Divergence preserving reconstruction of nodal components of the vector field from its normal components to the faces

Richard Liska[1], **Mikhail Shashkov**[2]

[1] *Faculty of Nuclear Sciences and Physical Engineering,*
Czech Technical University in Prague, Czech Republic

[2] *X-Computational Physics Division , Group XCP-4 ,*
Los Alamos National Laboratory, Los Alamos, USA

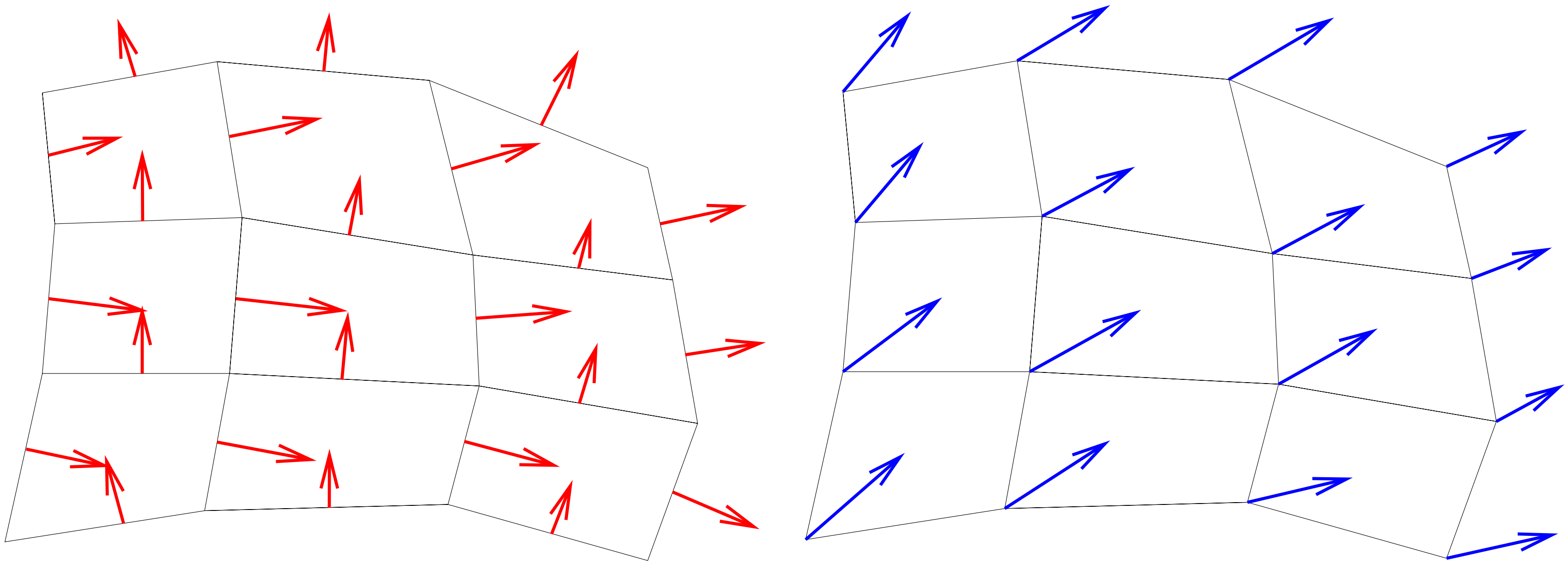
Multimat, Würzburg, 8 Sept. 2015

Overview

- motivation
- existing local methods – do not preserve divergence
- treating boundaries for local methods
- problem statement
- divergence preserving global methods
- treating boundaries for global methods
- numerical examples

Motivation

- **2 different representation of vector field on the mesh – normal components to the faces, vectors at nodes**

