Development of Spectral Element Methods for Compressible Flow Problems

David A. Kopriva

Department of Mathematics
The Florida State University

Spectral Multidomain Methods

1980's

D.A. Kopriva / A spectral multidomain method

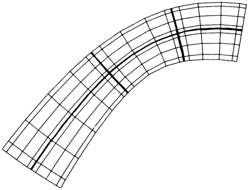
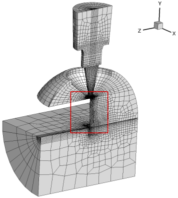
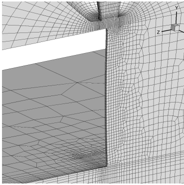


Fig. 10. Multidomain grid with six subdomains for the Ringleb problem.

- Limited
- Inflexible
- Complicated

Today



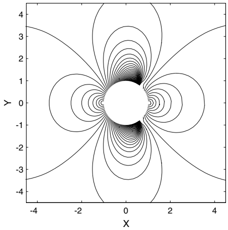


- Powerful,
- Robust
- Flexible

1983: Flow Over a Cylinder

- Problem (Hussaini):

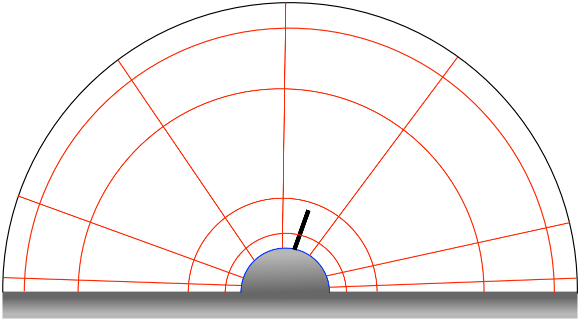
  Find, precisely, the transonic Mach number for a cylinder



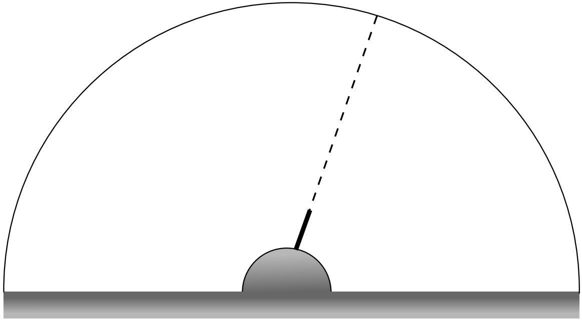
- Approach:
  - Chebyshev spectral method
  - Euler Gas-Dynamics equations

(Contour Plot: Hafez & Wahba, 2004)

