Finite Element method for fluid flow problems

To this point we have focused on elliptic PDE (only even order derivatives) that produce nice symmetric global systems. Standard Galerkin methods are can be shown to be “optimal” for these problems. On the other hand standard Galerkin methods do not work well for advection dominated problems where there are first derivative terms that are important. One simple advection equation is

$$\phi_{,t} + a_i \phi_{,i} = 0 \text{ in } \Omega$$

with appropriate boundary and initial conditions.
The most basic problem of general interest (that is it has both advection and diffusion) in this class is the “static” advection/diffusion equation

$$a_i \phi_{,i} + \kappa \phi_{,ii} - f \text{ in } \Omega \text{ subject to } \phi = g \text{ on } \Gamma$$

Most problems of interest also have time dependent terms. Since the time domain is almost always handled through semi-discretization (e.g., use finite elements for the spatial discretization and finite difference for the temporal discretization).

If you apply a standard Galerkin finite element method to these equations you will find the solutions will have large oscillations and at large over shoots and undershoots at discontinuities (which can happen in these classes of equations).

Thus a large number of Petrov-Galerkin methods have been developed to address this class of problem. The currently two most popular classes of methods are the:

- Discontinuous Galerkin (DG). One Reference (there are a large number of them): Cockburn, Bernardo, George E. Karniadakis, and Chi-Wang Shu. "The development of discontinuous Galerkin methods." In *Discontinuous Galerkin Methods*, pp. 3-50. Springer, Berlin, Heidelberg, 2000.
- Stabilized finite elements – developed heavily by Tom Hughes (an people that studied with him). There is not a goo book type reference on this method. A few well cited papers are:
    - Franca, L.P., Frey, S.L. and Hughes, T.J., 1992. Stabilized finite element methods: I. Application to the advective-diffusive model. *Computer Methods in Applied Mechanics and Engineering*, *95*(2), pp.253-276.
    - Franca, L.P. and Frey, S.L., 1992. Stabilized finite element methods: II. The incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, *99*(2-3), pp.209-233.
    - Tezduyar, T.E., 1991. Stabilized finite element formulations for incompressible flow computations. In *Advances in applied mechanics* (Vol. 28, pp. 1-44). Elsevier.
    - Whiting, C.H. and Jansen, K.E., 2001. A stabilized finite element method for the incompressible Navier–Stokes equations using a hierarchical basis. *International Journal for Numerical Methods in Fluids*, *35*(1), pp.93-116.

How do we achieve this ?

Let us <u>consider a few well known schemes</u> and their basic properties to understand what is needed.

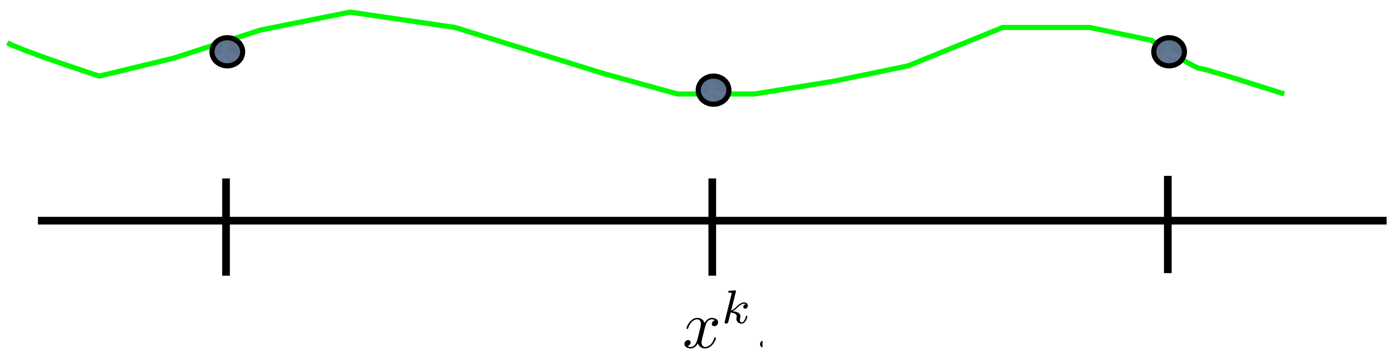
Consider the basic equation

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = g, \quad x \in \Omega,$$

All schemes involve two choices

- In which way does one approximate the solution ?
- In which way should the approximation satisfy the PDE ?

Finite difference methods



- The local approximation is a 1D polynomial
- The equation is satisfied in a pointwise manner

$$x \in [x^{k-1}, x^{k+1}] : \quad u_h(x,t) = \sum_{i=0}^{2} a_i(t)(x - x^k)^i, \quad f_h(x,t) = \sum_{i=0}^{2} b_i(t)(x - x^k)^i,$$

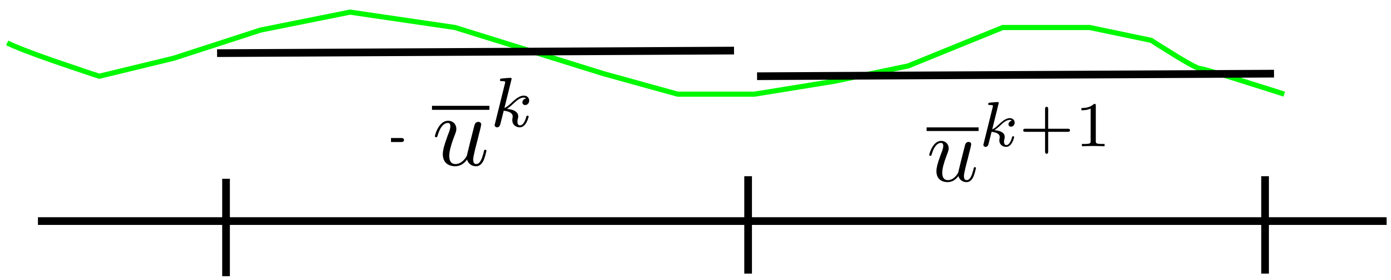
$$\frac{du_h(x^k,t)}{dt} + \frac{f_h(x^{k+1},t) - f_h(x^{k-1},t)}{h^k + h^{k-1}} = g(x^k,t),$$

Finite difference schemes

- Main benefits
  - Simple to implement and fast
  - High-order is feasible
  - Explicit in time
  - Direction can be exploited - upwind
  - Strong theory

- Main problem
  - Simple local approximation and geometric flexibility are not agreeable

