High-resolution finite element schemes for coupled problems

Matthias Möller

Institute of Applied Mathematics, LS III

Dortmund University of Technology, Germany

matthias.moeller@math.tu-dortmund.de

Joint work with Dmitri Kuzmin

Department of Mathematics

University of Houston, USA

kuzmin@math.uh.edu

Idealized Z-pinch implosion model *Banks, Shadid 2008*

- Generalized Euler system coupled with scalar tracer equation

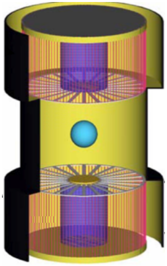
$$\frac{\partial}{\partial t}\begin{bmatrix}\rho \\ \rho\mathbf{v} \\ \rho E \\ \rho\lambda\end{bmatrix} + \nabla\cdot\begin{bmatrix}\rho\mathbf{v} \\ \rho\mathbf{v}\otimes\mathbf{v} + p\mathcal{I} \\ \rho E\mathbf{v} + p\mathbf{v} \\ \rho\lambda\mathbf{v}\end{bmatrix} = \begin{bmatrix}0 \\ \mathbf{f} \\ \mathbf{f}\cdot\mathbf{v} \\ 0\end{bmatrix}$$

- Non-dimensional Lorentz force

$$\mathbf{f} = (\rho\lambda)\left(\frac{I(t)}{I_{\max}}\right)^2 \frac{\hat{\mathbf{e}}_r}{r_{\text{eff}}}$$

$$I(t) = \sqrt{12(1-t^4)t^2}$$

$$r_{\text{eff}} = \max\{r/R_0, r_{\min}\}$$



Idealized Z-pinch implosion model *Banks, Shadid 2008*

- Generalized Euler system coupled with scalar tracer equation

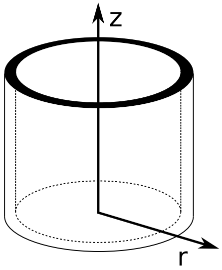
$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho\mathbf{v}\\ \rho E\\ \rho\lambda\end{bmatrix} + \nabla\cdot\begin{bmatrix}\rho\mathbf{v}\\ \rho\mathbf{v}\otimes\mathbf{v}+p\mathcal{I}\\ \rho E\mathbf{v}+p\mathbf{v}\\ \rho\lambda\mathbf{v}\end{bmatrix} = \begin{bmatrix}0\\ \mathbf{f}\\ \mathbf{f}\cdot\mathbf{v}\\ 0\end{bmatrix}$$

- Non-dimensional Lorentz force

$$\mathbf{f} = (\rho\lambda)\left(\frac{I(t)}{I_{\max}}\right)^2\frac{\hat{\mathbf{e}}_r}{r_{\text{eff}}}$$

$$I(t) = \sqrt{12(1-t^4)t^2}$$

$$r_{\text{eff}} = \max\{r/R_0, r_{\min}\}$$



Outline

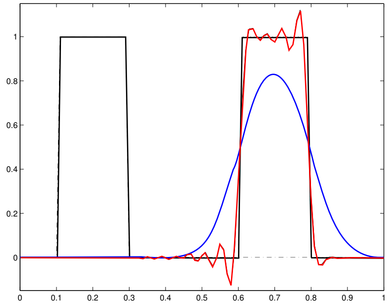
Numerical troubles

Convection in 1D

$$\begin{cases} \dfrac{\partial u}{\partial t} + v\dfrac{\partial u}{\partial x} = 0, & v > 0 \\ u(x,0) = u_0(x), & \forall x \in (0,1) \\ u(0,t) = 0, & \forall t \geq 0 \end{cases}$$

finite difference approximation

backbard Euler time stepping



- Qualitative properties: nonnegativity, no creation of new extrema
- Underresolved approximations: spurious wiggles, numerical diffusion

Design criteria I

- Model problem $M\frac{\mathrm{d}u}{\mathrm{d}t} = Cu, \quad M = \mathrm{diag}\{m_i\}, \quad C = \{c_{ij}\}$

Local extremum diminishing scheme, *Jameson '93*

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} c_{ij}(u_j - u_i), \qquad m_i > 0, \quad c_{ij} \geq 0, \quad \forall i, \forall j \neq i$$

Design criteria I

- Model problem $M\frac{\mathrm{d}u}{\mathrm{d}t} = Cu, \quad M = \mathrm{diag}\{m_i\}, \quad C = \{c_{ij}\}$

Local extremum diminishing scheme, *Jameson '93*

$$m_i\frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j\neq i} c_{ij}(u_j - u_i), \qquad m_i > 0, \quad c_{ij} \geq 0, \quad \forall i, \forall j \neq i$$

- Maximum/minimum values cannot increase/decrease

Design criteria I

- Model problem $M\frac{\mathrm{d}u}{\mathrm{d}t} = Cu, \quad M = \mathrm{diag}\{m_i\}, \quad C = \{c_{ij}\}$

Local extremum diminishing scheme, *Jameson '93*

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} c_{ij}(u_j - u_i), \qquad m_i > 0, \quad c_{ij} \geq 0, \quad \forall i, \forall j \neq i$$

- Maximum/minimum values cannot increase/decrease

Proof: maximum u_i at node i implies $u_i \geq u_j, \quad \forall j \neq i$

$$c_{ij}(u_j - u_i) \leq 0, \quad \forall j \neq i \quad \Rightarrow \quad \frac{\mathrm{d}u_i}{\mathrm{d}t} \leq 0$$

Design criteria I

- Model problem $\quad M\frac{\mathrm{d}u}{\mathrm{d}t} = Cu, \quad M = \mathrm{diag}\{m_i\}, \quad C = \{c_{ij}\}$

Local extremum diminishing scheme, *Jameson '93*

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} c_{ij}(u_j - u_i), \qquad m_i > 0, \quad c_{ij} \geq 0, \quad \forall i, \forall j \neq i$$

- Maximum/minimum values cannot increase/decrease

Proof: maximum u_i at node i implies $u_i \geq u_j, \quad \forall j \neq i$

$$c_{ij}(u_j - u_i) \leq 0, \quad \forall j \neq i \quad \Rightarrow \quad \frac{\mathrm{d}u_i}{\mathrm{d}t} \leq 0$$

- General form $\quad m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} c_{ij}(u_j - u_i) + u_i \sum_j c_{ij}$

Design criteria II

Positivity-preserving scheme

If A is a monotone matrix (i.e. $A^{-1} \geq 0$) and $B \geq 0$ then

$$Au^{n+1} = Bu^n, \quad u^n \geq 0 \quad \Rightarrow \quad u^{n+1} = A^{-1}Bu^n \geq 0$$

Design criteria II

Positivity-preserving scheme

If A is a monotone matrix (i.e. $A^{-1} \geq 0$) and $B \geq 0$ then

$$Au^{n+1} = Bu^n, \quad u^n \geq 0 \quad \Rightarrow \quad u^{n+1} = A^{-1}Bu^n \geq 0$$

- Sufficient conditions for regular matrix A to be monotone
  - A has positive diagonal coefficients $\quad a_{ii} > 0, \quad \forall i$
  - A has no positive off-diagonal entries $\quad a_{ij} \leq 0, \quad \forall j \neq i$
  - A is strictly diagonally dominant $\quad \sum_j a_{ij} \geq 0, \quad \forall i$

Design criteria II

Positivity-preserving scheme

If A is a monotone matrix (i.e. $A^{-1} \geq 0$) and $B \geq 0$ then

$$Au^{n+1} = Bu^n, \quad u^n \geq 0 \quad \Rightarrow \quad u^{n+1} = A^{-1}Bu^n \geq 0$$

- Sufficient conditions for regular matrix A to be monotone
  - A has positive diagonal coefficients $\qquad a_{ii} > 0, \quad \forall i$
  - A has no positive off-diagonal entries $\qquad a_{ij} \leq 0, \quad \forall j \neq i$
  - A is strictly diagonally dominant $\qquad \sum_j a_{ij} \geq 0, \quad \forall i$

- Two-level θ-scheme $\qquad M\frac{u^{n+1}-u^n}{\Delta t} = \theta C u^{n+1} + (1-\theta)Cu^n$

Constraints on $A = M - \theta\Delta t C$ and $B = M + (1-\theta)\Delta t C$

$(0 \leq \theta \leq 1)$ yield computable bounds on the time step Δt.

Galerkin finite element scheme

- Continuity equation $\int_\Omega w \left[\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v} u) \right] \, \mathrm{d}\mathbf{x} = 0, \quad \forall w$

Galerkin finite element scheme

- Continuity equation $\int_\Omega w \left[\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) \right] \, \mathrm{d}\mathbf{x} = 0, \quad \forall w$

- Group formulation $u \approx \sum_j u_j \varphi_j, \qquad \mathbf{v}u \approx \sum_j (\mathbf{v}_j u_j) \varphi_j$

Galerkin finite element scheme

- Continuity equation $\int_\Omega w\left[\frac{\partial u}{\partial t} + \nabla\cdot(\mathbf{v}u)\right]\,\mathrm{d}\mathbf{x} = 0, \quad \forall w$

- Group formulation $u \approx \sum_j u_j\varphi_j, \qquad \mathbf{v}u \approx \sum_j(\mathbf{v}_j u_j)\varphi_j$

- Galerkin FEM $M_C\frac{\mathrm{d}u}{\mathrm{d}t} = Ku, \quad M_C = \{m_{ij}\}, \quad K = \{k_{ij}\}$

$$m_{ij} = \int_\Omega \varphi_i\varphi_j\,\mathrm{d}\mathbf{x}, \qquad k_{ij} = -\mathbf{c}_{ij}\cdot\mathbf{v}_j, \qquad \mathbf{c}_{ij} = \int_\Omega \varphi_i\nabla\varphi_j\,\mathrm{d}\mathbf{x}$$