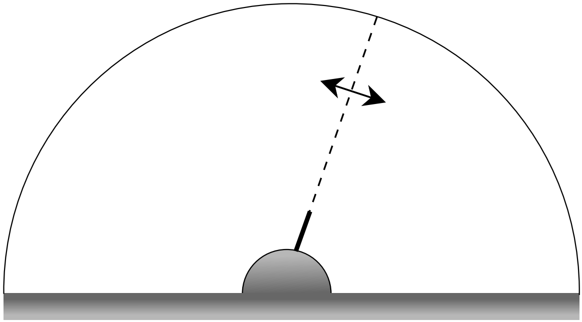
Chebyshev Spectral Collocation



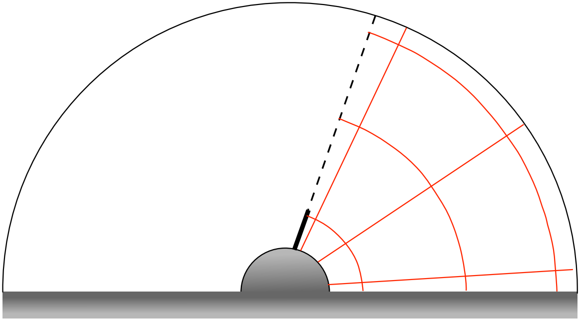
Origin of Spectral Multidomain



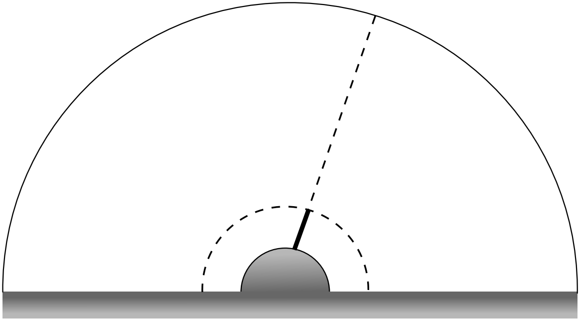
Origin of Spectral Multidomain



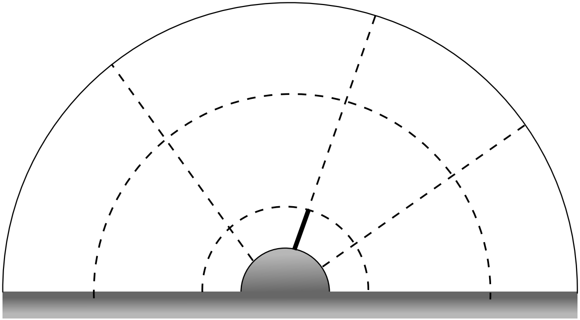
Origin of Spectral Multidomain



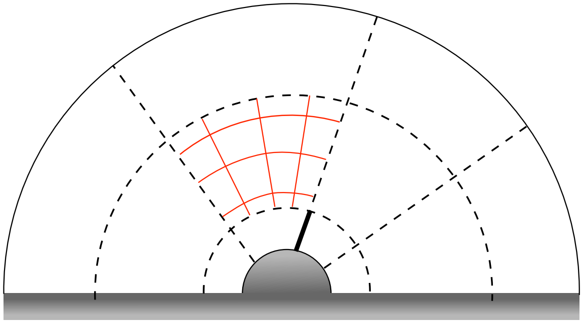
Origin of Spectral Multidomain



Origin of Spectral Multidomain



Origin of Spectral Multidomain



Timeline

- 1980's : Baby Steps
  - Strong form Chebyshev collocation
- 1990's : Search for the Ultimate Scheme™
  - Cell Average FCT (Karniadakis)
  - Penalty method (Hesthaven, Gottlieb, Funaro)
  - Staggered Grid (Kopriva)
- 1999+ : Rise of DG
- 2010s+: Large scale applications

Strong Form Chebyshev

Features:

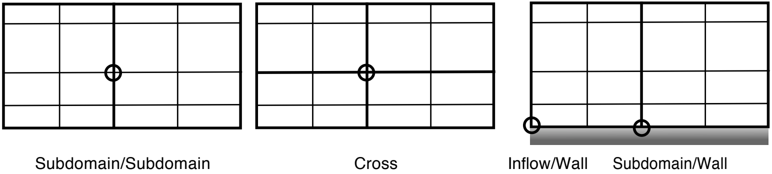
- Standard Chebyshev Collocation in interiors
- (Characteristic) Patching conditions at interfaces

Pros:

- Lower cost per DOF than single domain
- Spectral accuracy

Cons:

- Mesh required continuous metrics
- Complicated to implement
- Not robust



90's: Weak Imposition of BCs

Weak imposition gives full unstructured mesh flexibility.

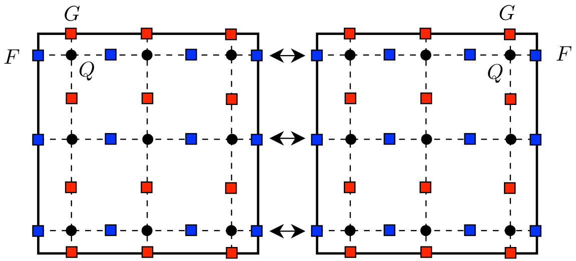
- Spectral Penalty method (Hesthaven)
  - (+) Natural imposition of conditions for advection-diffusion operators
  - (+) Stability proof for linearized compressible Navier-Stokes
  - (−) Penalty parameter
  - (−) Stiff
  - (−) Not Conservative
  - (−) Ad-Hoc treatment at corners and when advection speed vanishes
- Staggered Grid Method (Kopriva)
  - (+) Conservative
  - (+) Easy to implement
  - (+) No special corner point operations
  - (+) Robust
  - (−) Weak instability for periodic advection problems

Staggered Grid Approximation

- Solution and fluxes in different polynomial spaces

$$Q \in P^N \times P^N$$
$$F \in P^{N+1} \times P^N$$
$$G \in P^N \times P^{N+1}$$

- Only fluxes on boundaries
- Uses Riemann solvers on discontinuous solutions



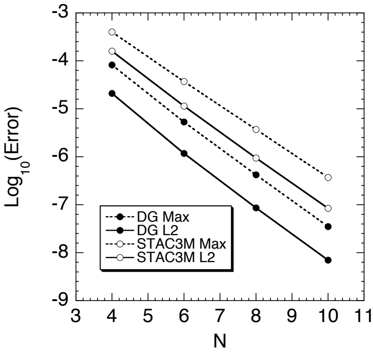
00's : DG Spectral Element Method (DGSEM)

DGSEM:

- Conservative
- Easy BCs
- Variational Formulation
- Broad Framework

Why DG over staggered grid?