Finite volume methods



- The local approximation is a cell average

$$\int_{x^{k-1/2}}^{x^{k+1/2}} u_h(x)\,dx = h^k \overline{u}^k,$$

- The equation is satisfied on conservation form

$$h^k \frac{d\overline{u}^k}{dt} + f^{k+1/2} - f^{k-1/2} = h^k \overline{g}^k,$$

Finite volume methods

The key challenge is one of reconstruction

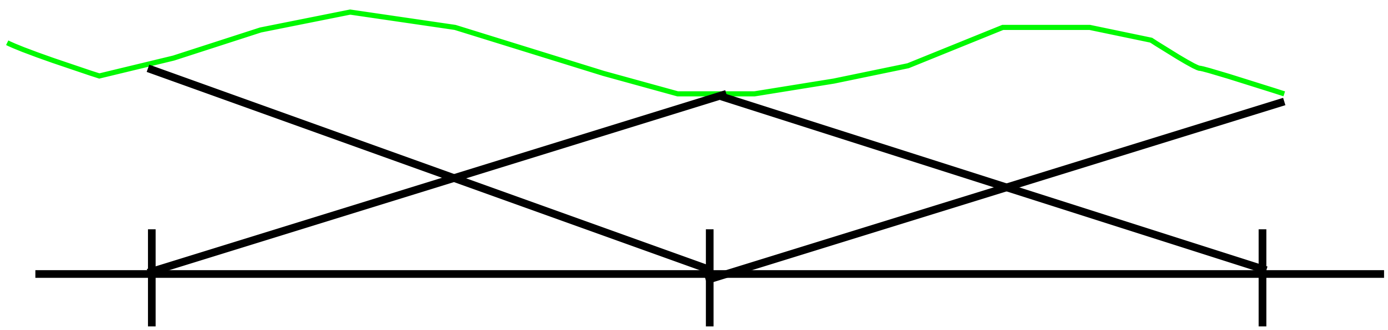
$$u^{k+1/2} = \frac{\overline{u}^{k+1} + \overline{u}^k}{2}, \quad f^{k+1/2} = f(u^{k+1/2}),$$

- Main benefit
  - Robust and fast due to locality
  - Complex geometries
  - Well suited for conservation laws
  - Explicit in time

- Main problem
  - Inability to achieve high-order on general grids due to extended stencils
  - Grid smoothness requirements

Finite element methods

We begin by splitting the solution into elements as



- The solution is defined in a nonlocal manner

$$u_h(x) = \sum_{k=1}^{K} u(x^k) N^k(x)$$

- The equation is satisfied globally

$$\int_{\Omega} \left(\frac{\partial u_h}{\partial t} + \frac{\partial f_h}{\partial x} - g_h \right) N^j(x) \, dx = 0,$$

Finite element methods

This yields the global equation

$$\mathcal{M}\frac{d\boldsymbol{u}_h}{dt} + \mathcal{S}\boldsymbol{f}_h = \mathcal{M}\boldsymbol{g}_h,$$

$$\mathcal{M}_{ij} = \int_\Omega N^i(x)N^j(x)\,dx, \quad \mathcal{S}_{ij} = \int_\Omega N^i(x)\frac{dN^j}{dx}\,dx,$$

- Main benefits
  - High-order accuracy and complex geometries can be combined

- Main problems
  - Implicit in time
  - Not well suited for problems with direction

Lets summarize the observations

	Complex geometries	High-order accuracy and hp-adaptivity	Explicit semi-discrete form	Conservation laws	Elliptic problems
FDM	×	✓	✓	✓	✓
FVM	✓	×	✓	✓	(✓)
FEM	✓	✓	×	(✓)	✓
DG-FEM	✓	✓	✓	✓	(✓)

What we need is a scheme that combines

- The local high-order/flexible element of FEM
- The local statement on the equation for FVM

These are exactly the components of the *Discontinuous Galerkin Finite Element Method*

Discontinuous Galerkin Finite Element Method: Survey and Recent Development

Taken form notes of Prof. Chi-Wang Shu

Division of Applied Mathematics

Brown University

Prof. F. Li in the math department covers these methods in her course

Introduction and history of the DG method

How does the method work — an example

I_{j-1} I_j I_{j+1}

$x_{j-1} = x_{(j-1)+1/2}$ $x_{j+1/2} = x_{(j+1)-1/2}$

To solve a hyperbolic conservation law:

$u_t = \frac{\partial u}{\partial t}$ $f(u)_x = \frac{\partial f(u)}{\partial x}$

$$u_t + f(u)_x = 0 \tag{1}$$

Multiplying with a test function v, integrate over a cell $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, and integrate by parts:

$$\int_{I_j} u_t v dx - \int_{I_j} f(u) v_x dx + f(u_{j+\frac{1}{2}}) v_{j+\frac{1}{2}} - f(u_{j-\frac{1}{2}}) v_{j-\frac{1}{2}} = 0$$

I_{j-1} I_j I_{j+1}

$u_{j-1/2}$ $u_{j+1/2}$

$u_{(j-1)+1/2}$ $u_{(j+1)-1/2}$

u, v are not continuous

$u_{(j-1)+1/2} \neq u_{j-1/2}$

Now assume both the solution u and the test function v come from a finite dimensional approximation space V_h, which is usually taken as the space of piecewise polynomials of degree up to k:

$$V_h = \left\{ v : v|_{I_j} \in P^k(I_j),\ j = 1, \cdots, N \right\}$$

However, the boundary terms $f(u_{j+\frac{1}{2}})$, $v_{j+\frac{1}{2}}$ etc. are not well defined when u and v are in this space, as they are discontinuous at the cell interfaces.

From the conservation and stability (upwinding) considerations, we take

- A single valued monotone numerical flux to replace $f(u_{j+\frac{1}{2}})$:

$$\hat{f}_{j+\frac{1}{2}} = \hat{f}(u^-_{j+\frac{1}{2}}, u^+_{j+\frac{1}{2}})$$

← a function of what is on the two sides structured considering

where $\hat{f}(u,u) = f(u)$ (consistency); $\hat{f}(\uparrow, \downarrow)$ (monotonicity) and $\hat{f}$ is Lipschitz continuous with respect to both arguments.

- Values from inside I_j for the test function v

$$v^-_{j+\frac{1}{2}}, \qquad v^+_{j-\frac{1}{2}}$$

"right" end of element   "left" end of element

Hence the DG scheme is: find $u \in V_h$ such that

$$\int_{I_j} u_t v dx - \int_{I_j} f(u) v_x dx + \hat{f}_{j+\frac{1}{2}} v^-_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}} v^+_{j-\frac{1}{2}} = 0 \qquad (2)$$

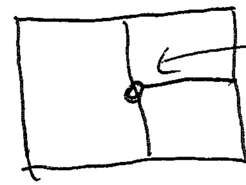
function of I_j only

function of I_j only

Function of I_j and I_{j+1}

Function of I_j and I_{j-1}

for all $v \in V_h$.



hanging node

Advantages of the DG method:

- Easy handling of complicated geometry and boundary conditions (common to all finite element methods). Allowing hanging nodes in the mesh;

- Compact. Communication only with immediate neighbors, regardless of the order of the scheme;

- Explicit. Because of the discontinuous basis, the mass matrix is local to the cell, resulting in explicit time stepping (no systems to solve);

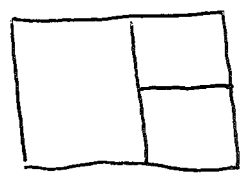
- Parallel efficiency. Achieves 99% parallel efficiency for static mesh and over 80% parallel efficiency for dynamic load balancing with adaptive meshes (Flaherty et al.) [this was like only 128 cores ~~with~~ an explicit method] – explicit easy to scale.

People have good parallel adaptive methods for the hanging node types of refinement – 10^7 cores

- Provable cell entropy inequality and L^2 stability, for arbitrary scalar equations in any spatial dimension and any triangulation, for any order of accuracy, without limiters;

- At least $(k+\frac{1}{2})$-th order accurate, and often $(k+1)$-th order accurate for smooth solutions when piecewise polynomials of degree k are used, regardless of the structure of the meshes.

