Two numerical algorithms for anisotropic curvature motions

Yen-Hsi Richard Tsai
The University of Texas at Austin

in collaboration with

S. Esedoglu and S. Ruuth
R. Takei, S. Osher, and A. Oberman

Sapporo, August 2010

Outline

- Merriman-Bence-Osher scheme
- Chambolle's algorithm for Ahlmgren-Taylor-Wang's algorithm
- The use of signed distance function
- Linear heat equations

- Accurate and efficient (high order) geometric motions of an interface
- Crystaline curvature

Related Work

- Merriman-Bence-Osher, Ruuth-Merriman
- [Deckelnik-Dziuk-Elliott:Acta-Numerica], Rumpf, Nochetto, Lakkis, Heintz, Garcke, Kimura
- Level set methods:
- Phase-field methods: Kobayashi, ...
- Almgren-Taylor-Wang, Chambolle, Novaga, Paolini, Bellettini
- [Esedoglu-Ruuth-Tsai]
- [Oberman-Osher-Takei-Tsai]

Anisotropic MCF

- Almgren-Taylor-Wang's discrete time variational formulation:

$$\mathcal{E}(C) = \int_C \gamma(\hat{n}_C) ds$$

$$\min_f \mathcal{E}(f(C)) - \mathcal{E}(C) + \frac{1}{2h}\langle f, f\rangle$$

- Chambolle's formulation:

$$u^{n+1} = \arg\min_u \int \gamma(\nabla u) + \frac{1}{2\Delta t}\int (u - \mathrm{dist}_{\{u^n=0\}})^2$$

Computationally nontrivial optimization due to nonlinearity.

Anisotropic MCF

- The level set method, with smooth anisotropy,

$$\frac{\partial u}{\partial t} = (\gamma + \gamma'')|\nabla u| \nabla \cdot \frac{\nabla u}{|\nabla u|}$$

- Difficult to generate efficient and accurate numerical schemes. No good multigrid method.
- This equation is not suitable for crystalline curvature flow.

Merriman-Bence-Osher scheme

$w^0 = \chi_\Omega$ (char. function)

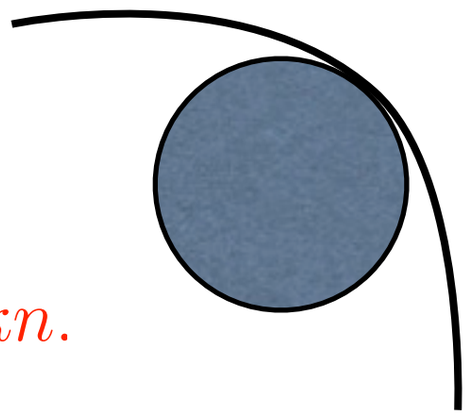
- Step 1: diffuse

$$u_\tau = \Delta u, 0 < \tau \leq \delta t$$
$$u(x,0) = w^n$$

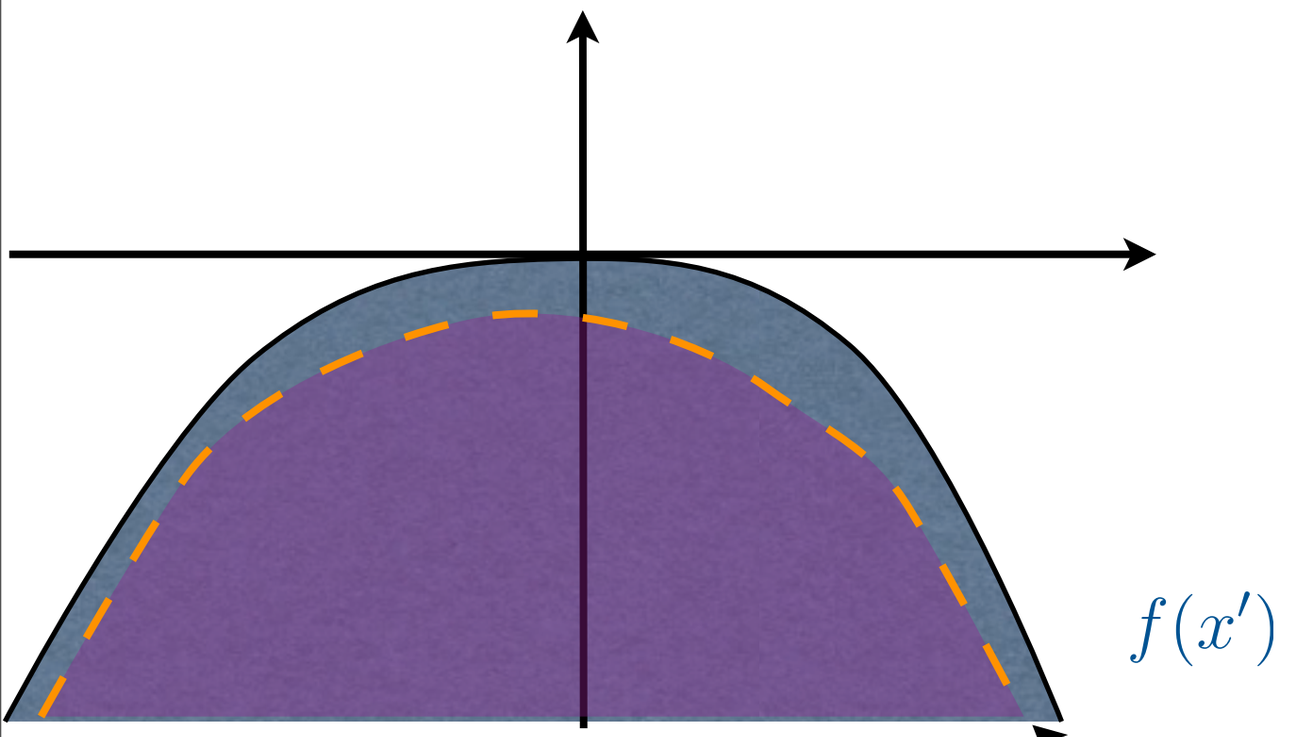
- Step 2: threshold

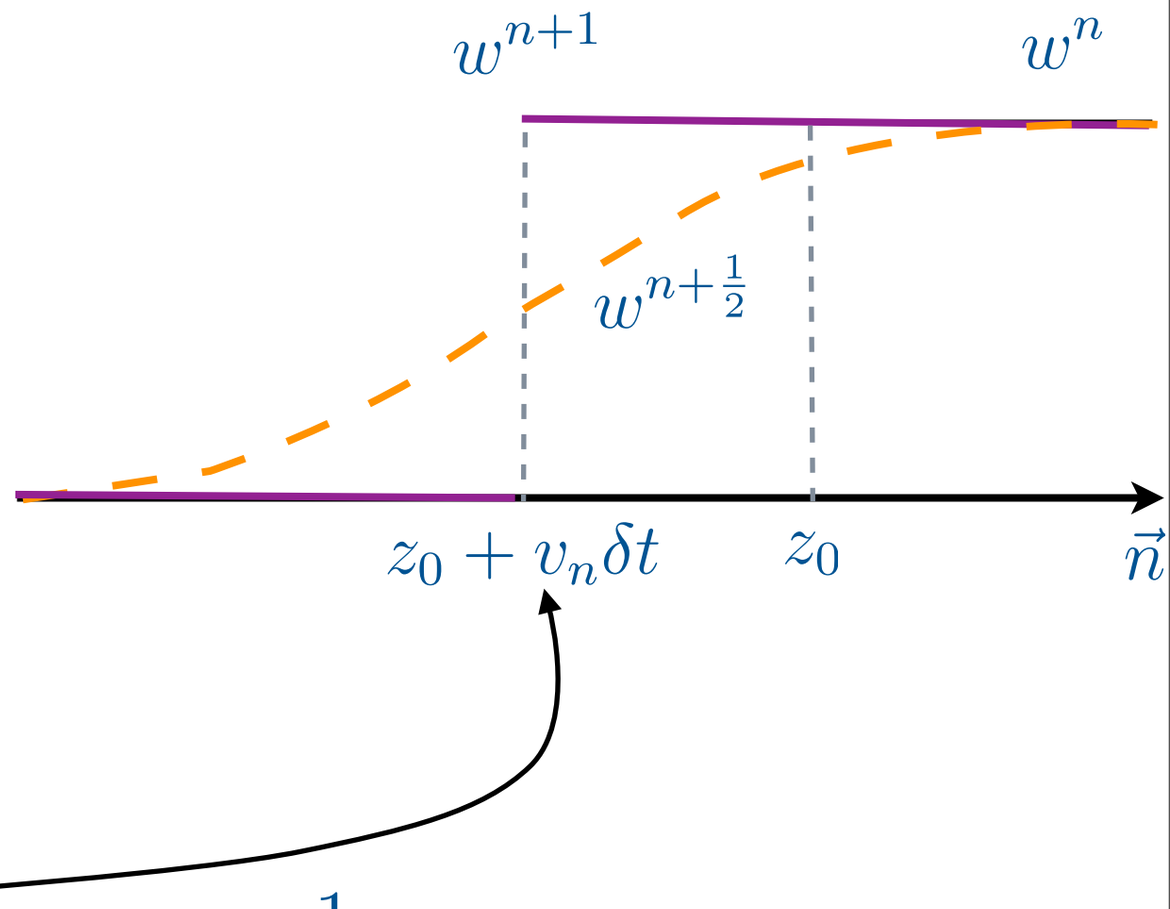
$$w^{n+1} = \chi_{\{u(\cdot,\delta t) \geq \frac{1}{2}\}}$$

- Iterate



In polar coordinates: $\Delta \sim \frac{1}{r}\partial_r + \partial_r^2$
In the limit, $\{u = \frac{1}{2}\}$ is moving by its curvature $-\kappa n$.



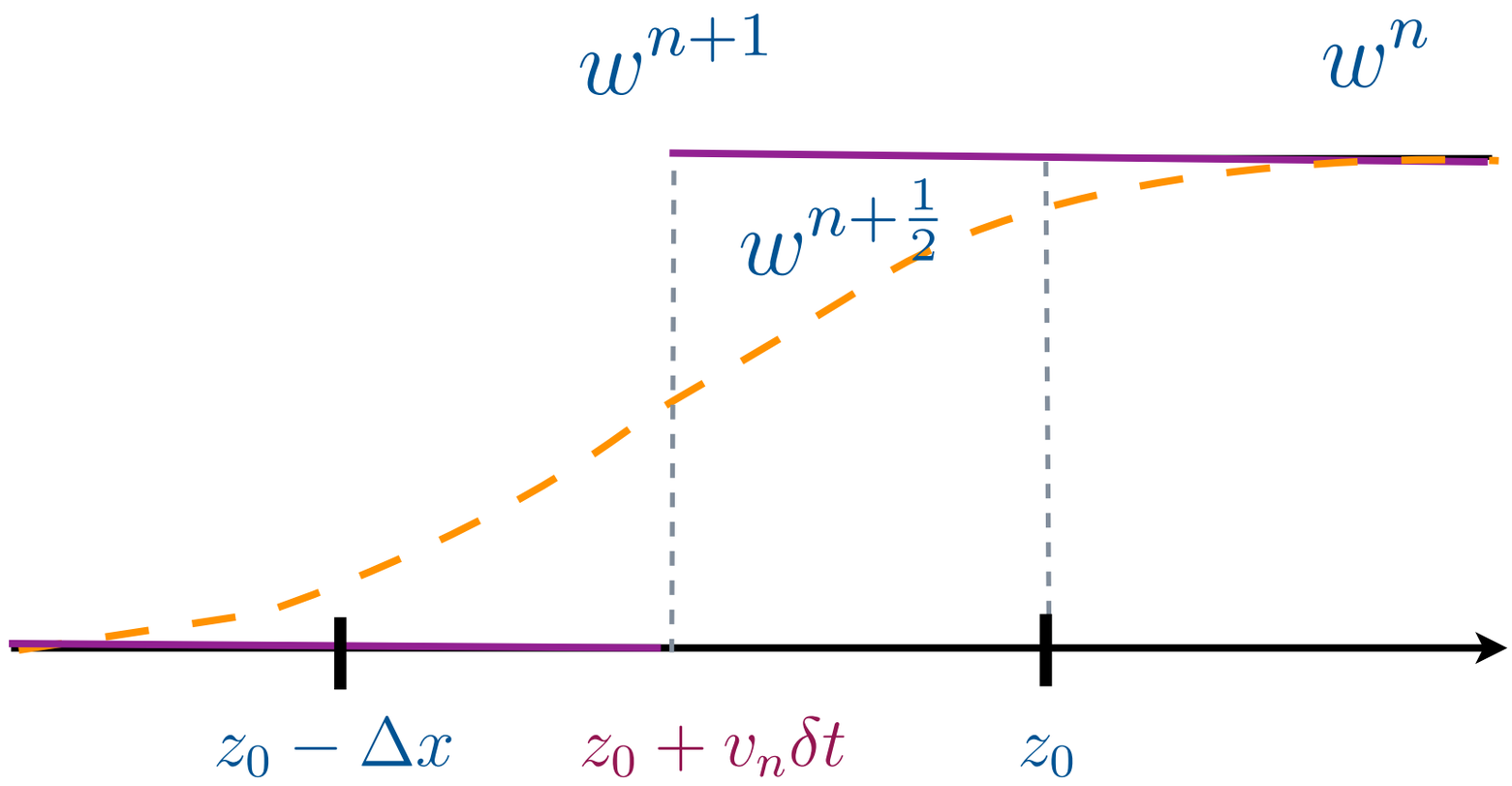


$$\{w^{n+1} = \frac{1}{2}\} = \{w^{n+\frac{1}{2}} = \frac{1}{2}\}$$

Converges to motion by mean curvature for $\partial\Omega(t)$

[Mascarenhas, Barles-Georgelin, Evans, Souganidis, Ishii]

Potential trapping on a grid



Non-stationary numerical solutions require: $\delta t > C_0 \Delta x$

How far can one go?

- Step 1: diffuse

$$w^{n+\frac{1}{2}} = G_{\delta t}[w^n]$$

Pro: optimal numerical solution.

- Step 2: threshold

$$w^{n+1} = \mathcal{T}[w^{n+\frac{1}{2}}]$$

Pros: simplicity and efficiency

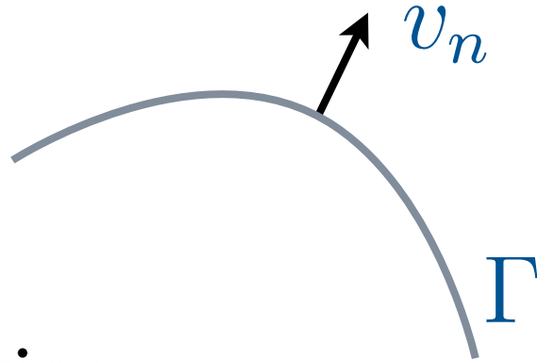
- Iterate

Pros: unconditional stable, efficiency.

Objectives

- Compute geometric motions of a curve

$$v_n = v_n(\vec{n}_\Gamma, \kappa_\Gamma, \cdots)$$



- Construct minimizers of a class of energies

$$\min_\Gamma J[\Gamma]$$

- Simple, efficient, and robust algorithms

Some energies

- Curve length/total variation
- Mumford-Shah energy (and variants) [Esedoglu-Tsai 2005]
- Elastic energy

Stability, efficiency, resolution and accuracy

- Algorithm stable for large time steps
- Efficiency in solving linear diffusion equation

- Diffusion kernels are wide: (curves cannot be too close) $G_{\sqrt{\delta t}}$ $\delta t = c_0 \Delta x$
- Truncation error: $|y(t_n) - y_n| = \mathcal{O}(\delta t^{\frac{3}{2}})$
- Grid refinement strategy: [Ruuth]

Threshold dynamics/ Diffusion generated motion

Use signed distance functions.

