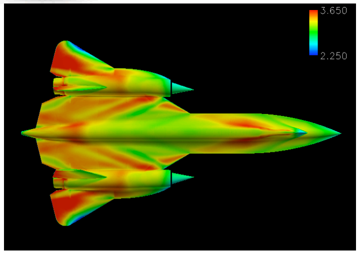
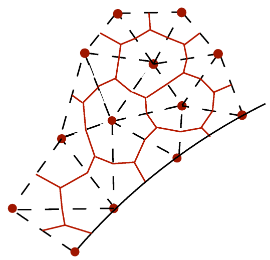
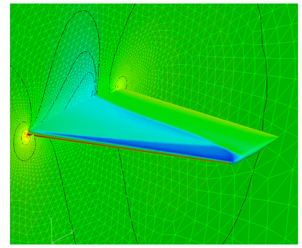
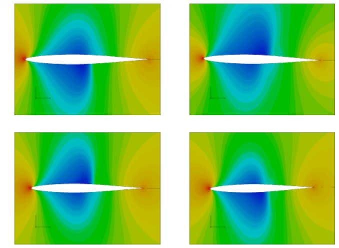
STANFORD UNIVERSITY

Aerospace Computing Laboratory









Solution Algorithms for Viscous Flow

Antony Jameson

Department of Aeronautics and Astronautics

Stanford University, Stanford CA

Indian Institute of Science

Bangalore, India

September 20, 2004

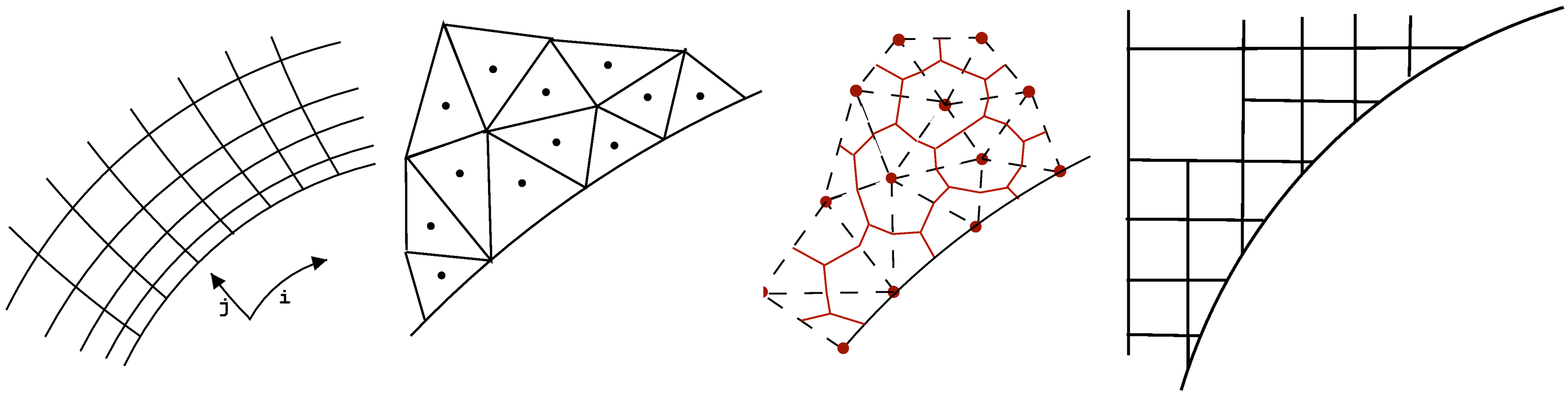
Flo3xx

Computational Aerodynamics on Arbitrary Meshes

Antony Jameson
Georg May

Support for Arbitrary Meshes

- Examples of mesh types which are being used in computational aerodynamics



Structured

Unstructured Cell-Centered

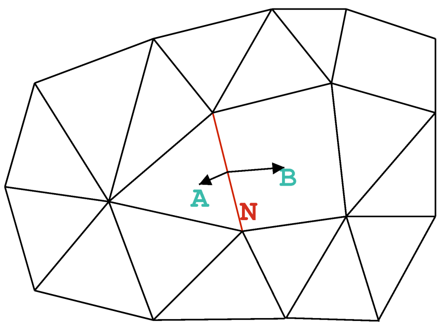
Unstructured Cell-Vertex

Nested Cartesian With Cut Cells

- In Flo3xx a unified mesh-blind formulation supports all of these in one code
- Designed to meet the following objectives:
  - Platform for automatic mesh adaptation
  - Migration path to emerging mesh generation technologies
  - A robust algorithm that is tolerant to bad meshes