- "Easy" h-p adaptivity.
  ↑ requires care



each element can have a different p

- Stable and convergent DG methods are now available for many nonlinear PDEs containing higher derivatives: convection diffusion equations, KdV equations, ...

→ The forget to mention the very large number of unknowns one gets — values on both sides

Three examples

We show three examples to demonstrate the excellent performance of the DG method.

The first example is the linear convection equation

$$u_t + u_x = 0, \qquad or \qquad u_t + u_x + u_y = 0,$$

on the domain $(0, 2\pi) \times (0, T)$ or $(0, 2\pi)^2 \times (0, T)$ with the characteristic function of the interval $(\frac{\pi}{2}, \frac{3\pi}{2})$ or the square $(\frac{\pi}{2}, \frac{3\pi}{2})^2$ as initial condition and periodic boundary conditions.

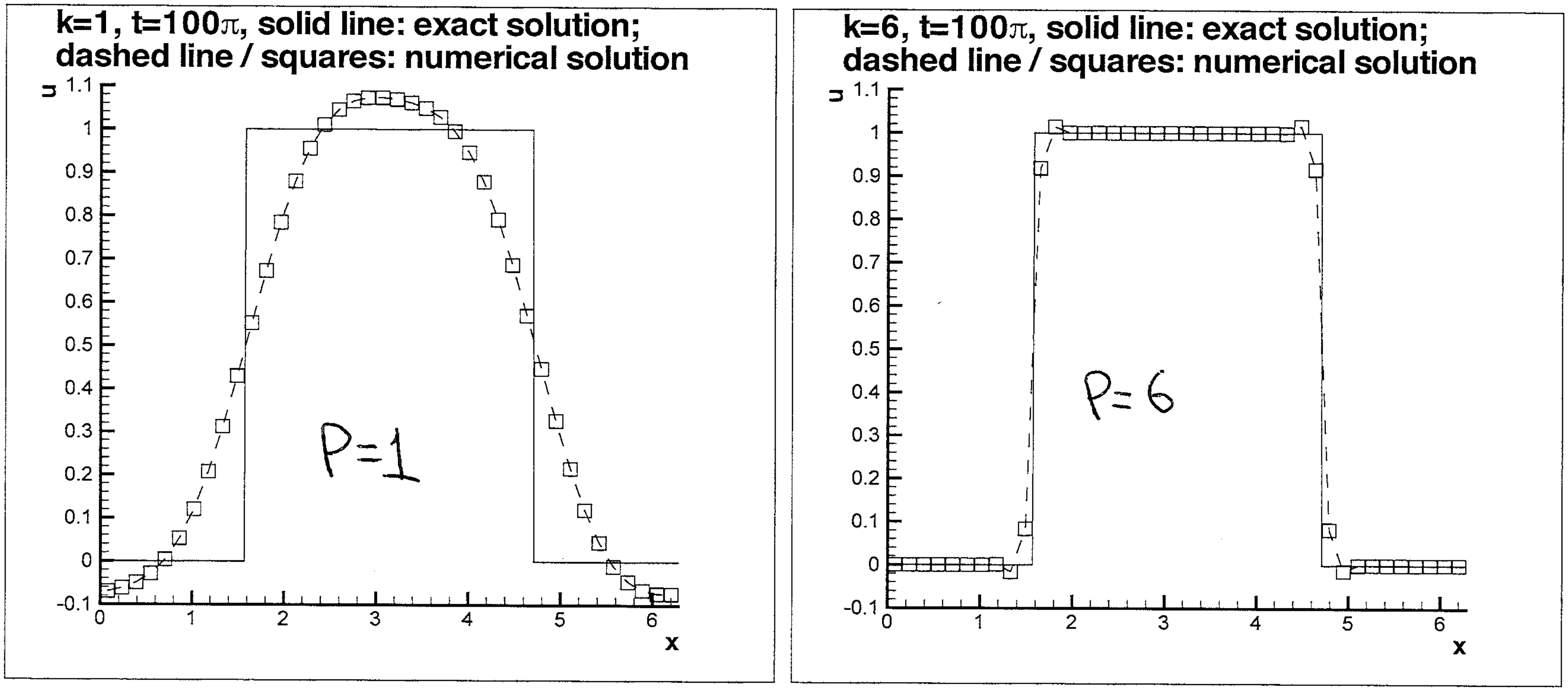


Figure 1: Transport equation: Comparison of the exact and the RKDG solutions at $T = 100\pi$ with second order (P^1, left) and seventh order (P^6, right) RKDG methods. One dimensional results with 40 cells, exact solution (solid line) and numerical solution (dashed line and symbols, one point per cell)