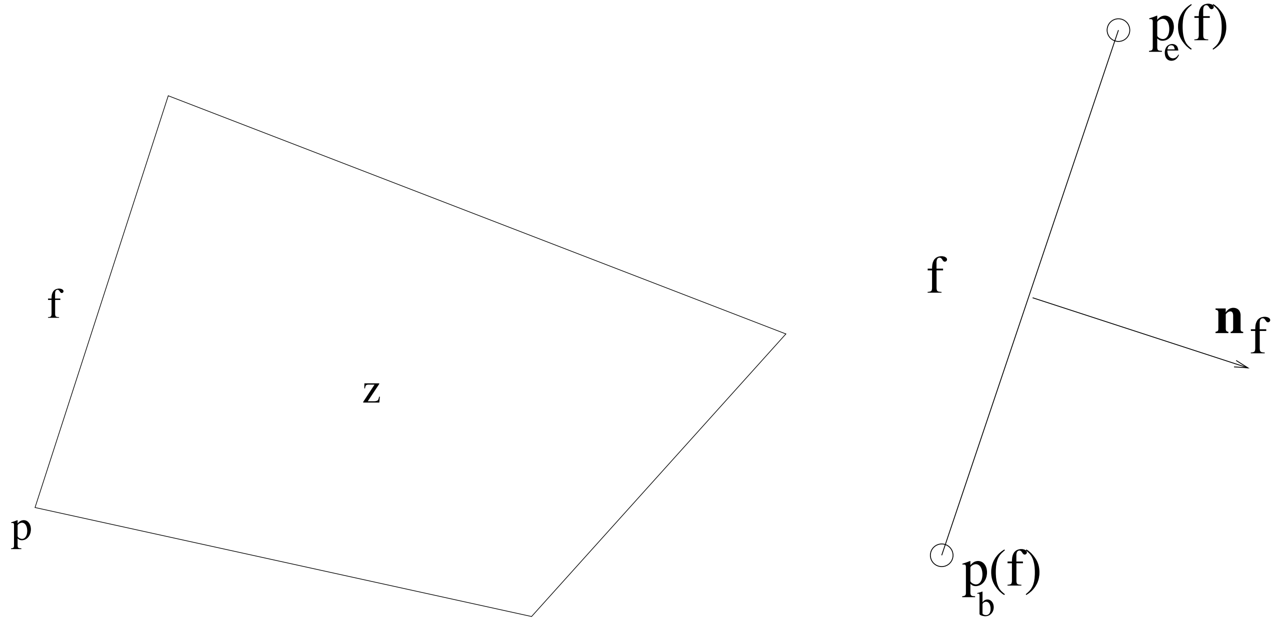


Motivation

- **part of discrete vector and tensor calculus**
- **Lagrangian gas dynamics discretizations based on Godunov method**
  - **the normal component of a vector on an edge between two cells is computed from the solution of a 1D Riemann problem in the direction orthogonal to the edge**
  - **but the Cartesian components are needed at nodes in order to compute the nodal motion**

Notations

- **zone (cell)** z**; point (vertex, node)** $\mathbf{p} = (x, y)$**; face (edge)** f**;** $\mathbf{p}_b(f)$ **and** $\mathbf{p}_e(f)$ **begin and end points of the face; oriented unit normal to the face** $\mathbf{n}_f = (n_f^x,\, n_f^y)$**.**



- **the center of the face** $\mathbf{f} = (\mathbf{p}_b(f) + \mathbf{p}_e(f))/2$

- **vector field** $\mathbf{w} = (u,\, v)$ **with nodal components** $\mathbf{w}_p = (u_p,\, v_p)$ **and normal face components** $w_f = (\mathbf{w},\, \mathbf{n}_f)$**.**

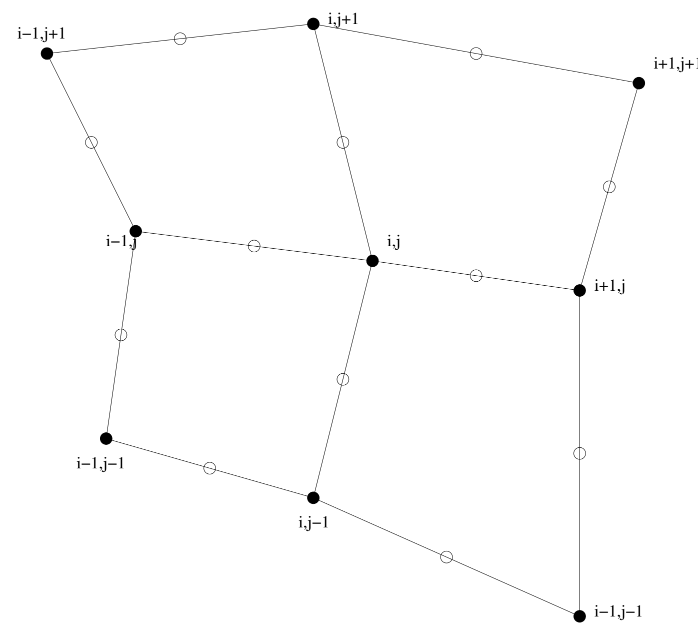
State of the art – local method

- [Shashkov, Swartz, Wendroff, JCP 1998] **1-st order method – minimization of the functional**