- Faster: 20% faster (Simpler Interpolations)
- More Accurate: 10× on test problem



DGSEM Framework: Conservation Laws

Problems modeled by a system of conservation laws:

$$\vec{q}_t + \nabla \cdot \vec{f} = 0$$

$$\vec{f} = \vec{f^i} + \vec{f^v}$$

Examples:

- Euler Equations

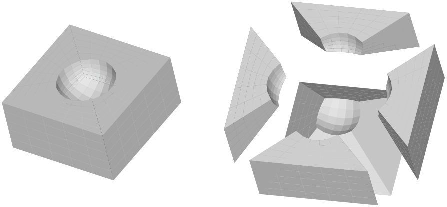
$$\vec{q} = \begin{bmatrix} \rho \\ \rho\vec{u} \\ \rho E \end{bmatrix}, \quad \vec{f^i} = \begin{bmatrix} \rho\vec{u} \\ \rho\vec{u} \otimes \vec{u} + pI \\ \rho u H \end{bmatrix}, \quad \vec{f^v} = 0$$

- Navier-Stokes Equations

$$\vec{f^v} = \begin{bmatrix} 0 \\ -\tau \\ \tau \cdot \vec{u} + k\nabla T \end{bmatrix}$$

Multi-Element Decomposition

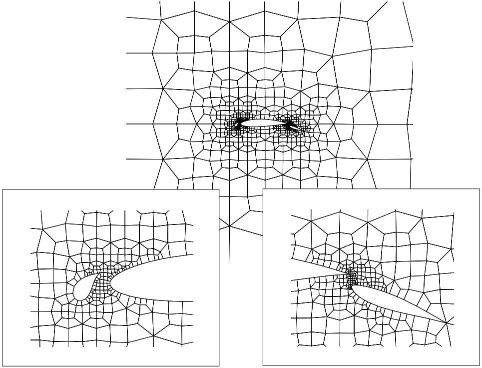
Subdivide domain into multiple elements



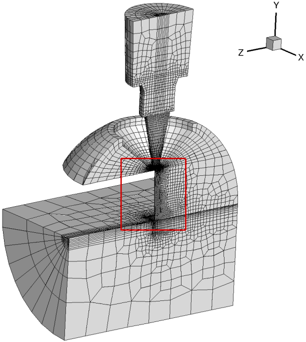
Multi-Element Decomposition

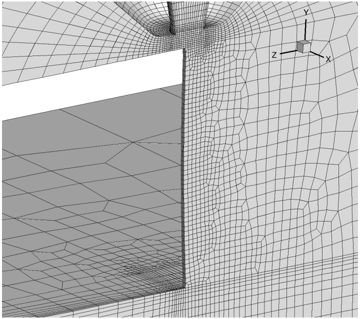
Decomposition:

- Arbitrarily complex
- Conforming or nonconforming
- Moving or stationary



Multi-Element Decomposition: 3D



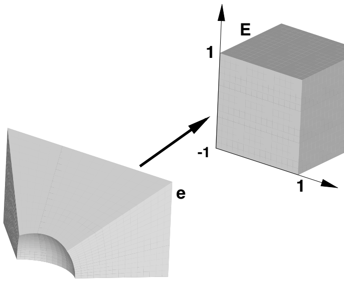


(Courtesy of G. Gassner)

Mapping to Reference Element

Transform:

$$\mathbf{x} = \mathbf{X}\left(\vec{\xi}, \tau\right)$$



Equations on Reference Element

Strong form of conservation law:

$$\tilde{q}_t + \nabla \cdot \tilde{f} = 0$$

where

$$\begin{aligned} \tilde{q} &= J\mathbf{q} \\ \tilde{f}^i &= J\mathbf{a}^i \cdot (\mathbf{f} - \mathbf{q}\mathbf{x}_\tau) \end{aligned}$$

Jacobian satisfies Geometric Conservation Law:

$$J_\tau + \nabla_\xi \cdot \tilde{\Psi}(J) = 0,$$

The DG Spectral Element Framework

Three characteristics:

1. Approximate

$$\tilde{q} \approx \tilde{Q} \in \mathbb{P}^N, \quad \tilde{f} \approx \tilde{F} \in \mathbb{P}^M on\ E$$

2. Weak form

$$\int_E \left(\tilde{Q}_t + \nabla \cdot \tilde{F} \right) \phi = 0$$

3. No continuity on $\phi \in \mathbb{P}^N$ between elements

DG Formulation

Integrate by parts

$$\int_E \tilde{Q}_t \phi d\xi + \int_{\partial E} \tilde{F} \cdot \hat{n}_\xi \phi dS - \int_E \tilde{F} \cdot \nabla \phi d\xi = 0$$

Replace boundary fluxes with Riemann solver

$$\int_E \tilde{Q}_t \phi d\xi + \int_{\partial E} \tilde{F}^* \cdot \hat{n}_\xi \phi dS - \int_E \tilde{F} \cdot \nabla \phi d\xi = 0 \quad Form\ I$$

Maybe integrate by parts again

$$\int_E \tilde{Q}_t \phi d\xi + \int_{\partial E} \left(\tilde{F} - \tilde{F}^* \cdot \hat{n}_\xi \right) \phi dS - \int_E \nabla \cdot \tilde{F} \phi d\xi = 0 \quad Form\ II$$

Choices, Choices, Choices

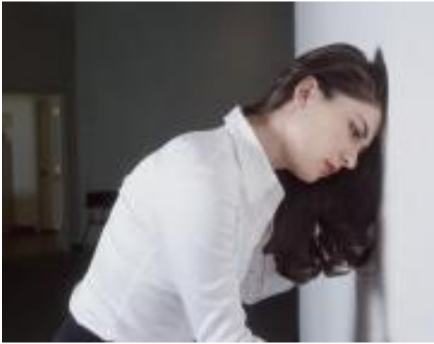
We actually have a *framework* from which to derive methods:

1. Quad/Hex or Tri/Tet elements?
2. Nodal or modal basis?
3. What polynomials?
4. Approximate boundaries with different orders?
5. Approximate solution and fluxes with different orders?
6. Exact integrals or quadrature?
7. Inexact or exact quadrature?
8. Form I or Form II?
9. ???

Too many choices can be overwhelming.

DG Spectral Element Approximation

It's Not That Hard!



Easy to Implement and Effective Approximation

"Classical" spectral element approximation:

1. Quadrilateral/ Hexahedral elements
   $\Rightarrow$ Efficient tensor product bases
2. Nodal basis
   $\Rightarrow$ Easy for nonlinear/variable coefficient/general complex geometry problems
3. All approximations at same polynomial order
   $\Rightarrow$ Simplifies coding
4. Legendre basis
   $\Rightarrow$ Spectral accuracy, conditioning
5. Gauss-Type quadrature

Implementation

Solution and fluxes by polynomials in (Lagrange) nodal form

$$\mathbf{Q} = \sum_{n=0}^{N}\sum_{m=0}^{N} \mathbf{Q}_{n,m}\ell_n(\xi)\ell_m(\eta)$$

$$\mathbf{F} = \sum_{n=0}^{N}\sum_{m=0}^{N} \left(\mathbf{F}_{n,m}\hat{x} + \mathbf{G}_{n,m}\right)\ell_n(\xi)\ell_m(\eta).$$

Integrate by parts 1x

$$\int_E \frac{\partial \mathbf{Q}}{\partial t}\phi_{i,j}d\xi + \int_{\partial E} \mathbf{F}^* \cdot \hat{n}\phi_{i,j}dS - \int_E \mathbf{F}\cdot\nabla\phi_{i,j}d\xi = 0$$

With $\phi_{i,j} = \ell_i(\xi)\ell_j(\eta)$.

Apply Quadrature to Each Integral

Time derivative integral

$$
\begin{aligned}
&\int_{-1,N}^{1} \frac{d\mathbf{Q}(\xi,\eta)}{dt}\ell_i(\xi)\ell_j(\eta)d\xi d\eta \\
&= \sum_{k=0}^{N}\sum_{l=0}^{N} \frac{d\mathbf{Q}(\xi_k,\eta_l)}{dt}\ell_i(\xi_k)\ell_j(\eta_l)w_k^{(\xi)}w_l^{(\eta)} \\
&= \frac{d\mathbf{Q}_{i,j}}{dt}w_i^{(\xi)}w_j^{(\eta)},
\end{aligned}
$$

etc.

Spatial Discretization

On each element we integrate

$$\frac{d\mathbf{Q}_{i,j}}{dt} + \left\{ \left[\tilde{\mathbf{F}}^*(1,\eta_j)\frac{\ell_i(1)}{w_i^{(\xi)}} - \tilde{\mathbf{F}}^*(-1,\eta_j)\frac{\ell_i(-1)}{w_i^{(\xi)}} \right] + \sum_{k=0}^{N} \tilde{\mathbf{F}}_{k,j}\hat{D}_{ik}^{(\xi)} \right\} + \left\{ \left[\tilde{\mathbf{G}}^*(\xi_i,1)\frac{\ell_j(1)}{w_j^{(\eta)}} - \tilde{\mathbf{G}}^*(\xi_i,-1)\frac{\ell_j(-1)}{w_j^{(\eta)}} \right] + \sum_{k=0}^{N} \tilde{\mathbf{G}}_{i,k}\hat{D}_{jk}^{(\eta)} \right\} = 0$$

Primary Work:

- Computation of fluxes $\tilde{\mathbf{F}}_{k,j}$ and $\tilde{\mathbf{G}}_{i,k}$ from solution
- Computation of Riemann solver $\tilde{\mathbf{F}}^*(\pm 1,\eta_j)$ and $\tilde{\mathbf{G}}^*(\xi_i,\pm 1)$
- Series of dot products (Gauss)
- Series of Matrix-Vector products

DGSEM Time Derivative Algorithm

Gauss-Lobatto Version:

for $j = 0$ **to** M **do**

$$\mathbf{F} = xFlux(\mathbf{Q}_j)$$
$$\mathbf{F}' = MatrixTimesVector(\hat{D}, \mathbf{F})$$
$$\dot{\mathbf{Q}}_j = -\mathbf{F}'$$
$$\dot{Q}_{0,j} = \dot{Q}_{0,j} - b_j^L * RiemannSolver(Q_j^{ext}, Q_{0,j}, \hat{n}_j^L)$$
$$\dot{Q}_{N,j} = \dot{Q}_{N,j} - b_j^R * RiemannSolver(Q_{N,j}, Q_j^{ext}, \hat{n}_j^R)$$

end

for $i = 0$ **to** N **do**

$$\mathbf{G} = yFlux(\mathbf{Q}_i)$$
$$\mathbf{G}' = MatrixTimesVector(\hat{D}, \mathbf{G})$$
$$\dot{\mathbf{Q}}_i = \dot{\mathbf{Q}}_i - \mathbf{G}'$$
$$\dot{Q}_{i,0} = \dot{Q}_{i,0} - b_i^B * RiemannSolver(Q_i^{ext}, Q_{i,0}, \hat{n}_i^B)$$
$$\dot{Q}_{i,M} = \dot{Q}_{i,M} - b_i^T * RiemannSolver(Q_{i,M}, Q_i^{ext}, \hat{n}_i^T)$$

end

See... It's Not That Bad!

What We Know

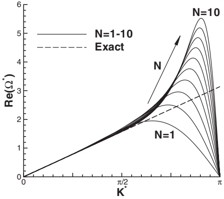
- Form I and Form II are algebraically identical
- Gauss has better Phase/Dissipation properties
- Gauss-Lobatto can take larger time steps
- Gauss is more robust
- Gauss is slightly more efficient than Gauss-Lobatto
- Mesh can be moved Free-Stream Preserving with spectral and full time accuracy
- Suitable for massive parallelization
- Can be used for industrial strength™ applications

Integrate By Parts 1X or 2X?

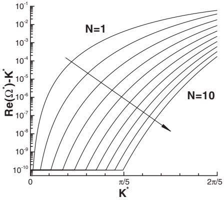
Theorem

(Kopriva and Gassner, 2010) For quadrilateral/hexahedral tensor product discontinuous Galerkin approximations to systems of hyperbolic conservation laws with either Gauss or Gauss-Lobatto quadratures the two forms are **algebraically equivalent** *as long as one uses global polynomial representations for the flux and solutions.*

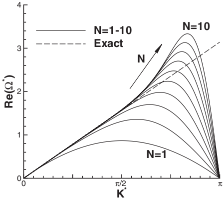
Gauss Has Better Dispersion Error



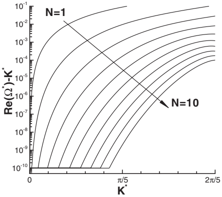
(a) Dispersion relation



(b) Logarithm of dispersion error

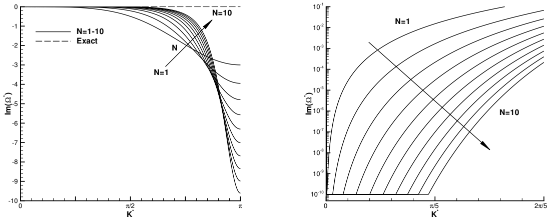


(a) Dispersion relation



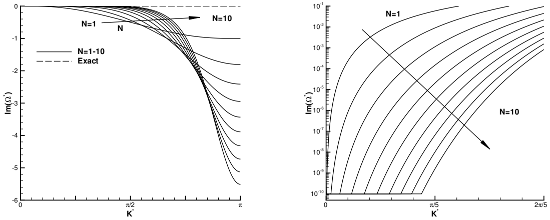
(b) Logarithm of dispersion error

Gauss Has Better Dissipation Error



(a) Dissipation relation (b) Logarithm of dissipation error

FIG. 6.1. *Imaginary part of the physical mode for the Gauss DGSEM scheme with* $N = 1$ *up to* $N = 10$*. In the logarithmic plot, the error is cut off at* 10^{-10} *to avoid numerical noise.*



(a) Dissipation relation (b) Logarithm of dissipation error

Gauss is Slightly More Efficient Overall

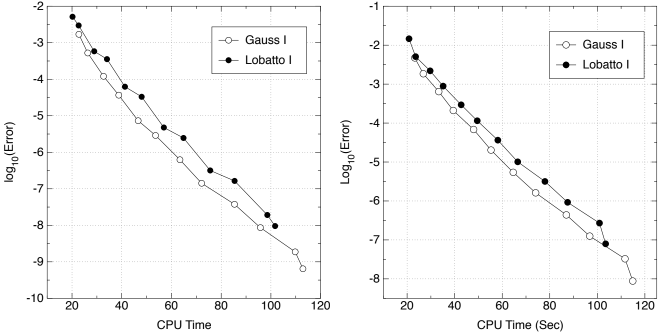


Figure: Maximum error as a function of work for the Gauss and Lobatto approximations. Left: Uniform mesh. Right: Non-Uniform Mesh

Free-Stream Preservation and the Geometric Conservation Law

Theorem

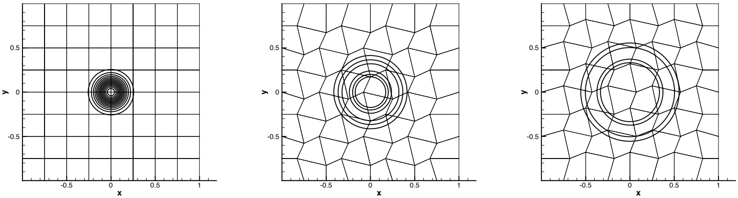
(Acosta & Kopriva, 2012) Suppose that at time τ^n, $\mathbf{Q}^n_{i,j} = \mathbf{c}$, where $\vec{c}$ is a constant vector. Define $\mathbf{Q}_{i,j} \equiv \tilde{Q}_{i,j}/\tilde{J}_{i,j}$, where $\tilde{J}_{i,j}$ is the solution of the GCL. Then

$$\mathbf{Q}^{n+1}_{ij} = \mathbf{c}.$$

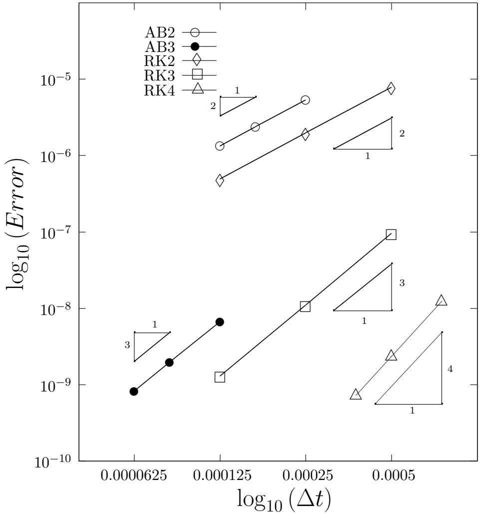
Spectral + High order time accuracy when moving mesh by:

- *Method 1*: Exact differentiation of the mapping.
- *Method 2*: Integration of an acceleration equation.
- *Method 3*: Numerical differentiation of the mesh position via the time integrator (Inverse operator).

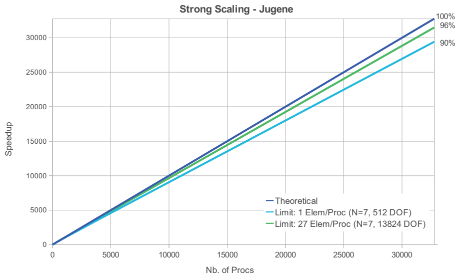
Example: Time Accurate Moving Mesh

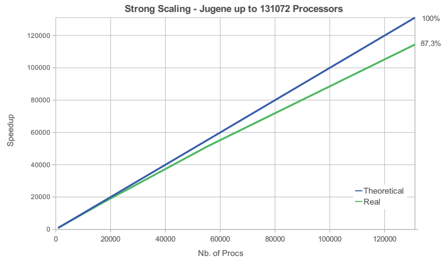


Example: Time Accuracy on Moving Mesh

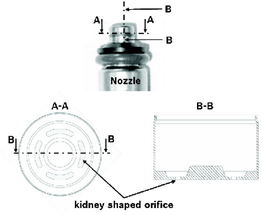


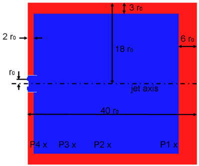
Massive Parallelization (G. Gassner)

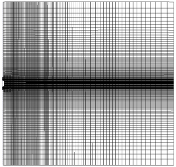


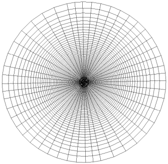


Industrial Strength Applications: Natural Gas Injector Acoustics



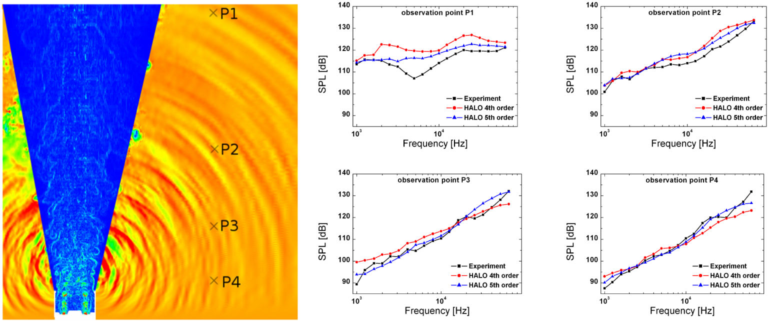






(Courtesy of G. Gassner)

Industrial Strength Applications: Natural Gas Injector Acoustics



(Courtesy of G. Gassner)

The Future: What We Still Want to Know

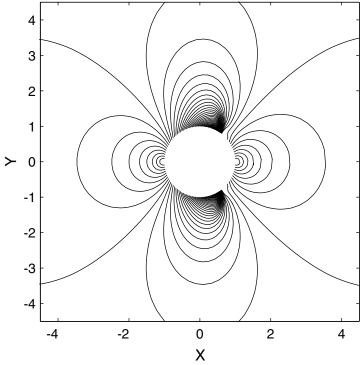
- How mesh affects accuracy and time step
- How to couple (moving) material interfaces
- How to move meshes efficiently
- How to solve time accurate problems efficiently
  - Implicit Schemes
  - Preconditioning
  - Local Time Stepping
- How to guarantee stability - Aliasing removal
- How to compute shocks
- Adaptation

AND ...

1983 + 30: Flow Over a Cylinder

- Problem (Hussaini):

  Find, precisely, the Mach number where flow over cylinder goes transonic.



(Contour Plot: Hafez & Wahba, 2004)

1983 + 30: Flow Over a Cylinder

STILL NOT DONE YET!