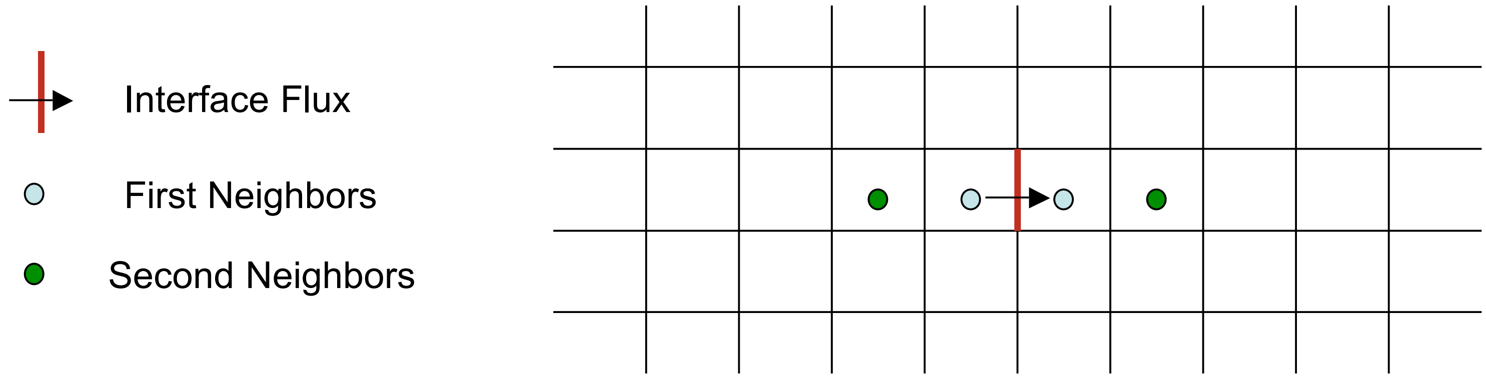
Support for Arbitrary Meshes

- Conservation laws are enforced on discrete control volumes
- Fluxes of conserved variables are exchanged through interfaces between these cells
- Independent of the mesh topology, each interface separates exactly two control volumes (on the right, face N separates cells A and B)



All algorithms are expressed in terms of a generic interface-based data structure

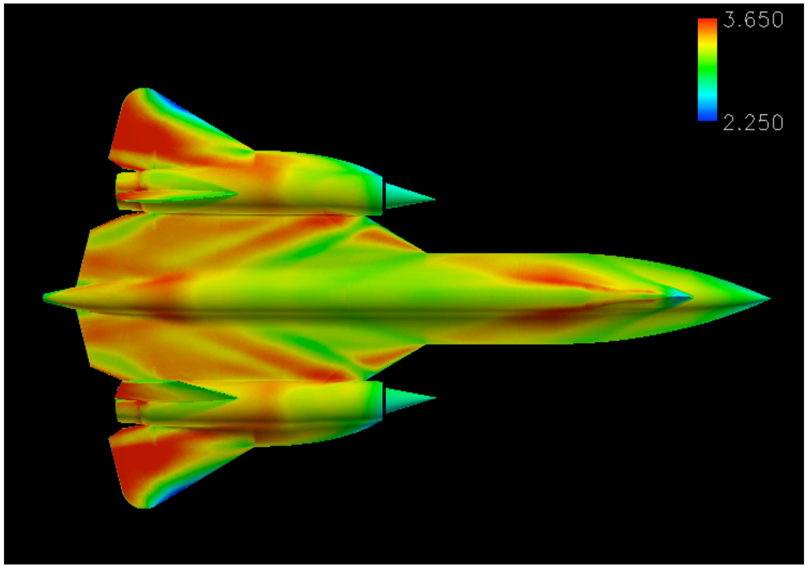
Treatment of Structured Meshes



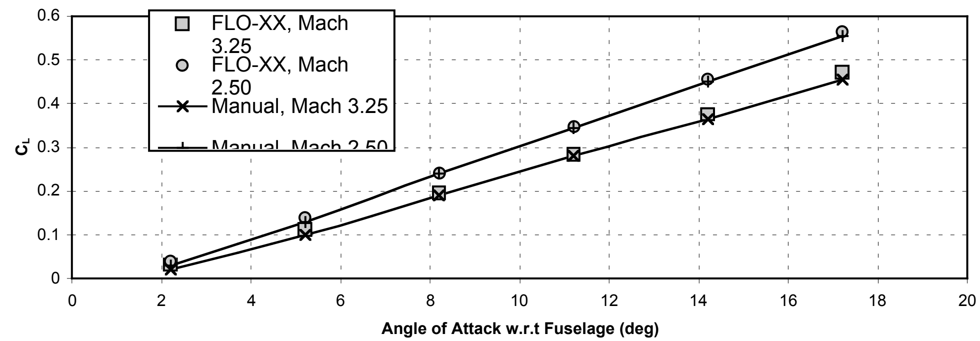
- Associate first and second neighbors with each face
- Allows implementation of standard schemes with five-point stencils (Jameson-Schmidt-Turkel JST, SLIP) in the same code
- Eliminates the need for gradient reconstruction
- Numerical experiments verify 25% overhead due to indirect addressing in comparison with standard structured-code implementation (FLO107)

