Computational Fluid Dynamics - Incompressible Flows

Doug Jacobsen

March 25, 2008
CFD - Incompressible Flows

- CFD is a Huge field
- Numerical Techniques for solving Navier Stokes
- Incompressible Constraints

- Different Techniques
CFD - Incompressible Flows

- CFD is a Huge field
- Numerical Techniques for solving Navier Stokes
- Incompressible Constraints
  - Constant Density
  - Constant Kinematic Viscosity
- Different Techniques
  - Spectral Elements
  - Finite Elements
Basis Functions

- Basis functions need to be evaluated a lot.
- Need to find “optimal” basis functions.
- Basis functions will be used in Galerkin Approximation
- A common choice are the Gauss-Lobatto Legendre Interpolants

\[ \varphi_i(\xi) = -\frac{(1 - \xi^2)L'_N(\xi)}{N(N+1)L_N(\xi_i)(\xi - \xi_i)} \]
Basis Functions - Contd.

There are two forms of bases one can use.

- Modal Basis Functions
  Basis functions with Increasing Degree
- Nodal Basis Functions
  Basis functions with Constant Degree ie. GLL Interpolants

![Basis Functions Diagram](image-url)
Discrete Equations

The equations involved with CFD make the system of equations very sparse. There are two main matrices that must be solved, and these are typically called the “mass” and “stiffness” matrices. Labeled $M$ and $A$ respectively.

These two matrices are both computed using the basis functions, and thus have some special properties, mainly $A$ though.

- $A$ is banded as a result of using local basis functions
- $A$ can be efficiently stored, computed and factored as a result
Basic Operations

We need to find cheap ways to perform a few basic operations, like:

- Integration
- Projection
- Differentiation
Integration is the easiest one to solve.
Most methods use Gaussian quadrature with weights:

\[(1 - \xi)^\alpha (1 + \xi)^\beta\]
Projection

Projection is known as the method of acquiring the quadrature points from a modal basis. Also it is used to determine the modal expansion coefficients from a set of nodal values. This process includes a lot of Linear Algebra, but after you finish, you can note that in the GLL nodal basis, the $M$ matrix is diagonal. This makes the problem a lot more simple since you can calculate its inverse trivially.

Figure 5.8: Definition of the standard quadrilateral domain $\mathbb{R}^2$. General curvilinear elements can always be mapped back to the standard element as shown.
Since the basis is formed from continuous functions, in principal, derivates can be evaluated. But we also only need them at the quadrature points, to do this, typically, one uses a Lagrangian derivative matrix formed by:

\[ D_{ip} = \left. \frac{d\varphi_p}{d\xi} \right|_{\xi_i} \]

This has a cost of \( O(N^2) \) for a 2-D problem.
Global Matrix Operations

All of the problems presented in the reading material require $C^0$ continuity, globally. This is only really an issue along the boundary. To insure this is true, one can minimize the difference in function values across each nonconforming interface.

$$\int_{\Gamma} (u - v)\psi ds$$

$v$ is taken to be the solution along some edge of some element, and $u$ is taken to be the solution along some edge of an adjacent nonconforming element. If implemented properly this method always converges to a continuous solution if one exists.
For the purposes of this paper, the global data was stored in a large unstructured array, along with an indexing array. However, Local data can be stored in any convenient format. Mesh storage will be discussed later. For high order elements the data is naturally partitioned into sets and can be operated on in this way, which is a big advantage.
Solution Techniques

A particularly successful method is Conjugate Gradient Iteration. Mainly because the structure allows the Helmholtz inner product to be evaluated using elemental matrices only. However, one needs to have a good preconditioner to help with rapid convergence. This preconditioner needs to be spectrally close to the full stiffness matrix yet easy to invert. Popular ones include Cholesky factorization and low-order approximations. Although they can be complicated to construct. Static Condensation is also used to reduce the complexity of the stiffness matrices.
Solution Techniques - Contd.

(a) Static Condensation

(b) Optimal Path

Figure 5.6 Static condensation form of the spectral element stiffness matrix. The vector $\phi = \phi_{\partial}$ represents the boundary (mortar) solution, while $\phi_{\Omega}$ represents the interior solution.

Figure 5.7 Bandwidth optimization for a spectral element mesh: (a) computational domain, (b) connectivity graph, and (c) an optimal path for numbering the boundary nodes in the mesh.
Adaptive Mesh Refinement

For the mesh, a common choice of storage is in a quad- or oct-tree. Note that for complex geometries one can use multiple trees at the coarse level to account for everything. Ideally, adaptive mesh refinement algorithms would take an error estimate as input, and produce a new mesh that reduces the error. However, in our case we don’t have an easy way to estimate the error, and there are unlimited ways that such an algorithm can improve our model. The first issue is solved by introducing a pseudo-heuristic error estimate based on the local polynomial spectrum and the second is solved by restricting “improvements” to propagating refinement down the tree.
For generation of a mesh, it is just a simple quad-tree, in the example. Although, refinements are limited to 1 level per neighbor. So every edge can only have 3 elements touching it at max. For refinement criteria, the easiest scheme is to refine everywhere the gradient is large, by requiring that:

$$\|\nabla u^{(k)}\| \leq \varepsilon \|u^h\|_1$$
Figure 5.9  A four-level quadtree mesh, expanded to show the elements that make up each level. Each leaf node $s^{(0)}$ has a unique integer key shown in binary. Daughter keys are generated from a parent’s key by a two-bit left shift, followed by a binary or in the range 00 to 11. The active elements $D^{(k)}$ that make up the current discretization are shown with a solid outline.