Galerkin finite element scheme

- Continuity equation $\int_\Omega w \left[\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u)\right] \mathrm{d}\mathbf{x} = 0, \quad \forall w$
- Group formulation $u \approx \sum_j u_j \varphi_j, \qquad \mathbf{v}u \approx \sum_j (\mathbf{v}_j u_j)\varphi_j$
- Galerkin FEM $M_C \frac{\mathrm{d}u}{\mathrm{d}t} = Ku, \quad M_C = \{m_{ij}\}, \quad K = \{k_{ij}\}$

$$m_{ij} = \int_\Omega \varphi_i \varphi_j \,\mathrm{d}\mathbf{x}, \qquad k_{ij} = -\mathbf{c}_{ij} \cdot \mathbf{v}_j, \qquad \mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \varphi_j \,\mathrm{d}\mathbf{x}$$

Lumped mass high-order scheme $M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Ku$

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} k_{ij}(u_j - u_i) + u_i \sum_j k_{ij}, \qquad m_i = \sum_j m_{ij}$$

Galerkin finite element scheme

- Continuity equation $\int_\Omega w \left[\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u)\right] \,\mathrm{d}\mathbf{x} = 0, \quad \forall w$
- Group formulation $u \approx \sum_j u_j \varphi_j, \qquad \mathbf{v}u \approx \sum_j (\mathbf{v}_j u_j)\varphi_j$
- Galerkin FEM $M_C \frac{\mathrm{d}u}{\mathrm{d}t} = Ku, \quad M_C = \{m_{ij}\}, \quad K = \{k_{ij}\}$

$$m_{ij} = \int_\Omega \varphi_i \varphi_j \,\mathrm{d}\mathbf{x}, \qquad k_{ij} = -\mathbf{c}_{ij} \cdot \mathbf{v}_j, \qquad \mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \varphi_j \,\mathrm{d}\mathbf{x}$$

Lumped mass high-order scheme $M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Ku$

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} k_{ij}(u_j - u_i) + u_i \sum_j k_{ij}, \qquad m_i = \sum_j m_{ij}$$

For P_1/Q_1 FEs $m_i > 0, \forall i$ but $k_{ij} < 0$ for some $j \neq i$. ↯LED

Discrete upwind scheme

Low-order scheme $M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Lu, \qquad L = K + D$

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} l_{ij}(u_j - u_i) + u_i \sum_j k_{ij}, \qquad l_{ij} \geq 0, \quad \forall j \neq i$$

Discrete upwind scheme

Low-order scheme $\quad M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Lu, \qquad L = K + D$

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} l_{ij}(u_j - u_i) + u_i \sum_j k_{ij}, \qquad l_{ij} \geq 0, \quad \forall j \neq i$$

- Properties of discrete diffusion operators $\qquad D = \{d_{ij}\}$

$$d_{ij} = d_{ji}, \qquad \sum_j d_{ij} = \sum_i d_{ij} = 0 \quad \Rightarrow \quad d_{ii} = -\sum_{j \neq i} d_{ij}$$

Discrete upwind scheme

Low-order scheme $\quad M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Lu, \qquad L = K + D$

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} l_{ij}(u_j - u_i) + u_i \sum_j k_{ij}, \qquad l_{ij} \geq 0, \quad \forall j \neq i$$

- Properties of discrete diffusion operators $\qquad D = \{d_{ij}\}$

$$d_{ij} = d_{ji}, \qquad \sum_j d_{ij} = \sum_i d_{ij} = 0 \quad \Rightarrow \quad d_{ii} = -\sum_{j \neq i} d_{ij}$$

- Choice of artificial diffusion coefficients

$$d_{ij} = \max\{-k_{ij}, 0, -k_{ji}\}$$

$$d_{ij} = \max\{|k_{ij}|, |k_{ji}|\}$$

Discrete upwind scheme

Low-order scheme $M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Lu, \qquad L = K + D$

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j\neq i} l_{ij}(u_j - u_i) + u_i \sum_j k_{ij}, \qquad l_{ij} \geq 0, \quad \forall j \neq i$$

- Properties of discrete diffusion operators $\qquad D = \{d_{ij}\}$

$$d_{ij} = d_{ji}, \qquad \sum_j d_{ij} = \sum_i d_{ij} = 0 \quad \Rightarrow \quad d_{ii} = -\sum_{j\neq i} d_{ij}$$

- Choice of artificial diffusion coefficients

$$d_{ij} = \max\{-k_{ij}, 0, -k_{ji}\} \qquad k_{ii} := k_{ii} - d_{ij} \quad k_{ij} + d_{ij} \geq 0$$

$$d_{ij} = \max\{|k_{ij}|, |k_{ji}|\} \qquad k_{ji} + d_{ij} \geq 0 \quad k_{jj} := k_{jj} - d_{ij}$$

Discrete upwind scheme

Low-order scheme $\quad M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Lu, \qquad L = K + D$

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} l_{ij}(u_j - u_i) + u_i \sum_j k_{ij}, \qquad l_{ij} \geq 0, \quad \forall j \neq i$$

- Properties of discrete diffusion operators $\qquad D = \{d_{ij}\}$

$$d_{ij} = d_{ji}, \qquad \sum_j d_{ij} = \sum_i d_{ij} = 0 \quad \Rightarrow \quad d_{ii} = -\sum_{j \neq i} d_{ij}$$

- Choice of artificial diffusion coefficients

$$d_{ij} = \max\{-k_{ij}, 0, -k_{ji}\} \qquad k_{ii} := k_{ii} - d_{ij} \quad k_{ij} + d_{ij} \geq 0$$

$$d_{ij} = \max\{|k_{ij}|, |k_{ji}|\} \qquad k_{ji} + d_{ij} \geq 0 \qquad k_{jj} := k_{jj} - d_{ij}$$

Linear monotone methods are at most first order accurate!

Algebraic flux correction

- Residual difference $f = (M_L - M_C)\frac{\mathrm{d}u}{\mathrm{d}t} - Du$

Algebraic flux correction

- Residual difference $$f = (M_L - M_C)\frac{\mathrm{d}u}{\mathrm{d}t} - Du$$
- Flux decomposition $$f_i = \sum_{j \neq i} f_{ij}, \quad f_{ji} = -f_{ij}$$
- Antidiffusive fluxes $$f_{ij} = \left[m_{ij}\frac{\mathrm{d}}{\mathrm{d}t} + d_{ij}\right](u_i - u_j)$$

Algebraic flux correction

- Residual difference $f = (M_L - M_C)\frac{\mathrm{d}u}{\mathrm{d}t} - Du$
- Flux decomposition $f_i = \sum_{j \neq i} f_{ij}, \quad f_{ji} = -f_{ij}$
- Antidiffusive fluxes $f_{ij} = \left[m_{ij}\frac{\mathrm{d}}{\mathrm{d}t} + d_{ij}\right](u_i - u_j)$
- Enforcing positivity constraints $\bar{f}_{ij} = \alpha_{ij} f_{ij}, \quad 0 \leq \alpha_{ij} \leq 1$
  - high-order approximation ($\alpha_{ij} = 1$) to be used in smooth regions
  - low-order approximation ($\alpha_{ij} = 0$) to be used near steep fronts

Algebraic flux correction

- Residual difference $f = (M_L - M_C)\frac{\mathrm{d}u}{\mathrm{d}t} - Du$
- Flux decomposition $f_i = \sum_{j \neq i} f_{ij}, \quad f_{ji} = -f_{ij}$
- Antidiffusive fluxes $f_{ij} = \left[m_{ij}\frac{\mathrm{d}}{\mathrm{d}t} + d_{ij}\right](u_i - u_j)$
- Enforcing positivity constraints $\bar{f}_{ij} = \alpha_{ij} f_{ij}, \quad 0 \leq \alpha_{ij} \leq 1$
  - high-order approximation ($\alpha_{ij} = 1$) to be used in smooth regions
  - low-order approximation ($\alpha_{ij} = 0$) to be used near steep fronts
- High-resolution scheme $M_L\frac{\mathrm{d}u}{\mathrm{d}t} = Lu + \bar{f}(u), \quad \bar{f}_i = \sum_{j \neq i} \bar{f}_{ij}$

Roadmap of nonlinear FEM-FCT schemes

Nonlinear algebraic system for the two-level θ-scheme, $\quad 0 < \theta \leq 1$

$$[M_L - \theta L]u^{n+1} = [M_L + (1-\theta)\Delta t L]u^n + \Delta t \bar{f}(u^{n+1}, u^n)$$

Roadmap of nonlinear FEM-FCT schemes

Nonlinear algebraic system for the two-level θ-scheme, $\quad 0 < \theta \leq 1$

$$[M_L - \theta L]u^{n+1} = [M_L + (1-\theta)\Delta t L]u^n + \Delta t \bar{f}(u^{n+1}, u^n)$$

1. Compute an explicit low-order approximation

$$M_L \tilde{u} = [M_L + (1-\theta)\Delta t L]u^n$$

Roadmap of nonlinear FEM-FCT schemes

Nonlinear algebraic system for the two-level θ-scheme, $\quad 0 < \theta \leq 1$

$$[M_L - \theta L]u^{n+1} = [M_L + (1-\theta)\Delta t L]u^n + \Delta t \bar{f}(u^{n+1}, u^n)$$

1. Compute an explicit low-order approximation

$$M_L \tilde{u} = [M_L + (1-\theta)\Delta t L]u^n$$

2. Compute successive approximations to u^{n+1} until convergence

   - Apply Zalesak's limiter to constrain antidiffusive fluxes

$$M_L \bar{u} = M_L \tilde{u} + \Delta t \bar{f}(u^{(m)}, u^n)$$

Roadmap of nonlinear FEM-FCT schemes

Nonlinear algebraic system for the two-level θ-scheme, $0 < \theta \leq 1$

$$[M_L - \theta L]u^{n+1} = [M_L + (1-\theta)\Delta t L]u^n + \Delta t \bar{f}(u^{n+1}, u^n)$$