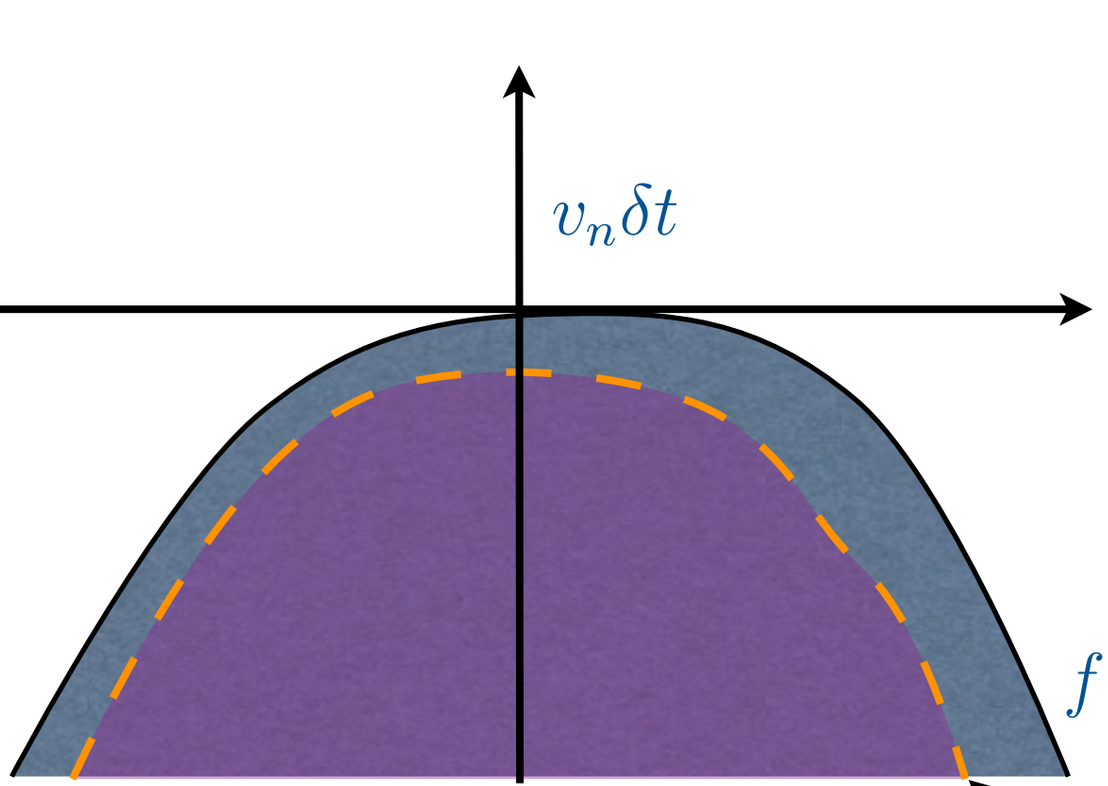
- Step 1: diffuse $w^{n+\frac{1}{2}} = G_{\delta t}[w^n]$

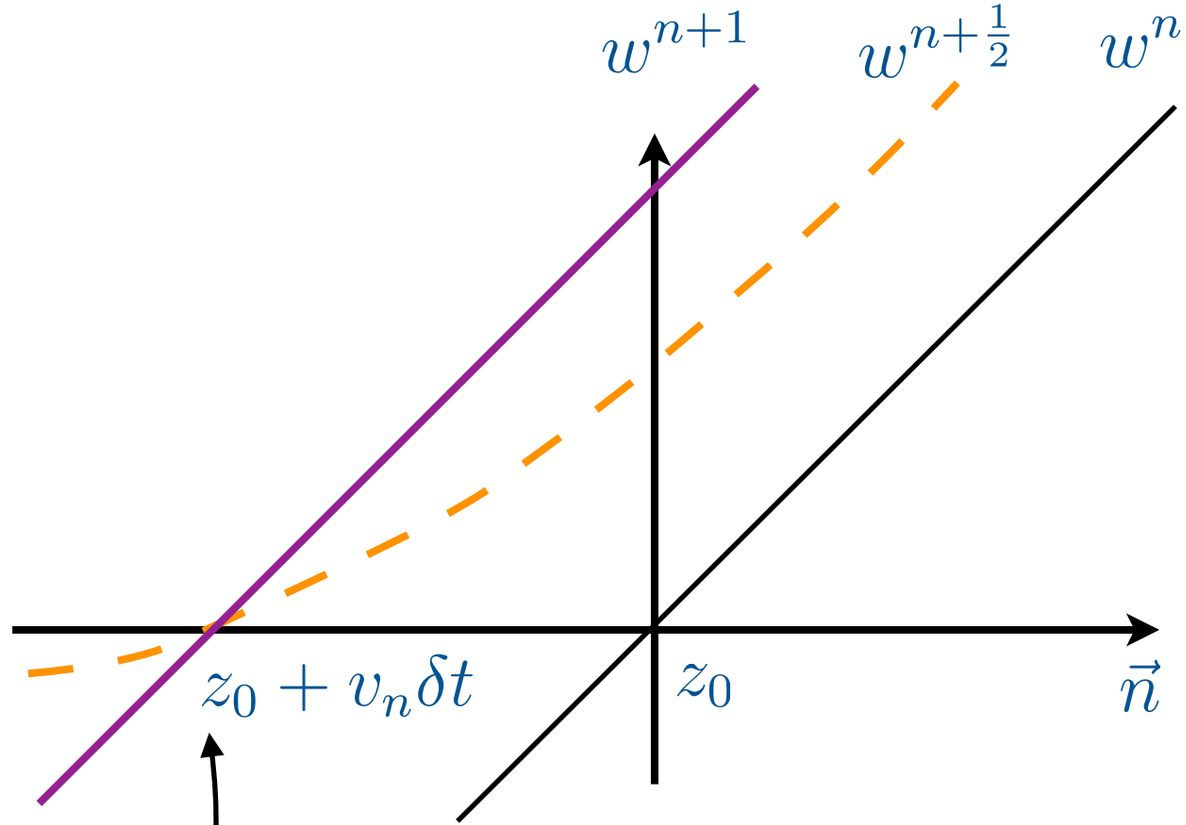
- Step 2: threshold $w^{n+1} = \mathcal{T}[w^{n+\frac{1}{2}}]$

Threshold operator → "redistancing"

- Iterate

No restriction on ~~$\delta t > C_0 \Delta x$~~





$$\{w^{n+\frac{1}{2}} = 0\} = \{w^{n+1} = 0\}$$

The set-up

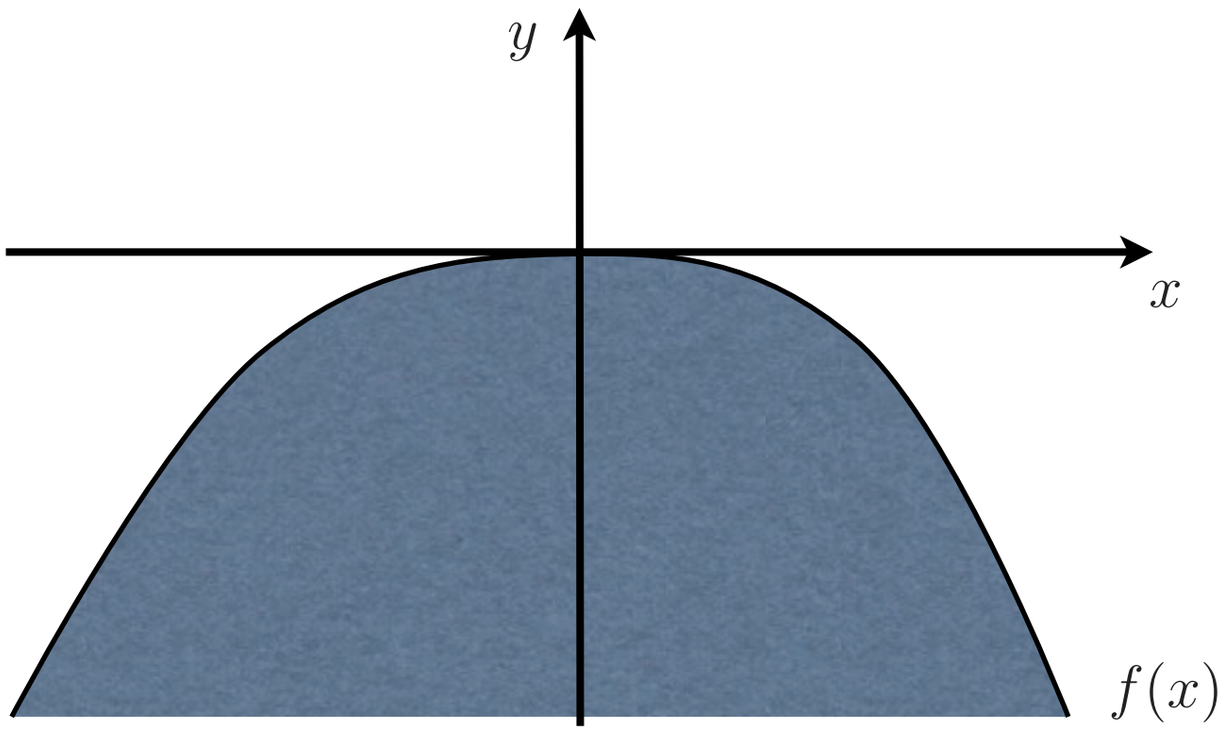
$$\Gamma = \{(x, f(x))\}$$

$$f(0) = f'(0) = 0, \text{ and } f''(0) = -\kappa(0).$$

$$|\nabla u(x, y)| = 1 \text{ such that } d(x, f(x)) = 0,$$

$$d(x, y) > 0 \quad \text{if} \quad y > f(x)$$

$$d(x, y) < 0 \quad \text{if} \quad y < f(x)$$



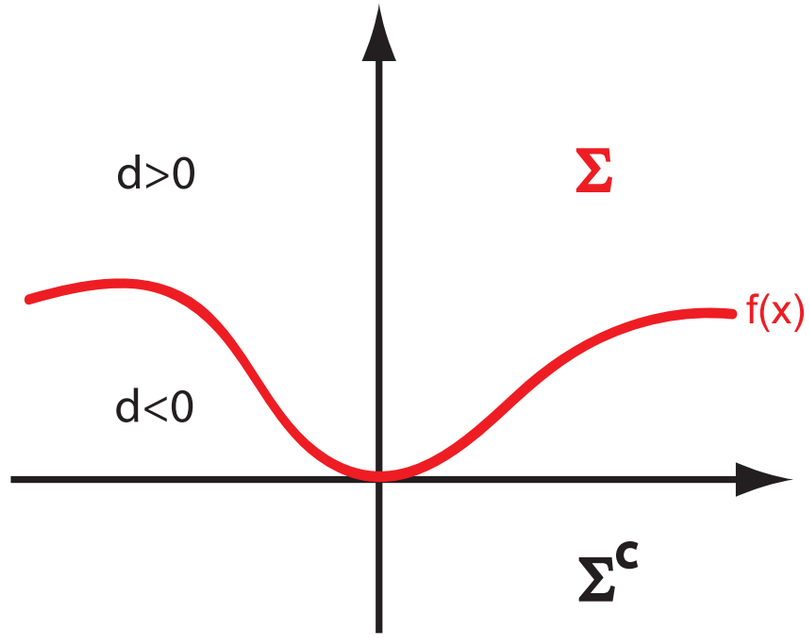
Asymptotics near the interface

$$|\nabla d| = 1$$

$$d(0,0) = 0, \text{ and } d_y(0,0) = 1, d_x(0,0) = 0.$$

Lemma. For sufficiently small y,

$$\begin{aligned} d_y(0,y) &= 1 \\ d_x(0,y) &= 0 \end{aligned}$$



$$d_{yy}(0,y) = d_{yyy}(0,y) = d_{yyyy}(0,y) = 0.$$

$$d_{xy}(0,y) = d_{xyy}(0,y) = d_{xyyy}(0,y) = 0.$$

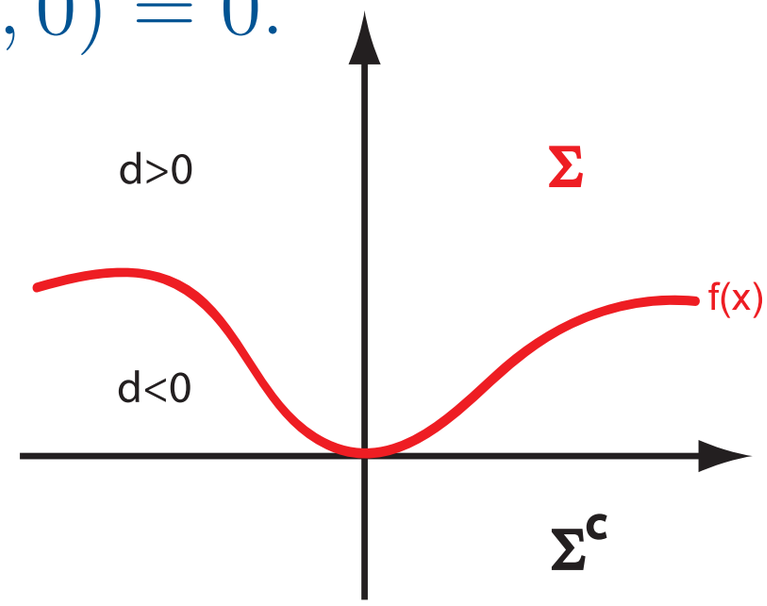
Asymptotics near the interface

$$|\nabla d| = 1$$

$$d(0,0) = 0, \text{ and } d_y(0,0) = 1, d_x(0,0) = 0.$$

Theorem:

$$\begin{aligned} d(x,y) &= y + \frac{1}{2}\kappa x^2 \\ &+ \frac{1}{6}(\kappa_x x^3 - 3\kappa^2 x^2 y) \\ &+ \frac{1}{24}((\kappa_{xx} - 3\kappa^3)x^4 - 12\kappa\kappa_x x^3 y + 12\kappa^3 x^2 y^2) \\ &+ \mathcal{O}(r^5). \end{aligned}$$



For $y = \mathcal{O}(\delta t)$

$$G_{\delta t} * d(0,y) = y + \kappa\delta t - \kappa^2 y\delta t + \frac{1}{2}(\kappa_{xx} + \kappa^3)\delta t^2 + \mathcal{O}(\delta t^3)$$

Curvature motion

For $y = \mathcal{O}(\delta t)$

$$G_{\delta t} * d(0, y) = y + \kappa \delta t - \kappa^2 y \delta t + \frac{1}{2}(\kappa_{xx} + \kappa^3)\delta t^2 + \mathcal{O}\delta t^3)$$

Solve:

$$G_{\delta t} * d(0, y) = 0$$

$$\Longrightarrow \text{ location of new zero level set}$$

$$y = -\kappa \delta t + \kappa^2 y \delta t + \cdots$$

Forward Euler in time

- Convolve (spatial discretization):

$$G_{\delta t} * d^n(0, y) = y + \kappa \delta t + \mathcal{O}(\delta t^2)$$

- Threshold:

$$d^{n+1} = \mathcal{T}[G_{\delta t} * d^n]$$

$$\{\mathcal{T}[u] = 0\} = \{u = 0\}$$

Improving the truncation error

$$G_t * d(0,y) = y + \kappa t - \kappa^2 y t + \frac{1}{2}(\kappa_{xx} + \kappa^3)t^2 + \mathcal{O}(t^3),$$

$$\tilde{G}_t * d := (2G_t - \frac{1}{2}G_{2t}) * d = \frac{3}{2}y + \kappa t + \kappa^2 y t + \mathcal{O}(t^3)$$

$$\begin{aligned} K_{\delta t}(0,y) &= \tilde{G}_{\delta t} * d - \frac{d}{\delta t}(\tilde{G}_{\delta t} * d - \frac{3}{2}d)^2 \\ &= \frac{3}{2}y + \kappa \delta t + \mathcal{O}(\delta t^3). \end{aligned}$$

Essential for building higher order accuracy in the evolution.

