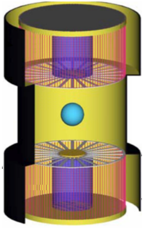
Failsafe FCT algorithms for strongly time-dependent flows with application to an idealized Z-pinch implosion model

Matthias Möller[1] joint work with: Dmitri Kuzmin[2]

[1] Institute of Applied Mathematics (LS3), Dortmund University of Technology, Germany, matthias.moeller@math.tu-dortmund.de

[2] Chair of Applied Mathematics III, University Erlangen-Nuremberg, Germany kuzmin@am.uni-erlangen.de

Motivation: *Z-pinch implosion*



- Phenomenological model by Banks and Shadid

  compressible Euler equations + source term

  coupled with tracer equation for Lorentz force

- Mathematical challenges

  high-resolution scheme for time-dependent conservation laws;

  positivity-preservation of density and pressure; failsafe strategy

High-resolution schemes for $\partial_t U + \mathbf{A} \cdot \nabla U = 0$

- High-order scheme

$$M_C \frac{\mathrm{d}U^H}{\mathrm{d}t} = KU^H$$

- Low-order scheme

$$M_L \frac{\mathrm{d}U^L}{\mathrm{d}t} = LU^L, \quad L = K + D$$

- **Predictor:** Compute low-order solution

$$M_L \frac{\mathrm{d}U^L}{\mathrm{d}t} = LU^L \qquad \Rightarrow \qquad \dot{U}^L \approx M_L^{-1} LU^L$$

- **Corrector:** Apply limited antidiffusion

$$M_L U = M_L U^L + \overline{F}, \qquad F = [M_L - M_C]\dot{U}^L - DU^L$$

High-resolution schemes for $\partial_t U + \mathbf{A} \cdot \nabla U = 0$

- High-order scheme

$$M_C \frac{\mathrm{d}U^H}{\mathrm{d}t} = KU^H$$

- Low-order scheme

$$M_L \frac{\mathrm{d}U^L}{\mathrm{d}t} = LU^L, \quad L = K + D$$

- **Predictor:** Compute low-order solution

$$M_L \frac{\mathrm{d}U^L}{\mathrm{d}t} = LU^L \qquad \Rightarrow \qquad \dot{U}^L \approx M_L^{-1} LU^L$$

- **Corrector:** Apply limited antidiffusion

$$M_L U = M_L U^L + \overline{F}, \qquad F = [M_L - M_C]\dot{U}^L - DU^L$$

⚠ Low-order scheme must satisfy physical constraints

Design principles of FCT schemes

Perform flux correction such that ***mass is conserved****.*

Design principles of FCT schemes

Perform flux correction such that ***mass is conserved****.*

- Conservative flux decomposition

$$m_i U_i = m_i U_i^L + \sum_{j \neq i} F_{ij}, \qquad F_{ji} = -F_{ij}$$

Design principles of FCT schemes

*Perform flux correction such that **mass is conserved**.*

- Conservative flux decomposition and limiting

$$m_i U_i = m_i U_i^L + \sum_{j \neq i} \alpha_{ij} F_{ij}, \qquad F_{ji} = -F_{ij}, \qquad \alpha_{ji} = \alpha_{ij}$$

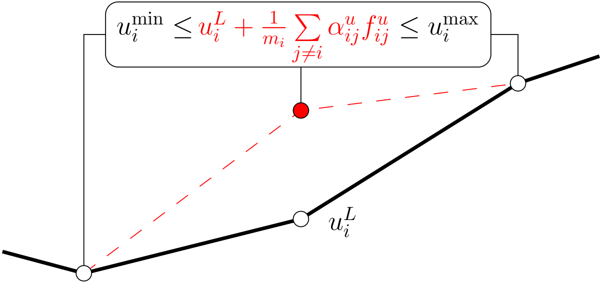
  - high-order approximation ($\alpha_{ij} = 1$) to be used in smooth regions
  - low-order approximation ($\alpha_{ij} = 0$) to be used near steep fronts

Design principles of FCT schemes, cont'd

Perform flux correction such that certain ***physical quantities are bounded*** *by the local extrema of the low-order solution.*

Design principles of FCT schemes, cont'd

Perform flux correction such that certain ***physical quantities are bounded*** *by the local extrema of the low-order solution.*



Fixed fraction flux limiter

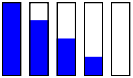
- Set $\alpha_{ij}^{(0)} := 1$ and repeat $r = 1, \dots, R$

- Mark all nodes i that violate the local FCT constraint

$$u_i^{\min} \leq u_i^L + \frac{1}{m_i} \sum_{j \neq i} \alpha_{ij}^{(r-1)} f_{ij}^u \leq u_i^{\max}$$

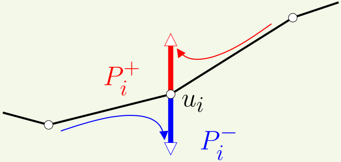
- Eliminate fixed fraction of unacceptable antidiffusion

$$\alpha_{ij}^{(r)} := \begin{cases} 1 - r/R & \text{if node } i \text{ or } j \text{ is marked} \\ \alpha_{ij}^{(r-1)} & \text{otherwise} \end{cases}$$



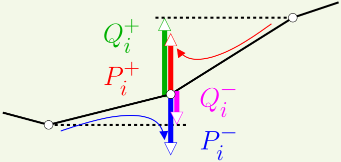
Low-order solution is recovered in the worst case: $U_i = U_i^L$

Zalesak's flux limiter



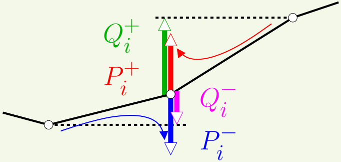
- Consider positive/negative anti-diffusive contributions separately

Zalesak's flux limiter



- Consider positive/negative anti-diffusive contributions separately
- Limit antidiffusion if it exceeds the distance to local maximum/minimum

Zalesak's flux limiter



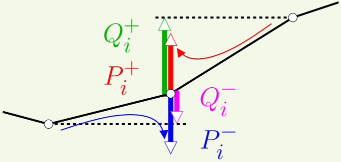
- Consider positive/negative anti-diffusive contributions separately
- Limit antidiffusion if it exceeds the distance to local maximum/minimum

- Nodal correction factors

$$R_i^+ = \min\{1, Q_i^+/P_i^+\} \quad \text{for positive fluxes into node } i$$

$$R_i^- = \min\{1, Q_i^-/P_i^-\} \quad \text{for negative fluxes into node } i$$

Zalesak's flux limiter



- Consider positive/negative anti-diffusive contributions separately
- Limit antidiffusion if it exceeds the distance to local maximum/minimum

- Nodal correction factors

$$R_i^+ = \min\{1, Q_i^+/P_i^+\} \quad \text{for positive fluxes into node } i$$

$$R_i^- = \min\{1, Q_i^-/P_i^-\} \quad \text{for negative fluxes into node } i$$

- Limit antidiffusive flux for edge ij by the minimum of R_i and R_j

Flux limiting for systems

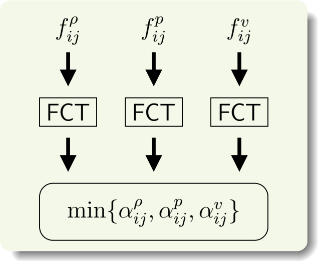
Apply flux limiter to a ***set of control variables****, e.g.,*

the primitive variables density, pressure and velocity.

Flux limiting for systems

Apply flux limiter to a ***set of control variables****, e.g.,*

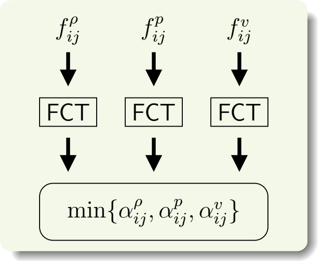
the primitive variables density, pressure and velocity.



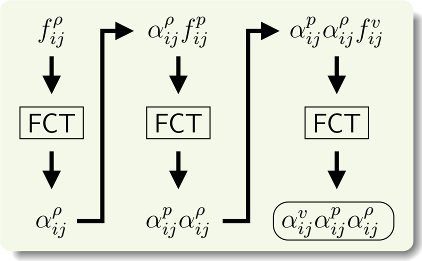
Flux limiting for systems

Apply flux limiter to a ***set of control variables****, e.g.,*

the primitive variables density, pressure and velocity.



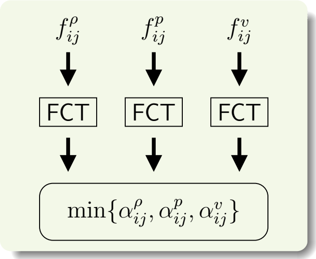
or



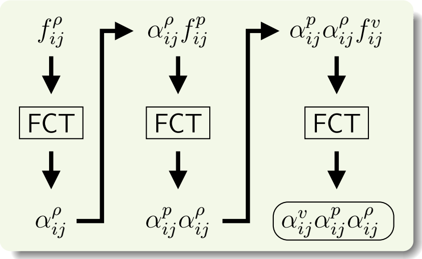
Flux limiting for systems

*Apply flux limiter to a **set of control variables**, e.g.,*

the primitive variables density, pressure and velocity.



or



Nodal transformation of variables $V_i = \mathcal{T}(U_i)U_i$ and $G_{ij} = \mathcal{T}(U_i)F_{ij}$

Failsafe flux correction algorithm

1. Compute low-order solution at time t^{n+1}

$$M_L \frac{U^L - U^n}{\Delta t} = \theta L U^L + (1-\theta) L U^n$$

prediction

2. Perform flux correction by Zalesak's limiter

$$m_i U_i^{(0)} = m_i U_i^L + \sum_{j \neq i} \alpha_{ij} F_{ij}$$

correction

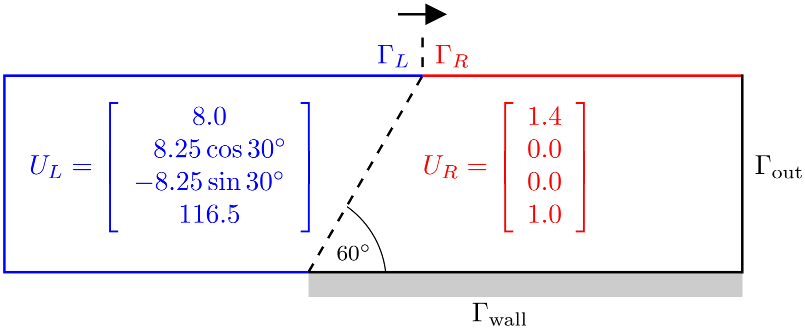
3. Eliminate spurious undershoots/overshoots

$$m_i U_i^{(r)} = m_i U_i^L + \sum_{j \neq i} \alpha_{ij}^{(r)} [\alpha_{ij} F_{ij}]$$

failsafe step

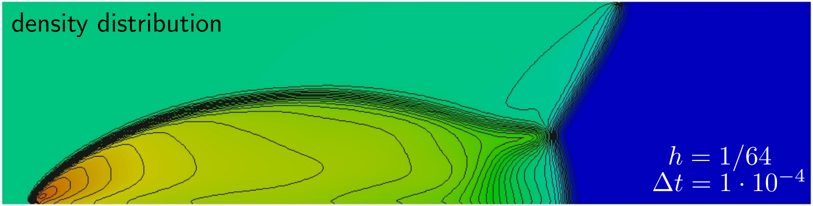
Double Mach reflection

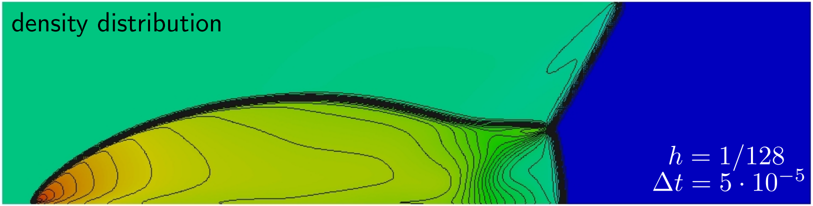
From: P.R. Woodward and P. Colella, JCP 54, 115 (1984)



Double Mach reflection, cont'd

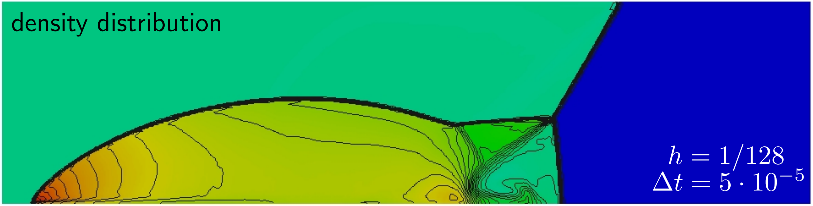
Solution at $T = 0.2$ computed by **low-order scheme** ($\alpha_{ij} \equiv 0$)





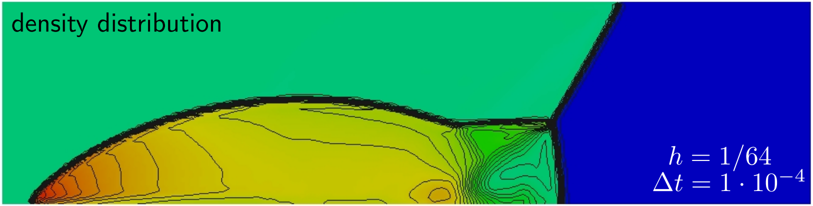
Double Mach reflection, cont'd

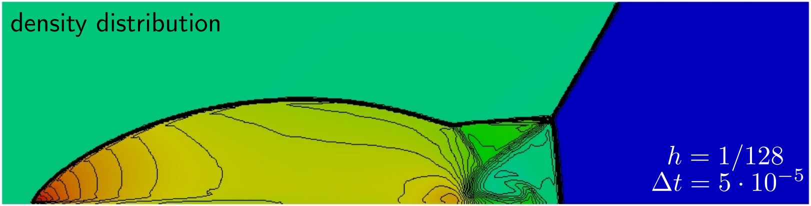
Solution at $T = 0.2$ computed by **fixed fraction flux limiter**



Double Mach reflection, cont'd

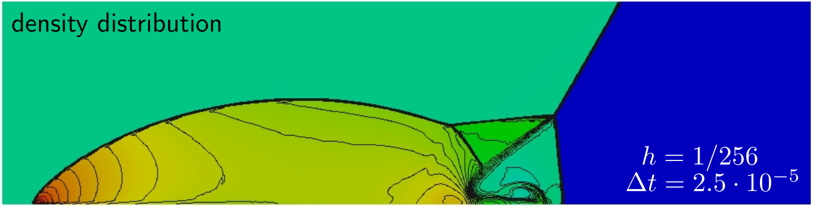
Solution at $T = 0.2$ computed by **Zalesak's flux limiter** ($\alpha_{ij} = \alpha_{ij}^p \alpha_{ij}^\rho$)





Double Mach reflection, cont'd

Solution at $T = 0.2$ computed by **Zalesak's flux limiter** ($\alpha_{ij} = \alpha_{ij}^{p}\alpha_{ij}^{\rho}$)



Numerical studies: D. Kuzmin, M. Shashkov, J. Shadid, M.M. (to appear in JCP)

Idealized Z-pinch implosion model by Banks and Shadid

- Generalized Euler system coupled with scalar tracer equation

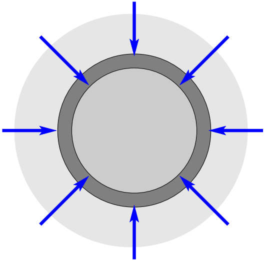
$$\frac{\partial}{\partial t}\begin{bmatrix} \rho \\ \rho\mathbf{v} \\ \rho E \\ \rho\lambda \end{bmatrix} + \nabla\cdot\begin{bmatrix} \rho\mathbf{v} \\ \rho\mathbf{v}\otimes\mathbf{v} + p\mathcal{I} \\ \rho E\mathbf{v} + p\mathbf{v} \\ \rho\lambda\mathbf{v} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{f}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

- Equation of state

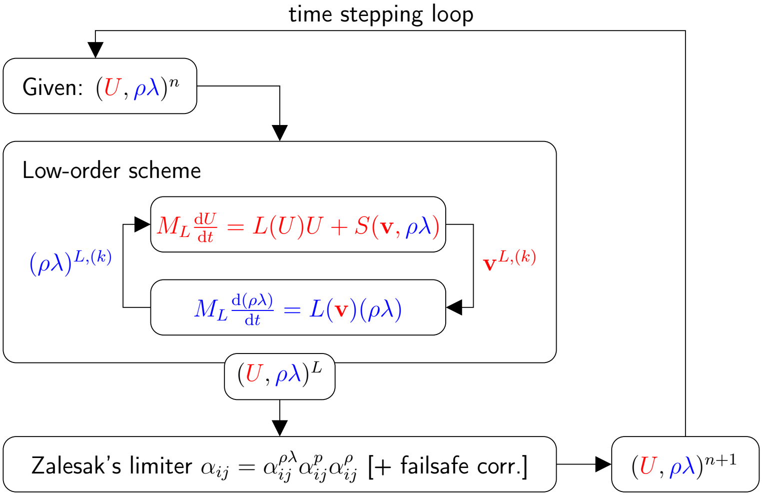
$$p = (\gamma - 1)\rho(E - 0.5|\mathbf{v}|^2)$$

- Non-dimensional Lorentz force

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

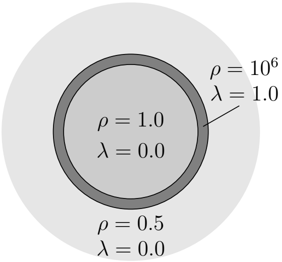


Coupled solution algorithm

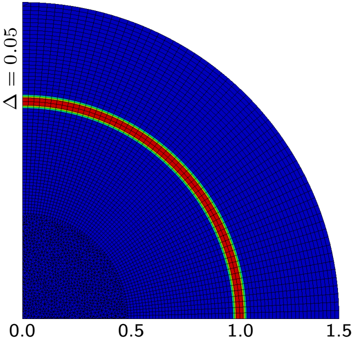


Idealized Z-pinch implosion

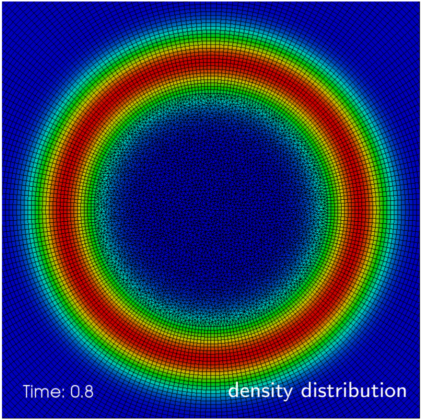
From: J.W. Banks and J.N. Shadid, JCP 61, 725 (2009)



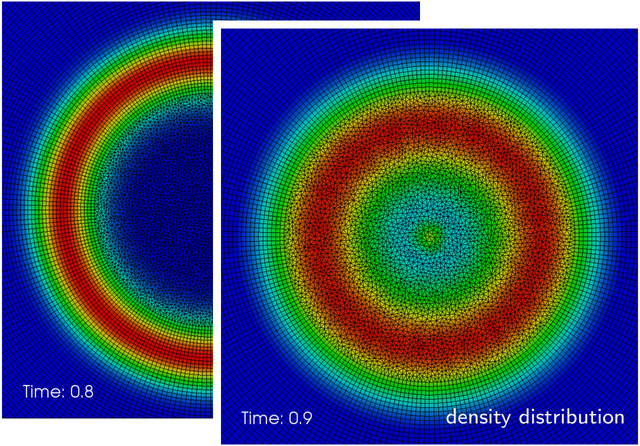
$\mathbf{v} = 0.0$, $p = 1.0$ everywhere



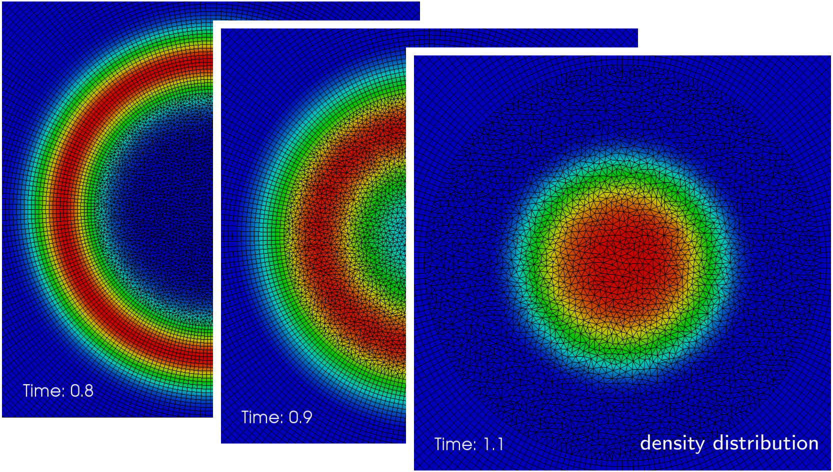
Idealized Z-pinch implosion, cont'd



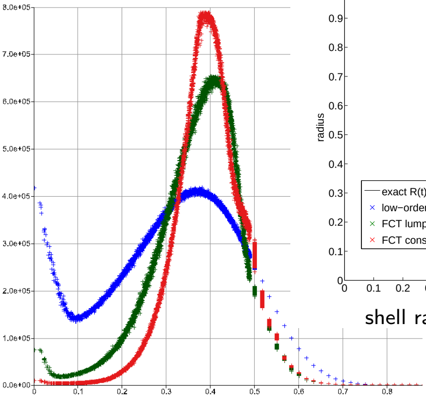
Idealized Z-pinch implosion, cont'd



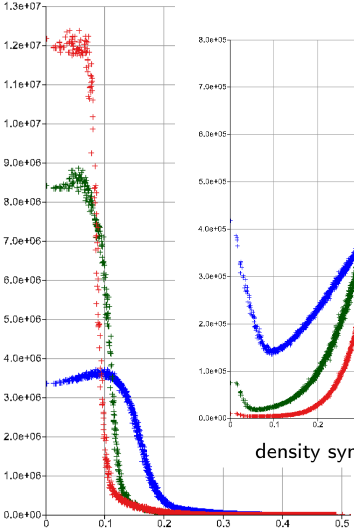
Idealized Z-pinch implosion, cont'd



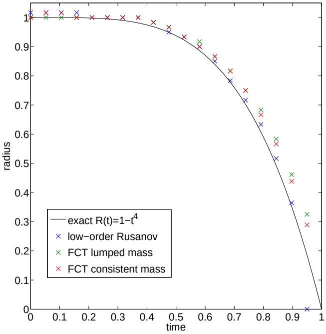
Idealized Z-pinch implosion, cont'd



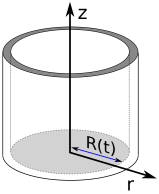
density symmetry plot, $t = 0.9$



density symmetry plot, $t = 1.05$

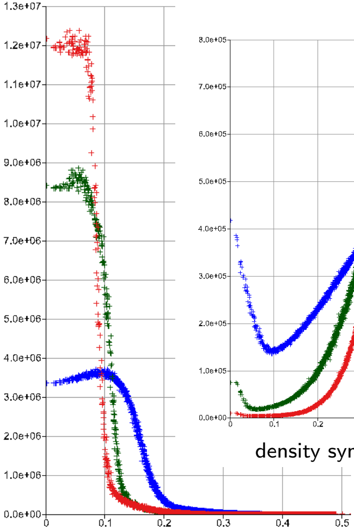


shell radius as a function of time

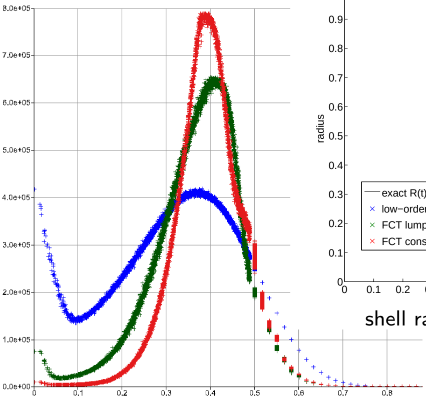


Idealized Z-pinch implosion, cont'd

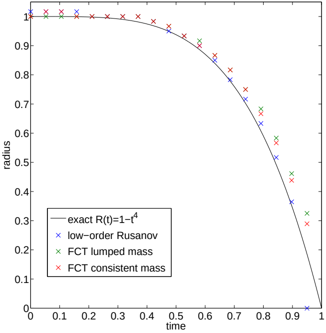
Initialization



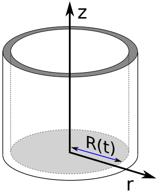
density symmetry plot, $t = 1.05$



density symmetry plot, $t = 0.9$



shell radius as a function of time



Conclusions and outlook

- Linearized flux correction algorithm for time-dependent flows

  mass conservation, boundedness of physical quantities,

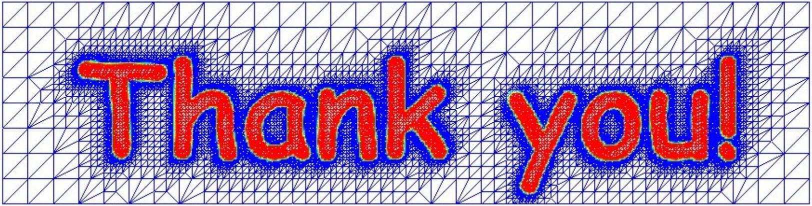
  failsafe strategy if density/pressure becomes negative

- Coupled solution algorithm for idealized Z-pinch implosions

  positivity and symmetry preservation on unstructured grids

- Todo: comparative study and extension to 'realistic' scenarios

  current drive, r-z plane, RT-instabilities, AMR


Thank you!

Appendix

Extended version of Zalesak's FCT limiter

Return

Input: auxiliary solution u^L and antidiffusive fluxes f_{ij}^u, where $f_{ji}^u \neq f_{ij}^u$

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

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

2. Upper/lower bounds based on the local extrema of u^L

$$Q_i^+ = m_i(u_i^{\max} - u_i^L), \qquad Q_i^- = m_i(u_i^{\min} - u_i^L)$$

3. Correction factors $\alpha_{ij}^u = \alpha_{ji}^u$ to satisfy the FCT constraints

$$\alpha_{ij}^u = \min\{R_{ij}, R_{ji}\}, \quad R_{ij} = \begin{cases} \min\{1, Q_i^+/P_i^+\} & \text{if } f_{ij}^u \geq 0 \\ \min\{1, Q_i^-/P_i^-\} & \text{if } f_{ij}^u < 0 \end{cases}$$

Node-based transformation of control variables

Return

- Conservative variables: density, momentum, total energy

$$U_i = [\rho_i, (\rho\mathbf{v})_i, (\rho E)_i], \qquad F_{ij} = \left[f_{ij}^{\rho}, \mathbf{f}_{ij}^{\rho v}, f_{ij}^{\rho E}\right], \qquad F_{ji} = -F_{ij}$$

- Primitive variables $V = TU$: density, velocity, pressure

$$V_i = [\rho_i, \mathbf{v}_i, p_i], \qquad \mathbf{v}_i = \frac{(\rho\mathbf{v})_i}{\rho_i}, \qquad p_i = (\gamma - 1)\left[(\rho E)_i - \frac{|(\rho\mathbf{v})_i|^2}{2\rho_i}\right]$$

$$G_{ij} = \left[f_{ij}^{\rho}, \mathbf{f}_{ij}^{v}, f_{ij}^{p}\right] = T(U_i)F_{ij}, \qquad T(U_j)F_{ji} = G_{ji} \neq -G_{ij}$$

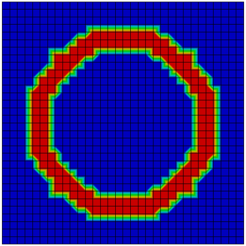
- Raw antidiffusive fluxes for the velocity and pressure

$$\mathbf{f}_{ij}^{v} = \frac{\mathbf{f}_{ij}^{\rho v} - \mathbf{v}_i f_{ij}^{\rho}}{\rho_i}, \qquad f_{ij}^{p} = (\gamma - 1)\left[\frac{|\mathbf{v}_i|^2}{2} f_{ij}^{\rho} - \mathbf{v}_i \cdot \mathbf{f}_{ij}^{\rho v} + f_{ij}^{\rho E}\right]$$

Constrained initialization

- Pointwise initialization

$$U(\mathbf{x}_i) = U_0(\mathbf{x}_i)$$



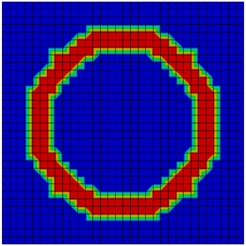
$$\rho = \begin{cases} 1.0 & \text{in } \Omega_1 \\ 0.01 & \text{in } \Omega_2 \end{cases}$$

$$u = v = 0.0,\ p = 1.0$$

Constrained initialization

- Pointwise initialization

$$U(\mathbf{x}_i) = U_0(\mathbf{x}_i)$$



$$\rho = \begin{cases} 1.0 & \text{in } \Omega_1 \\ 0.01 & \text{in } \Omega_2 \end{cases}$$

$u = v = 0.0$, $p = 1.0$

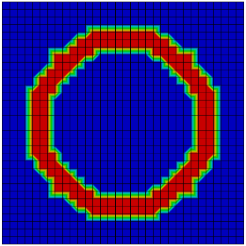
- Conservative initialization

$$\int_\Omega w U_h \,\mathrm{d}x = \int_\Omega w U_0 \,\mathrm{d}x$$

Constrained initialization

- Pointwise initialization

$$U(\mathbf{x}_i) = U_0(\mathbf{x}_i)$$



$$\rho = \begin{cases} 1.0 & \text{in } \Omega_1 \\ 0.01 & \text{in } \Omega_2 \end{cases}$$

$$u = v = 0.0,\ p = 1.0$$

- Conservative initialization

$$\int_\Omega w U_h \, \mathrm{d}x = \int_\Omega w U_0 \, \mathrm{d}x$$

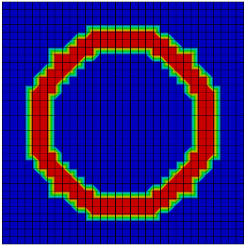
- Consistent L_2-projection

$$\sum_j m_{ij} U_j^H = \int_\Omega \varphi_i U_0 \, \mathrm{d}x$$

Constrained initialization

- Pointwise initialization

$$U(\mathbf{x}_i) = U_0(\mathbf{x}_i)$$



$$\rho = \begin{cases} 1.0 & \text{in } \Omega_1 \\ 0.01 & \text{in } \Omega_2 \end{cases}$$