Flo3xx in Action…

Mach Number - Upper



Angle-of-Attack sweep

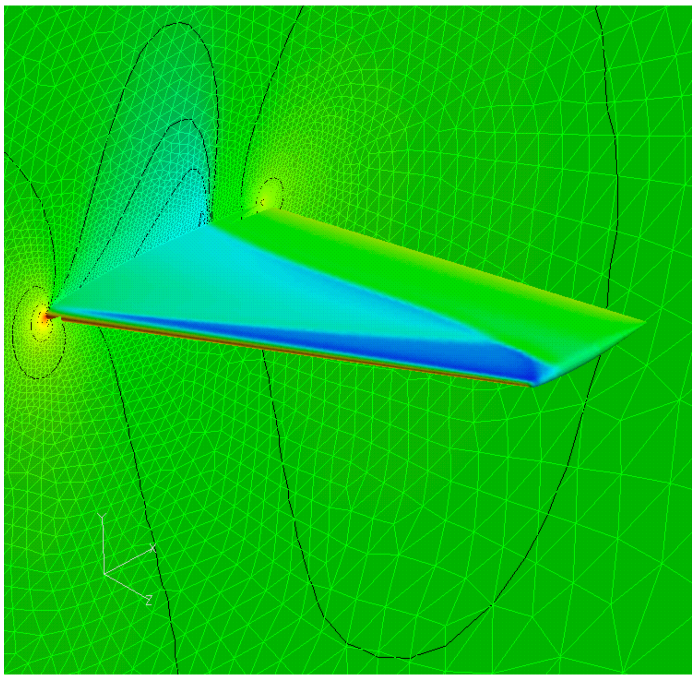


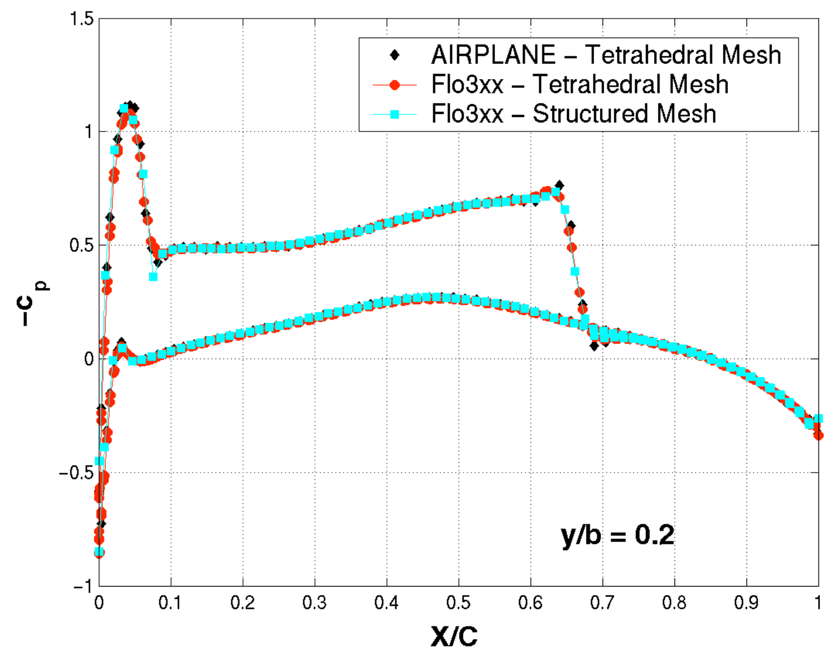
Geometry Courtesy of Lockheed Skunk Works

Lockheed SR71 at M= 3.2, - Euler calculation with 1.5 Million grid points

- **From IGES definition to completed result in one week, including CAD fixes, mesh generation**
- **We need to be able to compute extreme test cases**
- **This concerns both complexity of geometry and flow conditions**

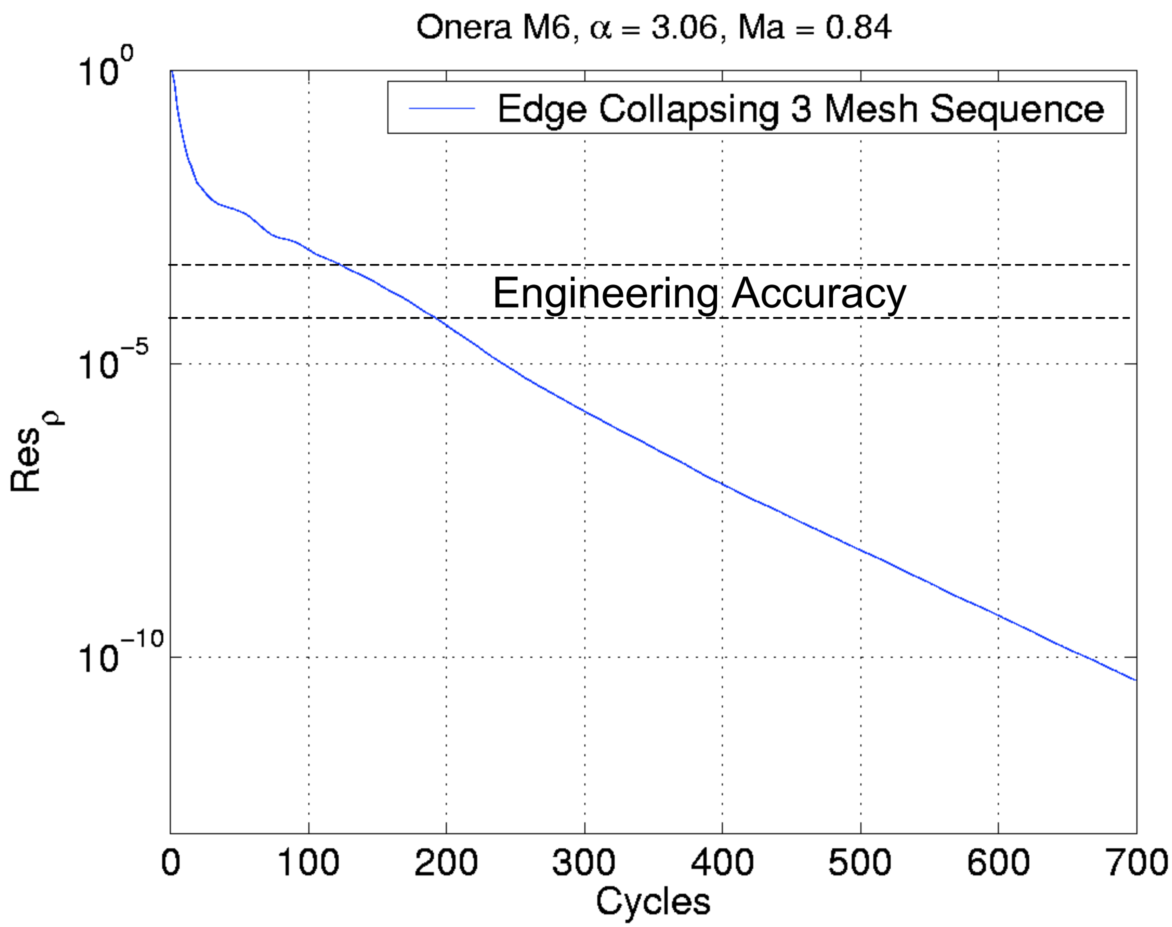
Validation of Unified Solution Algorithm in Flo3xx: Inviscid Transonic Flow