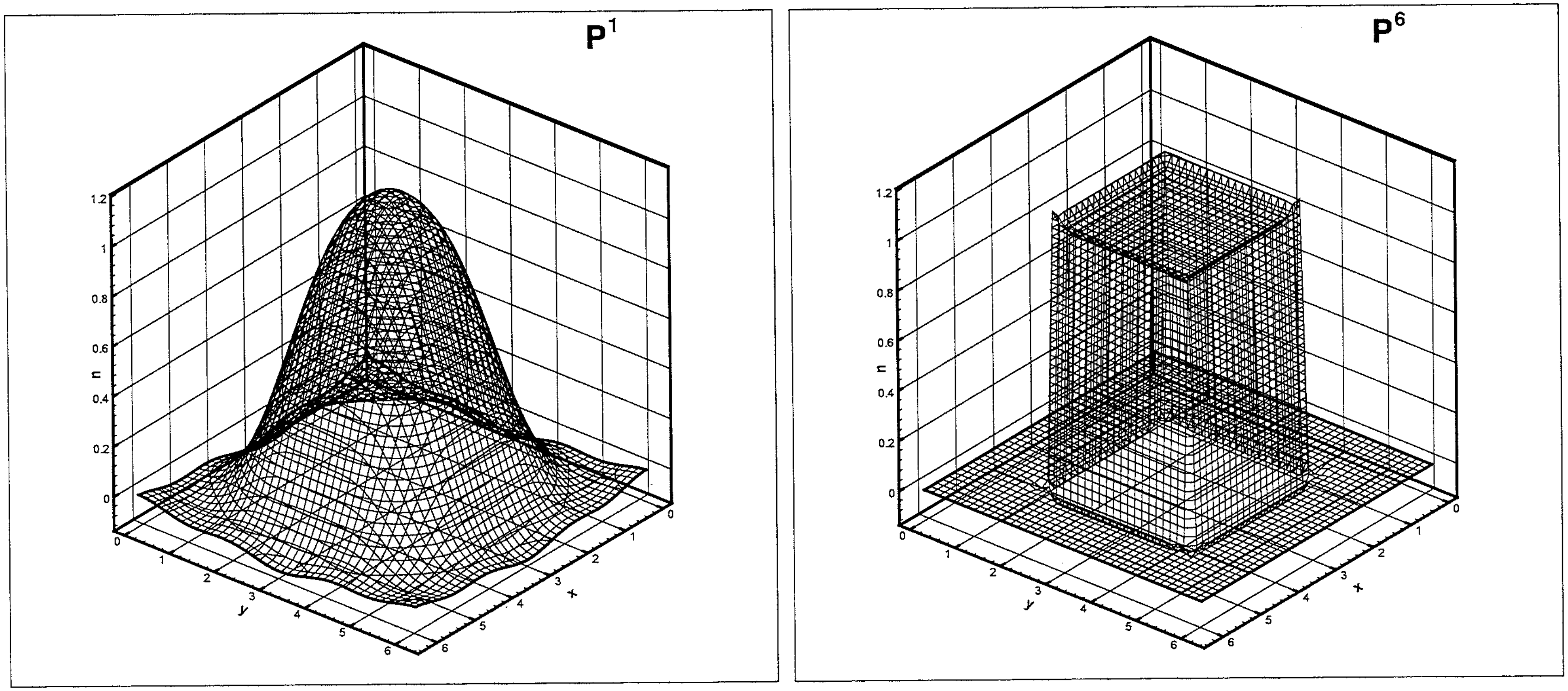


Figure 2: Transport equation: Comparison of the exact and the RKDG solutions at $T = 100\pi$ with second order (P^1, left) and seventh order (P^6, right) RKDG methods. Two dimensional results with 40×40 cells.

The second example is the double Mach reflection problem for the two dimensional compressible Euler equations.

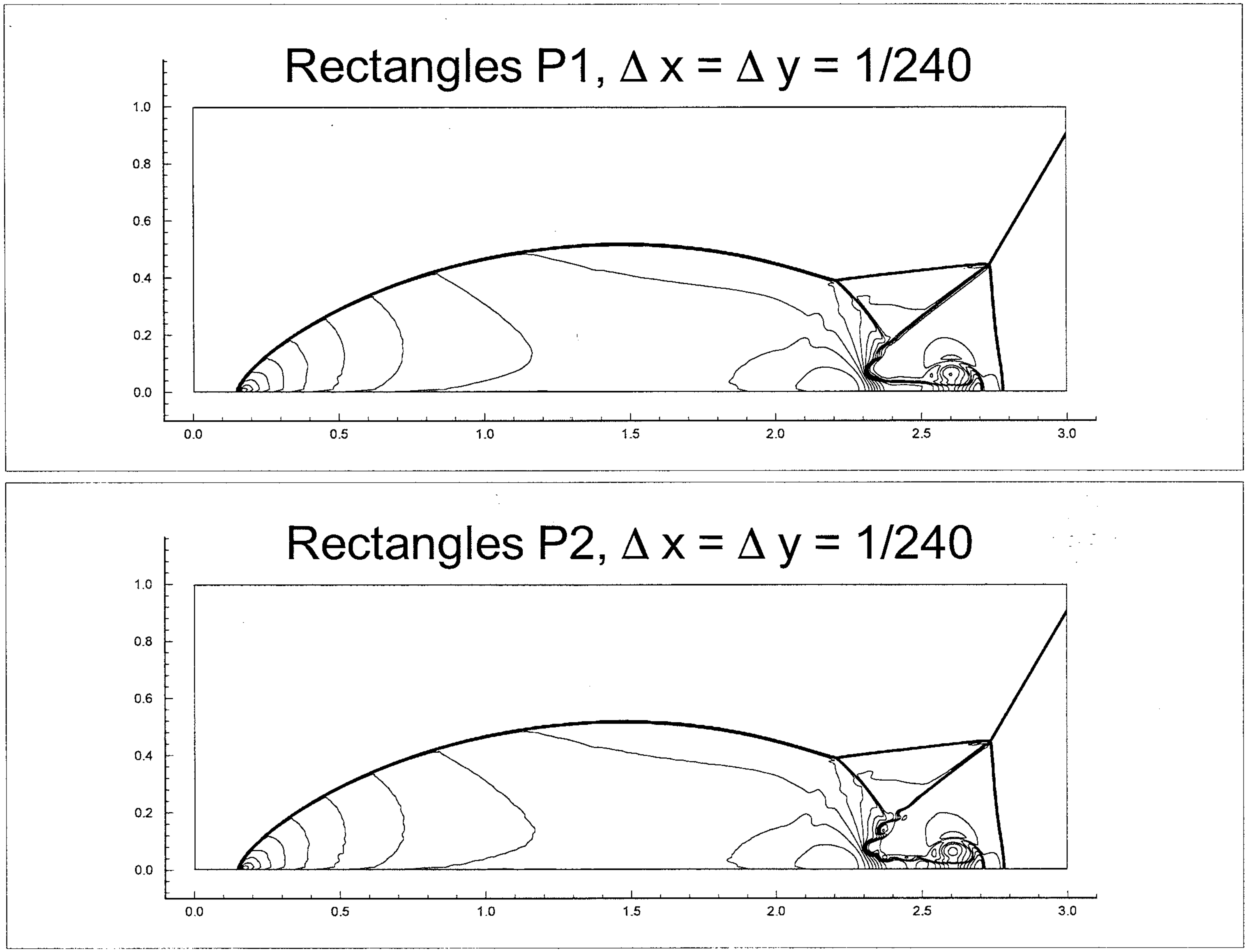
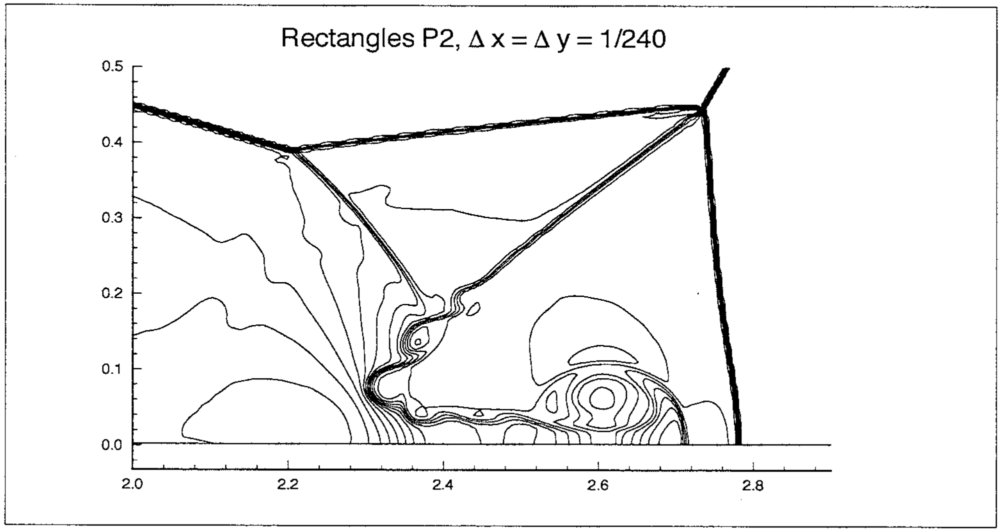


Figure 3: Double Mach reflection. $\Delta x = \Delta y = \frac{1}{240}$. Top: P^1; bottom: P^2