Higher order time stepping

Resolution	# time steps	Relative error (%)	Order
32×32	10	0.98	–
64×64	20	0.41	1.25
128×128	40	0.19	1.11
256×256	80	0.093	1.03
512×512	160	0.045	1.05

Resolution	# of time steps	Relative error (%)	Order
32×32	10	0.17	–
64×64	20	0.047	1.85
128×128	40	0.013	1.85
256×256	80	0.0033	1.98
512×512	160	0.00086	1.94

Results from shrinking a circle.

Curvature dependent motions

In a suitable coordinate system, near d(0,0)=0:

$$u(x,y) := K_{\delta t}[d] \quad s.t. \qquad u(0,y) = y + f(\kappa)dt$$

interface {u=0} moves by f(k)dt

Use the expansion:

$$G_{\delta t} * d(0,y) = y + \kappa\delta t - \kappa^2 y\delta t + \frac{1}{2}(\kappa_{xx} + \kappa^3)\delta t^2 + \mathcal{O}(\delta t^3)$$

$$u := d + f(\frac{G_{M\delta t} * d - d}{M\delta t})\delta t$$

$$\uparrow$$

$$\kappa + \mathcal{O}(\delta t)$$

Motion by f(k)

1. Form the function

$$A(\mathbf{x}) := d_j + (\delta t) f\left(\frac{1}{M(\delta t)}\Big\{G_{M(\delta t)} * d_j - d_j\Big\}\right) \tag{83}$$

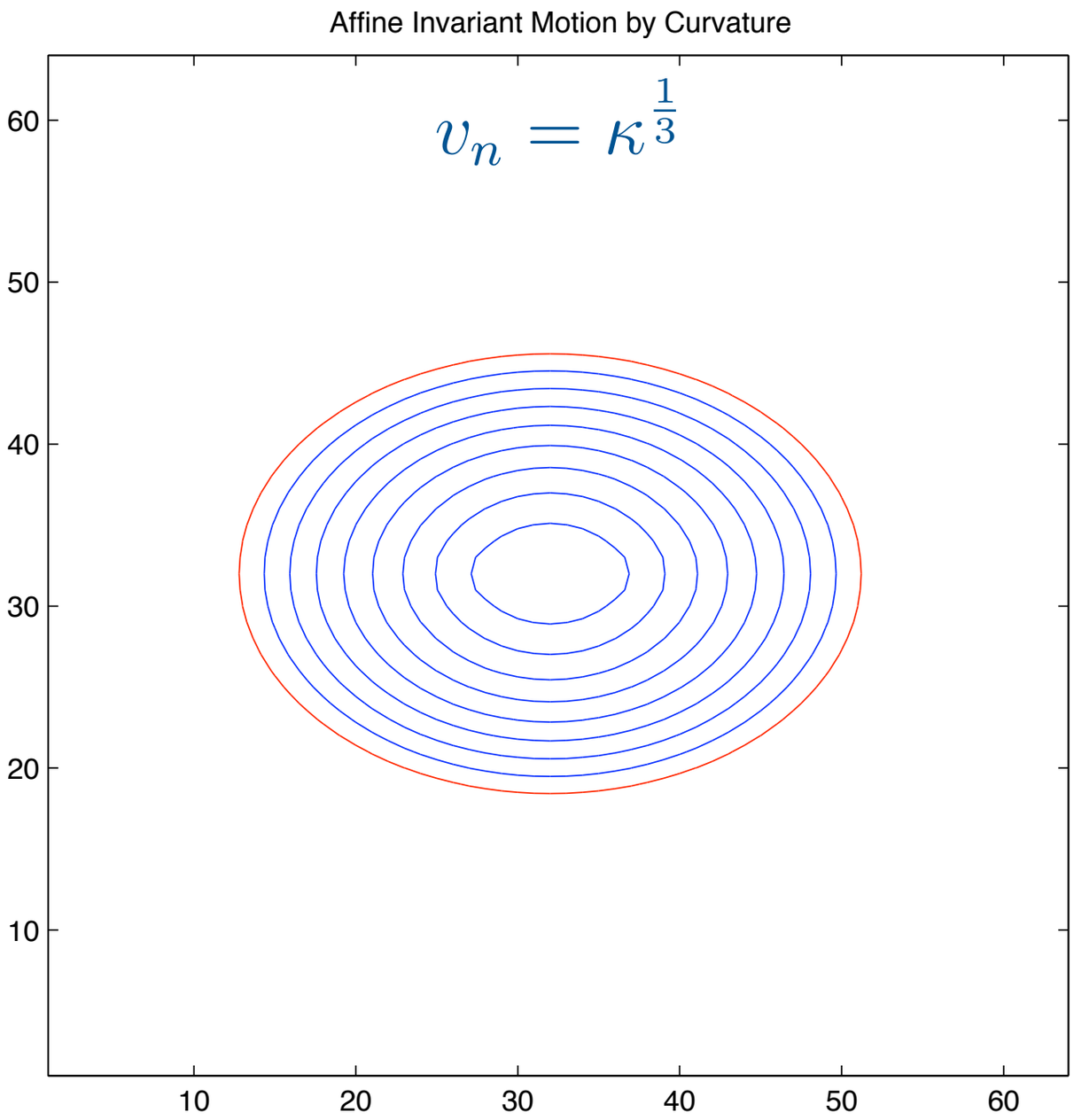
2. Construct distance function d_{j+1} by

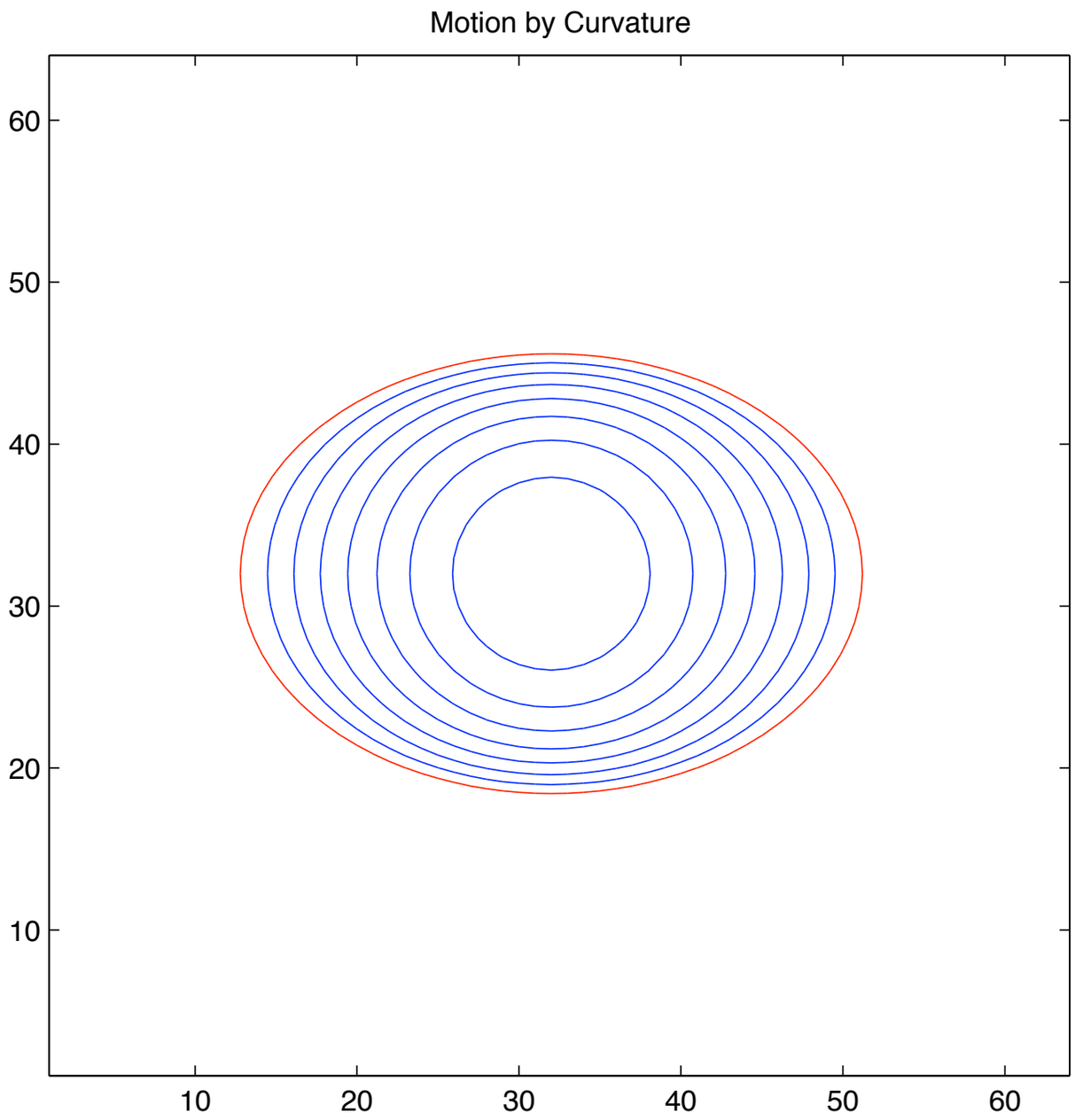
$$d_{j+1}(\mathbf{x}) = \mathbf{Redist}(A). \tag{84}$$

Proposition: f odd, increasing, and Lipschitz

If $M \geq L_f$, then algorithm (83) and (84) is monotone for any choice of time step size $\delta t > 0$.

Affine curvature motion





Notice the very coarse grid used in simulations.

Numerical convergence and accuracy

[Alvarez-Guichard-Lions-Morel]:

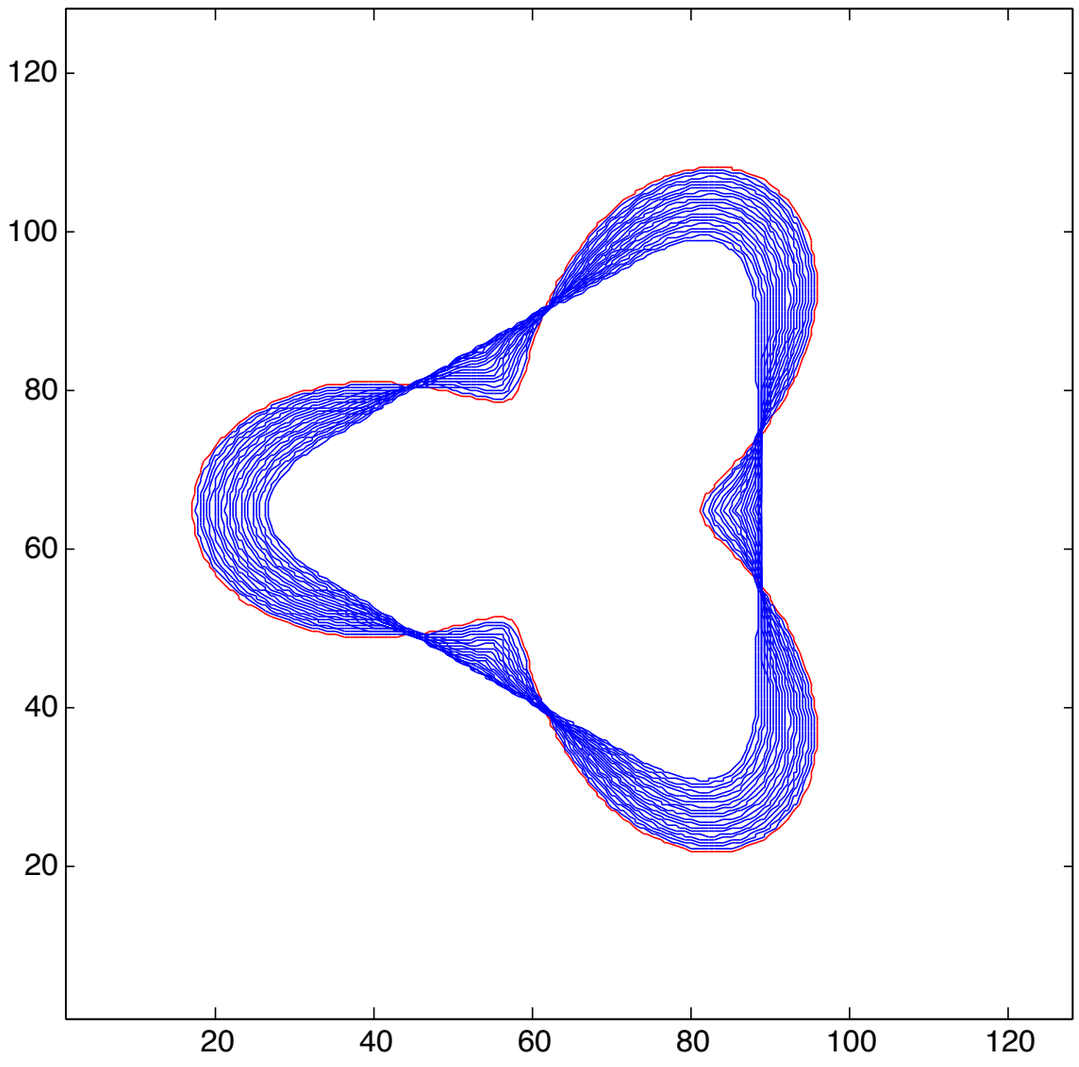
$$v_n = \kappa^{\frac{1}{3}}$$

Shrinking circle: $$r(t) = \left(r_0^{4/3} - \frac{4}{3} t \right)^{\frac{3}{4}} .$$

Refinement	s=1	2	4	8	16	32
Abs.err (1st order LTE)	1.5877e-3	7.6315e-4	3.4621e-4	1.6191e-4	7.5632e-5	3.3652e-5
Abs.err (2nd order LTE)	2.4445e-4	-2.7820e-4	-1.4523e-4	-6.0250e-5	-3.0562e-5	-1.8146e-5

Affine curvature motion

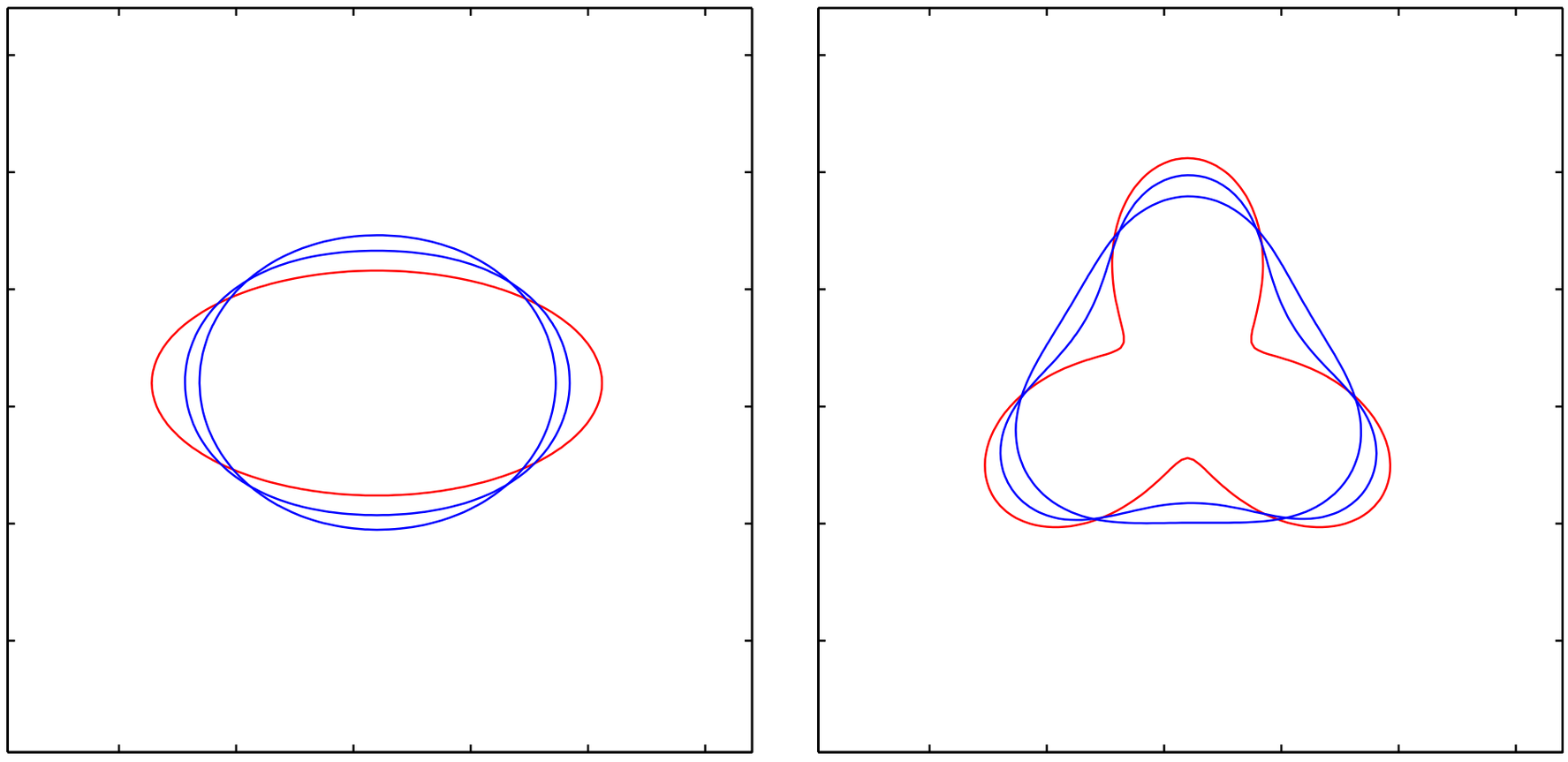
$$v_n = \kappa^{\frac{1}{3}}$$



Algorithm seems stable at inflection point

Surface diffusion

$$v_n = -\kappa_{xx}$$



Change in area at final time is around 4.25% with 4000 steps on a128x128 grid.