$$\Phi_p^{L1} = \sum_{f \in \mathcal{S}(p)} \left\{ \left([\mathbf{w}_p + (\mathbf{f} - \mathbf{p}) \cdot \nabla \mathbf{w}_p], \mathbf{n}_f \right) - w_f \right\}^2 ,$$

for unknowns $\mathbf{w}_p$ and its gradient $\nabla \mathbf{w}_p = \begin{pmatrix} \frac{\partial u_p}{\partial x} & \frac{\partial u_p}{\partial y} \\ \frac{\partial v_p}{\partial x} & \frac{\partial v_p}{\partial y} \end{pmatrix}$

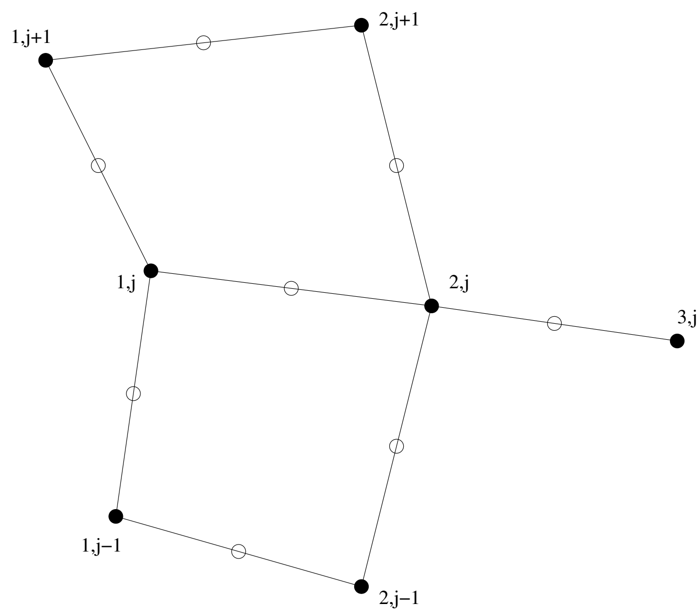
- **stencil $\mathcal{S}(p)$ of the functional Φ^{L1} – 12 edges**
- **differentiating of the functional Φ_p^{L1} with respect to 2 components of $\mathbf{w}_p$ and 4 components of $\nabla \mathbf{w}_p$ gives 6 linear equations for 6 unknowns**

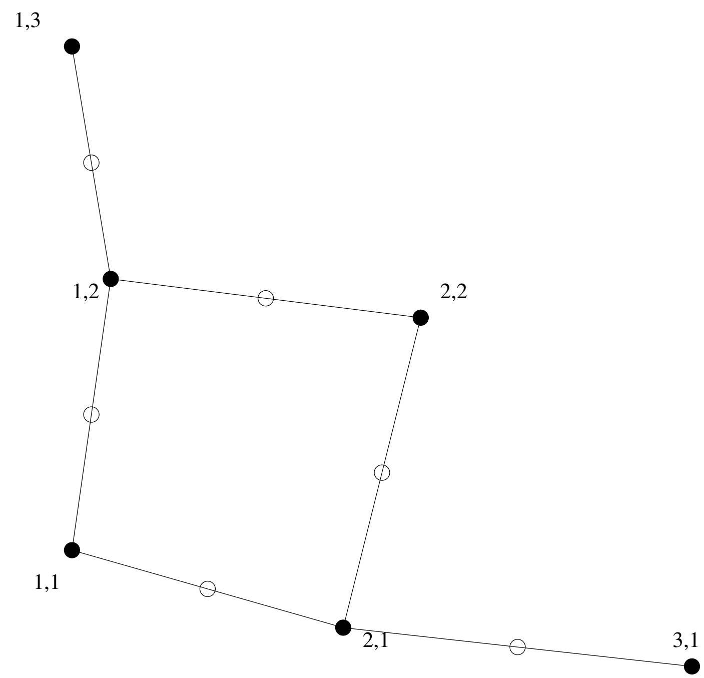


Treating boundaries for local method

- extending the 1-st order local method to treat also boundaries

- stencil on the left boundary extended by horizontal internal edge to be able to approximate $\frac{\partial v_p}{\partial x}$ on OG mesh





- stencil at the lower-left boundary corner extended by 2 edges

Piston/wall velocity boundary conditions

- assume left boundary is flat vertical piston moving with velocity U, i.e. at left boundary point $p_{1,j}$ the x-velocity component is $u_{1,j} = U$ and and its y derivative $\frac{\partial u_{1,j}}{\partial y} = 0$

- substitute these constant values into the functional $\Phi^{L1}_{1,j}$, which remains function of 4 unknowns, namely $v_{1,j}$ and 3 remaining components of $\nabla \mathbf{w}_{1,j}$

- differentiating $\Phi^{L1}_{1,j}$ with respect to the 4 unknowns gives 4 linear equations for 4 unknonws

- similarly on lower, upper and right walls – normal component and its transversal derivative are zero

- at corners both velocity components are given

Problem statement

- **divergence acting from faces to zones**

$$\left(\mathbf{DIV}^f\,\mathbf{w}\right)_z = \frac{\left(\mathbf{D}^f\,\mathbf{w}\right)_z}{V_z}, \quad \left(\mathbf{D}^f\,\mathbf{w}\right)_z = \sum_{f\in\mathcal{F}(z)} w_f\,S_f\,\psi_{f,z},$$

V_z – volume of the zone, S_f – length of the face, $\psi_{f,z} \in \{1,-1\}$ – orientation of the normal, $\mathcal{F}(z)$ – set of all faces of the zone z.

- **divergence acting from nodes to zones**

$$\left(\mathbf{DIV}^p\,\mathbf{w}\right)_z = \frac{\left(\mathbf{D}^p\,\mathbf{w}\right)_z}{V_z}, \quad \left(\mathbf{D}^p\,\mathbf{w}\right)_z = \sum_{f\in\mathcal{F}(z)} \left[\left(\frac{\mathbf{w}_{p_b(f)} + \mathbf{w}_{p_e(f)}}{2}, \mathbf{S}_{f,z}\right)\right]$$

where $\mathbf{S}_{f,z} = S_f\,\mathbf{n}_f\psi_{f,z}$

- **from face components w_f construct nodal components $\mathbf{w}_p$ preserving divergence $\left(\mathbf{D}^p\,\mathbf{w}\right)_z = \left(\mathbf{D}^f\,\mathbf{w}\right)_z$**

Divergence preserving method

- assume to have an accurate reference (target) approximation $\mathbf{w}_p^t$ of the vector field at the nodes

- minimize the functional

$$\Phi^t(\mathbf{w}_p, \lambda_z) = \frac{1}{2} \sum_{p \in \mathcal{P}} (\mathbf{w}_p - \mathbf{w}_p^t)^2 V_p + \sum_{z \in \mathcal{Z}} \left[\lambda_z \left((\mathbf{D}^p \mathbf{w}_p)_z - (\mathbf{D}^f w_f)_z\right)\right]$$

where $\mathcal{P}$ is a set of all mesh points and $\mathcal{Z}$ is a set of all cells and λ_z are Lagrange multipliers

- differentiate the functional with respect to unknowns $\mathbf{w}_p$ and λ_z give global system of linear equations

- differentiate $\Phi^t(\mathbf{w}_p, \lambda_z$ with respect to λ_z gives preservation of divergence $(\mathbf{D}^p\, \mathbf{w})_z = \left(\mathbf{D}^f\, \mathbf{w}\right)_z$

Rearranging divergence from nodes

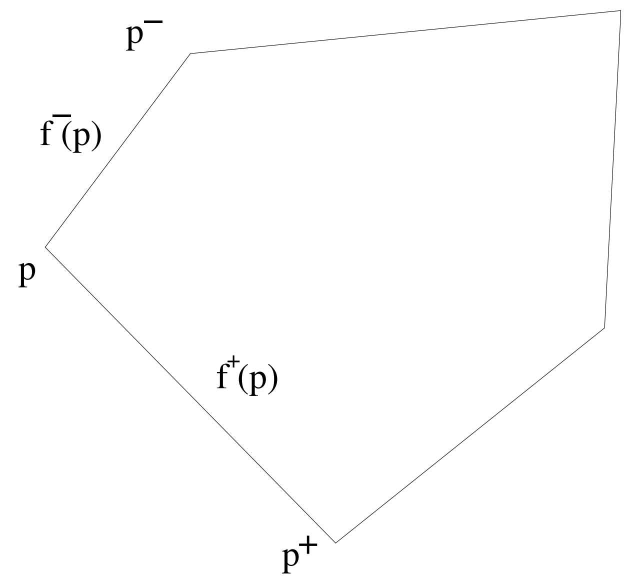
- divergence from nodes $\left(\mathbf{D}^p\,\mathbf{w}\right)_z = \sum_{f\in\mathcal{F}(z)}\left[\left(\frac{\mathbf{w}_{p_b(f)}+\mathbf{w}_{pe(f)}}{2},\,\mathbf{S}_{f,z}\right)\right]$
- rearrange to summation over points

$$\left(\mathbf{D}^p\,\mathbf{w}\right)_z = \sum_{p\in\mathcal{P}(z)}\left[\left(\mathbf{w}_p,\frac{\mathbf{S}_{f^-(p),z}+\mathbf{S}_{f^+(p),z}}{2}\right)\right] = \left(\delta_x\,u\right)_z + \left(\delta_y\,v\right)_z\,.$$

where operators δ_x and δ_y act on nodal functions u and v

$$\left(\delta_x\,u\right)_z = \frac{1}{2}\sum_{p\in\mathcal{P}(z)}\left(y_{p^+}-y_{p^-}\right)u_p$$

$$\left(\delta_y\,v\right)_z = -\frac{1}{2}\sum_{p\in\mathcal{P}(z)}\left(x_{p^+}-x_{p^-}\right)v_p\,,$$



- functional $\frac{1}{2}\sum_{p\in\mathcal{P}}(\mathbf{w}_p-\mathbf{w}_p^t)^2V_p+\sum_{z\in\mathcal{Z}}\left[\lambda_z\left((\mathbf{D}^p\mathbf{w}_p)_z-(\mathbf{D}^f w_f)_z\right)\right]$

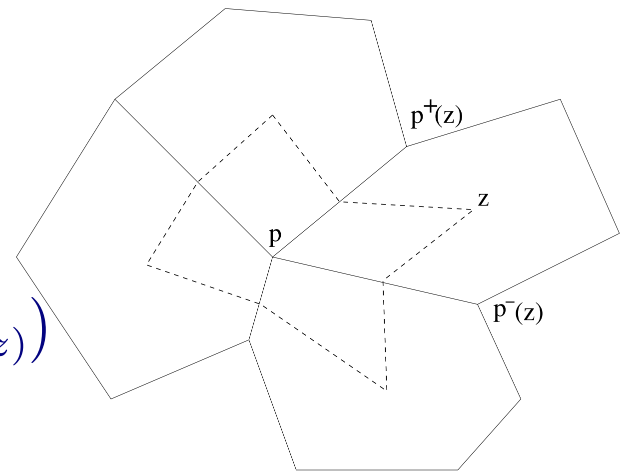
Differentiation with respect to $\mathbf{w}_p$

- a part of functional

$$\sum_{z\in\mathcal{Z}} \lambda_z (\mathbf{D}^p \mathbf{w}_p)_z = \frac{1}{2} \sum_{p\in\mathcal{P}} \sum_{z\in\mathcal{Z}(p)} \left[\left(y_{p^+(z)} - y_{p^-(z)}\right) u_p - \left(x_{p^+(z)} - x_{p^-(z)}\right) v_p \right]$$

- differentiation with respect to $\mathbf{w}_p = (u_p, v_p)$ produces

$$(\mathbf{D}^p)^\dagger_p \lambda_z = \begin{pmatrix} \left(\delta_x^\dagger \lambda_z\right)_p \\ \left(\delta_y^\dagger \lambda_z\right)_p \end{pmatrix}$$



where operators $\left(\delta_x^\dagger \lambda_z\right)_p$ and $\left(\delta_y^\dagger \lambda_z\right)_p$ act on zonal function λ_z

$$\left(\delta_x^\dagger \lambda_z\right)_p = -\frac{1}{2} \sum_{z\in\mathcal{Z}(p)} \lambda_z \left(y_{p^+(z)} - y_{p^-(z)}\right)$$

$$\left(\delta_y^\dagger \lambda_z\right)_p = \frac{1}{2} \sum_{z\in\mathcal{Z}(p)} \lambda_z \left(x_{p^+(z)} - x_{p^-(z)}\right)$$

Final system

- **functional** $\frac{1}{2}\sum_{p\in\mathcal{P}}(\mathbf{w}_p-\mathbf{w}_p^t)^2V_p+\sum_{z\in\mathcal{Z}}\left[\lambda_z\left((\mathbf{D}^p\mathbf{w}_p)_z-(\mathbf{D}^f w_f)_z\right)\right]$
- **derivative with respect to** $\mathbf{w}_p$

$$V_p(\mathbf{w}_p-\mathbf{w}_p^t)+\left((\mathbf{D}^p)^\dagger\,\lambda\right)_p=0$$

gives

$$\mathbf{w}_p=\mathbf{w}_p^t-\frac{1}{V_p}\left((\mathbf{D}^p)^\dagger\lambda_z\right)_p$$

- **substituting into the divergence constraint** $(\mathbf{D}^p\,\mathbf{w})_z=\left(\mathbf{D}^f\,\mathbf{w}\right)_z$

$$-\left[\left(\mathbf{D}^p\frac{1}{V_p}(\mathbf{D}^p)^\dagger\right)\lambda_z\right]_z=-(\mathbf{D}^p\mathbf{w}_p^t)_z+(\mathbf{D}^f w_f)_z,\quad\forall z$$

- **the system close to the system obtained for solving elliptic equation with cell centered discretization of the scalar function and nodal discretization of the vector function [Hyman, Shashkov, Steinberg, JCP 1997]**
- λ_z **by solving the SPD system, then** $\mathbf{w}_p=\mathbf{w}_p^t-\frac{1}{V_p}\left((\mathbf{D}^p)^\dagger\lambda_z\right)_p$

Piston/wall velocity boundary conditions

- **normal components are known at boundary points, thus at a boundary point P only the parallel component of**

$$\mathbf{w}_P = \mathbf{w}_P^t - \frac{1}{V_P}\left((\mathbf{D}^p)^\dagger \lambda_z\right)_P$$

is used when substituing into the divergence constraint $(\mathbf{D}^p\,\mathbf{w})_{z(P)} = \left(\mathbf{D}^f\,\mathbf{w}\right)_{z(P)},\;\; z(P) \in \mathcal{Z}(P)$

- **the part of $(\mathbf{D}^p\,\mathbf{w})_{z(P)}$ depending on the given normal component $\mathbf{w}_P^\perp$ is in**

$$-\left[\left(\mathbf{D}^p \frac{1}{V_p}(\mathbf{D}^p)^\dagger\right)\lambda_z\right]_{z(P)} = -(\mathbf{D}^p\mathbf{w}_p^t)_{z(P)} + (\mathbf{D}^f w_f)_{z(P)}$$