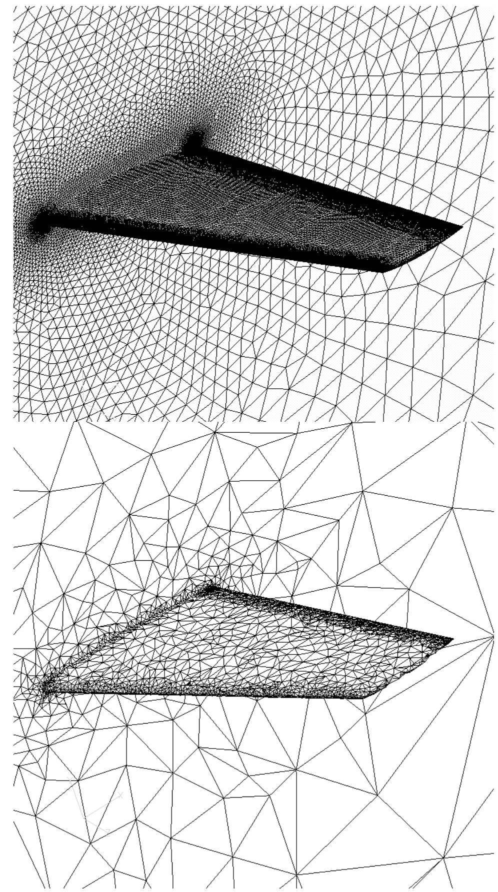




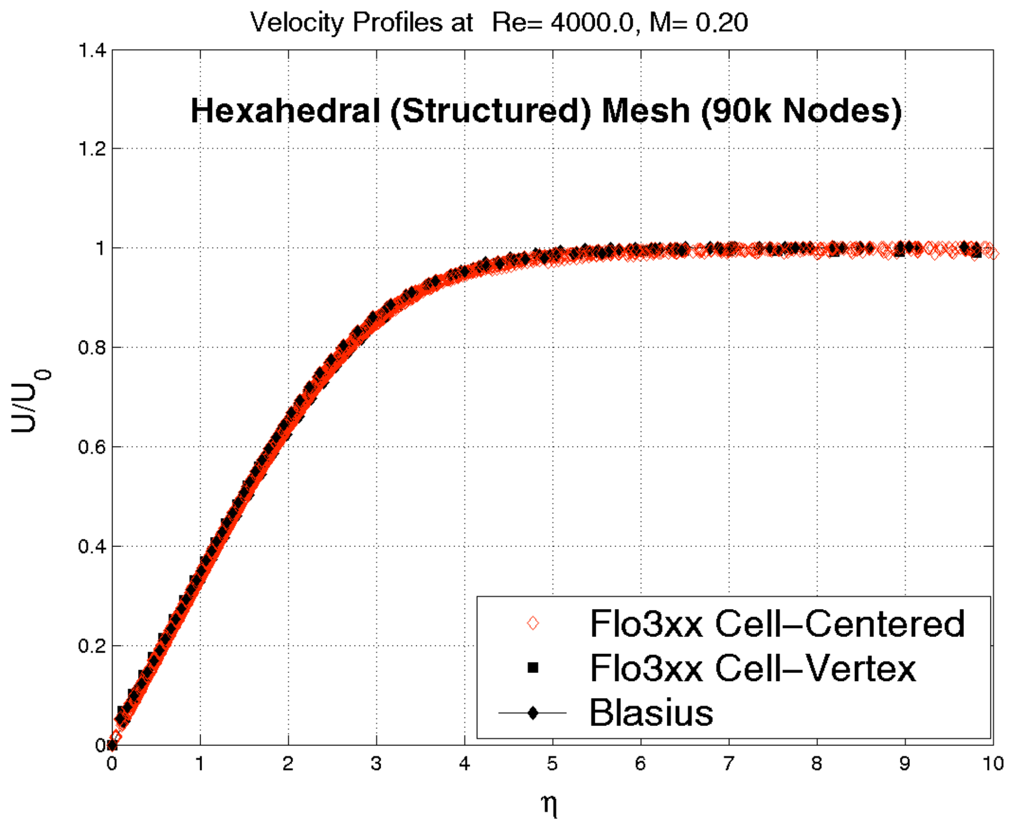
- Onera M6 Wing at M=0.84 and / / = 3.06 degrees