A word on complexity

- NxN grid
- Diffusions: $\mathcal{O}(N)$ or $\mathcal{O}(N \log N)$
- Highly accurate redistancing: $\mathcal{O}(N)$ or $\mathcal{O}(N \log N)$
- Solution in 0<t<T: T/δt. $M = T/\delta t \implies \mathcal{O}(MN^3 \log N)$

Second word on complexity

- Total cost: $\mathcal{O}(\frac{T}{\delta t} \cdot N^3 \log N)$

- How small should δt be?

  $\text{Err}=\mathcal{O}(\delta t^2)$

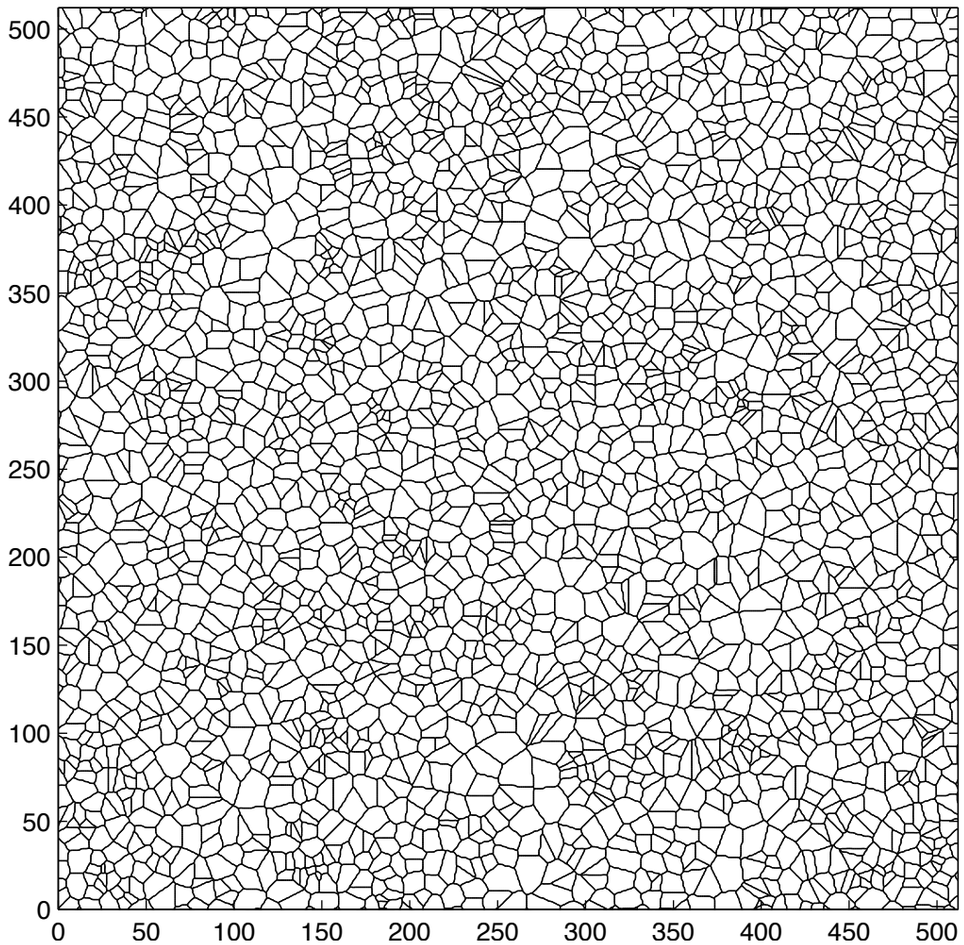
  $\text{Prescribed err}=\mathcal{O}(\Delta x^p) = \mathcal{O}(\delta t^2) \implies M = TN^{\frac{p}{2}}$

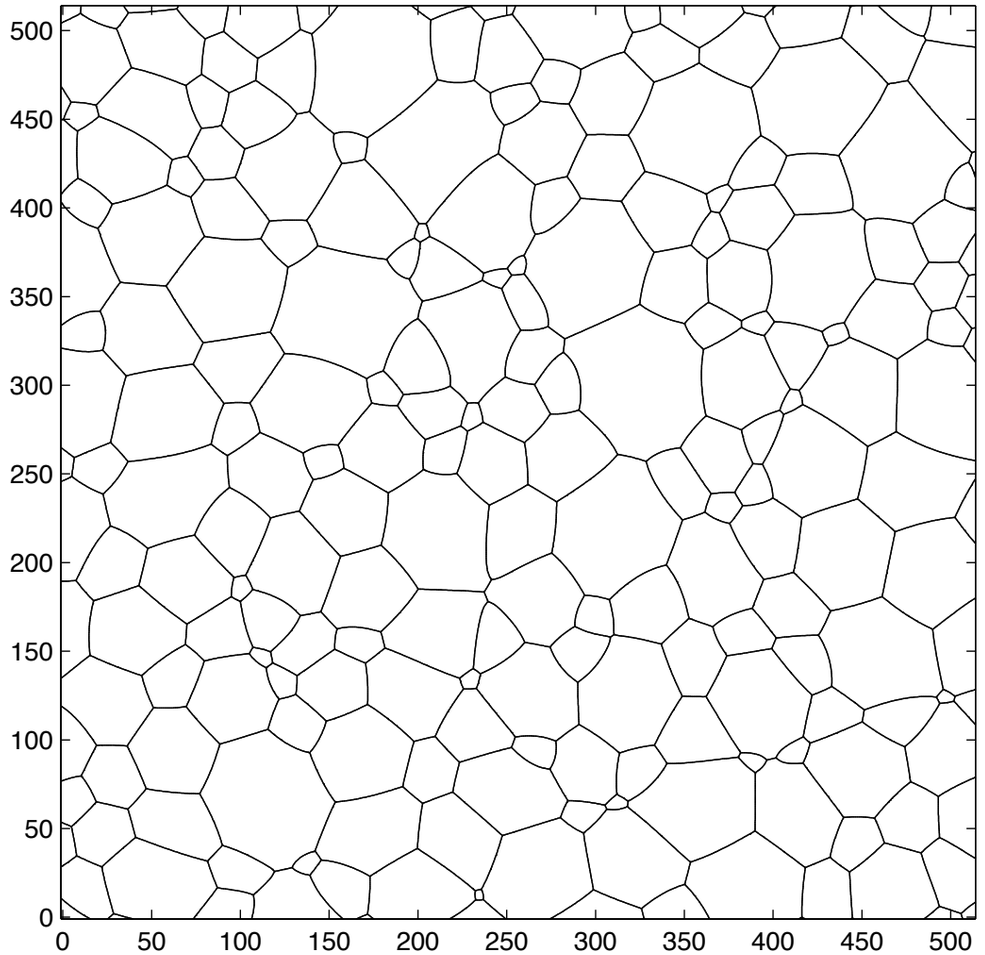
- Explicit Willmore flow:

  $\Delta t \leq C_0 \Delta x^4 \implies \mathcal{O}(N^4 \cdot N^2)$

Curvature motion of multiple junctions

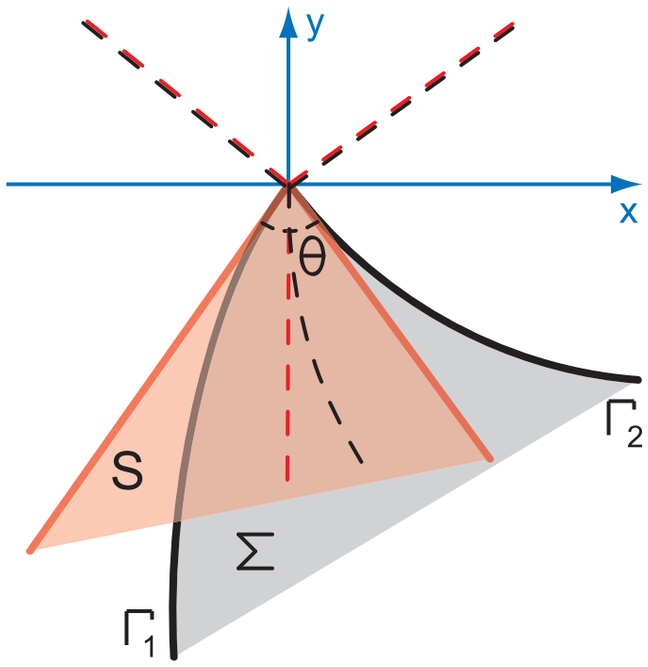
(ex: large scale coarse graining)

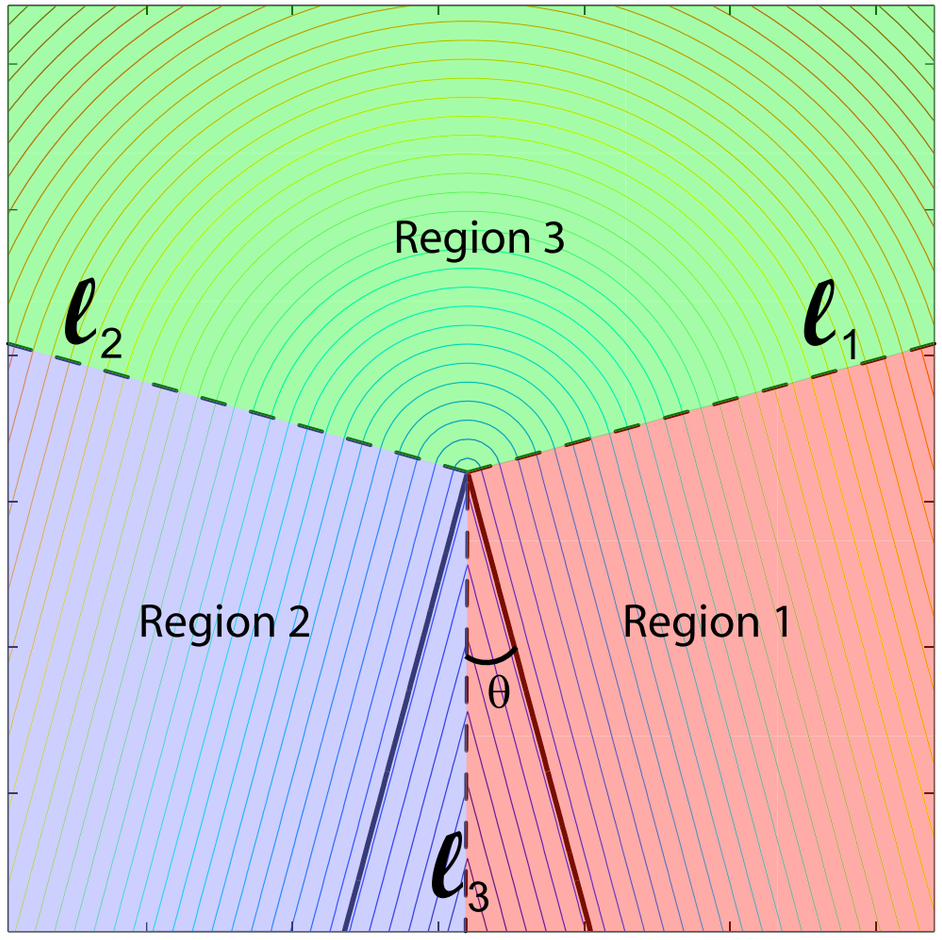




[Elsey, Esedoglu, Smereka, 2009]

The distance function to a corner





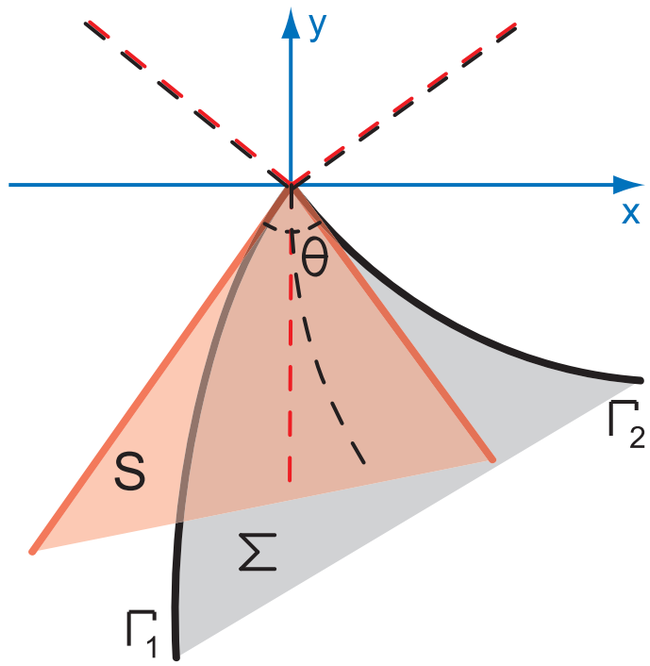
Two C^2 curves meeting at the origin.