1. Compute an explicit low-order approximation

   $$M_L \tilde{u} = [M_L + (1-\theta)\Delta t L]u^n$$

2. Compute successive approximations to u^{n+1} until convergence

   - Apply Zalesak's limiter to constrain antidiffusive fluxes

     $$M_L \bar{u} = M_L \tilde{u} + \Delta t \bar{f}(u^{(m)}, u^n)$$

   - Solve the linear system for the new solution $u^{(m+1)} \approx u^{n+1}$

     $$[M_L - \theta \Delta t L]u^{(m+1)} = M_L \bar{u}$$

Roadmap of linearized FEM-FCT schemes

Linearization of antidiffusive fluxes

$$f = (M_L - M_C)\dot{u}^L - Du^L, \qquad \dot{u}^L \approx \frac{\mathrm{d}u}{\mathrm{d}t}, \quad u^L \approx u^{n+1}$$

Roadmap of linearized FEM-FCT schemes

Linearization of antidiffusive fluxes

$$f = (M_L - M_C)\dot{u}^L - Du^L, \qquad \dot{u}^L \approx \frac{\mathrm{d}u}{\mathrm{d}t}, \quad u^L \approx u^{n+1}$$

1. Compute a provisional low-order solution

$$[M_L - \theta \Delta t L]u^L = [M_L + (1-\theta)\Delta t L]u^n$$

Roadmap of linearized FEM-FCT schemes

Linearization of antidiffusive fluxes

$$f = (M_L - M_C)\dot{u}^L - Du^L, \qquad \dot{u}^L \approx \frac{\mathrm{d}u}{\mathrm{d}t}, \quad u^L \approx u^{n+1}$$

1. Compute a provisional low-order solution

$$[M_L - \theta \Delta t L]u^L = [M_L + (1-\theta)\Delta t L]u^n$$

2. Compute an approximation to the time derivative

$$M_C \dot{u}^L = K u^L \quad \text{or} \quad M_L \dot{u}^L = L u^L$$

Roadmap of linearized FEM-FCT schemes

Linearization of antidiffusive fluxes

$$f = (M_L - M_C)\dot{u}^L - Du^L, \qquad \dot{u}^L \approx \frac{\mathrm{d}u}{\mathrm{d}t}, \quad u^L \approx u^{n+1}$$

1. Compute a provisional low-order solution

$$[M_L - \theta \Delta t L]u^L = [M_L + (1-\theta)\Delta t L]u^n$$

2. Compute an approximation to the time derivative

$$M_C \dot{u}^L = K u^L \quad \text{or} \quad M_L \dot{u}^L = L u^L$$

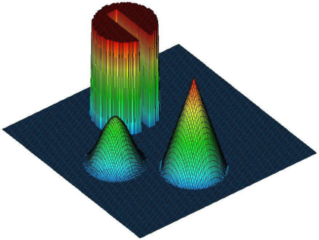
3. Apply Zalesak's limiter to constrain antidiffusive fluxes

$$M_L u^{n+1} = M_L u^L + \Delta t \bar{f}(u^L, u^n)$$

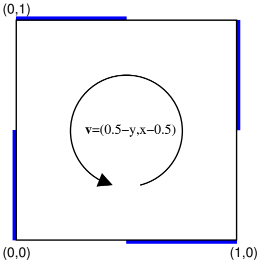
Solid body rotation

Crank-Nicolson time-stepping, Q_1 elements, $h = 1/128$, $\Delta t = 10^{-3}$

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) = 0 \quad \text{in} \quad \Omega = (0,1) \times (0,1), \qquad u = 0 \quad \text{on} \quad \Gamma_D$$

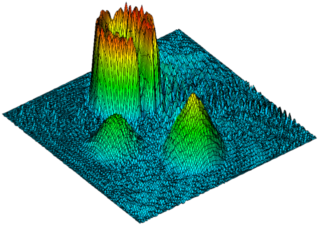


initial/exact solution $t = 2\pi k$

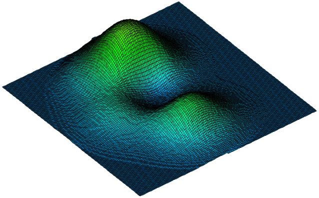


domain and velocity

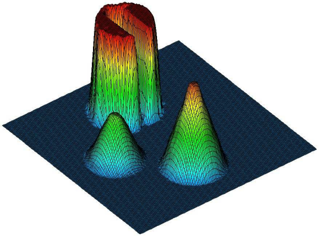
Solid body rotation $t = 2\pi$



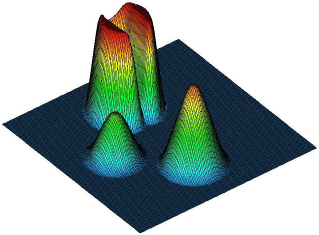
high-order solution



low-order solution



nonlinear FEM-FCT solution



linearized FEM-FCT solution

Remarks

- High-resolution schemes for scalar conservation can be based on the Galerkin method by enforcing mathematical constraints a posteriori
- FEM-FCT schemes based on flux linearization provide an efficient alternative to the nonlinear flux corrected transport algorithm
- Algebraic flux correction can be generalized to hyperbolic systems

Design criteria III

Reminder: LED criterion for scalar equations

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} c_{ij}(u_j - u_i), \qquad m_i > 0, \quad c_{ij} \geq 0, \quad \forall i, \forall j \neq i$$

Design criteria III

Reminder: LED criterion for scalar equations

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} c_{ij}(u_j - u_i), \qquad m_i > 0, \quad c_{ij} \geq 0, \quad \forall i, \forall j \neq i$$

- Model problem $\quad M \frac{\mathrm{dU}}{\mathrm{d}t} = C\mathrm{U}, \quad M = \mathrm{diag}\{\mathrm{M}_i\}, \quad C = \{\mathrm{C}_{ij}\}$

Generalized LED criterion for hyperbolic systems

$$\mathrm{M}_i \frac{\mathrm{dU}_i}{\mathrm{d}t} = \sum_{j \neq i} \mathrm{C}_{ij}(\mathrm{U}_j - \mathrm{U}_i), \quad \mathrm{M}_i > 0, \quad \mathrm{C}_{ij} \text{ p.s.d.}, \quad \forall i, \forall j \neq i$$

Design criteria III

Reminder: LED criterion for scalar equations

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} c_{ij}(u_j - u_i), \qquad m_i > 0, \quad c_{ij} \geq 0, \quad \forall i, \forall j \neq i$$

- Model problem $\quad M \frac{\mathrm{d}\mathrm{U}}{\mathrm{d}t} = C\mathrm{U}, \quad M = \mathrm{diag}\{\mathrm{M}_i\}, \quad C = \{\mathrm{C}_{ij}\}$

Generalized LED criterion for hyperbolic systems

$$\mathrm{M}_i \frac{\mathrm{d}\mathrm{U}_i}{\mathrm{d}t} = \sum_{j \neq i} \mathrm{C}_{ij}(\mathrm{U}_j - \mathrm{U}_i), \quad \mathrm{M}_i > 0, \quad \mathrm{C}_{ij} \text{ p.s.d.}, \quad \forall i, \forall j \neq i$$

- Condition for matrix C_{ij} to be positive semi-definite

$$\mathrm{X}^{\mathrm{T}} \mathrm{C}_{ij} \mathrm{X} \geq 0, \quad \forall \mathrm{X} \quad \Leftrightarrow \quad \text{all eigenvalues of } \mathrm{C}_{ij} \text{ are non-negative}$$

Hyperbolic conservation laws

Divergence form

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} = 0, \quad \nabla \cdot \mathbf{F} = \sum_d \frac{\partial F^d}{\partial x_d}$$

Quasi-linear form

$$\frac{\partial U}{\partial t} + \mathbf{A} \cdot \nabla U = 0, \quad A^d = \frac{\partial F^d}{\partial U}$$

Hyperbolic conservation laws

Divergence form

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} = 0, \quad \nabla \cdot \mathbf{F} = \sum_d \frac{\partial F^d}{\partial x_d}$$

Quasi-linear form

$$\frac{\partial U}{\partial t} + \mathbf{A} \cdot \nabla U = 0, \quad A^d = \frac{\partial F^d}{\partial U}$$

- Group finite element formulation $\quad U = \sum_j \mathrm{U}_j \varphi_j, \quad \mathbf{F} = \sum_j \mathbf{F}_j \varphi_j$

Hyperbolic conservation laws

Divergence form

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} = 0, \quad \nabla \cdot \mathbf{F} = \sum_d \frac{\partial F^d}{\partial x_d}$$

Quasi-linear form

$$\frac{\partial U}{\partial t} + \mathbf{A} \cdot \nabla U = 0, \quad A^d = \frac{\partial F^d}{\partial U}$$

- Group finite element formulation $\quad U = \sum_j \mathrm{U}_j \varphi_j, \quad \mathbf{F} = \sum_j \mathbf{F}_j \varphi_j$
- Lumped mass Galerkin method $\quad M_L \frac{\mathrm{dU}}{\mathrm{d}t} = K\mathrm{U}, \quad K = \{\mathrm{K}_{ij}\}$

Hyperbolic conservation laws

Divergence form

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} = 0, \quad \nabla \cdot \mathbf{F} = \sum_d \frac{\partial F^d}{\partial x_d}$$

Quasi-linear form

$$\frac{\partial U}{\partial t} + \mathbf{A} \cdot \nabla U = 0, \quad A^d = \frac{\partial F^d}{\partial U}$$

- Group finite element formulation $\quad U = \sum_j \mathrm{U}_j \varphi_j, \quad \mathbf{F} = \sum_j \mathbf{F}_j \varphi_j$
- Lumped mass Galerkin method $\quad M_L \frac{\mathrm{dU}}{\mathrm{dt}} = K\mathrm{U}, \quad K = \{\mathrm{K}_{ij}\}$

$$(K\mathrm{U})_i = -\sum_j \mathbf{c}_{ij} \cdot \mathbf{F}_j, \qquad \mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \varphi_j \, \mathrm{d}\mathbf{x}, \qquad \sum_j \mathbf{c}_{ij} = 0$$

Hyperbolic conservation laws

Divergence form

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} = 0, \quad \nabla \cdot \mathbf{F} = \sum_d \frac{\partial F^d}{\partial x_d}$$