Convergence Using Automatic Multigrid



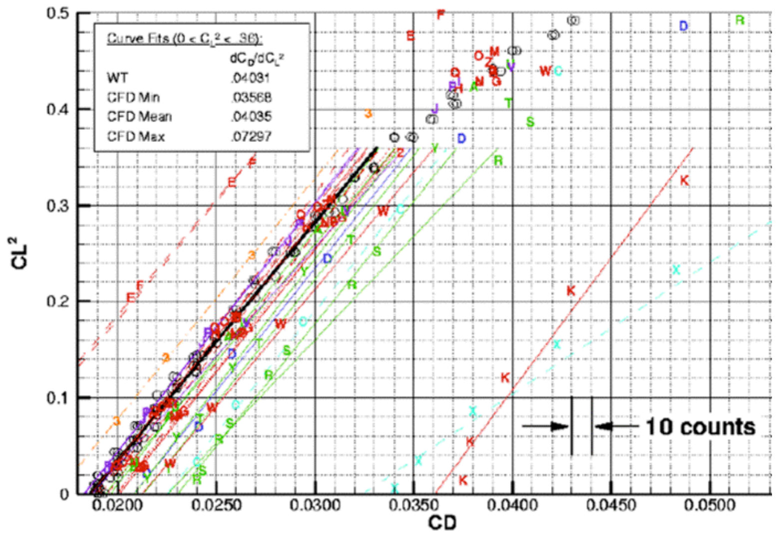


Initial Validation for Viscous Flow: Zero-Pressure-Gradient Boundary Layer

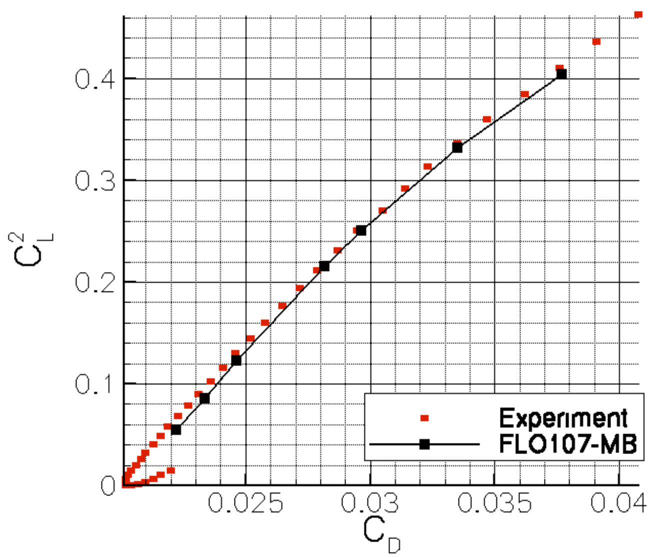


RANS Results Using FLO107-MB For Drag Prediction Workshop

Statistical Evaluation DPW1 – All Participants



Flo107-MB (DPW2)



- Accurate drag prediction for complex geometries in transonic flow is still very hard
- Flo3xx is currently in viscous validation phase.
- FLO107-MB has been thoroughly validated.
- Results of right figure were obtained with CUSP scheme and k-/ / turbulence model

Flo3xx Payoffs

- Highly flexible platform for all applied aerodynamics problems and other problems governed by conservation laws
- Fast turnaround through convergence acceleration techniques
- Framework can be used to support advanced research, such as the BGK method or the Time-Spectral Method, which will be addressed in this talk
- This means, take advanced research out of a laboratory setting and apply it to problems of practical engineering interest, which is ultimately the only way to make an impact on the state-of-the art

Non Linear Symmetric Gauss-Seidel Multigrid Scheme

Jameson + Caughey 2001
Evolved from LUSGS scheme
Yoon + Jameson (1986)
Rieger + Jameson (1986)
Achieved "Text Book" Multigrid Convergence

Nonlinear Symmetric Gauss-Seidel (SGS) Scheme

Forward and reverse sweeps:

For 1D case: $\frac{\partial w}{\partial t} + \frac{\partial}{\partial x} f(w) = 0$

Sweep (1): Increasing j →

$$w_j^{(1)} = w_j^{(0)} - |A|^{-1}\left[f^{(00)}_{j+\frac{1}{2}} - f^{(10)}_{j-\frac{1}{2}}\right]$$

$$f^{(01)}_{j-\frac{1}{2}} = f\left(w_j^{(0)}, w_{j-1}^{(1)}\right)$$

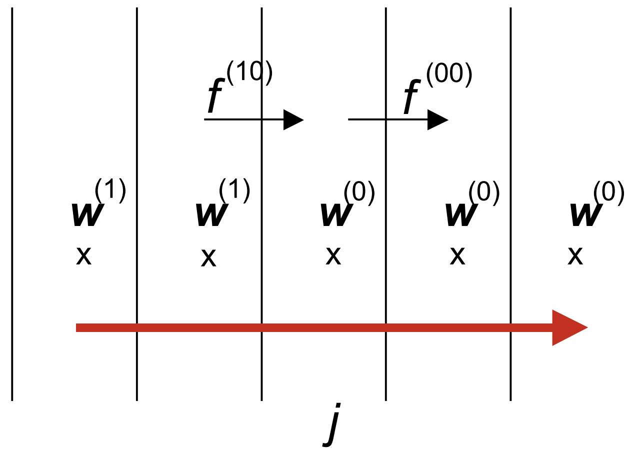
$$A = \frac{\partial f}{\partial w}$$

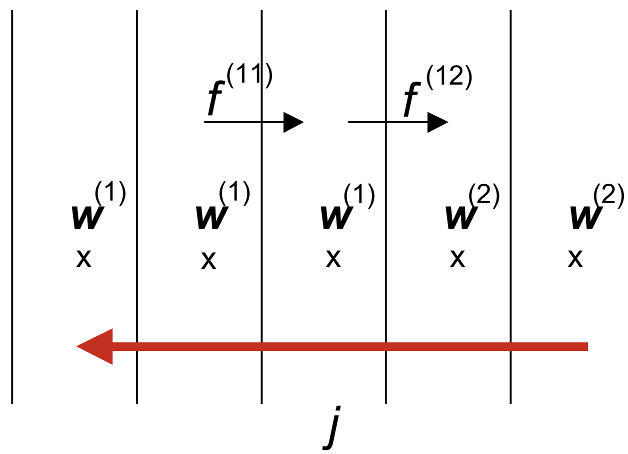
Sweep (2): Decreasing j →

$$w_j^{(2)} = w_j^{(1)} - |A|^{-1}\left[f^{(12)}_{j+\frac{1}{2}} - f^{(11)}_{j-\frac{1}{2}}\right]$$