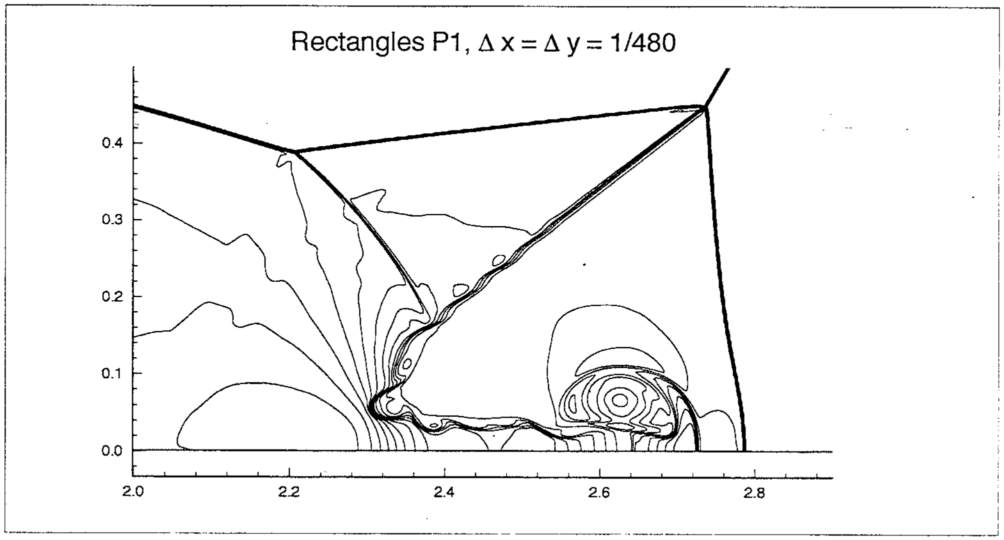


Figure 4: Double Mach reflection. Zoomed-in region. Top: P^2 with $\Delta x = \Delta y = \frac{1}{240}$; bottom: P^1 with $\Delta x = \Delta y = \frac{1}{480}$.

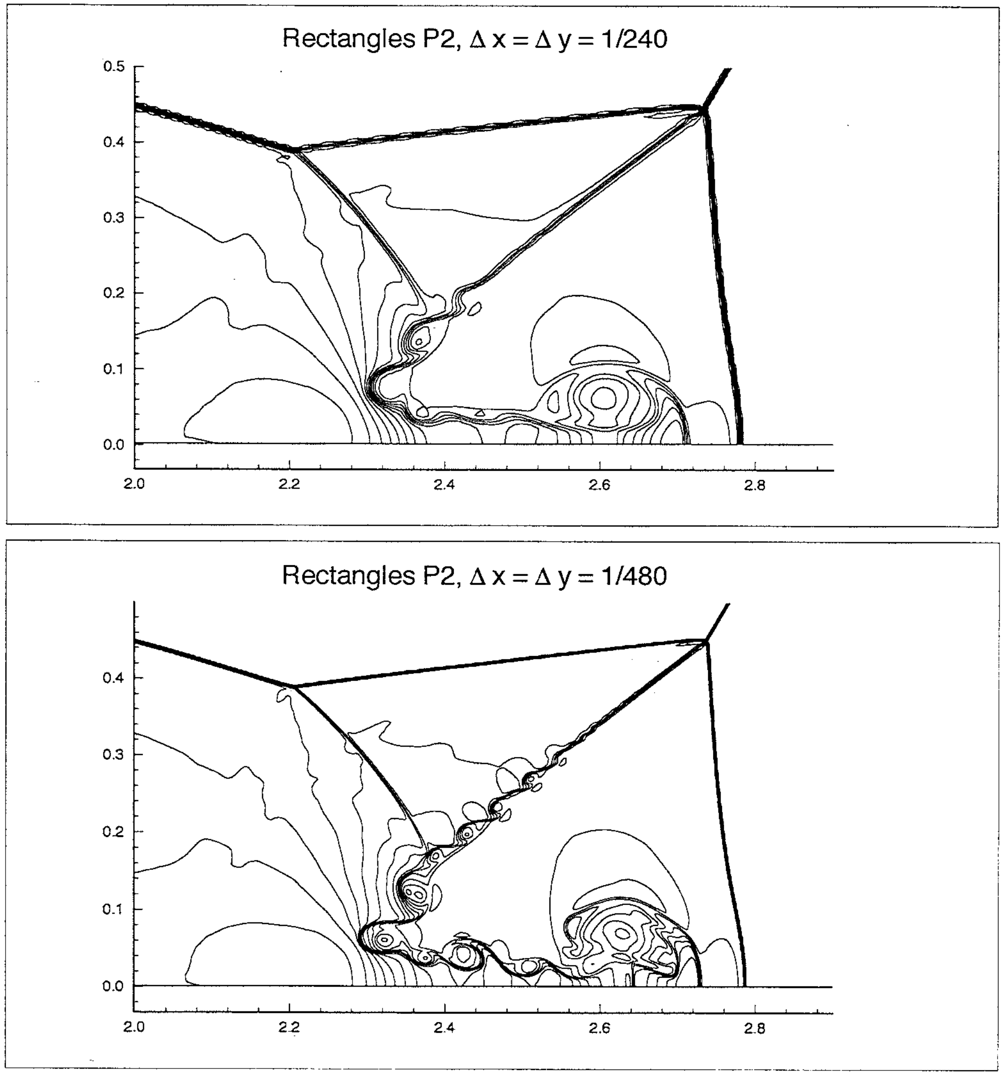
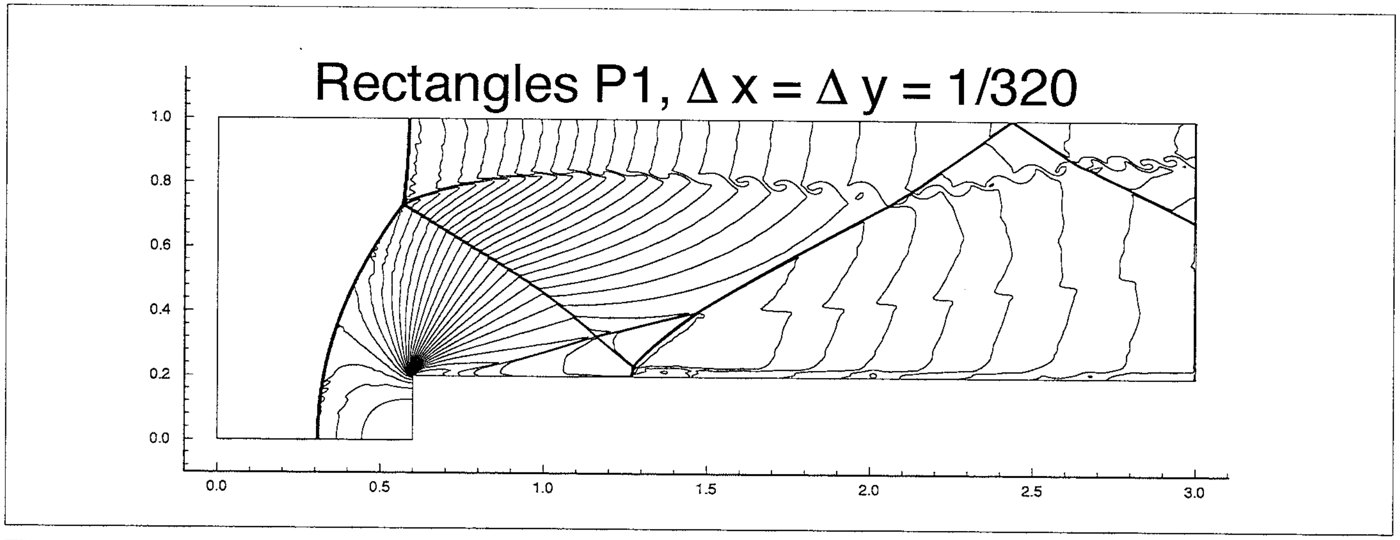


Figure 5: Double Mach reflection. Zoomed-in region. P^2 elements. Top: $\Delta x = \Delta y = \frac{1}{240}$; bottom: $\Delta x = \Delta y = \frac{1}{480}$.

