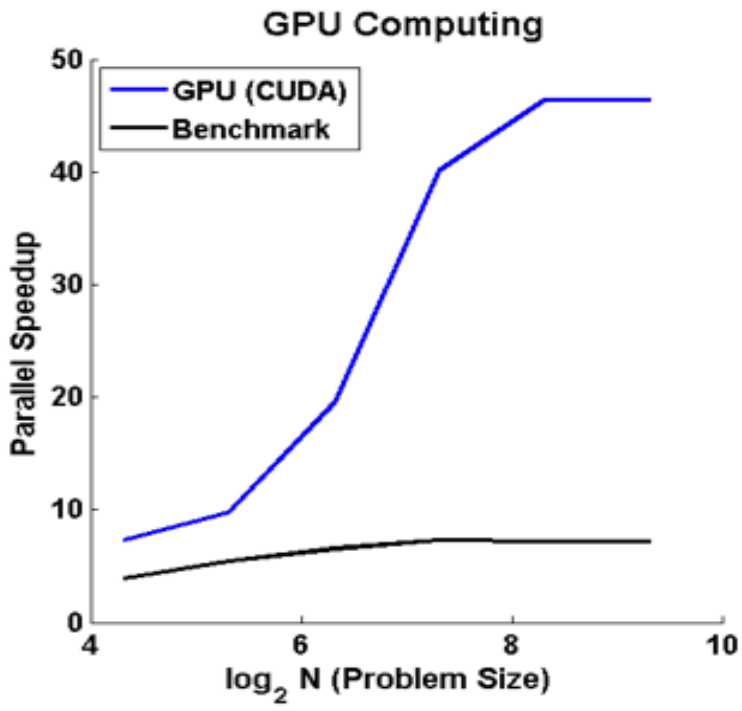
COMPUTATIONAL TIME



GPU Computing

- The existing parallel techniques have very limited parallel speedup.
- Our method doesn't suffer the same limitations
- Great speedup results on GPUs
- Makes larger problems tractable





Conclusions

- Sharp treatment for the interface conditions
- Numerical result agree with theoretical predictions and experimental observations
- Curvature and grain growth

Future work

- Fluid in solidification, parallel efficiency, grain growth, ...

Special thanks

- Students and Postdocs: Brun, Chen, Detrixhe, Guittet, Helgadottir, Min, Mirzadeh, Ng, Ouaknin, Papac, Theillard.
- Colleagues: Pollock, Min.
- Funding agency: