Moving Mesh Discontinuous Galerkin (DG) Methods to solve Hyperbolic Conservation Laws

Gero Schnücke
University of Jena, Germany
Email: gero.schnuecke@uni-jena.de

University of Strasbourg
September 19, 2023

Collaboration with:

Nico Krais*, Thomas Bolemann† and Gregor J. Gassner‡

*Carl Zeiss (Oberkochen)

† Bosch (Stuttgart)

‡ University of Cologne

Adaptive Strategies

Real world applications, e.g. weather forecast, simulation of supernovas, simulation of turbulent flows and so on, require adaptive methodologies to reduce the computational costs and degrees of freedoms (DOFs).

h-**Refinement**:
This methodology involves automatic refinement or coarsening of the spatial mesh based on error indicators.
The method contains two independent parts: An algorithm to solve the partial differential equation (PDE) and a mesh selection algorithm.

p-**Refinement**:
The algorithm to solve the PDE is bases on a polynomial approximation in each element. A *p*-refinement method adjust the polynomial degree in each element.

r-**Refinement**:
This method relocates grid points such that the nodes remain concentrated in regions of rapid variation of the solution. Since the grid point distribution is changed in each time step, the mesh must be recalculated in each time step. $\Rightarrow$ *Moving Mesh Method*

The basic idea of a moving mesh method is to construct a transformation from a reference domain to the physical domain.

The transformation is realized by solving moving mesh PDEs or minimization problems. For instance MMPDEs for algorithms to solve hyperbolic conservation laws have been constructed by Li and Tang[1].

Assumption: The mapping or distribution of the grid points is explicitly given for an upcoming time level by a suitable moving mesh methodology.

---

[1] R. Li, T. Tang . *J. Sci. Comput.* 27 (2006), 347-363.

Discontinuous Galerkin (DG) Method

- Method uses a finite dimensional trial function space given by discontinuous piecewise polynomial functions basis.
- Choice of discontinuous basis functions gives the method a local structure (elements only communicate with immediate neighbors) and the property to handle complex mesh geometries.
  $\Rightarrow$ Attractive for parallel and high performance computing.

System of Conservation Laws

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla_x \cdot \overset{\leftrightarrow}{\mathbf{f}} = 0, \qquad \text{in } \Omega \subseteq \mathbb{R}^3.$$

The state vector and block flux $\overset{\leftrightarrow}{\mathbf{f}}$ are given by

$$\mathbf{u} = [u_1, \ldots, u_p]^T, \qquad \nabla_x \cdot \overset{\leftrightarrow}{\mathbf{f}} = \sum_{j=1}^{3} \frac{\partial \mathbf{f}_j}{\partial x_j},$$

with $u_k : \Omega \to \mathbb{R}^d, k = 1, \ldots, p$ and $\mathbf{f}_j : \mathbb{R}^p \to \mathbb{R}^p$, j=1,2,3.

The domain Ω is divided in K non-overlapping **time-dependent curved hexahedral elements** $e_\kappa(t)$, $\kappa = 1, \ldots, K$.

Transformation on Reference Element $E = [-1,1]^3$

Each element $e_\kappa(t)$ is connected with $E = [-1,1]^3$ by a time-dependent mapping

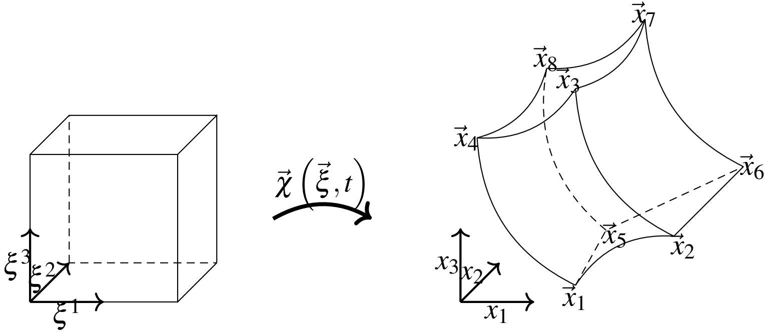
$$\vec{\chi} = [\chi_1, \chi_2, \chi_3]^T.$$

- Jacobian $J = \det \left[\begin{array}{ccc} \frac{\partial \vec{\chi}}{\partial \xi_1} & \frac{\partial \vec{\chi}}{\partial \xi_2} & \frac{\partial \vec{\chi}}{\partial \xi_3} \end{array} \right]$
- Grid velocity $\frac{\partial \vec{\chi}}{\partial t} = [v_1, v_2, v_3]^T = \vec{v}$
- Contravariant vectors:

$$J\vec{a}^1 = \frac{\partial \chi_2}{\partial \xi_2} \times \frac{\partial \chi_3}{\partial \xi_3}, \qquad J\vec{a}^2 = \frac{\partial \chi_3}{\partial \xi_3} \times \frac{\partial \chi_1}{\partial \xi_1}, \qquad J\vec{a}^3 = \frac{\partial \chi_1}{\partial \xi_1} \times \frac{\partial \chi_2}{\partial \xi_2}$$

Transformation on Reference Element $E = [-1,1]^3$

Each element $e_\kappa(t)$ is connected with the reference element $E = [-1,1]^3$ by a time-dependent mapping $\vec{\chi}$.



Transformation on Reference Element $E = [-1,1]^3$

For each element $e_\kappa(t)$ holds:

- The contravariant vectors satisfy the metric identities (MI):

$$\sum_{j=1}^{3} \frac{\partial J\vec{a}^j}{\partial \xi_j} = 0.$$

- The system becomes:

$$J\frac{\partial \mathbf{u}}{\partial t} + \nabla_\xi \cdot \overset{\leftrightarrow}{\tilde{\mathbf{f}}} = \mathbf{0}, \qquad in\ [-1,1]^3,$$
$$\overset{\leftrightarrow}{\tilde{\mathbf{f}}} = \left[\mathbf{L}\left(J\vec{a}^1, J\vec{a}^2, J\vec{a}^3\right) \otimes \mathbf{I}_p\right] \overset{\leftrightarrow}{\mathbf{f}},$$

where $\mathbf{I}_p$ is the $p \times p$ unit matrix,

$$\mathbf{L}\left(J\vec{a}^1, J\vec{a}^2, J\vec{a}^3\right) = \begin{bmatrix} J\vec{a}^1 & J\vec{a}^2 & J\vec{a}^3 \end{bmatrix} = \begin{bmatrix} J\vec{a}^1_1 & J\vec{a}^2_1 & J\vec{a}^3_1 \\ J\vec{a}^1_2 & J\vec{a}^2_2 & J\vec{a}^3_2 \\ J\vec{a}^1_3 & J\vec{a}^2_3 & J\vec{a}^3_3 \end{bmatrix}.$$

Transformation on Reference Element $E = [-1,1]^3$

For each element $e_\kappa(t)$ holds:

- Jacobian J and grid velocity $\vec{v}$ satisfy the geometric conservation law (GCL)

$$\frac{\partial J}{\partial t} = \nabla_\xi \cdot \vec{\tilde{v}}, \qquad in \ [-1,1]^3,$$
$$\vec{\tilde{v}} = \mathbf{L}\left(J\vec{a}^1, J\vec{a}^2, J\vec{a}^3\right)\vec{v}.$$

- The system becomes:

$$\frac{\partial \left(J\mathbf{u}\right)}{\partial t} + \nabla_\xi \cdot \overset{\leftrightarrow}{\tilde{\mathbf{g}}} = \mathbf{0}, \qquad in \ [-1,1]^3,$$
$$\overset{\leftrightarrow}{\tilde{\mathbf{g}}} = \left[\mathbf{L}\left(J\vec{a}^1, J\vec{a}^2, J\vec{a}^3\right) \otimes \mathbf{I}_p\right]\mathbf{g},$$
$$\mathbf{g} = \begin{bmatrix} \mathbf{f}_1 - v_1\mathbf{u} \\ \mathbf{f}_2 - v_2\mathbf{u} \\ \mathbf{f}_3 - v_3\mathbf{u} \end{bmatrix}.$$

Note: If the discretization for the contravariant vectors does not provide a discrete analogue for the MI, unphysical waves can be generated and to propagate through the grid.

$\Rightarrow$ Unphysical solution for the system!

Kopriva[1] introduced the **conservative curl form**:

$$\mathbb{J}a_1^j = -\vec{e}_j \cdot \nabla_\xi \times \left(\mathbb{I}_N\left(\chi_2 \nabla_\xi \chi_3\right)\right), \qquad j = 1,2,3,$$
$$\mathbb{J}a_2^j = -\vec{e}_j \cdot \nabla_\xi \times \left(\mathbb{I}_N\left(\chi_3 \nabla_\xi \chi_1\right)\right), \qquad j = 1,2,3,$$
$$\mathbb{J}a_3^j = -\vec{e}_j \cdot \nabla_\xi \times \left(\mathbb{I}_N\left(\chi_1 \nabla_\xi \chi_2\right)\right), \qquad j = 1,2,3.$$

with the Cartesian unit vectors $\vec{e}_j$ and interpolation operator $\mathbb{I}_N(\cdot)$.

Note $\mathbb{I}_N(\cdot)$ is computed by tensor product Lagrange polynomials with maximum degree N. The Lagrange polynomials are given by LGL points.

[1] D. A. Kopriva. *J. Sci. Comput.* 26 (2006), 301-327.

Discrete Contravariant Vectors

The discrete contravariant vectors $\mathbb{J}\vec{a}^j$ in conservative curl form satisfy

$$\sum_{j=1}^{3} \frac{\partial \mathbb{I}_N\left(\mathbb{J}\vec{a}^j\right)}{\partial \xi_j} = 0$$

Building Blocks for the DG Discretization

- Lagrange polynomials

$$\mathscr{L}_j(\xi) := \prod_{i=0, j\neq i}^{N} \frac{\xi - \xi_i}{\xi_j - \xi_i}, \quad j = 0, \ldots N,$$

  are computed with LGL quadrature points.
- The Jacobian J and the solution $J\mathbf{u}$ are approximated by tensor product Lagrange polynomials.
- Polynomial approximations are highlighted by capital letters, e.g.

$$\begin{aligned} J_h \in \mathbb{P}^N(E) \qquad &\longleftrightarrow \qquad J \\ V_j \in \mathbb{P}^N(E) \qquad &\longleftrightarrow \qquad v_j \\ J\mathbf{U}_h \in \mathbb{P}^N(E, \mathbb{R}^p) \qquad &\longleftrightarrow \qquad J\mathbf{u} \\ \mathsf{F}_j \in \mathbb{P}^N(E, \mathbb{R}^p) \qquad &\longleftrightarrow \qquad \mathbf{f}_j \end{aligned}$$

- Integrals are approximated by a tensor product extension of a $2N-1$ accurate LGL quadrature formula.

Discrete Integrals

Notation: The discrete volume integral is given by

$$\langle \boldsymbol{\psi}, \boldsymbol{\varphi} \rangle_N = \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=0}^{N} \omega_i \omega_j \omega_k \boldsymbol{\psi}_{ijk}^T \boldsymbol{\varphi}_{ijk}, \qquad \forall \boldsymbol{\psi}, \boldsymbol{\varphi} \in \mathbb{P}^N(E, \mathbb{R}^p).$$

For a block vector $\overleftrightarrow{\mathbf{F}}$ and $\boldsymbol{\varphi} \in \mathbb{P}^N(E, \mathbb{R}^p)$, the discrete surface integral is given by

$$\begin{aligned} \int_{\partial E, N} \boldsymbol{\varphi}^T \left\{ \overleftrightarrow{\mathbf{F}} \cdot \hat{n} \right\} dS := \quad & \sum_{j,k=0}^{N} \omega_j \omega_k \left(\boldsymbol{\varphi}_{Njk}^T (\mathbf{F}_1)_{Njk} - \boldsymbol{\varphi}_{0jk}^T (\mathbf{F}_1)_{0jk} \right) \\ & + \sum_{i,k=0}^{N} \omega_i \omega_k \left(\boldsymbol{\varphi}_{iNk}^T (\mathbf{F}_2)_{iNk} - \boldsymbol{\varphi}_{i0k}^T (\mathbf{F}_2)_{i0k} \right) \\ & + \sum_{i,j=0}^{N} \omega_i \omega_j \left(\boldsymbol{\varphi}_{ijN}^T (\mathbf{F}_3)_{ijN} - \boldsymbol{\varphi}_{ij0}^T (\mathbf{F}_3)_{ij0} \right). \end{aligned}$$

Here $\hat{n}$ is the unit outward normal of E.

For each element $e_\kappa(t)$:

$$\left\langle \frac{\partial J_h}{\partial t}, \varphi \right\rangle_N = \left\langle \nabla_\xi \cdot \tilde{\vec{V}}, \varphi \right\rangle_N + \int_{\partial E,N} \varphi \left(\tilde{V}_{\hat{n}}^{\mathsf{NSF}} - \tilde{V}_{\hat{n}} \right) dS, \qquad \forall \varphi \in \mathbb{P}^N(E).$$

Notation:

- $\tilde{\vec{V}} = \mathbf{L}\left(\mathbb{J}\vec{a}^1, \mathbb{J}\vec{a}^2, \mathbb{J}\vec{a}^3\right)\vec{V}$
- $\tilde{V}_{\hat{n}}^{\mathsf{NSF}}$ is a numerical two point flux. Consistent with $\tilde{V}_{\hat{n}} = \vec{V} \cdot \hat{n}$.
- For all $i, j, k = 0, \ldots, N$:

$$\left[\nabla_\xi \cdot \tilde{\vec{V}}\right]_{ijk} = \sum_{m=0}^{N} \left(\vec{V}_{mjk} \cdot \mathbb{J}\vec{a}^1_{mjk}\right) \mathscr{L}_m'(\xi_i) + \left(\vec{V}_{imk} \cdot \mathbb{J}\vec{a}^2_{imk}\right) \mathscr{L}_m'(\xi_j) + \left(\vec{V}_{ijm} \cdot \mathbb{J}\vec{a}^3_{ijm}\right) \mathscr{L}_m'(\xi_k).$$

For each element $e_\kappa(t)$:

$$\left\langle \frac{\partial\left(J\mathbf{U}_h\right)}{\partial t}, \boldsymbol{\varphi} \right\rangle_N = -\left\langle \nabla_\xi \cdot \overset{\leftrightarrow}{\tilde{\mathbf{G}}}, \boldsymbol{\varphi} \right\rangle_N - \int_{\partial E,N} \boldsymbol{\varphi}^T \left(\tilde{\mathbf{G}}_{\hat{n}}^{\mathsf{NSF}} - \tilde{\mathbf{G}}_{\hat{n}}\right) dS, \qquad \forall \boldsymbol{\varphi} \in \mathbb{P}^N\left(E, \mathbb{R}^p\right).$$

Notation:

- $\overset{\leftrightarrow}{\tilde{\mathbf{G}}} = \mathbf{L}\left(\mathbb{J}\vec{a}^1, \mathbb{J}\vec{a}^2, \mathbb{J}\vec{a}^3\right) \otimes \mathbf{I}_p \overset{\leftrightarrow}{\mathbf{G}}$
- $\tilde{\mathbf{G}}_{\hat{n}}^{\mathsf{NSF}}$ is a numerical two point flux. Consistent with $\tilde{\mathbf{G}}_{\hat{n}} = \overset{\leftrightarrow}{\mathbf{G}} \cdot \hat{n}$.
- For all $i, j, k = 0, \ldots, N$:

$$\left[\nabla_\xi \cdot \overset{\leftrightarrow}{\tilde{\mathbf{G}}}\right]_{ijk} = \sum_{m=0}^{N} \left(\overset{\leftrightarrow}{\mathbf{G}}_{mjk} \cdot \mathbb{J}\vec{a}^1_{mjk}\right) \mathscr{L}'_m\left(\xi_i\right) + \left(\overset{\leftrightarrow}{\mathbf{G}}_{imk} \cdot \mathbb{J}\vec{a}^2_{imk}\right) \mathscr{L}'_m\left(\xi_j\right) + \left(\overset{\leftrightarrow}{\mathbf{G}}_{ijm} \cdot \mathbb{J}\vec{a}^3_{ijm}\right) \mathscr{L}'_m\left(\xi_k\right).$$

Volume average operator: Let $\mathbf{A}_{ijk} \in \mathbb{R}$ for $i,j,k = 1,\ldots,N$. Then we define:

$$
\begin{aligned}
\{\{\mathbf{A}\}\}_{(i,m)jk} &= \frac{1}{2}\left(\mathbf{A}_{ijk} + \mathbf{A}_{mjk}\right), \\
\{\{\mathbf{A}\}\}_{i(jm)k} &= \frac{1}{2}\left(\mathbf{A}_{ijk} + \mathbf{A}_{imk}\right), \\
\{\{\mathbf{A}\}\}_{ij(k,m)} &= \frac{1}{2}\left(\mathbf{A}_{ijk} + \mathbf{A}_{ijm}\right).
\end{aligned}
$$

Aliasing free Split Form DG Approximation

For each element $e_\kappa(t)$:

$$\left\langle \frac{\partial J_h}{\partial t}, \varphi \right\rangle_N = \left\langle \vec{\mathbb{D}}_N \cdot \tilde{\vec{V}}, \varphi \right\rangle_N + \int\limits_{\partial E, N} \varphi \left(\tilde{V}_{\hat{n}}^{\mathsf{NSF}} - \tilde{V}_{\hat{n}} \right) dS, \qquad \forall \varphi \in \mathbb{P}^N(E),$$

$$\left\langle \frac{\partial (J\mathbf{U}_h)}{\partial t}, \boldsymbol{\varphi} \right\rangle_N = -\left\langle \vec{\mathbb{D}}_N \cdot \overset{\leftrightarrow}{\tilde{\mathbf{G}}}^{\mathsf{NVF}}, \boldsymbol{\varphi} \right\rangle_N - \int\limits_{\partial E, N} \boldsymbol{\varphi}^T \left(\tilde{\mathbf{G}}_{\hat{n}}^{\mathsf{NSF}} - \tilde{\mathbf{G}}_{\hat{n}} \right) dS, \qquad \forall \boldsymbol{\varphi} \in \mathbb{P}^N(E, \mathbb{R}^p).$$

For all $i, j, k = 0, \ldots, N$:

$$\begin{aligned} \vec{\mathbb{D}}_N \cdot \tilde{\vec{V}}_{ijk} = \sum_{m=0}^{N} \quad & 2\,\{\{\vec{V}\}\}_{(i,m)jk} \cdot \{\{\mathbb{J}\vec{a}^1\}\}_{(i,m)jk} \mathscr{L}'_m(\xi_i) \\ & + 2\,\{\{\vec{V}\}\}_{i(j,m)k} \cdot \{\{\mathbb{J}\vec{a}^2\}\}_{i(j,m)k} \mathscr{L}'_m(\xi_j) \\ & + 2\,\{\{\vec{V}\}\}_{ij(k,m)} \cdot \{\{\mathbb{J}\vec{a}^3\}\}_{ij(k,m)} \mathscr{L}'_m(\xi_k) \end{aligned}$$

[1] G. J. Gassner, A. R. Winters, D. A. Kopriva. *J. Comput.* 327 (2016), 39-66.

Aliasing free Split Form DG Approximation

For each element $e_\kappa(t)$:

$$\left\langle \frac{\partial J_h}{\partial t}, \varphi \right\rangle_N = \left\langle \vec{\mathbb{D}}_N \cdot \tilde{\vec{V}}, \varphi \right\rangle_N + \int_{\partial E, N} \varphi \left(\tilde{V}_{\hat{n}}^{\mathsf{NSF}} - \tilde{V}_{\hat{n}} \right) dS, \qquad \forall \varphi \in \mathbb{P}^N(E),$$

$$\left\langle \frac{\partial (J\mathbf{U}_h)}{\partial t}, \boldsymbol{\varphi} \right\rangle_N = - \left\langle \vec{\mathbb{D}}_N \cdot \overset{\leftrightarrow}{\tilde{\mathbf{G}}}{}^{\mathsf{NVF}}, \boldsymbol{\varphi} \right\rangle_N - \int_{\partial E, N} \boldsymbol{\varphi}^T \left(\tilde{\mathbf{G}}_{\hat{n}}^{\mathsf{NSF}} - \tilde{\mathbf{G}}_{\hat{n}} \right) dS, \qquad \forall \boldsymbol{\varphi} \in \mathbb{P}^N(E, \mathbb{R}^p).$$

- For all $i, j, k = 0, \ldots, N$:

$$\begin{aligned} \vec{\mathbb{D}}_N \cdot \overset{\leftrightarrow}{\tilde{\mathbf{G}}}{}^{\mathsf{NVF}}_{ijk} = \sum_{m=0}^{N} \quad & 2 \left(\overset{\leftrightarrow}{\mathbf{G}}{}^{\mathsf{NVF}} \left(\vec{V}_{ijk}, \vec{V}_{mjk}, \mathbf{U}_{ijk}, \mathbf{U}_{mjk} \right) \cdot \{\{\mathbb{J}\vec{a}^1\}\}_{(i,m)jk} \right) \mathscr{L}'_m(\xi_i) \\ & + 2 \left(\overset{\leftrightarrow}{\mathbf{G}}{}^{\mathsf{NVF}} \left(\vec{V}_{ijk}, \vec{V}_{imk}, \mathbf{U}_{ijk}, \mathbf{U}_{imk} \right) \cdot \{\{\mathbb{J}\vec{a}^2\}\}_{i(j,m)k} \right) \mathscr{L}'_m(\xi_j) \\ & + 2 \left(\overset{\leftrightarrow}{\mathbf{G}}{}^{\mathsf{NVF}} \left(\vec{V}_{ijk}, \vec{V}_{ijm}, \mathbf{U}_{ijk}, \mathbf{U}_{ijm} \right) \cdot \{\{\mathbb{J}\vec{a}^3\}\}_{ij(k,m)} \right) \mathscr{L}'_m(\xi_k) \end{aligned}$$

- $\overset{\leftrightarrow}{\mathbf{G}}{}^{\mathsf{NVF}}$ is symmetric and consistent with $\overset{\leftrightarrow}{\mathbf{G}}$.

[1] G. J. Gassner, A. R. Winters, D. A. Kopriva. *J. Comput.* 327 (2016), 39-66.

Theorem

Consider the split form moving mesh DG method with explicit Runge-Kutta method and assume:

- *Periodic boundary conditions.*
- $\forall\, e_{\kappa}(t_n)$ *the solution at time level* t_n *is given by*

$$\mathbf{U}_{ijk}^{n} = (c_1, \ldots, c_p)^T \in \mathbb{R}^p.$$

Then, the constant states c_k, $k = 1, \ldots, p$, *are preserved in each Runge-Kutta stage.*

Entropy Conservative (EC) Moving Mesh Flux

Assumption: The system is equipped with an entropy $s : \mathbb{R}^p \to \mathbb{R}$ and an entropy flux

$$\vec{f}^s = [f_1^s, f_2^s, f_3^s]^T, \qquad f_j^s : \mathbb{R}^p \to \mathbb{R}.$$

Then the entropy variables are given by $\mathbf{w} = \frac{\partial s}{\partial \mathbf{u}}$.

For $j = 1, 2, 3$ holds:

$$\frac{\partial s}{\partial x_j} = \left[\frac{\partial s}{\partial \mathbf{u}}\right]^T \frac{\partial \mathbf{u}}{\partial x_j} = \mathbf{w}^T \frac{\partial \mathbf{u}}{\partial x_j} \qquad \text{and} \qquad \frac{\partial f_j^s}{\partial x_j} = \mathbf{w}^T \frac{\partial \mathbf{f}_j}{\partial x_j}.$$

For $j = 1, 2, 3$ the chain rule formula provides:

$$\begin{aligned}
& \mathbf{w}^T \frac{\partial \mathbf{g}_j}{\partial x_j} = \mathbf{w}^T \frac{\partial \mathbf{f}_j}{\partial x_j} - \mathbf{w}^T \frac{\partial \left(v_j \mathbf{u} \right)}{\partial x_j} \\
\Leftrightarrow \quad & \mathbf{w}^T \frac{\partial \mathbf{g}_j}{\partial x_j} = \frac{\partial f_j^s}{\partial x_j} - v_j \frac{\partial s}{\partial x_j} - \frac{\partial v_j}{\partial x_j} \mathbf{w}^T \mathbf{u} \\
\Leftrightarrow \quad & \left[\frac{\partial \mathbf{w}}{\partial x_j} \right]^T \mathbf{g}_j = \frac{\partial \left(\mathbf{w}^T \mathbf{g}_j \right)}{\partial x_j} - \frac{\partial f_j^s}{\partial x_j} + v_j \frac{\partial s}{\partial x_j} + \frac{\partial v_j}{\partial x_j} \mathbf{w}^T u \\
\Leftrightarrow \quad & \left[\frac{\partial \mathbf{w}}{\partial x_j} \right]^T \mathbf{g}_j = \frac{\partial \left(\mathbf{w}^T \mathbf{f}_j - f_j^s \right)}{\partial x_j} - v_j \frac{\partial \left(\mathbf{w}^T u - s \right)}{\partial x_j}
\end{aligned}$$

Entropy Conservative (EC) Moving Mesh Flux

The orientated jump and averaged operator for primary "$-$" and secondary "$+$" states along an interface are defined by

$$[\![\cdot]\!] := (\cdot)^+ - (\cdot)^-, \quad \text{and} \quad \{\{\cdot\}\} := \frac{1}{2}\left[(\cdot)^+ + (\cdot)^-\right].$$

When applied to vectors, the average and jump operators are evaluated separately for each vector component.

$$\underbrace{\left[\frac{\partial \mathbf{w}}{\partial x_j}\right]^T \mathbf{g}_j}_{\approx [\![\mathbf{W}]\!]^T \mathbf{G}_j^*} = \underbrace{\frac{\partial\left(\mathbf{w}^T \mathbf{f}_j - f_j^s\right)}{\partial x_j}}_{\approx [\![\mathbf{W^T F_j} - \mathbf{F_j^s}]\!]^T} - \underbrace{v_j}_{\approx \{\{V_j\}\}} \underbrace{\frac{\partial\left(\mathbf{w}^T u - s\right)}{\partial x_j}}_{\approx [\![\mathbf{W}^T U - S]\!]}$$

Entropy Conservative (EC) Moving Mesh Flux

- $\mathbf{G}_j^{\mathsf{EC}}$ is an EC flux, if

$$[\![\mathbf{W}]\!]^T \mathbf{G}_j^{\mathsf{EC}} = [\![\mathbf{W}^{\mathbf{T}}\mathbf{F}_{\mathbf{j}} - \mathbf{F}_{\mathbf{j}}^{\mathbf{s}}]\!]^T - \{\{V_j\}\} [\![\mathbf{W}^T U - S]\!].$$

- $\mathbf{G}_j^{\mathsf{ES}}$ is an entropy stable (ES) flux, if

$$[\![\mathbf{W}]\!]^T \mathbf{G}_j^{\mathsf{ES}} \leq [\![\mathbf{W}^{\mathbf{T}}\mathbf{F}_{\mathbf{j}} - \mathbf{F}_{\mathbf{j}}^{\mathbf{s}}]\!]^T - \{\{V_j\}\} [\![\mathbf{W}^T U - S]\!].$$

Entropy Conservative (EC) Moving Mesh Flux

An EC moving mesh Flux can be constructed by Abgrall's[1] framework

$$\mathbf{G}_j^{\mathsf{EC}} = \{\{\mathbf{G}_j\}\} - \alpha[\![\mathbf{W}]\!],$$

with the correction parameter

$$\alpha = \frac{[\![\mathbf{W}]\!]^T\{\{\mathbf{G}_j\}\} - [\![\mathbf{W}^T\mathbf{F}_j - F_j^s]\!]^T + \{\{V_j\}\}[\![\mathbf{W}^T\mathbf{U} - s]\!]}{[\![\mathbf{W}]\!]^T[\![\mathbf{W}]\!]}.$$

1 R. Abgrall. *J. Comput.* 327 (2018), 640-666.

For a hyperbolic system, the flux Jacobian matrices $\frac{\partial \mathbf{f}_j}{\partial \mathbf{u}}, j = 1, 2, 3,$ are diagonalizable and have real eigenvalues

$$\left\{\lambda_k^j(\mathbf{u})\right\}_{k=1}^p \subseteq \mathbb{R}.$$

Thus, the flux matrices $\frac{\partial \mathbf{g}_j}{\partial \mathbf{u}}$ are diagonalizable with the same eigenvector basis.

Assume that $\mathbf{R}_j$ contains these right eigenvectors. The corresponding real eigenvectors are

$$\left\{\lambda_k^j(\mathbf{u}) - v_j\right\}_{i=1}^p \subseteq \mathbb{R}.$$

Define the matrix.

$$\Lambda_j = \text{diag}\left(\lambda_1^j(\mathbf{u}) - v_j, \ldots, \lambda_p^j(\mathbf{u}) - v_j\right)$$

Eigenvector scaling theorem[1]:
$\exists$ symmetric scaling matrices $\mathbf{T}_j$ with

$$\mathbf{S}_j = \mathbf{R}_j \mathbf{T}_j \qquad \Rightarrow \qquad \begin{cases} \frac{\partial \mathbf{g}_j}{\partial \mathbf{u}} = \mathbf{S}_j \Lambda_l \mathbf{S}_j^{-1}, \\ \frac{\partial \mathbf{u}}{\partial \mathbf{w}} = \mathbf{S}_j \mathbf{S}_j^T. \end{cases}$$

Chain rule formula provides

$$\frac{\partial \mathbf{g}_j}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{x}} = \frac{\partial \mathbf{g}_j}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{w}} \frac{\partial \mathbf{w}}{\partial x} = \left[\mathbf{S}_j \Lambda_j \mathbf{S}_j^T \right] \frac{\partial \mathbf{w}}{\partial x}.$$

Motivates in x_j-direction the dissipation term:

$$\frac{1}{2} \hat{\mathbf{S}}_j \left| \hat{\Lambda} \right|_j \hat{\mathbf{S}}_j^T [\![\mathbf{W}]\!], \qquad \hat{\mathbf{S}}_j = \hat{\mathbf{R}}_j \hat{\mathbf{T}}_j.$$

[1] T. J. Barth. In: An Introduction to Recent Developments in Theory and Numerics for Conservation Laws. Springer (1999), p. 195-285.

Note:

- $\hat{\mathbf{R}}_j$ is consistent with $\mathbf{R}_j$
- $\hat{\mathbf{T}}_j$ is consistent with $\mathbf{T}_j$
- $\hat{\mathbf{S}}_j = \hat{\mathbf{R}}_j \hat{\mathbf{T}}_j$ needs to be regular and $\left|\hat{\Lambda}\right|_j$ needs to be a SPD matrix.
  $\Rightarrow \hat{\mathbf{S}}_j \left|\hat{\Lambda}\right|_j \hat{\mathbf{S}}_j^T$ is a SPD matrix, too.
- The matrix dissipation operator $\hat{\mathbf{S}}_j \left|\hat{\Lambda}\right|_j \hat{\mathbf{S}}_j^T$ can affect the accuracy of the split form DG method.

Entropy Stable Moving Mesh Flux

Consider the EC flux $\mathbf{G}_j^{\mathsf{EC}}$ and the dissipation operator $\hat{\mathbf{S}}_j \left|\hat{\Lambda}\right|_j \hat{\mathbf{S}}_j^T$.
The flux

$$\mathbf{G}_j^{\mathsf{ES}} = \mathbf{G}_j^{\mathsf{EC}} - \frac{1}{2}\hat{\mathbf{S}}_j \left|\hat{\Lambda}\right|_j \hat{\mathbf{S}}_j^T [\![\mathbf{W}]\!]$$

satisfies

$$\begin{aligned}
[\![\mathbf{W}]\!]^T \mathbf{G}_j^{\mathsf{ES}} = &\ [\![\mathbf{W^T F_j} - \mathbf{F_j^s}]\!]^T - \{\!\{V_j\}\!\} [\![\mathbf{W}^T U - S]\!] \\
&- \frac{1}{2} [\![\mathbf{W}]\!]^T \hat{\mathbf{S}}_j \left|\hat{\Lambda}\right|_j \hat{\mathbf{S}}_j^T [\![\mathbf{W}]\!] \\
\leq &\ [\![\mathbf{W^T F_j} - \mathbf{F_j^s}]\!]^T - \{\!\{V_j\}\!\} [\![\mathbf{W}^T U - S]\!],
\end{aligned}$$

since $\left|\hat{\Lambda}\right|_j$ is a SPD matrix.

Theorem

Consider the semi-discrete split form moving mesh DG method with:

- *Periodic boundary conditions.*
- *The volume fluxes* $\mathbf{G}_j^{NVF}$*,* $j = 1, 2, 3$*, are EC.*
- *The surface fluxes* $\mathbf{G}_j^{NSF}$*,* $j = 1, 2, 3$*, are ES.*

Then the method satisfies the following discrete entropy inequality

$$\frac{d}{dt}\int_{\Omega(t)} s\,dx \approx \frac{d}{dt}\sum_{\kappa} \langle S, J_h \rangle_N \big|_{e_\kappa} \leq 0.$$

- 3D Euler Equations for an Ideal Gas
- (5,4) Low-Storage RK Method[1]
- Grid point distribution is given by;

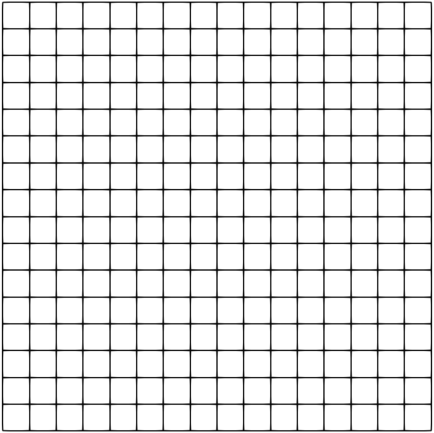
$$\vec{x}_\kappa(0) = \left(x_1^\kappa(0), x_2^\kappa(0), x_3^\kappa(0)\right)^T \in e_\kappa(0)$$

$$\vec{x}_\kappa(t) = \vec{x}_\kappa(0) + 0.05L\sin(2\pi t)\sin\left(\frac{2\pi}{L}x_1^\kappa(0)\right)\sin\left(\frac{2\pi}{L}x_2^\kappa(0)\right)\sin\left(\frac{2\pi}{L}x_3^\kappa(0)\right),$$
$$L := x_{\mathsf{max}} - x_{\mathsf{min}}.$$

[1] C. A. Kennedy, M. H. Carpenter and R. M. Lewis. *Appl. Numr. Math.* 35 (2000): 177-219.

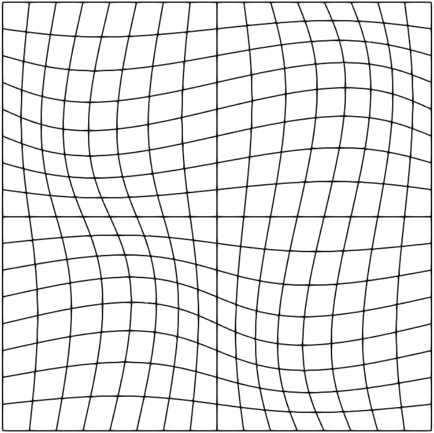
Slice through 3D Mesh

Mesh with $K = 16^3$ elements at initial time.



Slice through 3D Mesh

Mesh with $K = 16^3$ elements at its maximal distortion.



$$
\begin{aligned}
\mathbf{G}^{\mathsf{CH}}_{1,j} &= \{\!\{\rho\}\!\}^{\log}\{\!\{u_j - v_j\}\!\},\\
\mathbf{G}^{\mathsf{CH}}_{k+1,j} &= \{\!\{\rho\}\!\}^{\log}\{\!\{u_j - v_j\}\!\}\{\!\{u_k\}\!\} + \frac{\{\!\{\rho\}\!\}}{\{\!\{\frac{\rho}{p}\}\!\}}\delta_{jk}, \qquad k = 1,2,3,\\
\mathbf{G}^{\mathsf{CH}}_{5,j} &= \{\!\{\rho\}\!\}^{\log}\left[\frac{1}{\{\!\{\frac{1}{e}\}\!\}^{\log}} + \frac{1}{2}\sum_{k=1}^{3} 2\{\!\{u_k\}\!\}^2 - \{\!\{u_k^2\}\!\}\right]\{\!\{u_j - v_j\}\!\}\\
&\quad + \frac{\{\!\{\rho\}\!\}}{\{\!\{\frac{\rho}{p}\}\!\}}\{\!\{u_\iota\}\!\},
\end{aligned}
$$

where

$$
\{\!\{a\}\!\}^{\log} := \begin{cases} \frac{[\![a]\!]}{[\![\log(a)]\!]}, & a^L \neq a^R,\\ \{\!\{a\}\!\}, & a^L = a^R. \end{cases}
$$

---

[1] P. Chandrashekar. *J. Comput.* 14 (2013), 1252-1286.

Matrix Dissipation Term

Dissipation term in x_1-direction:

$$\hat{\mathbf{R}}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ \{\{u_1\}\} - \bar{c} & \{\{u_1\}\} & 0 & 0 & \{\{u_1\}\} + \bar{c} \\ \{\{u_2\}\} & \{\{u_2\}\} & 1 & 0 & \{\{u_2\}\} \\ \{\{u_3\}\} & \{\{u_3\}\} & 0 & 1 & \{\{u_3\}\} \\ \bar{h} - \{\{u_1\}\}\bar{c} & \frac{1}{2}\overline{|\vec{u}|^2} & \{\{u_2\}\} & \{\{u_3\}\} & \bar{h} + \{\{u_1\}\}\bar{c} \end{bmatrix},$$

$$\hat{\mathbf{T}}_1 = \operatorname{diag}\left(\sqrt{\frac{\{\{\rho\}\}^{\log}}{2\gamma}}, \sqrt{\frac{(\gamma-1)}{\gamma}\{\{\rho\}\}^{\log}}, \sqrt{\frac{\{\{\rho\}\}}{2\{\{\beta\}\}}}, \sqrt{\frac{\{\{\rho\}\}}{2\{\{\beta\}\}}}, \sqrt{\frac{\{\{\rho\}\}^{\log}}{2\gamma}}\right),$$

$$\left|\hat{\Lambda}\right|_1 = \operatorname{diag}\left(|\{\{u_1 - v_1\}\} - \bar{c}|, |\{\{u_1 - v_1\}\}|, |\{\{u_1 - v_1\}\}|, |\{\{u_1 - v_1\}\}|, |\{\{u_1 - v_1\}\} + \bar{c}|\right),$$

where

$$\bar{c} := \sqrt{\frac{\gamma\{\{\rho\}\}}{2\{\{\rho\}\}^{\log}\{\{\beta\}\}}}, \qquad \bar{h} := \frac{\gamma}{2(\gamma-1)\{\{\beta\}\}^{\log}} + \frac{1}{2}\sum_{k=1}^{3} 2\{\{u_k\}\}^2 - \{\{u_k^2\}\}.$$

---

[1] A. R. Winters, D. Derigs, G. J. Gassner, S. Walch. J. Comput. Phys. 332 (2017), 274-289.

Free Stream Preservation

Consider the domain $\Omega = [0, 2\pi]^3$ and the initial data

$$\mathbf{u}(x,t) = \begin{bmatrix} \rho(x,t) \\ \rho u_1(x,t) \\ \rho u_2(x,t) \\ \rho u_3(x,t) \\ E(x,t) \end{bmatrix} = \begin{bmatrix} 1 \\ 0.3 \\ 0 \\ 0 \\ 17 \end{bmatrix}.$$

Test with:
$\rightarrow$ $K = 16^3$ elements
$\rightarrow$ Periodic BC
$\rightarrow$ Final time $T = 20$

Free Stream Preservation $N = 3$

C_{CFL}	$\mathrm{L}^{\infty}(\rho)$	$\mathrm{L}^{\infty}(\rho u_1)$	$\mathrm{L}^{\infty}(\rho u_2)$	$\mathrm{L}^{\infty}(\rho u_3)$	$\mathrm{L}^{\infty}(E)$
0.95	2.47E-14	1.40E-12	4.46E-12	4.48E-12	1.33E-12
0.5	2.47E-14	1.40E-12	4.46E-12	4.48E-12	1.33E-12
0.25	2.70E-14	1.40E-12	4.46E-12	4.48E-12	1.36E-12
0.125	3.10E-14	1.40E-12	4.46E-12	4.48E-12	1.43E-12
0.0625	3.78E-14	1.40E-12	4.46E-12	4.48E-12	1.56E-12

Free Stream Preservation $N = 4$

C_{CFL}	$\mathrm{L}^\infty(\rho)$	$\mathrm{L}^\infty(\rho u_1)$	$\mathrm{L}^\infty(\rho u_2)$	$\mathrm{L}^\infty(\rho u_3)$	$\mathrm{L}^\infty(E)$
0.95	2.07E-14	1.24E-12	5.28E-12	5.21E-12	1.12E-12
0.5	2.49E-14	1.24E-12	5.28E-12	5.21E-12	1.30E-12
0.25	2.81E-14	1.24E-12	5.28E-12	5.21E-12	1.34E-12
0.125	3.32E-14	1.24E-12	5.28E-12	5.21E-12	1.40E-12
0.0625	4.24E-14	1.24E-12	5.28E-12	5.21E-12	1.59E-12

Experimental Convergence Rates

Consider the domain $\Omega = [-1,1]^3$ with the manufactured solution

$$\mathbf{u}(\vec{x},t) = \begin{bmatrix} \rho(\vec{x},t) \\ \rho u_1(\vec{x},t) \\ \rho u_2(\vec{x},t) \\ \rho u_3(\vec{x},t) \\ E(\vec{x},t) \end{bmatrix} = \begin{bmatrix} 2+0.1\sin(\pi(x_1+x_2+x_3-2\cdot 0.3t)) \\ 2+0.1\sin(\pi(x_1+x_2+x_3-2\cdot 0.3t)) \\ 2+0.1\sin(\pi(x_1+x_2+x_3-2\cdot 0.3t)) \\ 2+0.1\sin(\pi(x_1+x_2+x_3-2\cdot 0.3t)) \\ [2+0.1\sin(\pi(x_1+x_2+x_3-2\cdot 0.3t))]^2 \end{bmatrix}.$$

Test with:
$\rightarrow$ Periodic BC
$\rightarrow$ Final time $T = 5$

Experimental Convergence Rates $N = 3$

K	$L^2(\rho)$	$EOC(\rho)$	$L^2(\rho u_1)$	$EOC(\rho u_1)$	$L^2(E)$	$EOC(E)$
2^3	4.16E-02	-	3.73E-02	-	5.61E-02	-
4^3	3.77E-03	3.46	3.52E-03	3.41	6.06E-03	3.21
8^3	1.99E-04	4.25	1.75E-04	4.33	3.24E-04	4.23
16^3	5.37E-06	5.21	4.91E-06	5.16	1.20E-05	4.75
32^3	2.18E-07	4.62	2.07E-07	4.57	5.83E-07	4.36
64^3	1.45E-08	3.92	1.34E-08	3.95	3.95E-08	3.88

Note: The errors and EOCs for the x_1 and x_2 components for the momentum $\rho\vec{u}$ are not given to make the table more compact.

Experimental Convergence Rates $N = 4$

K	$L^2(\rho)$	$EOC(\rho)$	$L^2(\rho u_1)$	$EOC(\rho u_1)$	$L^2(E)$	$EOC(E)$
2^3	1.02E-02	-	9.06E-03	-	1.45E-02	-
4^3	4.53E-04	4.50	4.13E-04	4.46	7.18E-04	4.33
8^3	1.10E-05	5.37	1.02E-05	5.35	1.86E-05	5.27
16^3	1.91E-07	5.85	1.72E-07	5.88	3.81E-07	5.61
32^3	7.28E-09	4.71	6.33E-09	4.77	1.38E-08	4.78
64^3	2.79E-10	4.71	2.38E-10	4.74	5.40E-10	4.68

Note: The errors and EOCs for the x_1 and x_2 components for the momentum $\rho\vec{u}$ are not given to make the table more compact.

Simulation of flow past a plunging SD7003 airfoil

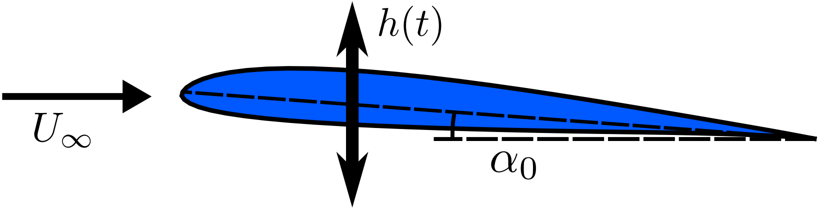


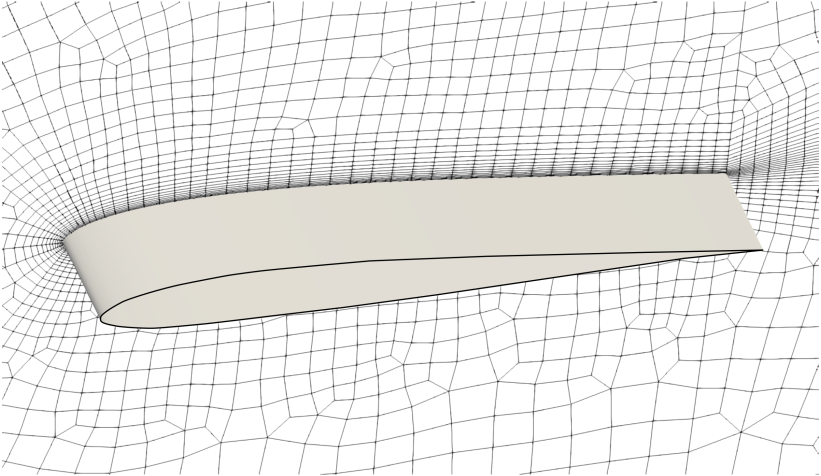
Figure: Schematic view of the forced plunging motion.

- Experimental Data[1]
- Numerical Data[2]

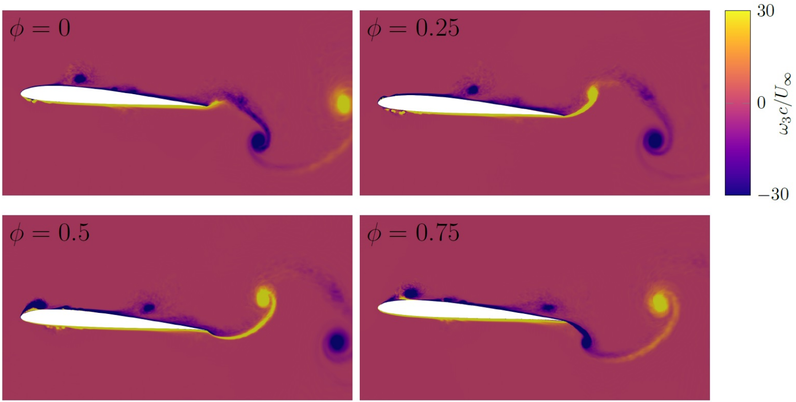
[1] G. McGowan, A. Gopalarathnam, M. Ol, J. Edwards, D. Fredberg. 46th AIAA Aerospace Sciences Meeting and Exhibit (2008).

[2] M. R. Visbal. High-fidelity simulation of transitional fows past a plunging airfoil. AIAA Journal 47 (2009), 2685-2697.

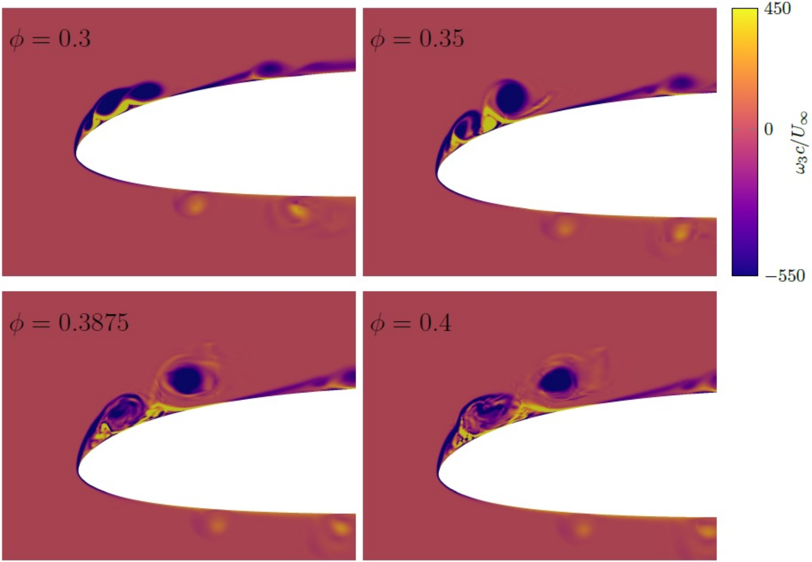
SD7003 Test: Slice through the Mesh



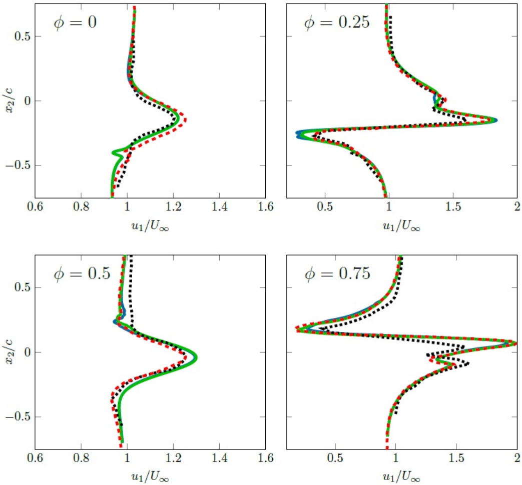
SD7003 Test: Vorticity during different Phases



SD7003 Test: Vorticity during different Phases



SD7003 Test: Comparison



- **Black** dashed experimental data from McGowan et al.
- Red dashed numerical data from Visbal
- **Green** ES moving mesh DG
- **Blue** kinetic energy dissipative (KED) moving mesh DG

SD7003 Test: PID for Moving Mesh DG Methods

The performance index (PID), defined as

$$\text{PID} = \frac{\text{wall clock time} \cdot \#\text{cores}}{\#\text{DOF} \cdot \#\text{time steps}}.$$

It measures the time it takes to advance a single DOF by one time step with the Runge-Kutta scheme.

SD7003 Test: PID for Moving Mesh DG Methods

	consistent integration	ES fluxes	KED fluxes
PID $[10^{-6}s]$	8.03	3.27	2.84

- Cray XC40 system Hazel Hen at the High-Performance Computing Center Stuttgart (HLRS).
- Open source high order DG solver FLEXI[1].

[1] www.flexi-project.org

Concluding Remarks

Assume that a methodology for the temporal grid point distribution is given. Then the constructed split form moving mesh DG method has the following properties:

- By a proper approximation of the GCL and contravariant vectors free stream preservation is ensured.
- The semi-discrete method satisfies a discrete total entropy inequality, if the split from DG framework is used with EC volume and ES surface fluxes.
  Note: The discrete total entropy inequality is not enough to avoid the Gibbs phenomenon.
- The expected accuracy $N+1$ for tensor product polynomials with maximum degree N can be reached, if the surface flux is uses with a proper matrix dissipation operator.

Merci!