The third example is the flow past a forward-facing step problem for the two dimensional compressible Euler equations. No special treatment is performed near the corner singularity.



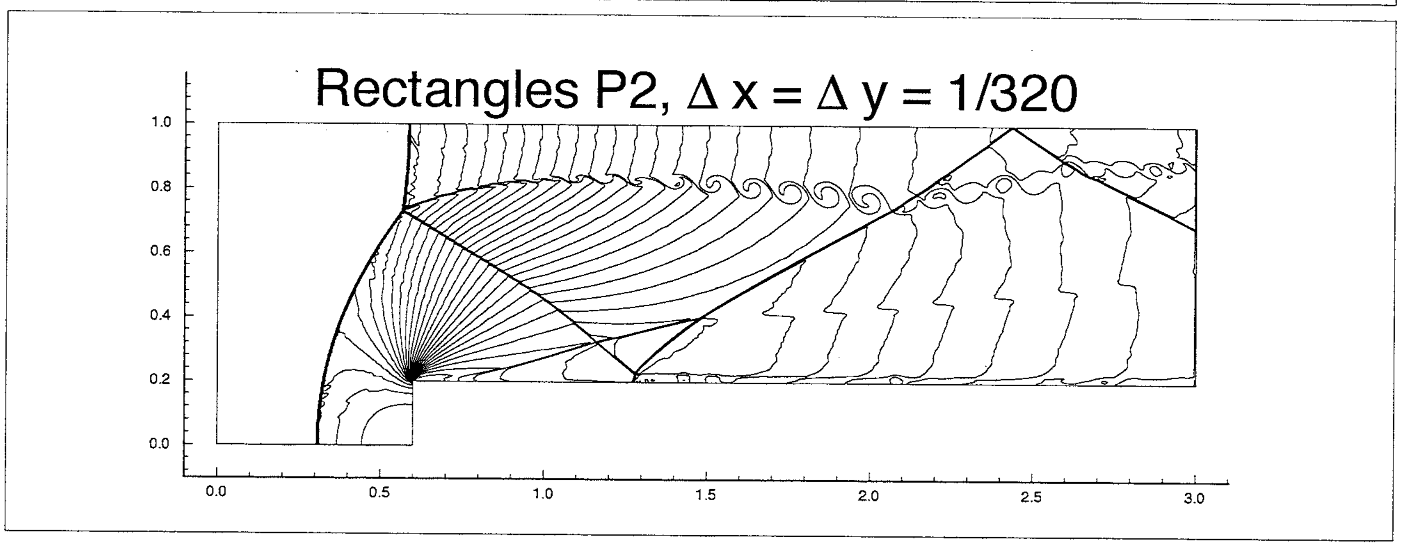


Figure 6: Forward facing step. Zoomed-in region. $\Delta x = \Delta y = \frac{1}{320}$. Left: P^1 elements; right: P^2 elements.

Stabilized Finite Elements

Consider the simple advection-diffusion equation

$$a_i \phi_{,i} + K \phi_{,ii} = f \quad \text{in } \Omega$$
$$\phi = g \quad \text{on } \Gamma \quad \text{(to keep it simple)}$$

Standard Galerkin

Given a_i, K, f find $\phi \in \mathcal{S}$ such that

$$\int_\Omega w\, a_i \phi_{,i}\, d\Omega - \int_\Omega w_{,i} K \phi_{,i}\, d\Omega = \int_\Omega w f\, d\Omega \quad \forall w \in \mathcal{V}$$

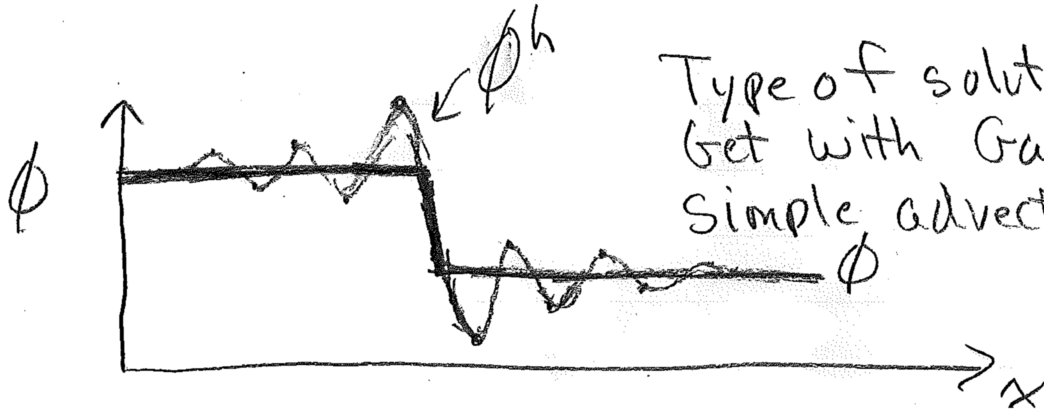
integrate the second term by parts once (boundary term $= 0$ since $w = 0$ on Γ for this fully essential BC case)

$\mathcal{S} = \{\phi \mid \phi \in H^1, \phi = g \text{ on } \Gamma\}$
$\mathcal{V} = \{w \mid w \in H^1, w = 0 \text{ on } \Gamma\}$

$$(w, a_i \phi_{,i}) - (w_{,i} K \phi_{,i}) = (w, f)$$

With this we would select C^0 shape functions for constructing $\mathcal{S}^h \subset \mathcal{S}, \mathcal{V}^h \subset \mathcal{V}$

We would find that this leads to poor solutions—



Type of solution you get with Galerkin for simple advective problem

Note if we look at putting our F.E. solution back to the original equation we have

$$a_i \phi^h_{,i} - K \phi^h_{,ii} - f = R \Leftarrow \text{a non-zero residual}$$

Note that if ϕ^h is exact (ie., $\phi^h = \phi$) the residual is zero ($R=0$).

Borrowing from things commonly done for problems where the discretized solution has overshoots and undershoots - we are looking to add an "artificial diffusion" term to our weak form. Note that if that artificial diffusion term is constructed as a function of the residual, R, it is consistent in that that term goes to zero as we approach the exact solution. (Many schemes use artificial diffusion terms that do not go to zero this way.)

One somewhat obvious way to proceed would be to add an integral least squares of the residual term to the weak form.

There would be some challenges in this when dealing with going from integrals over the domain to integrals over elements- The boundary terms would have to be included when using C^0 functions since we have $\phi_{,ii}$ in the residual

In addition it is the avective term that is the problematic term - thus we can "weight" by the avective part - not the whole residual.