Quasi-linear form

$$\frac{\partial U}{\partial t} + \mathbf{A} \cdot \nabla U = 0, \quad A^d = \frac{\partial F^d}{\partial U}$$

- Group finite element formulation $\quad U = \sum_j \mathrm{U}_j \varphi_j, \quad \mathbf{F} = \sum_j \mathbf{F}_j \varphi_j$
- Lumped mass Galerkin method $\quad M_L \frac{\mathrm{dU}}{\mathrm{d}t} = K\mathrm{U}, \quad K = \{\mathrm{K}_{ij}\}$

$$(K\mathrm{U})_i = -\sum_j \mathbf{c}_{ij} \cdot \mathbf{F}_j, \qquad \mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \varphi_j \, \mathrm{d}\mathbf{x}, \qquad \sum_j \mathbf{c}_{ij} = 0$$

- Decomposition of the right-hand side into edge contributions

$$\mathbf{c}_{ii} = -\sum_{j \neq i} \mathbf{c}_{ij} \quad \Rightarrow \quad (K\mathrm{U})_i = -\sum_{j \neq i} \mathbf{c}_{ij} \cdot (\mathbf{F}_j - \mathbf{F}_i)$$

Upwinding for the Euler equations

- Representation based on homogeneity property $\quad F^d = A^d U, \quad \forall d$

$$(K\mathrm{U})_i = \sum_{j\neq i} \mathrm{K}_{ij}(\mathrm{U}_j - \mathrm{U}_i) + \mathrm{U}_i \sum_j \mathrm{K}_{ij}, \qquad \mathrm{K}_{ij} = -\mathbf{c}_{ij} \cdot \mathbf{A}_j$$

Upwinding for the Euler equations

- Representation based on homogeneity property $F^d = A^d U, \quad \forall d$

$$(K\mathrm{U})_i = \sum_{j \neq i} \mathrm{K}_{ij}(\mathrm{U}_j - \mathrm{U}_i) + \mathrm{U}_i \sum_j \mathrm{K}_{ij}, \qquad \mathrm{K}_{ij} = -\mathbf{c}_{ij} \cdot \mathbf{A}_j$$

- Eigenvalues of matrix $\mathrm{K}_{ij} = \mathrm{R}_{ij} \Lambda_{ij} \mathrm{R}_{ij}^{-1}, \qquad \Lambda_{ij} \in \mathbb{R}^5$

$$\lambda_1 = v_{ij} + |\mathbf{c}_{ij}|\, c_j, \qquad \lambda_{2,3,4} = v_{ij}, \qquad \lambda_5 = v_{ij} - |\mathbf{c}_{ij}|\, c_j$$

$$v_{ij} = -\mathbf{c}_{ij} \cdot \mathbf{v}_j, \qquad c_j = \sqrt{(\gamma - 1)\left(E_j + p_j/\rho_j - \tfrac{1}{2}|\mathbf{v}_j|^2\right)}$$

Upwinding for the Euler equations

- Representation based on homogeneity property $\quad F^d = A^d U, \quad \forall d$

$$(K\mathrm{U})_i = \sum_{j\neq i} \mathrm{K}_{ij}(\mathrm{U}_j - \mathrm{U}_i) + \mathrm{U}_i \sum_j \mathrm{K}_{ij}, \qquad \mathrm{K}_{ij} = -\mathbf{c}_{ij} \cdot \mathbf{A}_j$$

- Eigenvalues of matrix $\qquad \mathrm{K}_{ij} = \mathrm{R}_{ij}\Lambda_{ij}\mathrm{R}_{ij}^{-1}, \qquad \Lambda_{ij} \in \mathbb{R}^5$

$$\lambda_1 = v_{ij} + |\mathbf{c}_{ij}|\, c_j, \qquad \lambda_{2,3,4} = v_{ij}, \qquad \lambda_5 = v_{ij} - |\mathbf{c}_{ij}|\, c_j$$

$$v_{ij} = -\mathbf{c}_{ij} \cdot \mathbf{v}_j, \qquad c_j = \sqrt{(\gamma - 1)\left(E_j + p_j/\rho_j - \tfrac{1}{2}|\mathbf{v}_j|^2\right)}$$

- Rusanov-type artificial viscosities $\qquad \mathrm{L}_{ij} = \mathrm{K}_{ij} + \mathrm{D}_{ij}, \quad \mathrm{D}_{ij} = d_{ij}\mathrm{I}$

$$d_{ij} = \max\{|v_{ij}| + |\mathbf{c}_{ij}|\, c_j, |v_{ji}| + |\mathbf{c}_{ji}|\, c_i\}, \qquad v_{ji} = -\mathbf{c}_{ji} \cdot \mathbf{v}_i$$

Flux correction for the Euler equations

- High-resolution scheme $M_L \frac{\mathrm{dU}}{\mathrm{d}t} = L\mathrm{U} + \bar{\mathrm{F}}(\mathrm{U})$

Flux correction for the Euler equations

- High-resolution scheme $M_L \frac{\mathrm{dU}}{\mathrm{d}t} = L\mathrm{U} + \bar{\mathrm{F}}(\mathrm{U})$

- Dimensional-split flux limiting based on characteristic variables

$$\bar{\mathrm{F}}_i = \sum_d \sum_{j \neq i} \mathrm{R}_{ij}^d \bar{\mathrm{G}}_{ij}^d, \quad \bar{\mathrm{G}}_{ij}^d = \mathrm{diag}\{\alpha_{ij}^{d,k}\} \mathrm{G}_{ij}^d, \quad \mathrm{G}_{ij}^d = [\mathrm{R}_{ij}^d]^{-1} \mathrm{F}_{ij}^d$$

Flux correction for the Euler equations

- High-resolution scheme $\quad M_L \frac{\mathrm{dU}}{\mathrm{d}t} = L\mathrm{U} + \bar{\mathrm{F}}(\mathrm{U})$

- Dimensional-split flux limiting based on characteristic variables

$$\bar{\mathrm{F}}_i = \sum_d \sum_{j \neq i} \mathrm{R}_{ij}^d \bar{\mathrm{G}}_{ij}^d, \quad \bar{\mathrm{G}}_{ij}^d = \mathrm{diag}\{\alpha_{ij}^{d,k}\}\mathrm{G}_{ij}^d, \quad \mathrm{G}_{ij}^d = [\mathrm{R}_{ij}^d]^{-1}\mathrm{F}_{ij}^d$$

- Synchronized flux limiting based on indicator variables, *Löhner '87*

$$\bar{\mathrm{F}}_i = \sum_{j \neq i} \alpha_{ij}\mathrm{F}_{ij}, \quad \alpha_{ij} = \min\{\alpha_{ij}^{\rho}, \alpha_{ij}^{\rho E}\} \quad \text{or} \quad \alpha_{ij} = \min\{\alpha_{ij}^{\rho}, \alpha_{ij}^{p}\}$$

Flux correction for the Euler equations

- High-resolution scheme $M_L \frac{\mathrm{dU}}{\mathrm{d}t} = L\mathrm{U} + \bar{\mathrm{F}}(\mathrm{U})$

- Dimensional-split flux limiting based on characteristic variables

$$\bar{\mathrm{F}}_i = \sum_d \sum_{j \neq i} \mathrm{R}_{ij}^d \bar{\mathrm{G}}_{ij}^d, \quad \bar{\mathrm{G}}_{ij}^d = \mathrm{diag}\{\alpha_{ij}^{d,k}\}\mathrm{G}_{ij}^d, \quad \mathrm{G}_{ij}^d = [\mathrm{R}_{ij}^d]^{-1}\mathrm{F}_{ij}^d$$

- Synchronized flux limiting based on indicator variables, *Löhner '87*

$$\bar{\mathrm{F}}_i = \sum_{j \neq i} \alpha_{ij}\mathrm{F}_{ij}, \quad \alpha_{ij} = \min\{\alpha_{ij}^{\rho}, \alpha_{ij}^{\rho E}\} \quad \text{or} \quad \alpha_{ij} = \min\{\alpha_{ij}^{\rho}, \alpha_{ij}^{p}\}$$

Choosing the 'best' strategy is quite an art which requires a good knowledge of the physics of the problem to be solved.

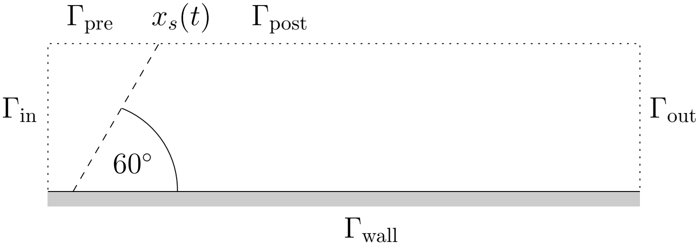
Double Mach reflection

Pre- and post-shock initial conditions in $\Omega = (0,4) \times (0,1)$

$$\begin{bmatrix} \rho_{\text{pre}} \\ u_{\text{pre}} \\ v_{\text{pre}} \\ p_{\text{pre}} \end{bmatrix} = \begin{bmatrix} 8.0 \\ 8.25\cos(30^\circ) \\ -8.25\sin(30^\circ) \\ 116.5 \end{bmatrix} \qquad \begin{bmatrix} \rho_{\text{post}} \\ u_{\text{post}} \\ v_{\text{post}} \\ p_{\text{post}} \end{bmatrix} = \begin{bmatrix} 1.4 \\ 0.0 \\ 0.0 \\ 1.0 \end{bmatrix}$$

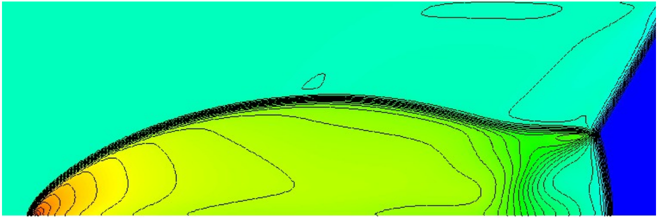
Time-dependend boundary conditions $x_s(t) = \frac{1}{6} + \frac{1+20t}{\sqrt{3}}$

$$\Gamma_{\text{pre}} = \{x < x_s(t), y = 1\}, \qquad \Gamma_{\text{post}} = \{x \geq x_s(t), y = 1\}$$