Approx. the signed distance function by that of a sector.

$$|d(\mathbf{x}) - \tilde{d}(\mathbf{x})| = \mathcal{O}(|\mathbf{x}|^2) \quad \text{as} \quad \mathbf{x} \longrightarrow \mathbf{0}.$$

Approximation of corners

$$
\begin{aligned}
|d * G_t(\mathbf{0}) - \tilde{d} * G_t(\mathbf{0})| &\leq \int_{\mathbb{R}^2} |(d - \tilde{d})(-\mathbf{x})| G_t(\mathbf{x})\, d\mathbf{x} \\
&\leq \int_{\mathbb{R}^2} O(|\mathbf{x}|^2) G_t(\mathbf{x})\, d\mathbf{x} \\
&= O(t) \text{ as } t \to 0.
\end{aligned}
$$



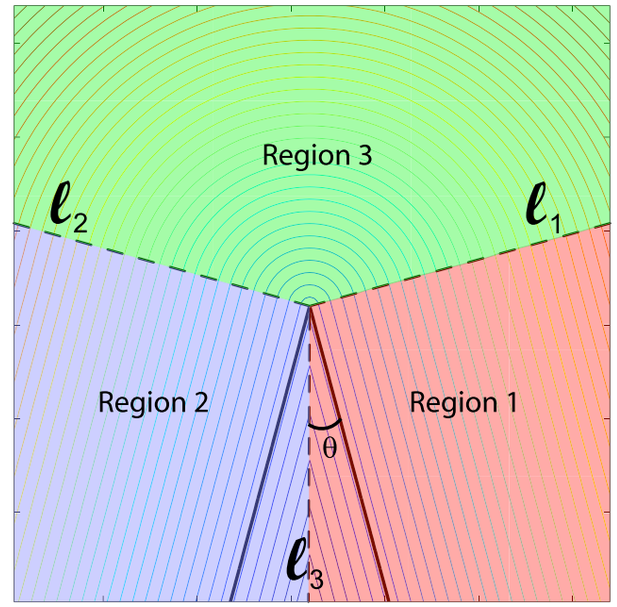
$$
\begin{aligned}
\left| \frac{\partial}{\partial x_j}(d * G_t)(\mathbf{0}) - \frac{\partial}{\partial x_j}(\tilde{d} * G_t)(\mathbf{0}) \right| &= \left| \left(d * \partial_{x_j} G_t \right)(\mathbf{0}) - \left(\tilde{d} * \partial_{x_j} G_t \right)(\mathbf{0}) \right| \\
&\leq \int_{\mathbb{R}^2} O(|\mathbf{x}|^2) \left| \partial_{x_j} G_t \right| d\mathbf{x} \\
&= O(\sqrt{t}) \text{ as } t \to 0.
\end{aligned}
$$

$$
\left| \frac{\partial^2}{\partial x_i \partial x_j}(d * G_t)(0,0) - \frac{\partial^2}{\partial x_i \partial x_j}(\tilde{d} * G_t)(0,0) \right| = \mathcal{O}(1)
$$

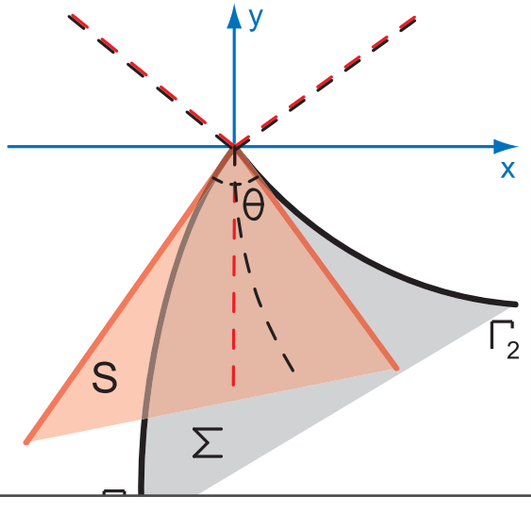
Expansion near a corner

$$\int_{\mathbb{R}^2} d(x',y')G_t(x-x',y-y')\,dx'\,dy'$$

$$=\frac{\sqrt{t}}{\pi}\left(\theta-\frac{\pi}{2}-\cos\theta+C_1(t)\right)$$
$$+C_2(t)\,x$$
$$+\frac{1}{2\pi}\left(4\cos^3\theta-\pi\sin\theta-2\theta\sin\theta-6\cos\theta+C_3(t)\right)y$$
$$+\frac{1}{\sqrt{t}}\frac{1}{16\sqrt{\pi}}\left(2\theta-4\cos\theta-\sin 2\theta-\pi+C_3(t)\right)x^2$$
$$+\frac{1}{\sqrt{t}}\frac{1}{16\sqrt{\pi}}\left(\sin 2\theta+2\theta-\pi+C_4(t)\right)y^2$$
$$+\frac{1}{\sqrt{t}}C_5(t)xy+H.O.T.$$



The coefficients $C_j(t)$ *satisfy* $C_j(t)=O(\sqrt{t})$ *as* $t\to 0^+$.



Algorithm for multiple junctions

1. Form the convolutions

$$L_j^k := K_{\delta t} * d_j^k \tag{92}$$

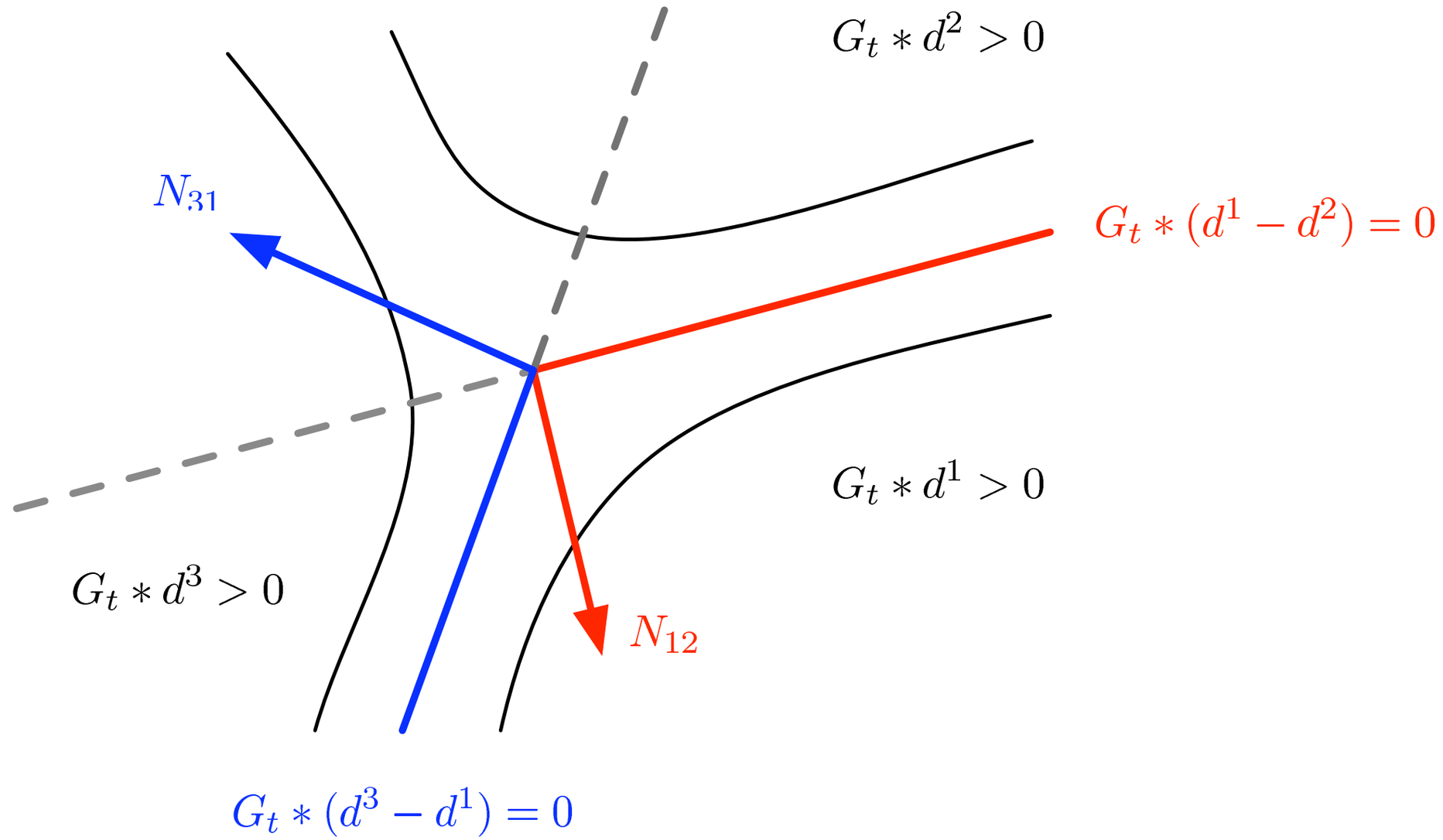
for $k = 1, \ldots, m$ where K_t is one of the kernels:

$$K_t = G_t \text{ or } K_t = \frac{1}{3}\Big(4G_{\frac{3}{2}t} - G_{3t}\Big).$$

2. Construct the signed distance functions d_{j+1}^k for $k = 1, \ldots, m$ according to

$$d_{j+1}^k = \mathbf{Redist}\Big(L_j^k - \max\{L_j^\ell \,:\, \ell \neq k\}\Big). \tag{93}$$

Phase re-assignment



[Baldo, Bronsard et al, Osher et al.]

Where does the junction go?

Phase reassignment: $d^1 \longleftarrow d^1 * G_t - \max(d^2 * G_t, d^3 * G_t)$

$$\begin{aligned}
\left(d^1 * G_t - d^2 * G_2\right)(0, y) &= \left(A(\theta_1) - A(\theta_2)\right)\sqrt{t} \\
&+ \left(B(\theta_1) - \cos(\theta_1 + \theta_2) B(\theta_2)\right) y \\
&+ \frac{1}{\sqrt{t}} \left(Q(\theta_1) - Q(\theta_2)\right) y^2 \\
&+ \mathcal{O}(t)
\end{aligned}$$

New interface location along y-axis:

$$\theta_1 \neq \theta_2 : y^2 + b\sqrt{t}\, y + a\, t = 0 \qquad y \simeq C_0 \sqrt{t} = \left(C_0 \frac{1}{\sqrt{t}}\right) t$$

v

Where does the junction go?

New triple junction location:

$$(d^1 * G_t)(x_*, y_*) = (d^2 * G_t)(x_*, y_*) = (d^3 * G_t)(x_*, y_*)$$

Near the 'Mercedes' angle:

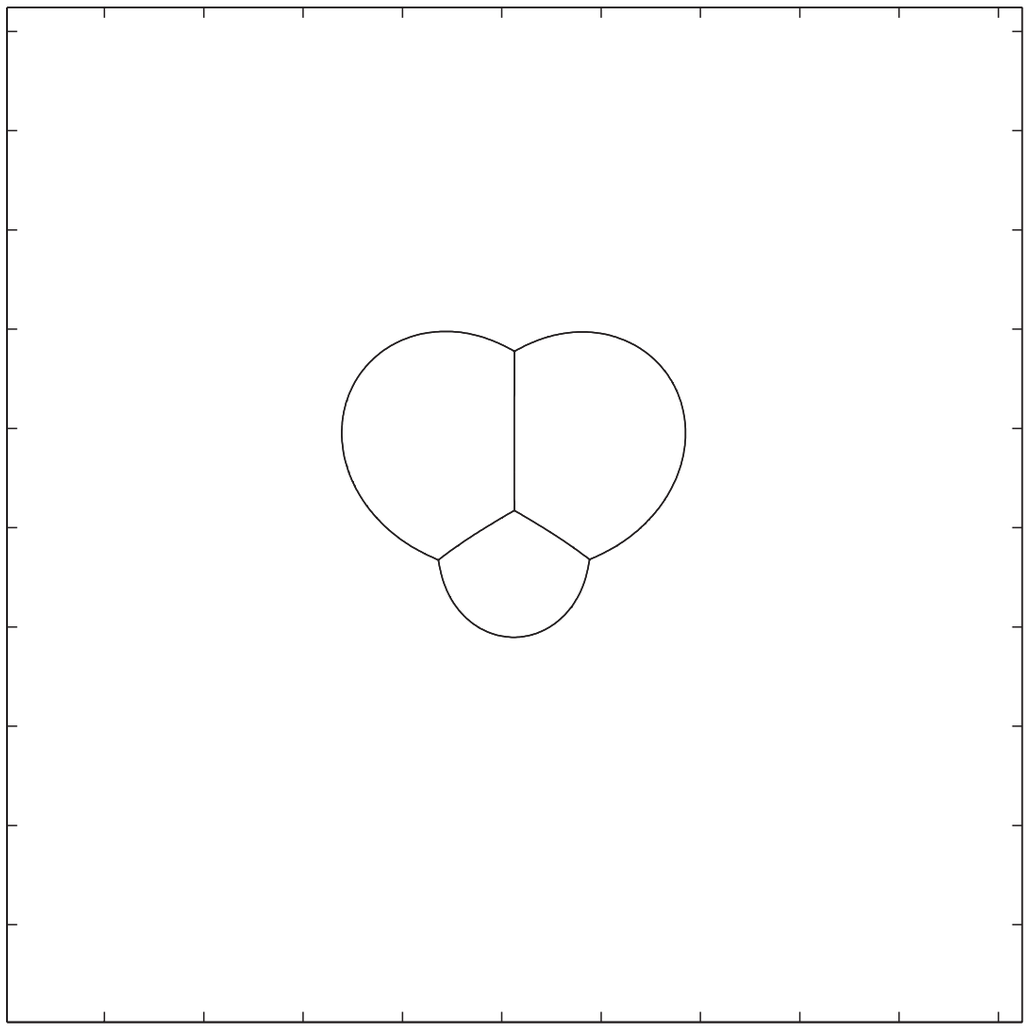
$$x_* = -\frac{\sqrt{3}}{3}\frac{A'(\frac{\pi}{3})}{B(\frac{\pi}{3})}\sqrt{t}\big(2\theta_2 + \theta_1 - \pi\big) + \text{H.O.T.}$$

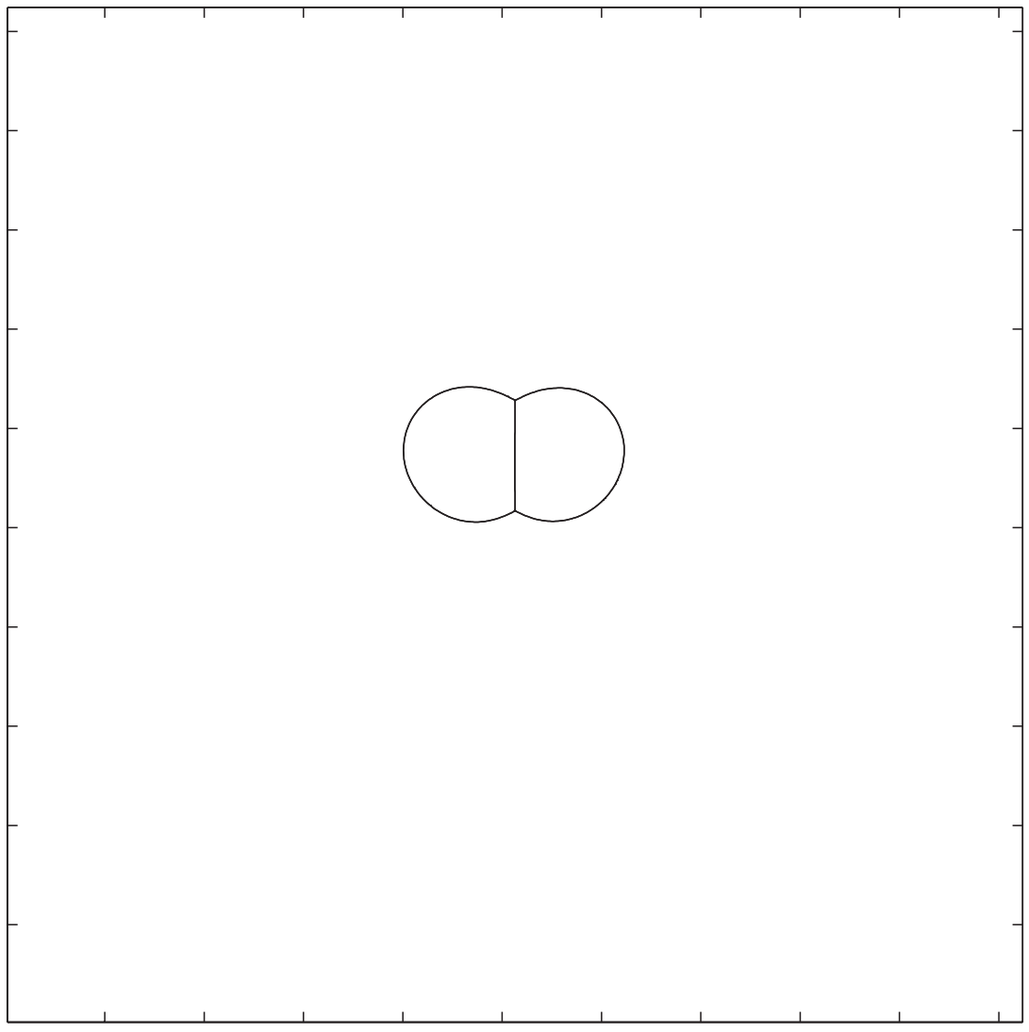
$$= \frac{2(3+2\sqrt{3})}{5(3+\pi\sqrt{3})}\sqrt{t}\big(2\theta_2 + \theta_1 - \pi\big) + \text{H.O.T.}$$

$$y_* = -\frac{A'(\frac{\pi}{3})}{B(\frac{\pi}{3})}\sqrt{t}\left(\theta_1 - \frac{\pi}{3}\right) + \text{H.O.T.}$$