An option for this that has been developed, and proven to be appropriate is to add the following to the Galerkin weak form

$$\sum_{e=1}^{n_{el}} \int_{\Omega^e} \underbrace{a_i w_{,i}}_{\text{advective part}} \tau \underbrace{(a_i \phi_{,i} - k\phi_{,ii} - f)}_{\text{residual}} d\Omega^e$$

Note- the τ is an important term- theory gives guidelines of its selection

Adding this term yields the following weak form

Given a_i, K, f, g, τ find $\phi \in S$ such that

$$(w, a_i \phi_{,i}) - (w_{,i}, K \phi_{,i}) + \sum_{e=1}^{n_{el}} \left((\tau a_i w_{,i}), (a_i \phi_{,i} - K \phi_{,ii}) \right)_{\Omega^e} =$$

$$(w, f) + \sum_{e=1}^{n_{el}} \left((\tau a_i w_{,i}), f \right)_{\Omega^e}$$

(no Γ_h terms simply because we have no Γ_h for this case - will be there as normal when $\Gamma_h \neq 0$)

$$(w, a_i \phi_{,i}) = \int_\Omega w a_i \phi_{,i} \, d\Omega$$

$$(w_{,i} K \phi_{,i}) = \int_\Omega w_{,i} K \phi_{,i} \, d\Omega$$

$$((\tau a_i w_{,i}), (a_i \phi_{,i} - K \phi_{,ii}))_{\Omega^e} = \int_{\Omega^e} \tau a_i w_{,i} (a_i \phi_{,i} - K \phi_{,ii}) \, d\Omega^e$$

$$(w, f) = \int_\Omega w f \, d\Omega$$

$$((\tau a_i w_{,i}), f)_{\Omega^e} = \int_{\Omega^e} \tau a_i w_{,i} f \, d\Omega$$

Note - when using picewise linear finite elements one needs to apply a reconstruction process to determine a useful $\phi_{,ii}$ for use in the residual (using 0 does not work) - reconstruction may be desired even for higher order)

Note - an L_2 projection constructed in a form simular to what we saw for "stress" recovery methods is on the order of what is used for the linear elements.

A critical ingredient of these methods is the τ term.

Error analysis provides guidance on how to select this term. For the advection-diffusion problem we are looking at

$$\tau(x_i, P_e) = \frac{h_e}{2|\underset{\sim}{a}|} \xi(P_e)$$

magnitude of vector

P_e - local Peclet number for element e

$\underset{\sim}{a}$ - advective coefficients

h^e - element diameter

K - diffusion coefficient

$$P_e = \frac{m_p |\underset{\sim}{a}| h_e}{2K}$$

$$\xi(P_e) = \begin{cases} P_e & 0 \le P_e \le 1 \\ 1 & P_e \ge 1 \end{cases}$$

$$m_p = \min\left(1/3, 2C_k\right)$$

P_e is a measure of:

$$\frac{\text{advective transport rate}}{\text{diffusive transport rate}}$$

C_k - element order constant - bounds from inverse estimates

Of course things get much more complex with the real equations of interest – Navier-Stokes equations

Strong form of NS for unsteady (conserative form)

$$\rho_{,t} + [\rho u_i]_{,i} = 0 \quad \text{Continuity, Mass conservation}$$

$$[\rho u_j]_{,t} + [\rho u_i u_j]_{,i} + P_{,j} = \tau_{ij,i} + b_j \quad \text{momentum}$$

$$[\rho e_{tot}]_{,t} + [\rho u_i e_{tot}]_{,i} + [u_i P]_{,i} + [\tau_{ij} u_i]_{,j} + b_j u_j + r - q_{i,i}$$

energy balance

ρ = density, P = pressure

u_i - velocity component

e_{tot} = total energy $= e + \frac{u_i u_i}{2}$, e = internal energy

τ_{ij} = viscous stress $= 2\mu S_{ij} + \lambda \delta_{ij} S_{kk}$

S_{ij} = Strain rate tensor $= \frac{u_{i,j} + u_{j,i}}{2}$

q_i = heat flux vector, $= -k T_{,i}$, k-conductivity

b_i = body force vector

r = heat supplied

Collecting unknowns into a vector we have

$$\underset{\sim}{U} = \rho \begin{Bmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ e_{tot} \end{Bmatrix} = \rho \begin{Bmatrix} 1 \\ u_j \\ e_{tot} \end{Bmatrix}$$

with this we get a form that looks like

$\sim$ - vector

$\approx$ - matrix

$$\underset{\sim}{U}_{,t} + \underset{\sim}{F}_{i,i} = \underset{\sim}{f} = \begin{Bmatrix} 0 \\ b_j \\ b_i u_i + r \end{Bmatrix}$$

$$\underset{\sim}{F}_i = \underbrace{\rho \begin{Bmatrix} u_i \\ u_i u_j \\ u_i e_{tot} \end{Bmatrix} + \begin{Bmatrix} 0 \\ p\delta_{ij} \\ p u_i \end{Bmatrix}}_{F_i^{adv}} \underbrace{- \begin{Bmatrix} 0 \\ \tau_{ij} \\ u_i \tau_{ij} \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ q \end{Bmatrix}}_{F_i^{diff}}$$

Can proceed with MWR - goal is semi-discrete form - will still have time derivatives

$$\int_\Omega \underset{\sim}{w} \cdot \left\{ \underset{\sim}{U}_{,t} + \underset{\sim}{F}_{i,i} - f \right\} d\Omega = 0$$

Integrate by parts

$$\int_\Omega \left[\underset{\sim}{w} \cdot \{ U_{,t} - f \} - w_{,i} \cdot F_i \right] d\Omega + \int_\Gamma \underset{\sim}{w} \cdot \underset{\sim}{F}_i n_i \, d\Gamma = 0$$

That looks fine - but having ρ as part of unknowns not good - particularly for incompressible where it is constant

Thus a change of variables is desired - Can be done inside the integrals to be a vector $\underset{\sim}{Y}$ (there are various versions of terms used for Y + compressible different than in compressible)

With out being specific on the transformation to go from $\underset{\sim}{U}$ to $\underset{\sim}{Y}$ we need to pick-up chain rule terms

$$\underset{\sim}{U}_{,t} = \frac{\partial \underset{\sim}{U}}{\partial Y} \frac{\partial \underset{\sim}{Y}}{\partial t} = \underset{\approx}{A}_0 \, \underset{\sim}{Y}_{,t} \qquad A_0 - 5\times 5 \text{ matrix}$$

$$F^{adv}_{L,i} = \frac{\partial \underset{\sim}{F}_i^{adv}}{\partial \underset{\sim}{Y}} \frac{\partial Y}{\partial x_i} = \underset{\sim}{A}_i \, \underset{\sim}{Y}_{,i} \qquad \text{Jacobian WRT } Y - 3 \; 5\times 5 \text{ matrices}$$

$$\underset{\sim}{F}^{diff}_i = -\underset{\approx}{K}_{ij} \frac{\partial \underset{\sim}{Y}}{\partial x_j} = -\underset{\approx}{K}_{ij} \, \underset{\sim}{Y}_{,j} \qquad 5\times 5 \text{ and } 3\times 3 \text{ of them}$$