Double Mach reflection, density profile at t=0.2

Characteristic FEM-FCT vs. low-order solution



Remarks

- High-resolution schemes for hyperbolic systems can be based on a generalized local extremum diminishing criterion
- Flux correction can be performed in characteristic variables
- Conservative flux limiting based on a set of indicator variables requires the synchronization of correction factors

Idealized Z-pinch implosion model revisited

- Generalized Euler system coupled with scalar transport equation

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F}(U) = S(U, \xi), \qquad \frac{\partial \xi}{\partial t} + \nabla \cdot (\xi \mathbf{v}) = 0$$

- Conservative variables, fluxes and source term

$$U = \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{bmatrix}, \quad \xi = \rho\lambda, \quad \mathbf{F} = \begin{bmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + p\mathcal{I} \\ \rho E \mathbf{v} + p\mathbf{v} \end{bmatrix}, \quad S = \begin{bmatrix} 0 \\ \mathbf{f} \\ \mathbf{f} \cdot \mathbf{v} \end{bmatrix}$$

- EOS for an ideal gas and non-dimensional Lorentz force term

$$p = (\gamma - 1)\rho \left(E - \frac{1}{2}|\mathbf{v}|^2 \right), \qquad \mathbf{f} = \xi \left(\frac{I(t)}{I_{\max}} \right)^2 \frac{\hat{\mathbf{e}}_r}{r_{\text{eff}}}$$

Coupled solution algorithm

For $n = 0, 1, \ldots, \bar{n} - 1$ time-stepping loop

Coupled solution algorithm

For $n = 0, 1, \ldots, \bar{n} - 1$ time-stepping loop

For $k = 0, 1, \ldots, \bar{k} - 1$ outer coupling loop

Coupled solution algorithm

For $n = 0, 1, \ldots, \bar{n} - 1$ time-stepping loop

For $k = 0, 1, \ldots, \bar{k} - 1$ outer coupling loop

- Update the low-order solution to the Euler system

For $l = 0, 1, \ldots, \bar{l} - 1$ defect correction loop

$$\frac{\mathrm{U}^{(k+1,l+1)} - \mathrm{U}^n}{\Delta t} + \theta \nabla \cdot \mathbf{F}(\mathrm{U}^{(k+1,l+1)}) + (1-\theta)\nabla \cdot \mathbf{F}(\mathrm{U}^n) = \theta \mathrm{S}(\mathbf{v}^{(k+1,l+1)}, \xi^{(k)}) + (1-\theta)\mathrm{S}(\mathbf{v}^n, \xi^n)$$

Coupled solution algorithm

For $n = 0, 1, \ldots, \bar{n} - 1$ — time-stepping loop

For $k = 0, 1, \ldots, \bar{k} - 1$ — outer coupling loop

- Update the low-order solution to the Euler system

For $l = 0, 1, \ldots, \bar{l} - 1$ — defect correction loop

$$\frac{\mathrm{U}^{(k+1,l+1)} - \mathrm{U}^n}{\Delta t} + \theta \nabla \cdot \mathbf{F}(\mathrm{U}^{(k+1,l+1)}) + (1-\theta)\nabla \cdot \mathbf{F}(\mathrm{U}^n) = \theta \mathrm{S}(\mathbf{v}^{(k+1,l+1)}, \xi^{(k)}) + (1-\theta)\mathrm{S}(\mathbf{v}^n, \xi^n)$$

- Update the low-order solution to the tracer equation

$$\frac{\xi^{(k+1)} - \xi^n}{\Delta t} + \theta \nabla \cdot (\xi^{(k+1)} \mathbf{v}^{(k+1,\bar{l})}) + (1-\theta)\nabla \cdot (\xi^n \mathbf{v}^n) = 0$$

Coupled solution algorithm

For $n = 0, 1, \dots, \bar{n} - 1$ — time-stepping loop

For $k = 0, 1, \dots, \bar{k} - 1$ — outer coupling loop

- Update the low-order solution to the Euler system

For $l = 0, 1, \dots, \bar{l} - 1$ — defect correction loop

$$\frac{\mathrm{U}^{(k+1,l+1)} - \mathrm{U}^n}{\Delta t} + \theta \nabla \cdot \mathbf{F}(\mathrm{U}^{(k+1,l+1)}) + (1-\theta)\nabla \cdot \mathbf{F}(\mathrm{U}^n) = \theta \mathrm{S}(\mathbf{v}^{(k+1,l+1)}, \xi^{(k)}) + (1-\theta)\mathrm{S}(\mathbf{v}^n, \xi^n)$$

- Update the low-order solution to the tracer equation

$$\frac{\xi^{(k+1)} - \xi^n}{\Delta t} + \theta \nabla \cdot (\xi^{(k+1)} \mathbf{v}^{(k+1,\bar{l})}) + (1-\theta)\nabla \cdot (\xi^n \mathbf{v}^n) = 0$$

- Apply synchronized FCT correction to $\mathrm{U}^L = \mathrm{U}^{(\bar{k},\bar{l})}, \quad \xi^L = \xi^{(\bar{k})}$

Finishing touches

- Uniform dissipation for the Euler system and the tracer equation

$$\mathrm{D}_{ij} = d_{ij}\mathrm{I}, \quad d_{ij} = \max\{|-\mathbf{c}_{ij}\cdot\mathbf{v}_j| + |\mathbf{c}_{ij}|\, c_j, |-\mathbf{c}_{ji}\cdot\mathbf{v}_i| + |\mathbf{c}_{ji}|\, c_i\}$$

Finishing touches

- Uniform dissipation for the Euler system and the tracer equation

$$\mathrm{D}_{ij} = d_{ij}\mathrm{I}, \quad d_{ij} = \max\{|-\mathbf{c}_{ij}\cdot\mathbf{v}_j| + |\mathbf{c}_{ij}|\,c_j, |-\mathbf{c}_{ji}\cdot\mathbf{v}_i| + |\mathbf{c}_{ji}|\,c_i\}$$

- Flux limiting based on density, pressure and tracer variables

$$\bar{\mathrm{F}}_i = \sum_{j\neq i}\alpha_{ij}\mathrm{F}_{ij}, \qquad \bar{f}_i = \sum_{j\neq i}\alpha_{ij}f_{ij}, \qquad \alpha_{ij} = \min\{\alpha_{ij}^{\rho}, \alpha_{ij}^{p}, \alpha_{ij}^{\xi}\}$$

Finishing touches

- Uniform dissipation for the Euler system and the tracer equation

$$\mathrm{D}_{ij} = d_{ij}\mathrm{I}, \quad d_{ij} = \max\{|-\mathbf{c}_{ij}\cdot\mathbf{v}_j| + |\mathbf{c}_{ij}|\,c_j, |-\mathbf{c}_{ji}\cdot\mathbf{v}_i| + |\mathbf{c}_{ji}|\,c_i\}$$

- Flux limiting based on density, pressure and tracer variables

$$\bar{\mathrm{F}}_i = \sum_{j\neq i}\alpha_{ij}\mathrm{F}_{ij}, \qquad \bar{f}_i = \sum_{j\neq i}\alpha_{ij}f_{ij}, \qquad \alpha_{ij} = \min\{\alpha_{ij}^{\rho}, \alpha_{ij}^{p}, \alpha_{ij}^{\xi}\}$$

- Pressure-based flux correction implies non-negative energy

$$p \geq 0 \quad \Rightarrow \quad \rho E = p/(\gamma-1) + \tfrac{1}{2}|\mathbf{v}|^2 \geq 0$$

Finishing touches

- Uniform dissipation for the Euler system and the tracer equation

$$\mathrm{D}_{ij} = d_{ij}\mathrm{I}, \quad d_{ij} = \max\{|-\mathbf{c}_{ij}\cdot\mathbf{v}_j| + |\mathbf{c}_{ij}|\,c_j, |-\mathbf{c}_{ji}\cdot\mathbf{v}_i| + |\mathbf{c}_{ji}|\,c_i\}$$

- Flux limiting based on density, pressure and tracer variables

$$\bar{\mathrm{F}}_i = \sum_{j\neq i}\alpha_{ij}\mathrm{F}_{ij}, \qquad \bar{f}_i = \sum_{j\neq i}\alpha_{ij}f_{ij}, \qquad \alpha_{ij} = \min\{\alpha_{ij}^{\rho}, \alpha_{ij}^{p}, \alpha_{ij}^{\xi}\}$$

- Pressure-based flux correction implies non-negative energy

$$p \geq 0 \quad \Rightarrow \quad \rho E = p/(\gamma - 1) + \tfrac{1}{2}|\mathbf{v}|^2 \geq 0$$

- No additional fail-safe post-processing is required

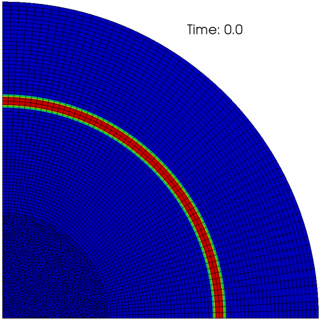
Idealized Z-pinch implosion

Non-dimensional initial conditions

movie

$$\rho' = \begin{cases} 1.0 & \text{if } r < R_0 \\ 10^6 & \text{if } r \in [R_0, R_0 + \Delta] \\ 0.5 & \text{if } r > R_0 + \Delta \end{cases} \qquad \xi' = \begin{cases} 10^6 & \text{if } r \in [R_0, R_0 + \Delta] \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{v}' = 0.0, \quad p' = 1.0, \quad R_0 = 1, \quad \Delta = 0.05, \quad r_{\text{eff}} = 10^{-4}, \quad I_{\max} = 1.0$$