moved to its RHS

Implementation issues

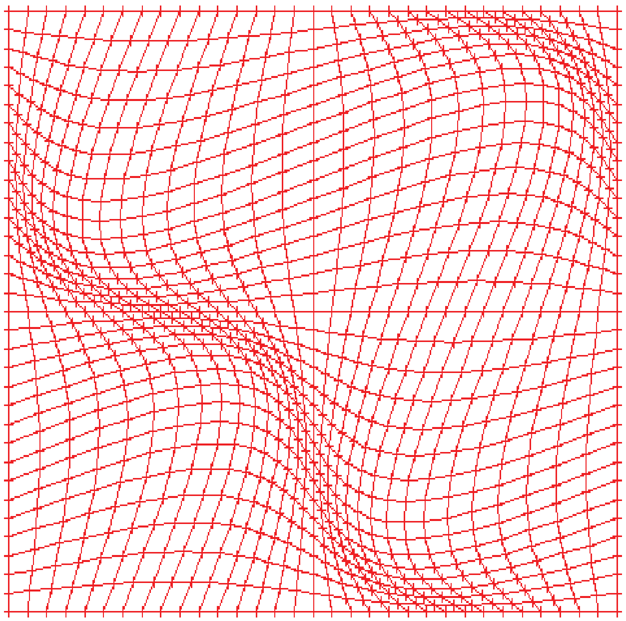
- implemented on logically rectangular mesh
- different stencils and coefficients of local functionals, different stencils of many global operators for cells/points inside, on the left/right/bottom/top boundaries and at 4 corners – quite complicated formula processing both for local and global methods
- 2 types of boundary conditions – given by w_F on boundary faces F; piston/wall boundary conditions with given normal component $\mathbf{w}_P^\perp$ for boundary points P
- computer algebra system REDUCE [A.C. Hearn, 2004] used to derive all formulas and for automatic code generation
- over 5 000 lines (300 kB) of Fortran source code generated automatically
- local methods – LAPACK used to solve small (up to 6×6) systems of linear algebraic equations
- global system for λ_z solved by the conjugate gradient method

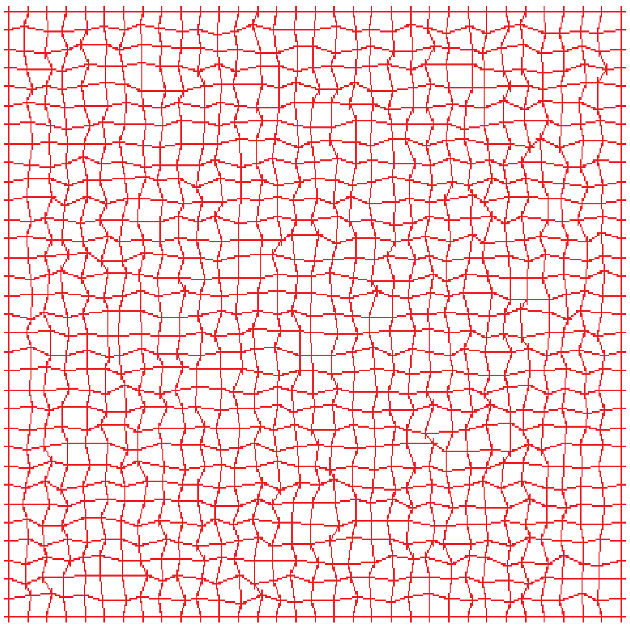
Numerical results – meshes

- **computational domain is the square** $(x, y) \in [-1/2, 1/2]^2$
- **smooth non-orthogonal mesh obtained from an uniform grid in** (ξ, η) **by the mapping**

$$
\begin{aligned}
x(\xi, \eta) &= \xi + 0.1 \sin(2\pi\xi) \sin(2\pi\eta) \\
y(\xi, \eta) &= \eta + 0.1 \sin(2\pi\xi) \sin(2\pi\eta)
\end{aligned}
$$

- **random non-smooth non-orthogonal mesh by randomly moving all internal nodes in squares with sides** $h/4$





Smooth test from [Shashkov etal. JCP 1998]

- max errors for vector field $\mathbf{w} = (x - y + x^2 - y^2, x + y + x^2 + y^2)$

grid		smooth			non-smooth		
method	M	$\mathbf{w}$	DIV	CURL	$\mathbf{w}$	DIV	CURL
local	32	0.32E-2	0.10E-1	0.98E-2	0.20E-2	0.39E-1	0.21E-1
	64	0.80E-3	0.26E-2	0.26E-2	0.54E-3	0.23E-1	0.14E-1
	128	0.20E-3	0.64E-3	0.65E-3	0.14E-3	0.11E-1	0.74E-2
	256	0.50E-4	0.16E-3	0.16E-3	0.34E-4	0.64E-2	0.42E-2
div-presr	32	0.31E-2	0.65E-13	0.97E-2	0.21E-2	0.17E-13	0.40E-1
	64	0.80E-3	0.99E-13	0.26E-2	0.57E-3	0.28E-13	0.33E-1
	128	0.20E-3	0.18E-12	0.65E-3	0.14E-3	0.77E-13	0.17E-1
	256	0.50E-4	0.43E-12	0.16E-3	0.39E-4	0.18E-12	0.94E-2