This yields the weak form

$$\int_\Omega \underset{\sim}{W} \cdot \{ \underset{\approx}{A}_0 \, \underset{\sim}{Y}_{,t} - \mathcal{F}(Y) \} - \underset{\sim}{W}_{,i} \cdot \underset{\sim}{F}_i(Y)] \, d\Omega$$

$$+ \int_\Gamma \underset{\sim}{W} \cdot \underbrace{\underset{\sim}{F}_i(\underset{\sim}{Y})}_{\text{the whole thing}} n_i \, d\Gamma$$

One can rewrite the strong form based on terms '$\underset{\sim}{Y}$ variables' - written in a quasi-linear form this looks like

$$\underset{\sim}{A}_0 \underset{\sim}{Y}_{,t} + \underset{\approx}{A}_i \underset{\sim}{Y}_{,i} - [\underset{\approx}{K}_{ij} \underset{\sim}{Y}_{,j}]_{,i} = \mathcal{F}(\underset{\sim}{Y})$$

in short - $\mathcal{L} \underset{\sim}{Y} = \mathcal{F}(\underset{\sim}{Y})$

$$\mathcal{L} = \underset{\approx}{A}_0 \frac{\partial}{\partial t} + \underset{\approx}{A}_i \frac{\partial}{\partial x_i} - \frac{\partial \left[K_{ij} \frac{\partial}{\partial x_j} \right]}{\partial x_i}$$

Can use different sets of variables for different classes of flows

primitive variables $\left\{\begin{matrix} p \\ u_i \\ T \end{matrix}\right\}$ better for incompressible

entropy variables $\frac{1}{T}\left\{\begin{matrix} \mu_c - u_i u_i/2 \\ u_j \\ -1 \end{matrix}\right\}$ good for highly compressible

We can proceed with this "standard Galerkin" but it is unstable if you use equal order interpolants for all variables.

To stabilize -

1). Interpolate pressure 1 order lower than velocity (for primative variables) - works well at high Re

2) Add a stabilizing operator

⇛ We will look at adding a stabilization term There is theory, etc. used to define valid ones - We will simply state a popular one - We will add

$$\sum_{e=1}^{n_{el}} \int_{\Omega^e} \left[\hat{\mathcal{L}} \underset{\sim}{W} \cdot \underset{\approx}{\tau} \left(\underset{\sim}{\mathcal{L}} Y - \mathcal{F} \right) \right] d\Omega^e$$

notes -

- Sum over element integrals - not global integral
- The strong form is in the integrand - its a weighted residual
- There is a matrix of weights, $\underset{\approx}{\tau}$, they must be carefully constructed - see literature and F.E. for fluids course

$\hat{\mathcal{L}} W$ - a modified weighting function from what one would get for a weighted least squares of the residual

options:

$\hat{\mathcal{L}} = \mathcal{L}^T$ would be least squars (GLS)

$\hat{\mathcal{L}} = \mathcal{L}^{adv^T} = \underbrace{\underset{\approx}{A}_i^T \frac{\partial}{\partial x_i}}_{\text{focus on "problem" term}}$ Popular SUPG - streamline Upwing Petrov Galerkin method

$\hat{\mathcal{L}} = -\mathcal{L}^{*^T} = \underset{\approx}{A_0}^T \frac{\partial}{\partial t} + \underset{\approx}{A}_i^T \frac{\partial}{\partial x_i} + \frac{\partial}{\partial x_i}\left[K_{ij}^T \frac{\partial}{\partial x_j}\right]$

$\mathcal{L}^*$ - adjoint of $\mathcal{L}$

Spatial discretization (discretize time later using "finite difference" methods)

$\underset{\sim}{W} = \sum_{A=1}^{n_{np}} N_A \underset{\sim}{w}_A$ $\quad \underset{\sim}{w}_A$ - nodal vector of length five for node A

$\underset{\sim}{Y} = \sum_{B=1}^{n_{np}} N_B \underset{\sim}{Y}_B$ $\quad \underset{\sim}{Y}_B$ vector of length 5 of solution values at node B

$\underset{\sim}{Y}_{,t} = \sum_{B=1}^{n_{np}} N_B \underset{\sim}{Y}_{B,t}$ $\quad \underset{\sim}{Y}_{B,t}$ time derivative of "variables" at node B

Plug this into the weak form and you get the usual form

$$\sum_{A=1}^{n_{np}} \underset{\sim}{W}_A \cdot \underset{\sim}{G}_A = 0$$

$$\mathbf{G}_A = \int_\Omega \left[N_A \left[\mathbf{A}_0 \sum_{B=1}^{n_{nd}} N_B \mathbf{Y}_{B,t} - \mathbf{\mathcal{F}} \right] - N_{A,i}\, \mathbf{F}_i \right] d\Omega \; +$$

$$\int_\Gamma N_A\, \mathbf{F}_i\, n_i \, d\Gamma \quad +$$

$$\sum_{e=1}^{n_{el}} \int_{\Omega^e} N_{A,i}\, \mathbf{A}_i\, \tau \left\{ \mathbf{A}_0 \sum_{B=1}^{n_{nd}} N_B \mathbf{Y}_{B,t} + \mathbf{A}_i \sum_{B=1}^{n_{nd}} N_{B,i} \mathbf{Y}_B - \left[\mathbf{K}_{ij} \sum_{B=1}^{n_{nd}} N_{B,j} \mathbf{Y}_B \right]_{,i} - \mathbf{\mathcal{F}} \right\} d\Omega^e$$

Since $\mathbf{W}_A$ is arbitrary

$\mathbf{G}_A = 0 \quad A = 1(1)\, n_{np}$

$\Rightarrow$ As with other F.E. we do things an element at a time and fully account for local nature of the shape functions.

The semi discrete form after formulation will look like

$$\mathbf{M}\, \check{\mathbf{Y}}_{,t} + \mathbf{S}\, \check{\mathbf{Y}} \qquad \check{\mathbf{Y}} \text{ is the full set of dof (with BC. accounted for)}$$

$\mathbf{M}$ and $\mathbf{S}$ matrices defined from terms above

This is a matrix ODE in time

Various time discretization methods are possible –

An appropriate $\underset{\sim 5\times 5}{\tau}$ matrix is needed

One option is a diagional matrix:

$$\begin{bmatrix} \tau_c & 0 & 0 & 0 & 0 \\ 0 & \tau_m & 0 & 0 & 0 \\ 0 & 0 & \tau_m & 0 & 0 \\ 0 & 0 & 0 & \tau_m & 0 \\ 0 & 0 & 0 & 0 & \tau_e \end{bmatrix}$$

$$\tau_c = \frac{|\underset{\sim}{u}|\, h_1^e}{2} \min(1, Re^h_e)$$

$$\tau_m = \min\left(\frac{\Delta t}{\rho}, \frac{h_2^e}{2\rho|\underset{\sim}{u}|}, \frac{m^p (h_3^e)^2}{4\mu}\right)$$

$$\tau_e = \min\left(\frac{\Delta t}{\rho C_v}, \frac{h_3^e}{2\rho C_v |\underset{\sim}{u}|}, \frac{m^p (h_3^e)^2}{4k}\right)$$

$$Re^h = \frac{\rho |\underset{\sim}{u}| h_1^e}{2\mu}, \quad m^p = \min(1/3, 2C_k)$$

h_1^e = "continuity scaled" edge length
h_2^e = "momentum scalled" edge length
h_3^e = "energy scaled" edge length