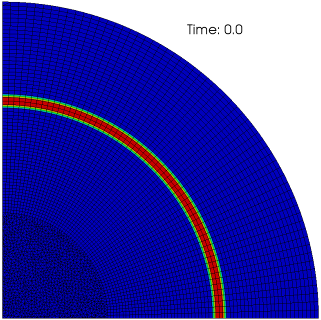
Idealized Z-pinch implosion

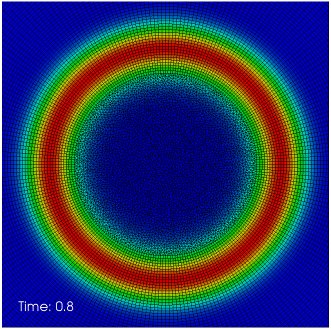
Non-dimensional initial conditions

movie

$$\rho' = \begin{cases} 1.0 & \text{if } r < R_0 \\ 10^6 & \text{if } r \in [R_0, R_0 + \Delta] \\ 0.5 & \text{if } r > R_0 + \Delta \end{cases} \qquad \xi' = \begin{cases} 10^6 & \text{if } r \in [R_0, R_0 + \Delta] \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{v}' = 0.0, \quad p' = 1.0, \quad R_0 = 1, \quad \Delta = 0.05, \quad r_{\text{eff}} = 10^{-4}, \quad I_{\max} = 1.0$$





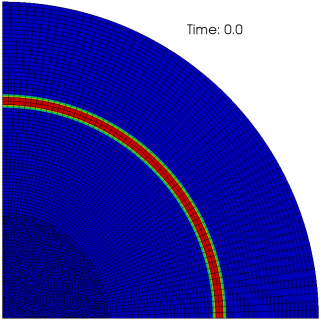
Idealized Z-pinch implosion

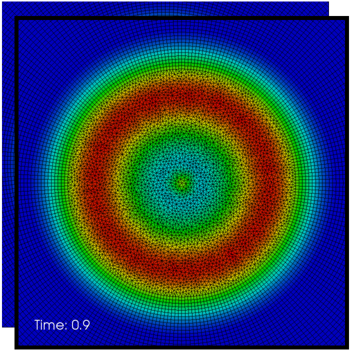
Non-dimensional initial conditions

movie

$$\rho' = \begin{cases} 1.0 & \text{if } r < R_0 \\ 10^6 & \text{if } r \in [R_0, R_0 + \Delta] \\ 0.5 & \text{if } r > R_0 + \Delta \end{cases} \qquad \xi' = \begin{cases} 10^6 & \text{if } r \in [R_0, R_0 + \Delta] \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{v}' = 0.0, \quad p' = 1.0, \quad R_0 = 1, \quad \Delta = 0.05, \quad r_{\text{eff}} = 10^{-4}, \quad I_{\max} = 1.0$$





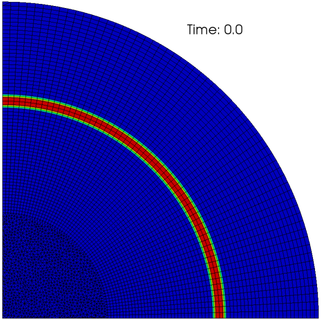
Idealized Z-pinch implosion

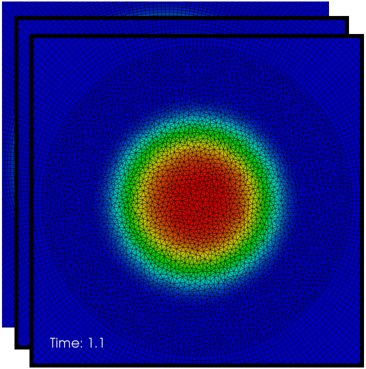
Non-dimensional initial conditions

movie

$$\rho' = \begin{cases} 1.0 & \text{if } r < R_0 \\ 10^6 & \text{if } r \in [R_0, R_0 + \Delta] \\ 0.5 & \text{if } r > R_0 + \Delta \end{cases} \qquad \xi' = \begin{cases} 10^6 & \text{if } r \in [R_0, R_0 + \Delta] \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{v}' = 0.0, \quad p' = 1.0, \quad R_0 = 1, \quad \Delta = 0.05, \quad r_{\text{eff}} = 10^{-4}, \quad I_{\max} = 1.0$$

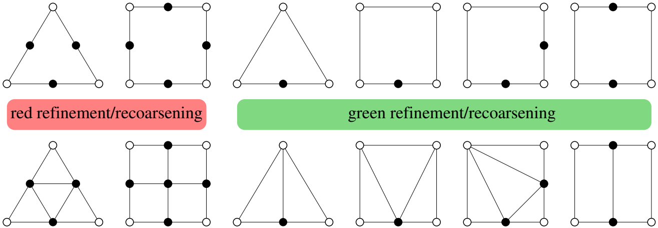




Dynamic mesh adaptation

Conformal refinement algorithm *Bank, et al. '83*

1. subdivide marked elements regularly (red rule)
2. eliminate 'hanging nodes' by transition cells (green rule)

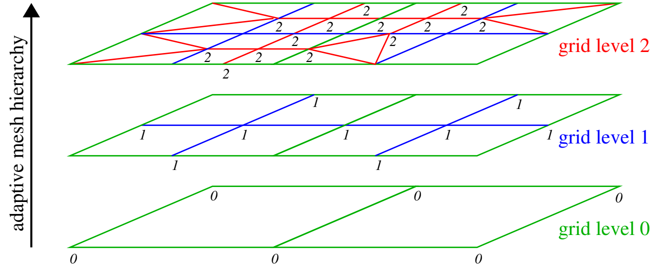


Dynamic mesh adaptation, cont'd

- Vertex-locking algorithm is used to reverse mesh refinement
- Nodal generation function provides all necessary information: *element type, inter-element relationship, refinement level, ...*

Dynamic mesh adaptation, cont'd

- Vertex-locking algorithm is used to reverse mesh refinement
- Nodal generation function provides all necessary information:
  element type, inter-element relationship, refinement level, ...



Solid body rotation

Crank-Nicolson scheme, P_1/Q_1 elements, $h_{\min} = 1/512$, $\Delta t = 10^{-3}$

$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) = 0 \quad \text{in} \quad \Omega = (0,1) \times (0,1), \qquad u = 0 \quad \text{on} \quad \Gamma_D$

Conclusions and Outlook

- Algebraic flux correction techniques can be used to compute highly accurate and symmetric solutions to prototypical Z-pinch implosions
- Globally coupled solution strategy makes it possible to treat the Euler system and the scalar tracer equation one after the other
- Mesh adaptation is a handy tool to compensate the artificial diffusion of the low-order method in the vicinity of steep fronts

Future plans for the Z-pinch implosions model

- Analysis of the coupled solution vs. operator splitting approach
- Mesh adaptation based on reliable error indicators/estimators

Constraints on the time step size

Two-level θ-time stepping scheme, $\quad 0 \leq \theta \leq 1$

$$m_i \frac{u_i^{n+1} - u_i^n}{\Delta t} = \theta \sum_j c_{ij} u_j^{n+1} + (1-\theta) \sum_j c_{ij} u_j^n, \qquad c_{ij} \geq 0, \quad \forall j \neq i$$

- System matrices $\quad A = M_L - \theta \Delta t C, \qquad B = M_L + (1-\theta)\Delta t C$
- Off-diagonal entries $\quad a_{ij} = -\theta \Delta t c_{ij} \leq 0, \quad b_{ij} = (1-\theta)\Delta t c_{ij} \geq 0$
- Diagonal coefficient $\quad a_{ii} = m_i - \theta \Delta t c_{ii} > \theta \Delta t \sum_{j \neq i} c_{ij} \geq 0, \quad \forall i$
- Diagonal coefficient $\quad b_{ii} = m_i + (1-\theta)\Delta t c_{ii} \geq 0, \quad \forall i$

Zalesak's multidimensional FCT limiter

Input: raw antidiffusive fluxes f_{ij}, auxiliary solution $\tilde{u}$

1. Sums of positive/negative antidiffusive fluxes into node i

$$P_i^+ = \sum_{j \neq i} \max\{0, f_{ij}\}, \qquad P_i^- = \sum_{j \neq i} \min\{0, f_{ij}\}$$

Zalesak's multidimensional FCT limiter

Input: raw antidiffusive fluxes f_{ij}, auxiliary solution $\tilde{u}$

1. Sums of positive/negative antidiffusive fluxes into node i

$$P_i^+ = \sum_{j \neq i} \max\{0, f_{ij}\}, \qquad P_i^- = \sum_{j \neq i} \min\{0, f_{ij}\}$$

2. Upper/lower bounds based on the local extrema of $\tilde{u}$

$$Q_i^+ = \frac{m_i}{\Delta t}(\tilde{u}_i^{\max} - \tilde{u}_i), \qquad Q_i^- = \frac{m_i}{\Delta t}(\tilde{u}_i^{\min} - \tilde{u}_i)$$

Zalesak's multidimensional FCT limiter

Input: raw antidiffusive fluxes f_{ij}, auxiliary solution $\tilde{u}$

1. Sums of positive/negative antidiffusive fluxes into node i

$$P_i^+ = \sum_{j \neq i} \max\{0, f_{ij}\}, \qquad P_i^- = \sum_{j \neq i} \min\{0, f_{ij}\}$$

2. Upper/lower bounds based on the local extrema of $\tilde{u}$

$$Q_i^+ = \frac{m_i}{\Delta t}(\tilde{u}_i^{\max} - \tilde{u}_i), \qquad Q_i^- = \frac{m_i}{\Delta t}(\tilde{u}_i^{\min} - \tilde{u}_i)$$

3. Correction factors for the pair of fluxes f_{ij} and $f_{ji} = -f_{ij}$

$$R_i^\pm = \min\left\{1, \frac{Q_i^\pm}{P_i^\pm}\right\}, \qquad \alpha_{ij} = \begin{cases} \min\{R_i^+, R_j^-\} & \text{if } f_{ij} \geq 0 \\ \min\{R_i^-, R_j^+\} & \text{if } f_{ij} < 0 \end{cases}$$