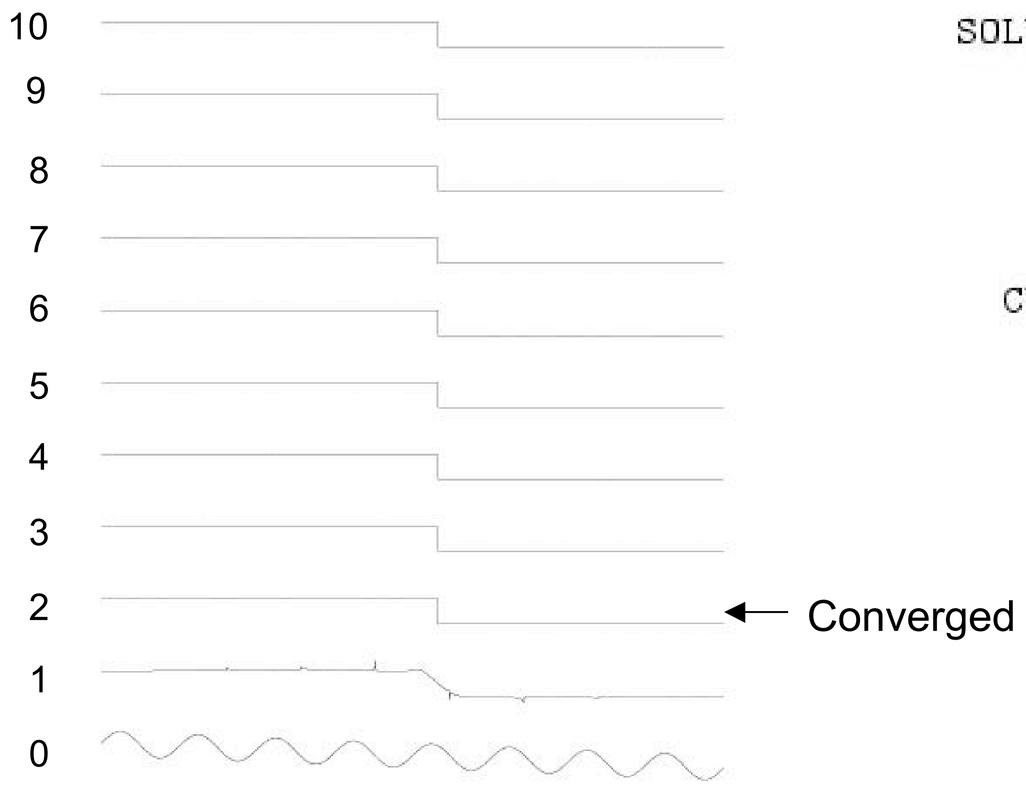
4 Flux evaluations in each double sweep

Cost per iteration similar to 4 - stage Runge - Kutta scheme





Solution of Burgers Equation on 131,072 Cells in Two Steps With 15 Levels of Multigrid



SOLUTION OF BURGERS EQUATION BY SYMMETRIC RELAXATION

131072 CELLS 15 LEVELS

CFL 1.000 RAVG 0.0

SOLUTION OF UT +UUX = 0.