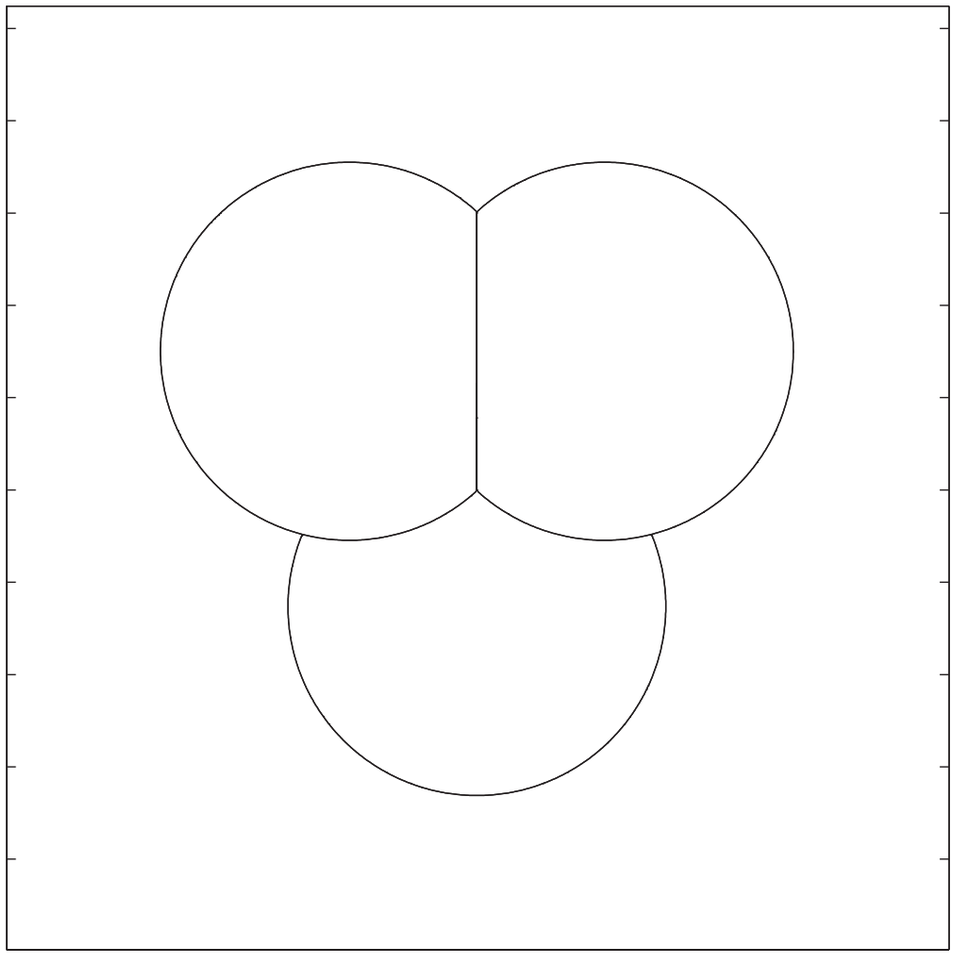
$$= \frac{6(2+\sqrt{3})}{5(3+\pi\sqrt{3})}\sqrt{t}\left(\theta_1 - \frac{\pi}{3}\right) + \text{H.O.T.}$$

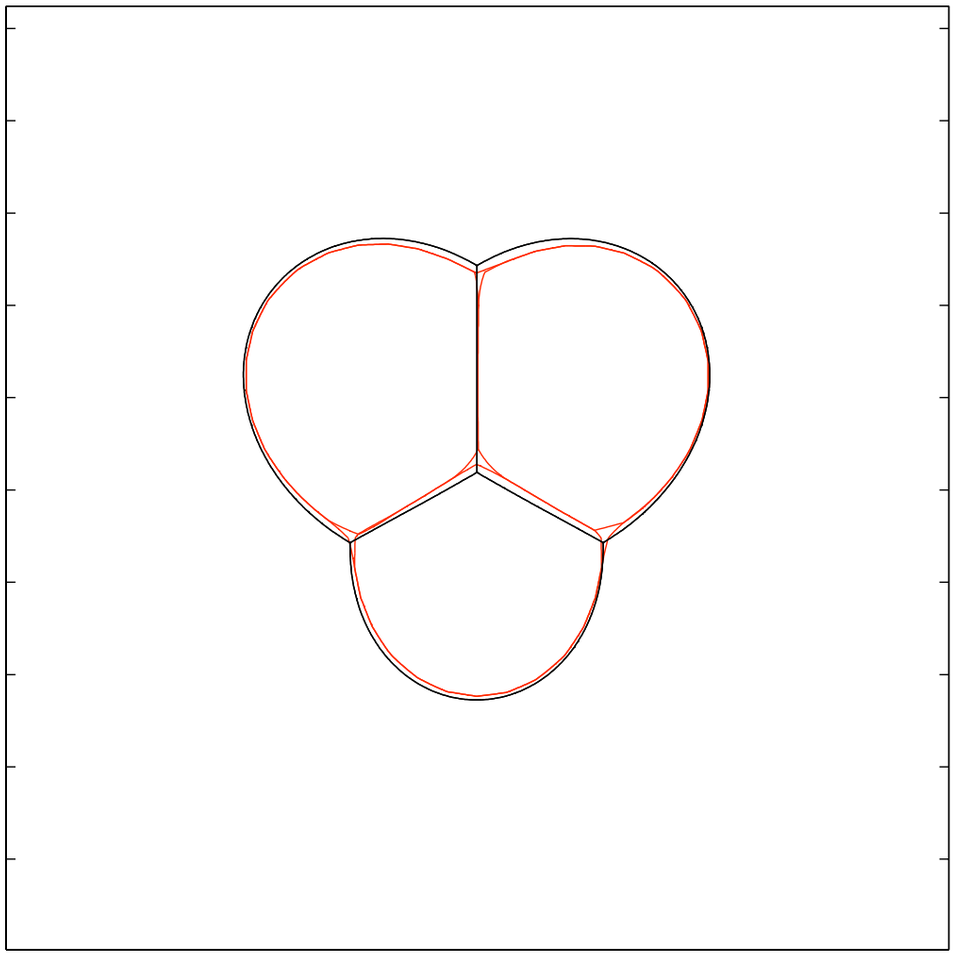
Topological changes





Good resolution on coarse grids

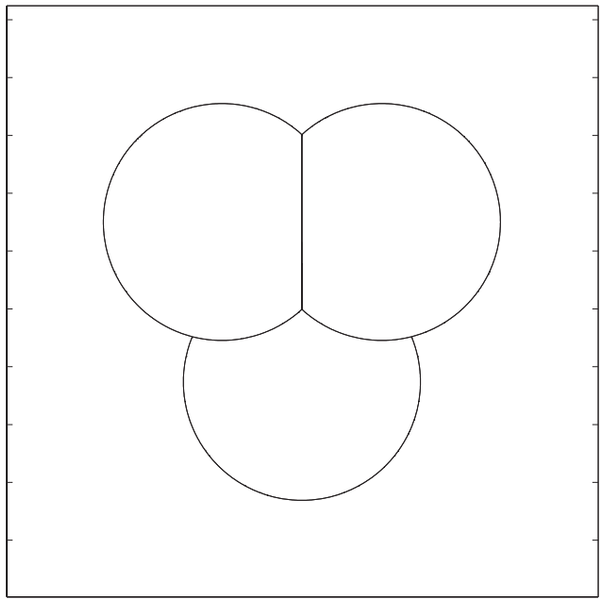


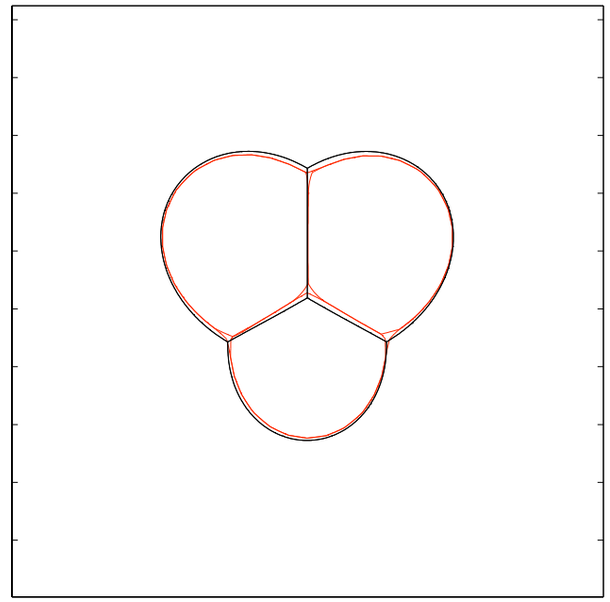


30 steps on a 32x32 grid v.s. 480 steps on a 512x512 grid.

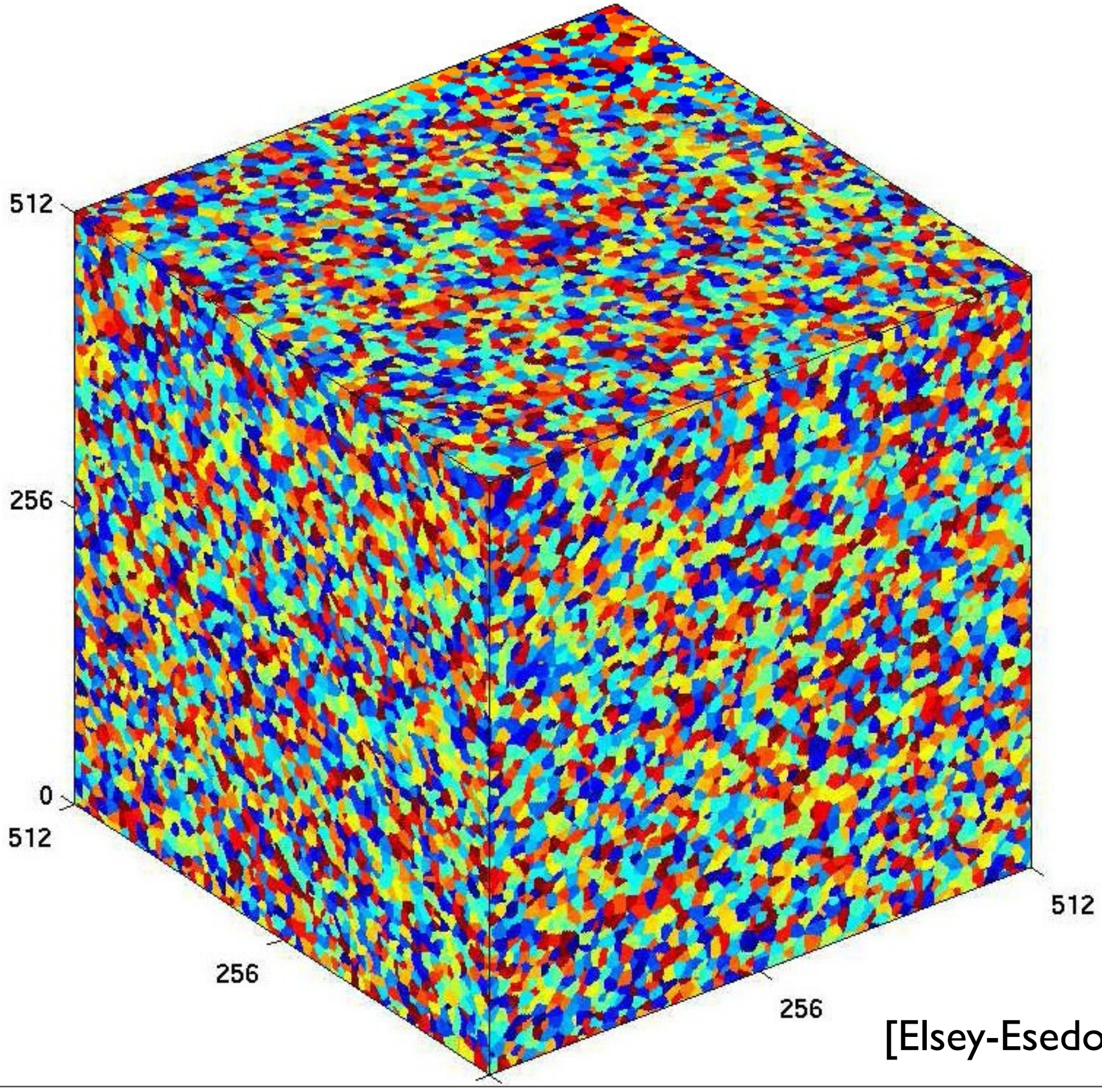
Accuracy w.r.t von Neumann Law

$$\frac{d}{dt}A(t) = \frac{\pi}{3}(n-6) = -\pi$$

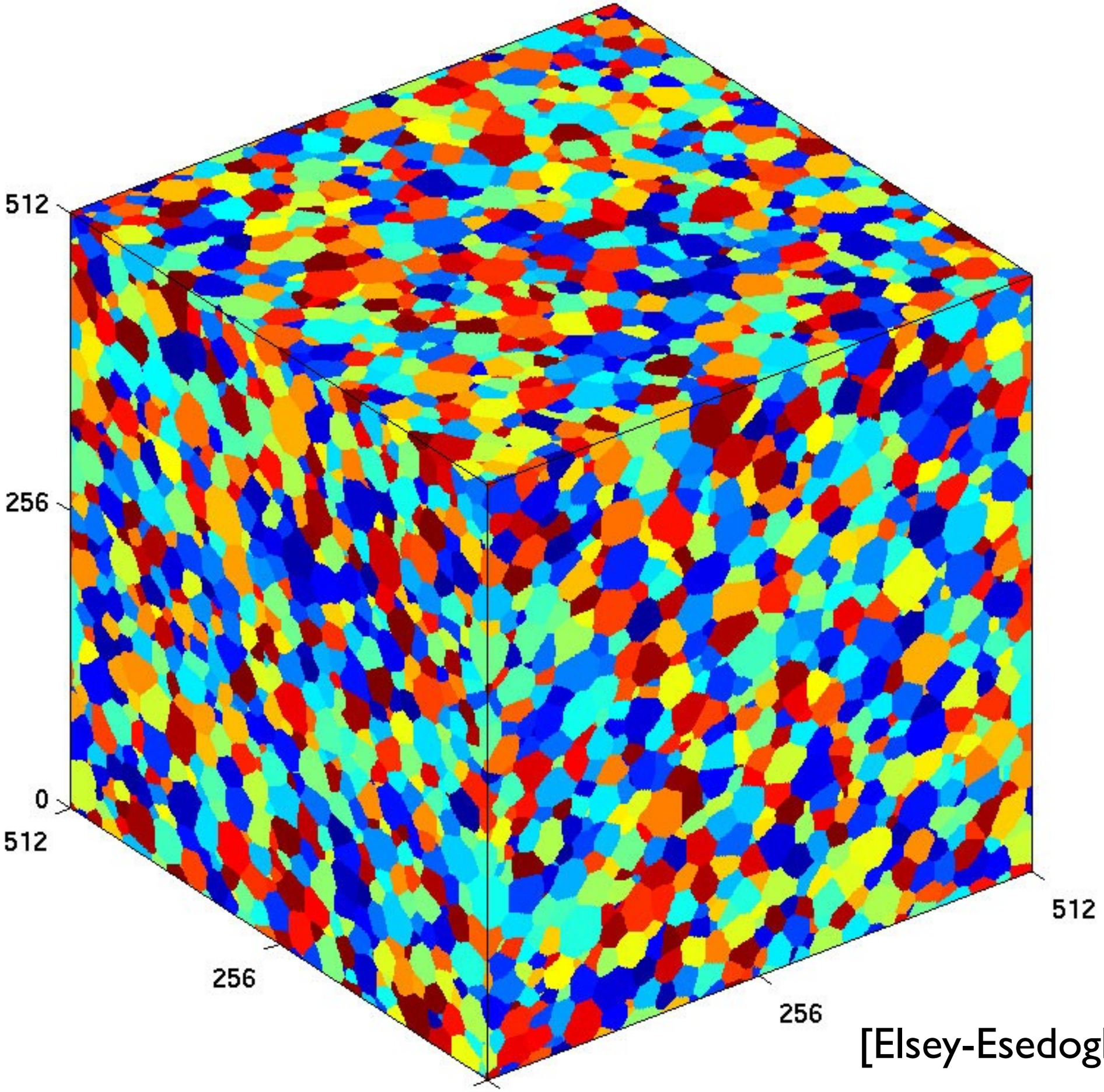




Resolution	# of Time Steps	Relative Error	Order
32×32	30	3.91%	–
64×64	60	2.07%	0.918
128×128	120	1.28%	0.693
256×256	240	0.84%	0.608
512×512	480	0.53%	0.664