Mesh genealogy

Triangulation $\mathcal{T}_m(\mathcal{E}_m, \mathcal{V}_m)$, $m = 0, 1, 2, \ldots$ consists of

$$\mathcal{E}_m = \{\Omega_k \,:\, k = 1, \ldots, N_{\mathrm{E}}\} \quad \text{and} \quad \mathcal{V}_m = \{\mathrm{v}_i \,:\, i = 1, \ldots, N_{\mathrm{V}}\}$$

Mesh genealogy

Triangulation $\mathcal{T}_m(\mathcal{E}_m, \mathcal{V}_m)$, $m = 0, 1, 2, \ldots$ consists of

$$\mathcal{E}_m = \{\Omega_k \,:\, k = 1, \ldots, N_{\mathrm{E}}\} \quad \text{and} \quad \mathcal{V}_m = \{\mathrm{v}_i \,:\, i = 1, \ldots, N_{\mathrm{V}}\}$$

- nodal **generation function** $g : \mathcal{V}_m \to \mathbb{N}_0$ is defined recursively

$$g(\mathrm{v}_i) := \begin{cases} 0 & \text{if} \quad \mathrm{v}_i \in \mathcal{V}_0 \\ \max\limits_{\mathrm{v}_j \in \Gamma_{kl}} g(\mathrm{v}_j) + 1 & \text{if} \quad \mathrm{v}_i \in \Gamma_{kl} := \bar{\Omega}_k \cap \bar{\Omega}_l \\ \max\limits_{\mathrm{v}_j \in \partial\Omega_k} g(\mathrm{v}_j) + 1 & \text{if} \quad \mathrm{v}_i \in \Omega_k \setminus \partial\Omega_k \end{cases}$$

Mesh genealogy

Triangulation $\mathcal{T}_m(\mathcal{E}_m, \mathcal{V}_m)$, $m = 0, 1, 2, \ldots$ consists of

$$\mathcal{E}_m = \{\Omega_k \,:\, k = 1, \ldots, N_\mathrm{E}\} \quad \text{and} \quad \mathcal{V}_m = \{\mathrm{v}_i \,:\, i = 1, \ldots, N_\mathrm{V}\}$$

- nodal **generation function** $g : \mathcal{V}_m \to \mathbb{N}_0$ is defined recursively

$$g(\mathrm{v}_i) := \begin{cases} 0 & \text{if} \quad \mathrm{v}_i \in \mathcal{V}_0 \\ \max\limits_{\mathrm{v}_j \in \Gamma_{kl}} g(\mathrm{v}_j) + 1 & \text{if} \quad \mathrm{v}_i \in \Gamma_{kl} := \bar{\Omega}_k \cap \bar{\Omega}_l \\ \max\limits_{\mathrm{v}_j \in \partial\Omega_k} g(\mathrm{v}_j) + 1 & \text{if} \quad \mathrm{v}_i \in \Omega_k \setminus \partial\Omega_k \end{cases}$$

Mesh genealogy

Triangulation $\mathcal{T}_m(\mathcal{E}_m, \mathcal{V}_m)$, $m = 0, 1, 2, \ldots$ consists of

$$\mathcal{E}_m = \{\Omega_k \,:\, k = 1, \ldots, N_{\mathrm{E}}\} \quad \text{and} \quad \mathcal{V}_m = \{\mathrm{v}_i \,:\, i = 1, \ldots, N_{\mathrm{V}}\}$$

- nodal **generation function** $g : \mathcal{V}_m \to \mathbb{N}_0$ is defined recursively

$$g(\mathrm{v}_i) := \begin{cases} 0 & \text{if} \quad \mathrm{v}_i \in \mathcal{V}_0 \\ \max\limits_{\mathrm{v}_j \in \Gamma_{kl}} g(\mathrm{v}_j) + 1 & \text{if} \quad \mathrm{v}_i \in \Gamma_{kl} := \bar{\Omega}_k \cap \bar{\Omega}_l \\ \max\limits_{\mathrm{v}_j \in \partial\Omega_k} g(\mathrm{v}_j) + 1 & \text{if} \quad \mathrm{v}_i \in \Omega_k \setminus \partial\Omega_k \end{cases}$$

Mesh genealogy

Triangulation $\mathcal{T}_m(\mathcal{E}_m, \mathcal{V}_m)$, $m = 0, 1, 2, \ldots$ consists of

$$\mathcal{E}_m = \{\Omega_k \,:\, k = 1, \ldots, N_\mathrm{E}\} \quad \text{and} \quad \mathcal{V}_m = \{\mathrm{v}_i \,:\, i = 1, \ldots, N_\mathrm{V}\}$$

- nodal **generation function** $g : \mathcal{V}_m \to \mathbb{N}_0$ is defined recursively

$$g(\mathrm{v}_i) := \begin{cases} 0 & \text{if} \quad \mathrm{v}_i \in \mathcal{V}_0 \\ \max\limits_{\mathrm{v}_j \in \Gamma_{kl}} g(\mathrm{v}_j) + 1 & \text{if} \quad \mathrm{v}_i \in \Gamma_{kl} := \bar{\Omega}_k \cap \bar{\Omega}_l \\ \max\limits_{\mathrm{v}_j \in \partial\Omega_k} g(\mathrm{v}_j) + 1 & \text{if} \quad \mathrm{v}_i \in \Omega_k \setminus \partial\Omega_k \end{cases}$$

Mesh genealogy

Triangulation $\mathcal{T}_m(\mathcal{E}_m, \mathcal{V}_m)$, $m = 0, 1, 2, \ldots$ consists of

$$\mathcal{E}_m = \{\Omega_k \,:\, k = 1, \ldots, N_\mathrm{E}\} \quad \text{and} \quad \mathcal{V}_m = \{\mathrm{v}_i \,:\, i = 1, \ldots, N_\mathrm{V}\}$$

- nodal **generation function** $g : \mathcal{V}_m \to \mathbb{N}_0$ is defined recursively

$$g(\mathrm{v}_i) := \begin{cases} 0 & \text{if} \quad \mathrm{v}_i \in \mathcal{V}_0 \\ \max\limits_{\mathrm{V}_j \in \Gamma_{kl}} g(\mathrm{v}_j) + 1 & \text{if} \quad \mathrm{v}_i \in \Gamma_{kl} := \bar{\Omega}_k \cap \bar{\Omega}_l \\ \max\limits_{\mathrm{V}_j \in \partial\Omega_k} g(\mathrm{v}_j) + 1 & \text{if} \quad \mathrm{v}_i \in \Omega_k \setminus \partial\Omega_k \end{cases}$$

- represents number of subdivisions $\Rightarrow$ prescribe maximum depth
- characterizes elements and their relation to neighboring cells

Mesh re-coarsening

Re-coarsening algorithms (vertex-based approach)

1. ‘lock’ vertices step-by-step which must not be removed
2. delete ‘free’ vertices/elements and restore macro cells

Mesh re-coarsening

Re-coarsening algorithms (vertex-based approach)

1. 'lock' vertices step-by-step which must not be removed
2. delete 'free' vertices/elements and restore macro cells

1. initialize $d(\mathrm{v}_i) := g(\mathrm{v}_i)$, $\forall \mathrm{v}_i \in \mathcal{V}_m \quad \Rightarrow \quad d(\mathrm{v}_i) = 0$, $\forall \mathrm{v}_i \in \mathcal{V}_0$

Mesh re-coarsening

Re-coarsening algorithms (vertex-based approach)

1. 'lock' vertices step-by-step which must not be removed
2. delete 'free' vertices/elements and restore macro cells

1. initialize $d(\mathrm{v}_i) := g(\mathrm{v}_i),\ \forall \mathrm{v}_i \in \mathcal{V}_m \quad \Rightarrow \quad d(\mathrm{v}_i) = 0,\ \forall \mathrm{v}_i \in \mathcal{V}_0$
2. vertex $\mathrm{v}_i \in \mathcal{V}_m$ is locked, i.e. $d(\mathrm{v}_i) := -|d(\mathrm{v}_i)|$ if
   - v_i belongs to an element which is marked for refinement
   - v_i belongs to a red element which should not be coarsened
   - there is an edge ij such that $g(\mathrm{v}_i) < g(\mathrm{v}_j)$ for some $\mathrm{v}_j \in \mathcal{V}_m$

Mesh re-coarsening

Re-coarsening algorithms (vertex-based approach)

1. 'lock' vertices step-by-step which must not be removed
2. delete 'free' vertices/elements and restore macro cells

1. initialize $d(\mathrm{v}_i) := g(\mathrm{v}_i),\ \forall \mathrm{v}_i \in \mathcal{V}_m \quad \Rightarrow \quad d(\mathrm{v}_i) = 0,\ \forall \mathrm{v}_i \in \mathcal{V}_0$
2. vertex $\mathrm{v}_i \in \mathcal{V}_m$ is locked, i.e. $d(\mathrm{v}_i) := -|d(\mathrm{v}_i)|$ if
   - v_i belongs to an element which is marked for refinement
   - v_i belongs to a red element which should not be coarsened
   - there is an edge ij such that $g(\mathrm{v}_i) < g(\mathrm{v}_j)$ for some $\mathrm{v}_j \in \mathcal{V}_m$
3. vertices are locked to preclude the creation of blue elements

Mesh re-coarsening

Re-coarsening algorithms (vertex-based approach)

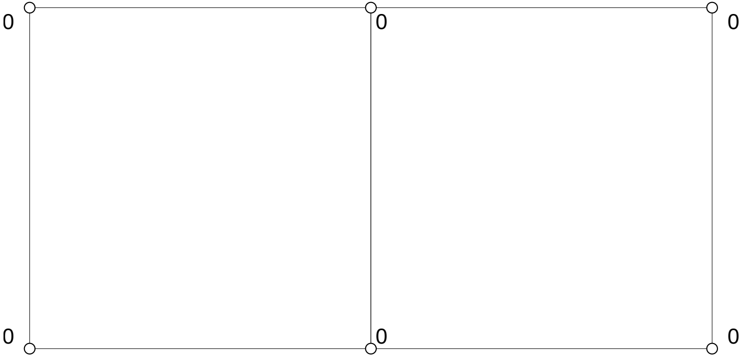
1. 'lock' vertices step-by-step which must not be removed
2. delete 'free' vertices/elements and restore macro cells

1. initialize $d(\mathrm{v}_i) := g(\mathrm{v}_i),\ \forall \mathrm{v}_i \in \mathcal{V}_m \quad \Rightarrow \quad d(\mathrm{v}_i) = 0,\ \forall \mathrm{v}_i \in \mathcal{V}_0$
2. vertex $\mathrm{v}_i \in \mathcal{V}_m$ is locked, i.e. $d(\mathrm{v}_i) := -|d(\mathrm{v}_i)|$ if
   - v_i belongs to an element which is marked for refinement
   - v_i belongs to a red element which should not be coarsened
   - there is an edge ij such that $g(\mathrm{v}_i) < g(\mathrm{v}_j)$ for some $\mathrm{v}_j \in \mathcal{V}_m$
3. vertices are locked to preclude the creation of blue elements

Result: Vertex v_i is locked if $d(\mathrm{v}_i) \leq 0$; otherwise it can be deleted.
All vertices of the initial mesh are locked by construction!