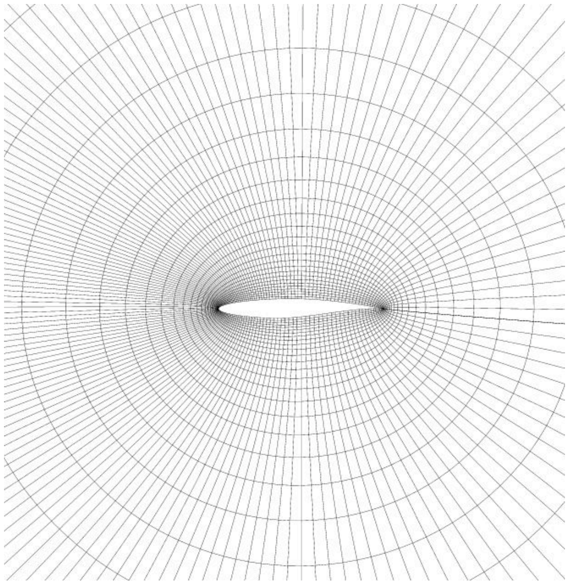
NX	N MESH
131072	15

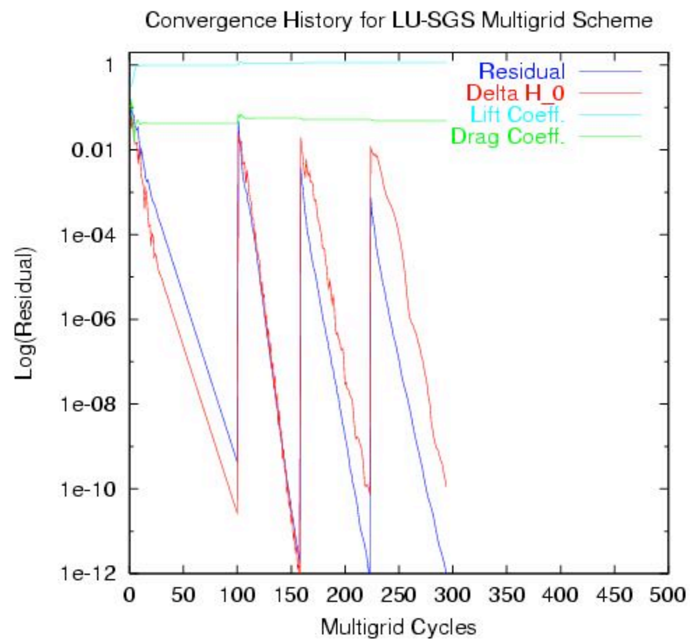
CFL

1.0000

CYCLE	AVG RESIDL	MAX RESIDL	IMAX
1	0.8038E-02	0.1295E+01	57667
2	0.3413E-04	0.4337E+00	70962
3	0.2773E-06	0.9391E-01	70961
4	0.1576E-14	0.9567E-05	70959
5	0.3037E-28	0.7905E-13	70960
6	0.3037E-28	0.7905E-13	70960
7	0.3037E-28	0.7905E-13	70960
8	0.3037E-28	0.7905E-13	70960
9	0.3037E-28	0.7905E-13	70960
10	0.3037E-28	0.7905E-13	70960

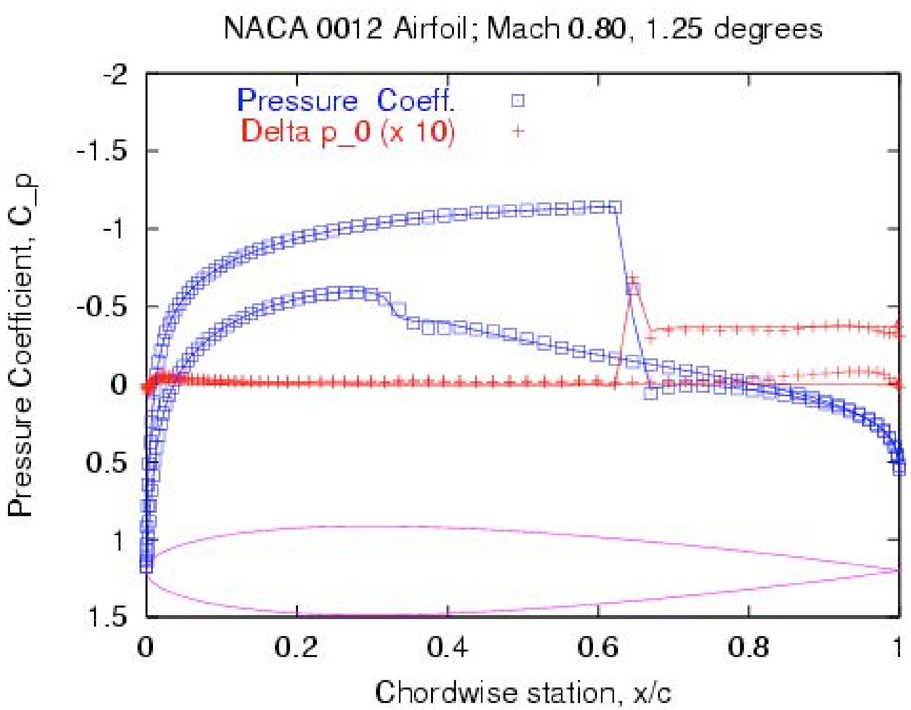
Solution of 2D Euler Equations: Convergence for NACA0012



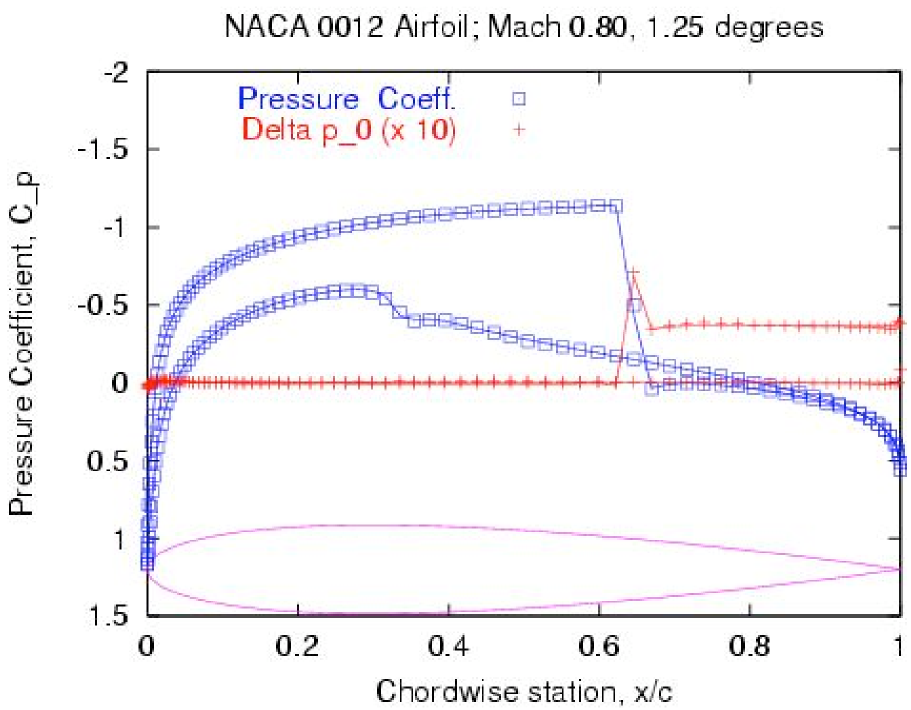


- The convergence history shows the successive computation on meshes of different sizes
- The convergence rate is independent of the mesh size
- Convergence rate ~ .75 per cycle