$$u = v = 0.0,\ p = 1.0$$

- Conservative initialization

$$\int_\Omega w U_h \,\mathrm{d}x = \int_\Omega w U_0 \,\mathrm{d}x$$

- Consistent L_2-projection

$$\sum_j m_{ij} U_j^H = \int_\Omega \varphi_i U_0 \,\mathrm{d}x$$

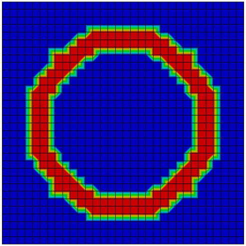
- Mass-lumped L_2-projection

$$m_i U_i^L = \int_\Omega \varphi_i U_0 \,\mathrm{d}x$$

Constrained initialization

- Pointwise initialization

$$U(\mathbf{x}_i) = U_0(\mathbf{x}_i)$$



$$\rho = \begin{cases} 1.0 & \text{in } \Omega_1 \\ 0.01 & \text{in } \Omega_2 \end{cases}$$

$$u = v = 0.0,\ p = 1.0$$

- Conservative initialization

$$\int_\Omega w U_h \,\mathrm{d}x = \int_\Omega w U_0 \,\mathrm{d}x$$

- Consistent L_2-projection

$$\sum_j m_{ij} U_j^H = \int_\Omega \varphi_i U_0 \,\mathrm{d}x$$

- Mass-lumped L_2-projection

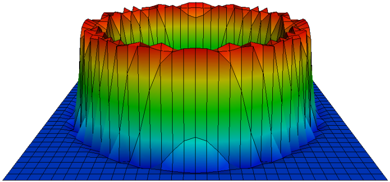
$$m_i U_i^L = \int_\Omega \varphi_i U_0 \,\mathrm{d}x$$

- Limited L_2-projection $(0 \le \alpha_{ij} \le 1)$

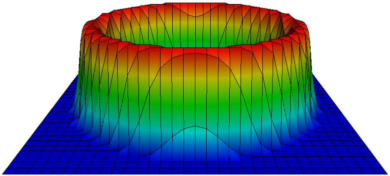
$$m_i U_i = m_i U_i^L + \sum_{j \neq i} \alpha_{ij} m_{ij} (U_i^L - U_j^L)$$

Initialization for bilinear elements

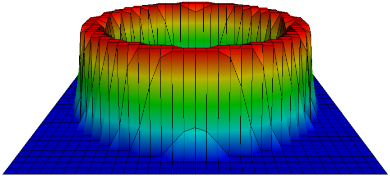
(a) consistent L_2-projection



(b) lumped L_2-projection



(c) L_2-projection, $\alpha_{ij} = \alpha_{ij}^{\rho}$

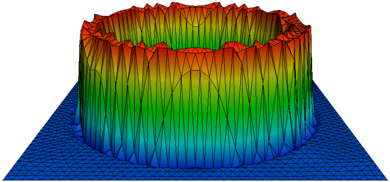


	bilinear elements, 3×3 Gauss rule		
	$\|\rho - \rho_h\|_2$	$\min(\rho_h)$	$\max(\rho_h)$
(a)	1.048e-1	-1.031e-1	1.098
(b)	1.168e-1	1.000e-2	1.000
(c)	1.103e-1	1.000e-2	1.000

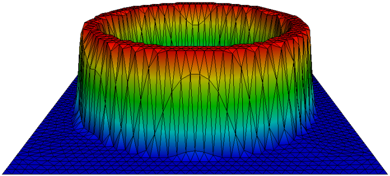
computed by adaptive cubature formulae

Initialization for linear elements Conclusions

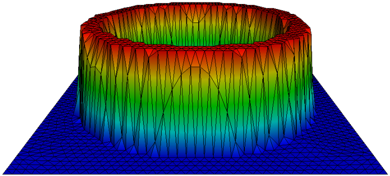
(a) consistent L_2-projection



(b) lumped L_2-projection



(c) L_2-projection, $\alpha_{ij} = \alpha_{ij}^{\rho}$



	linear elements, 3-point Gauss rule		
	$\|\rho - \rho_h\|_2$	$\min(\rho_h)$	$\max(\rho_h)$
(a)	1.206e-1	-7.143e-2	1.088
(b)	1.357e-1	1.000e-2	1.000
(c)	1.259e-1	1.000e-2	1.000

computed by adaptive cubature formulae