[Elsey-Esedoglu-Smereka]



[Elsey-Esedoglu-Smereka]

Anisotropic MCF

- Almgren-Taylor-Wang's discrete time variational formulation:

$$\mathcal{E}(C) = \int_C \gamma(\hat{n}_C) ds$$

$$\min_f \mathcal{E}(f(C)) - \mathcal{E}(C) + \frac{1}{2h}\langle f, f\rangle$$

- Chambolle's formulation:

$$u^{n+1} = \arg\min_u \int \gamma(\nabla u) + \frac{1}{2\Delta t}\int (u - \mathrm{dist}_{\{u^n=0\}})^2$$

Split Bregman Method (I)

$$\min_u \int \gamma(\nabla u) + \frac{1}{2h} \int (u - f)^2 \qquad f = d_{\Gamma(t_n)}$$

$$\longrightarrow \quad \min_{u,d} \int \gamma(d) + \frac{1}{2h} \int (u - f)^2 \quad s.t. \quad d = \nabla u$$

Bregman iterative algorithm:

Constrained minimization solved by a sequence of unconstrained problems:

$$\min_{u,d} \int \gamma(d) + \frac{1}{2h} \int (u - f)^2 + \frac{\lambda}{2} \int |d - \nabla u - b^k|^2$$

$$\longrightarrow \quad u^{k+1}, d^{k+1} \rightarrow b^{k+1}$$

Split Bregman Method (II)

$$\min_{u,d} \int \gamma(d) + \frac{1}{2h} \int (u-f)^2 + \frac{\lambda}{2} \int |d - \nabla u - b^k|^2$$

Iterate the alternative minimizations of u and d:

$$u^{k+1} \leftarrow u - h\lambda\Delta u = f + h\lambda\nabla \cdot (d^k - b^k)$$

Optimal solution for linear diffusion type equations

$$d^{k+1} \leftarrow \min \int \gamma(d) + \frac{\lambda}{2}||d - \nabla u - b^k||^2$$

derive simple update formula

$$b^{k+1} = b^k + (\nabla u^{k+1} - d^{k+1})$$

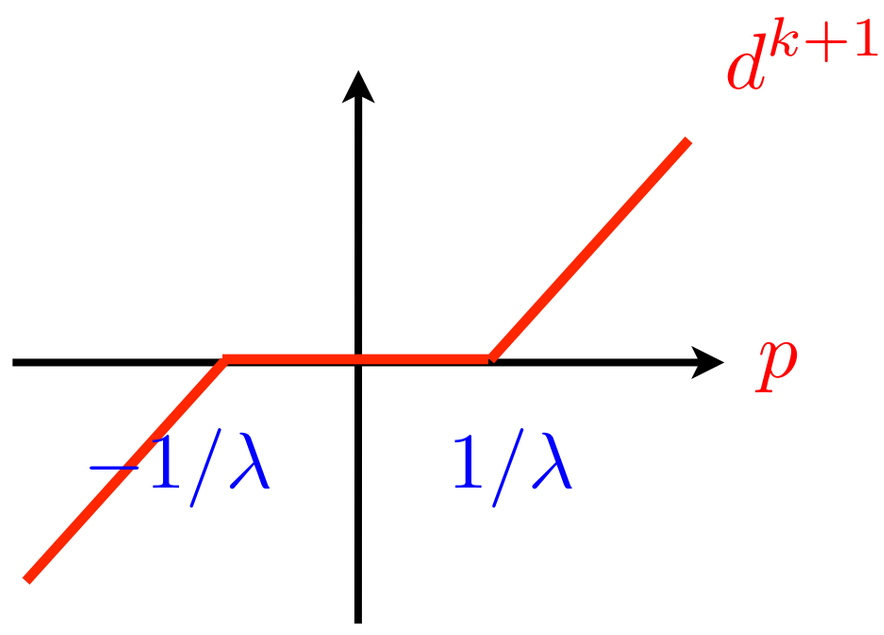
Shrinkage formulae (I)

$$d^{k+1} \leftarrow \min_d \int \gamma(d) + \frac{\lambda}{2}||d-p||^2$$

Pointwise shrinkage problem: $\min_{d \in \mathbb{R}^d} \gamma(d) + \frac{\lambda}{2}|d-p|^2$

$|\gamma(d)| = |d|$ (1-D)

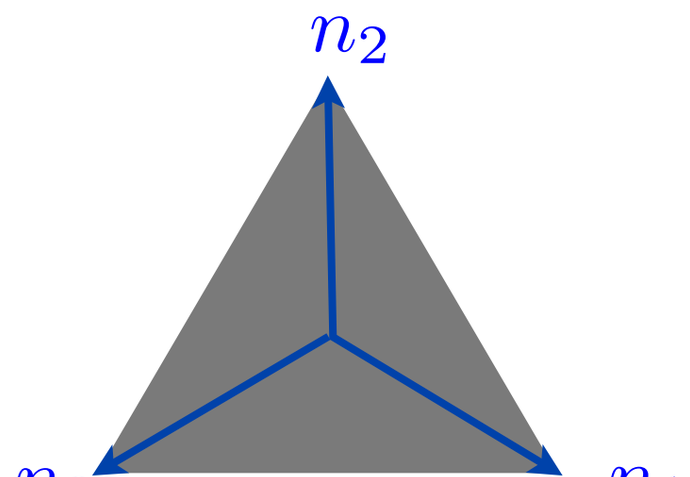
$d^{k+1} = \text{shrink}(p, 1/\lambda)$



Shrinkage formulae (II)

$$d^*(p) = \arg\min_d \{\gamma(d) + \frac{1}{2}|d-p|^2\}$$

$$\gamma(d) = \max_i \ \{n_i \cdot d\} \iff \gamma(d) = \sup_{q \in W} d \cdot q$$



$$\min_d \{\gamma(d) + \frac{1}{2}|d-p|^2\}$$

$$\Longrightarrow \min_{d \in \mathbb{R}^n} \max_{q \in W} \{d \cdot q + \frac{1}{2}|d-p|^2\}$$

$$\Longrightarrow \max_{q \in W} \min_{d \in \mathbb{R}^n} \{d \cdot q + \frac{1}{2}|d-p|^2\} \qquad \text{(by convexity)}$$

$$d = p - q$$

$$\Longrightarrow \max_{q \in W} \{-\frac{1}{2}|q|^2 + p \cdot q\} = \min_{q \in W} \{\frac{1}{2}|q-p|^2\}$$

Shrinkage formulae (II)

$$d^*(p) = \arg\min_d \{\gamma(d) + \frac{1}{2}|d-p|^2\}$$

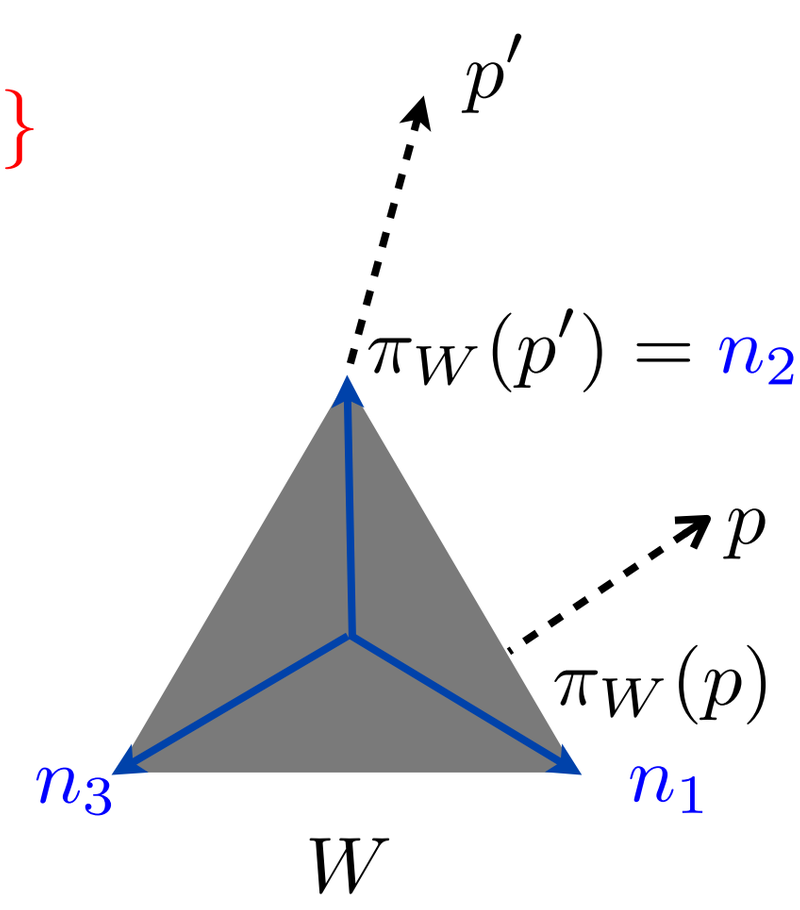
$$\gamma(d) = \max_i \ \{n_i \cdot d\} \iff \gamma(d) = \sup_{q \in W} d \cdot q$$

$$\Longrightarrow \max_{q \in W}\{-\frac{1}{2}|q|^2 + p \cdot q\} = \min_{q \in W}\{\frac{1}{2}|q-p|^2\}$$

$$d^*(p) = p - \pi_W(p)$$

Simple, explicit expression for the minimizer,
involving the vertices and the faces of the polyhedron

Fast and accurate numerical projection: [Tsai, 2002]



Crystalline MCF algorithm

$$\min_u \int \gamma(\nabla u) + \frac{1}{2h}\int (u-f)^2 \qquad f = d_{\Gamma(t_n)}$$

- Find the minimizer (u,d):

$$u^{k+1} \leftarrow u - h\lambda\Delta u = f + h\lambda\nabla\cdot(d^k - b^k)$$

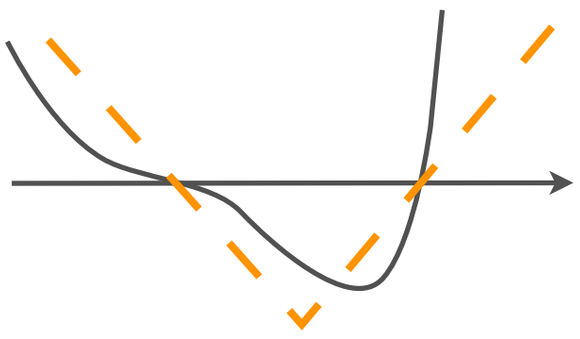
$$d^{k+1} \leftarrow \min \int \gamma(d) + \frac{\lambda}{2}||d - \nabla u - b^k||^2$$

$$d^*(p) = p - \pi_W(p)$$

$$b^{k+1} = b^k + (\nabla u^{k+1} - d^{k+1})$$

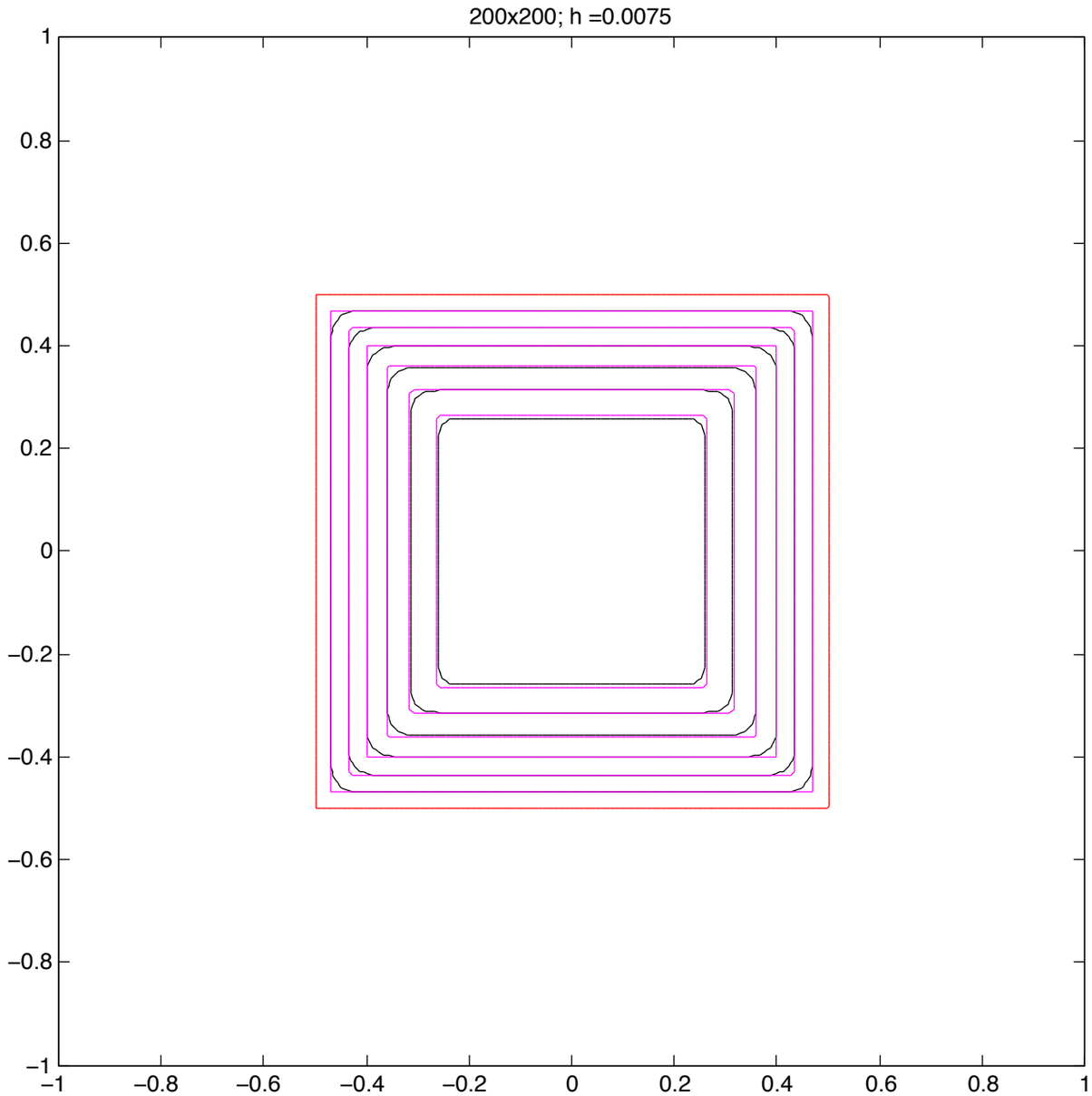
- Redistancing: $f := \text{redistance}(u^{k+1})$