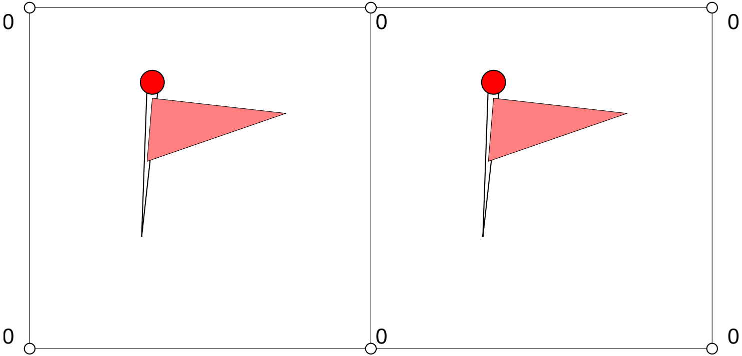
Step-by-step illustration

Refinement algorithm: initial mesh



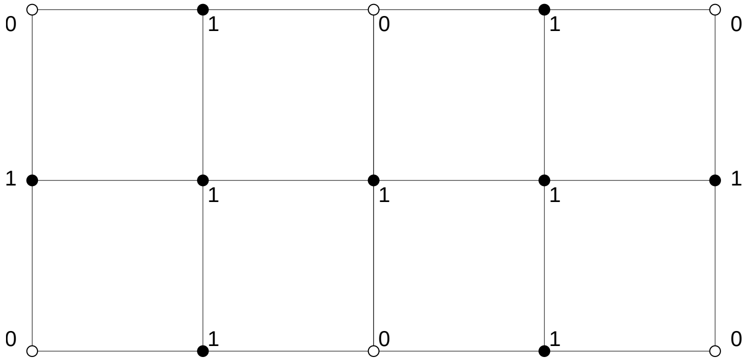
Step-by-step illustration

Refinement algorithm: mark elements for regular refinement



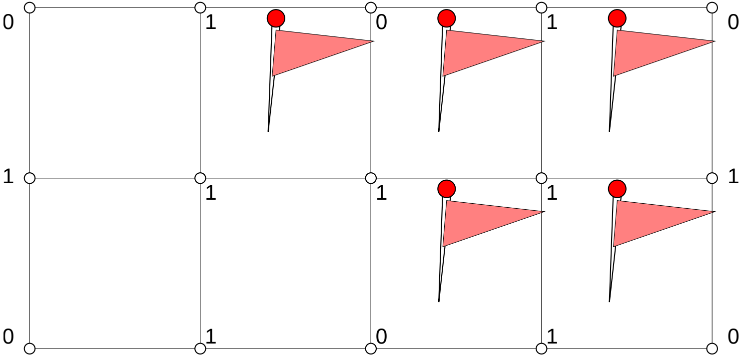
Step-by-step illustration

Refinement algorithm: perform regular refinement



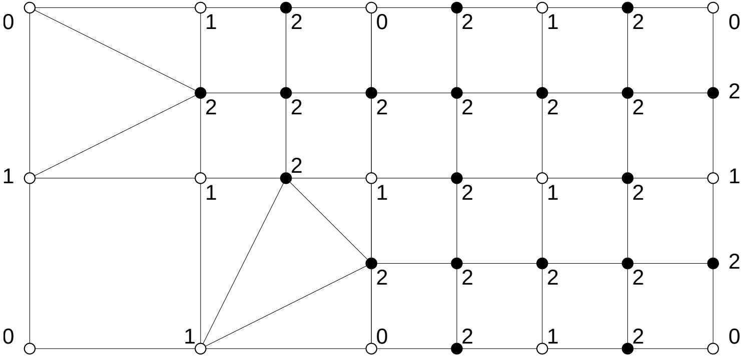
Step-by-step illustration

Refinement algorithm: mark elements for regular refinement



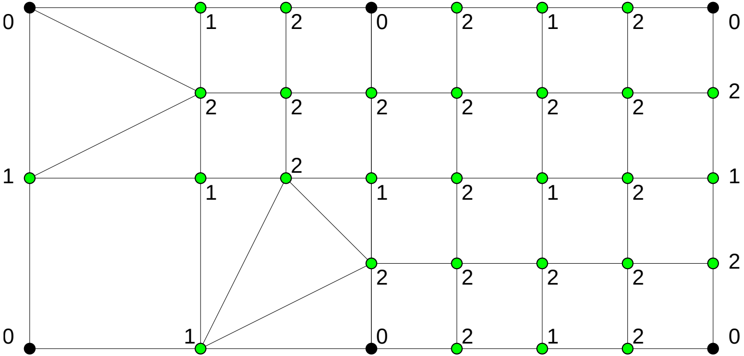
Step-by-step illustration

Refinement algorithm: perform regular refinement + transition cells



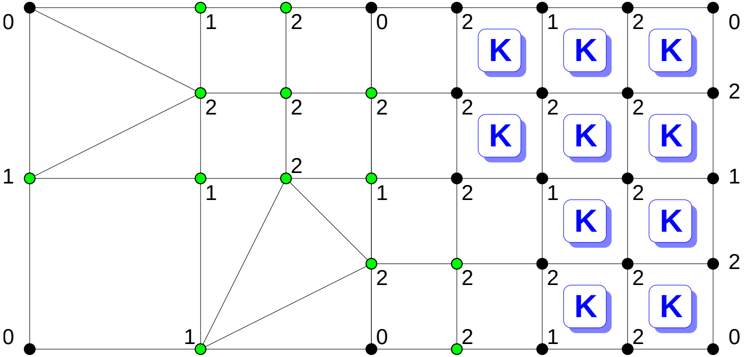
Step-by-step illustration

Re-coarsening algorithm: vertices from initial mesh are locked



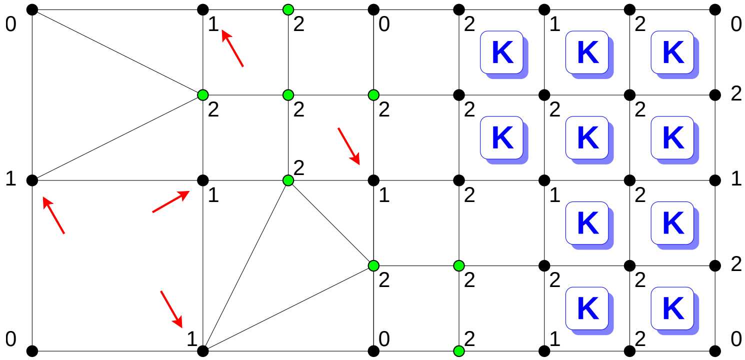
Step-by-step illustration

Re-coarsening algorithm: keep cells and lock connected vertices



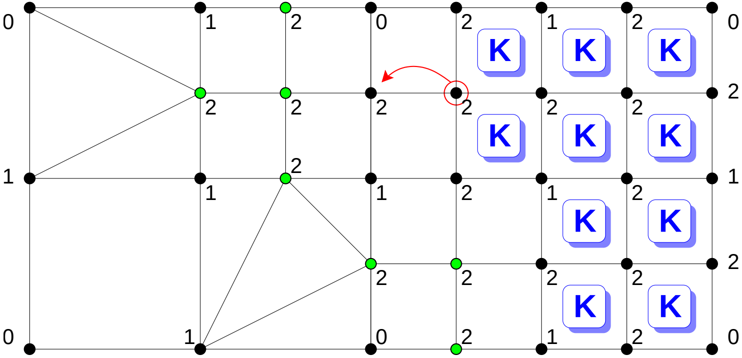
Step-by-step illustration

Re-coarsening algorithm: lock vertices if there are younger neighbors



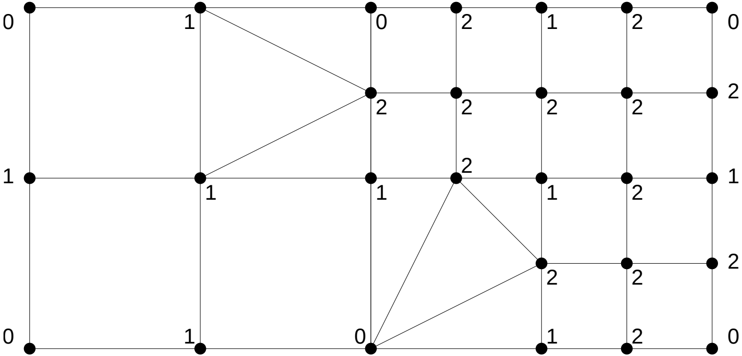
Step-by-step illustration

Re-coarsening algorithm: lock vertices to preclude blue elements



Step-by-step illustration

Re-coarsening algorithm: remove vertices and update elements



Dynamic mesh adaptation for the Euler equations