- local method repeats results from [Shashkov etal. JCP 1998]

- global method preserves divergence exactly and keeps the convergence of the vector field and curl

Shock-like test from [Shashkov etal. JCP 1998]

- max errors for vector field $\mathbf{w} = (e^{20x}/(1+e^{20x}), 0)$

grid		smooth			non-smooth		
method	M	$\mathbf{w}$	DIV	CURL	$\mathbf{w}$	DIV	CURL
local	32	0.27E-1	0.54E+0	0.15E+0	0.19E-1	0.37E+0	0.20E+0
	64	0.77E-2	0.19E+0	0.47E-1	0.50E-2	0.12E+0	0.15E+0
	128	0.20E-2	0.52E-1	0.13E-1	0.14E-2	0.81E-1	0.72E-1
	256	0.51E-3	0.13E-1	0.32E-2	0.35E-3	0.41E-1	0.46E-1
div-presr	32	0.24E-1	0.65E-13	0.15E+0	0.17E-1	0.17E-13	0.19E+0
	64	0.65E-2	0.93E-13	0.47E-1	0.45E-2	0.30E-13	0.17E+0
	128	0.17E-2	0.18E-12	0.13E-1	0.13E-2	0.74E-13	0.91E-1
	256	0.41E-3	0.32E-12	0.32E-2	0.32E-3	0.16E-12	0.47E-1
div-presr	32	0.74E-2	0.82E-13	0.15E+0	0.61E-2	0.22E-13	0.19E+0
normal	64	0.20E-2	0.71E-13	0.47E-1	0.19E-2	0.29E-13	0.17E+0
BCs	128	0.52E-3	0.17E-12	0.13E-1	0.50E-3	0.65E-13	0.91E-1
	256	0.13E-3	0.32E-12	0.32E-2	0.14E-3	0.19E-12	0.45E-1

- local method with normal BCs give very similar results
- global method preserves divergence exactly and keeps the convergence of the vector field; with normal BCs improves the errors

Noh-like velocity field

- **radially symmetric inward directed vector field** $\mathbf{w} = -|\mathbf{w}|\frac{(x,y)}{r}$

$$|\mathbf{w}| = \begin{cases} 0 & \text{for } r \le r_{min} \\ \frac{1-\cos(\pi(r-r_{min})/a)}{2} & \text{for } r_{min} \le r \le r_{min}+a \\ 1 & \text{for } r \ge r_{min}+a \end{cases}, \quad r_{min} = 0.2, a = 0.1$$

grid		smooth			non-smooth		
method	M	w	dir(w)	DIV	w	dir(w)	DIV
local	32	0.15E+0	0.32E-1	0.67E+01	0.13E+0	0.31E-1	0.50E+1
	64	0.55E-1	0.11E-1	0.27E+01	0.41E-1	0.96E-2	0.28E+1
	128	0.16E-1	0.29E-2	0.14E+01	0.12E-1	0.27E-2	0.16E+1
	256	0.43E-2	0.76E-3	0.71E+00	0.33E-2	0.78E-3	0.75E+0
div-presr	32	0.72E-1	0.37E-1	0.85E-13	0.52E-1	0.50E-1	0.25E-13
	64	0.22E-1	0.12E-1	0.17E-12	0.18E-1	0.18E-1	0.42E-13
	128	0.68E-2	0.32E-2	0.42E-12	0.61E-2	0.59E-2	0.70E-13
	256	0.17E-2	0.83E-3	0.46E-12	0.18E-2	0.17E-2	0.15E-12

Conclusion

- treating boundaries for local method
- global divergence preserving method including boundaries
- piston and wall normal boundary conditions for local and global methods
- numerical examples – global method preserves divergence and keeps convergence for vector field