$$\Gamma(t_{n+1}) := \{f = 0\}$$

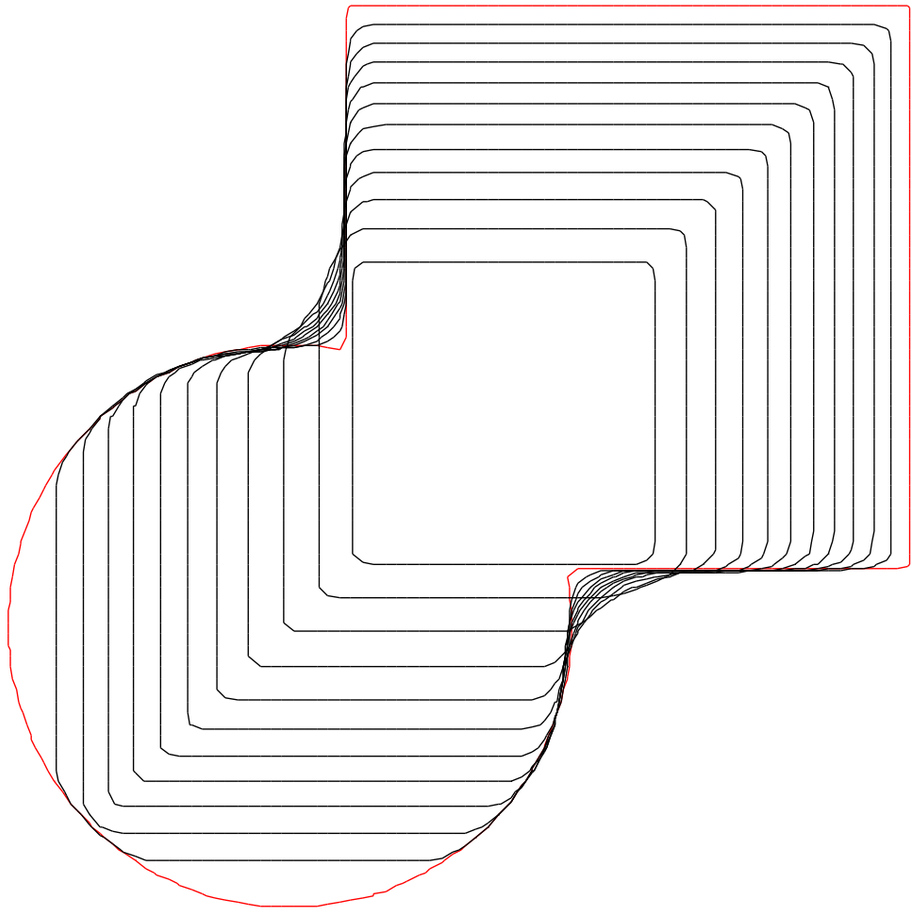


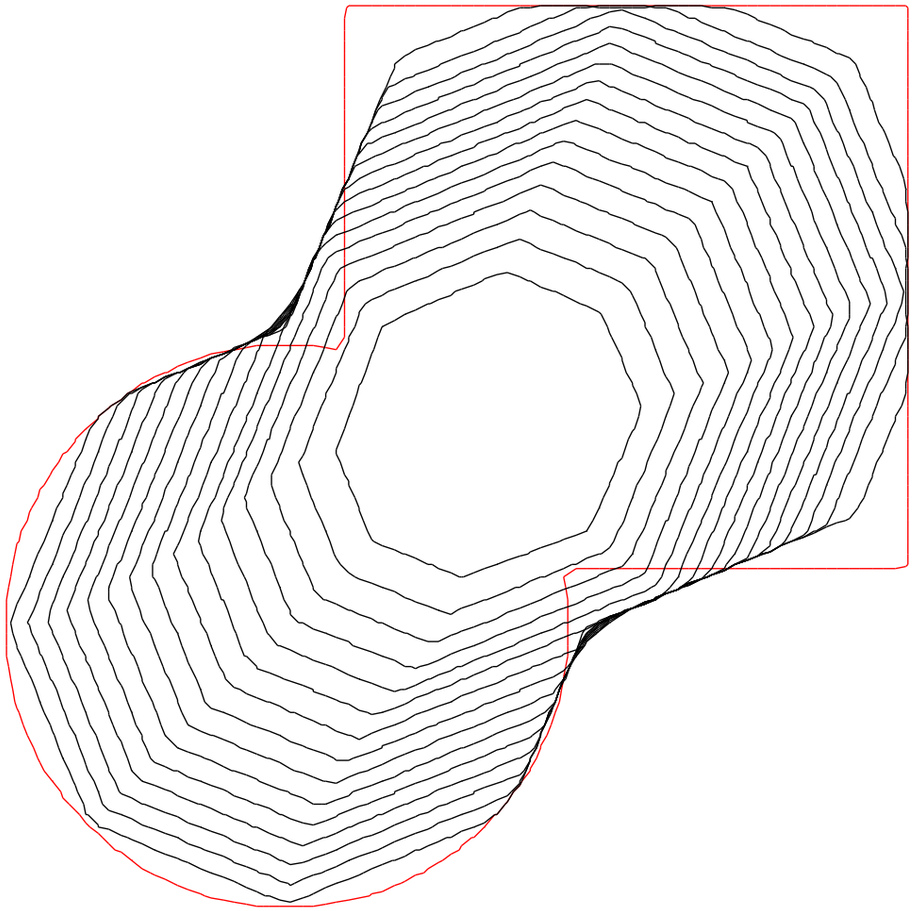
Fast, and high-order approach by eikonal flow: [Cheng-Tsai]

Result - self similar solutions

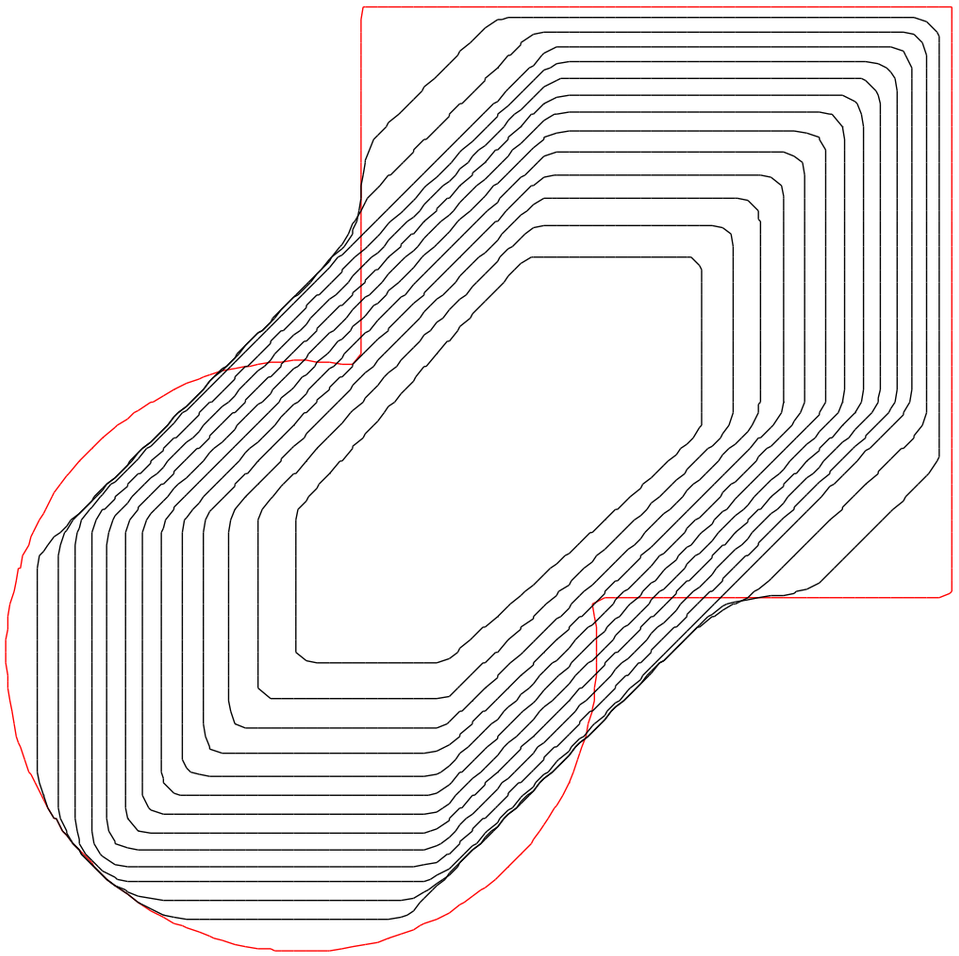


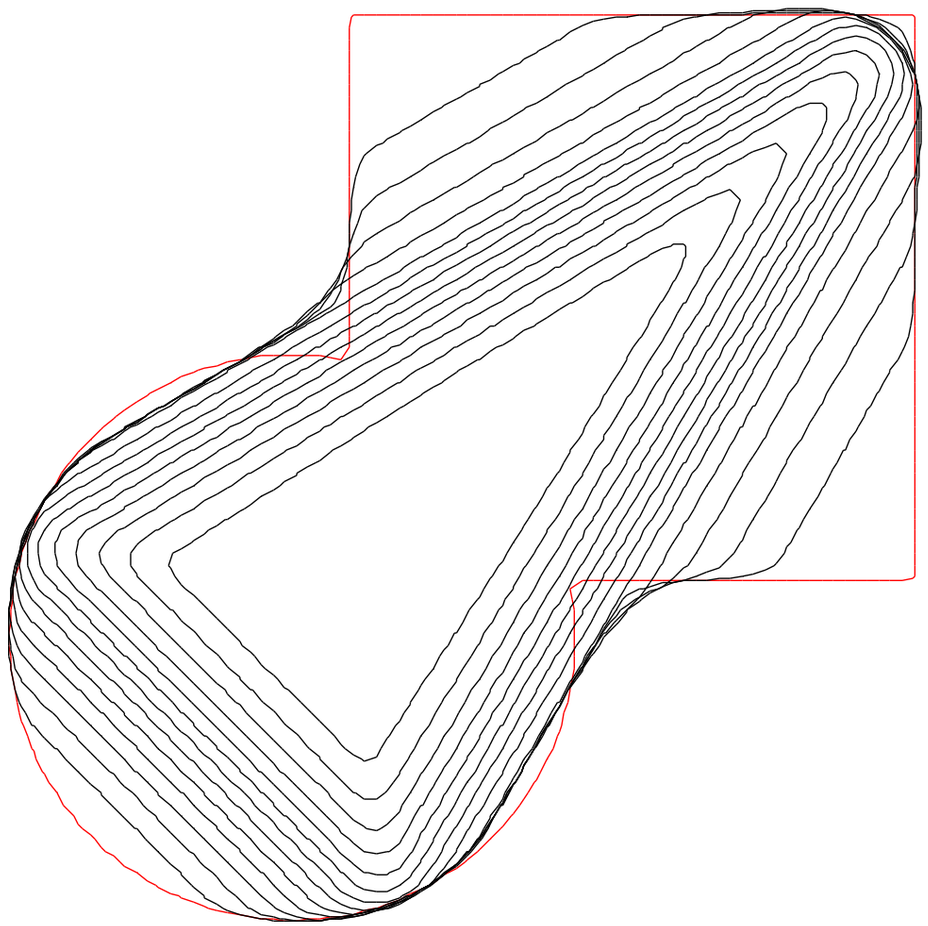
Results





Some results





Diffusion generated anisotropic motion

For smooth anisotropic curvatures:

- Step 1: diffuse $w^{n+\frac{1}{2}} = G_{\delta t}[w^n]$

- Step 2: threshold $w^{n+1} = \mathcal{T}[w^{n+\frac{1}{2}}]$

  "redistancing to a suitably weighted signed distance function"

- Iterate

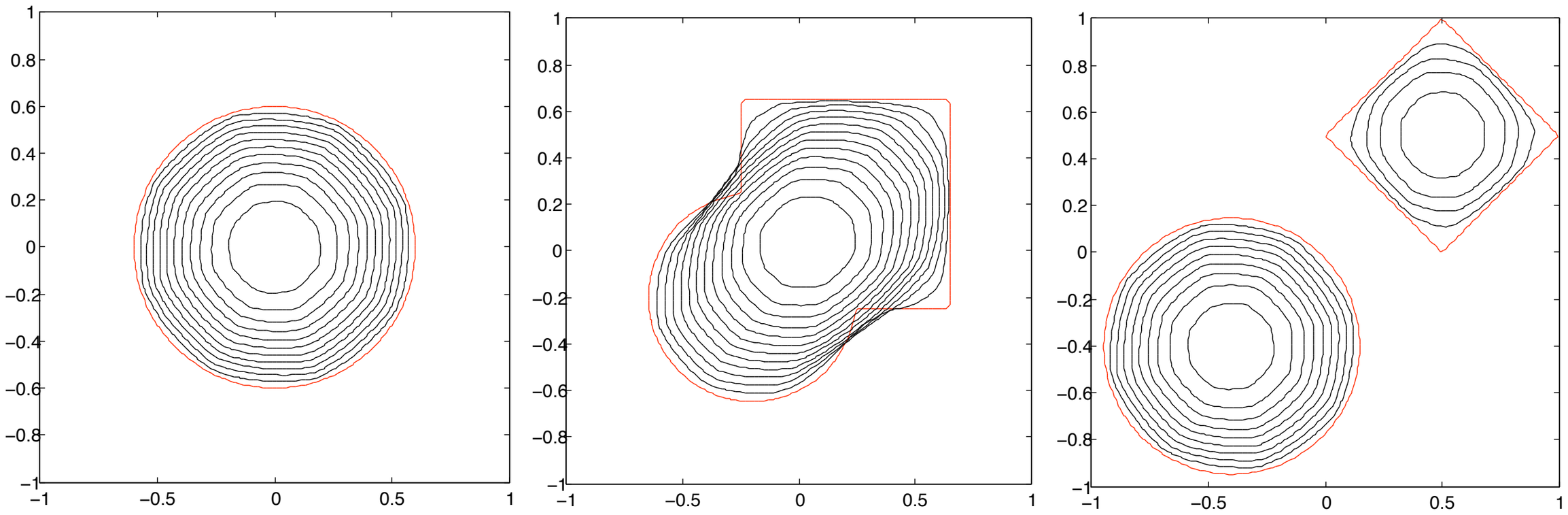
Smooth anisotropy

$$\gamma(\theta) = (64 + \sin(8\theta/50))$$

$$\frac{\partial \theta}{\partial s} = \frac{p^{\perp} \cdot \nabla u}{\gamma + \gamma''},$$

$$\frac{\partial u}{\partial s} = h(\Delta u - \nabla \cdot d),$$

$$R = p \cdot \nabla u, \qquad d = Rp.$$



Summary

- Almgren-Taylor-Wang, Chambolle algorithms: difficulty of solving nonlinear equations.
- MBO scheme: solve a simple linear heat equation
- Asymptotics for distance functions
- High resolution and accuracy algorithms
- Algorithm stable for large time steps
- Efficiency in solving linear diffusion equation
- Direct application to 3D