Solution of 2D Euler Equations
NACA0012 Airfoil



Solution after 3 multigrid cycles

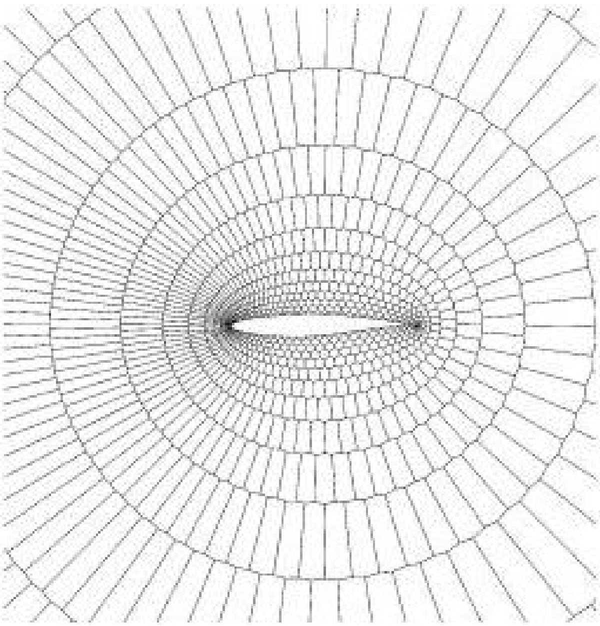


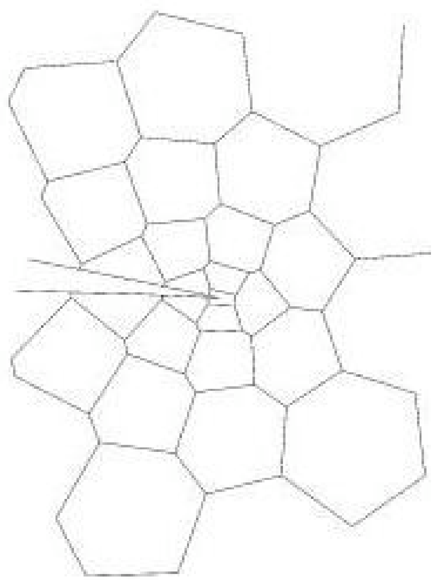
Solution after 5 multigrid cycles

Solid lines: fully converged result

Face-based Gauss Seidel (FBGS) Scheme

(Following a suggestion by John Vassberg)





- On an arbitrary grid, loop over faces instead of looping over cells
- Update the cells adjacent to a face as you go along
- Updated state will be used on next visit to a cell

The Finite-Volume BGK Scheme

Using Statistical Mechanics to Enhance Computational Aerodynamics

Balaji Srinivasan
Georg May
Antony Jameson

A Major Conceptual Difference Between Continuum Mechanics and Statistical Mechanics

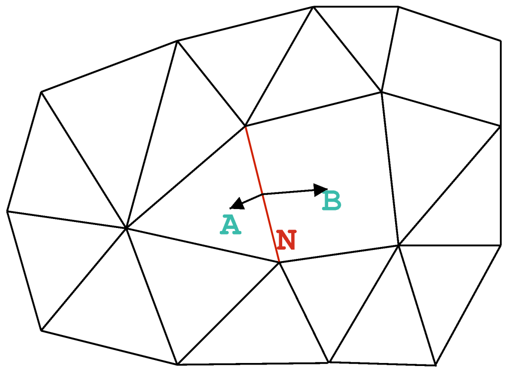
- In continuum mechanics the unknown solution variables are defined “pointwise” with precise values:

$$U = U(x, y, z, t)$$

- In statistical mechanics the solution variables exist only as moments of a statistical distribution in physical and phase space, or as “expectation values”:

$$U = \int u\, f(x, y, z, u, v, w, \square, t)\, du\, dv\, dw\, d\square$$

The Key Idea of the Finite-Volume BGK Scheme



- Compute the fluxes for the Navier-Stokes equations at interface N from the distribution functions in cells A and B
- A time-dependent distribution function needs to be constructed at each time step for each cell

Finding the Distribution Function

- The equilibrium distribution function
  is known from Boltzmann statistics:

$$f_{eq} = g(x, y, z, u, v, w, \xi) = A(x, y, z) e^{-\lambda(x,y,z)\left\{(U-u)^2 + (V-v)^2 + (W-w)^2 + \xi^2\right\}}$$

- The nonequilibrium distribution function
  is unknown, but its evolution is given by the Boltzmann equation:

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} + v\frac{\partial f}{\partial y} + w\frac{\partial f}{\partial z} = Q(f, f)$$

Collision Integral

- Global numerical solution infeasible, because of high dimensionality

A Crucial Simplification
(Bhatnagar, Gross & Krook - BGK)

- Replace the Collision Integral Q with a linear relaxation term:

$$Q = -\frac{f - g}{\tau} \quad | \, | \quad \boxed{\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} + v\frac{\partial f}{\partial y} + w\frac{\partial f}{\partial z} = -\frac{f-g}{\tau}}$$

Collision Time

- This equation can be solved analytically:

$$f(\vec{x},\vec{u},t,\xi) = \int_0^t g(\vec{x} - \vec{u}(t - t'),\vec{u},t',\xi)\, e^{-\frac{(t-t')}{\tau}}\, dt' + e^{-\frac{t}{\tau}} f_0(\vec{x} - \vec{u}t,\vec{u},\xi)$$

A Key Observation

- By Chapman-Enskog expansion the Navier Stokes equations can be recovered from the BGK equation, with the viscosity coefficient

$$\mu = \tau p$$

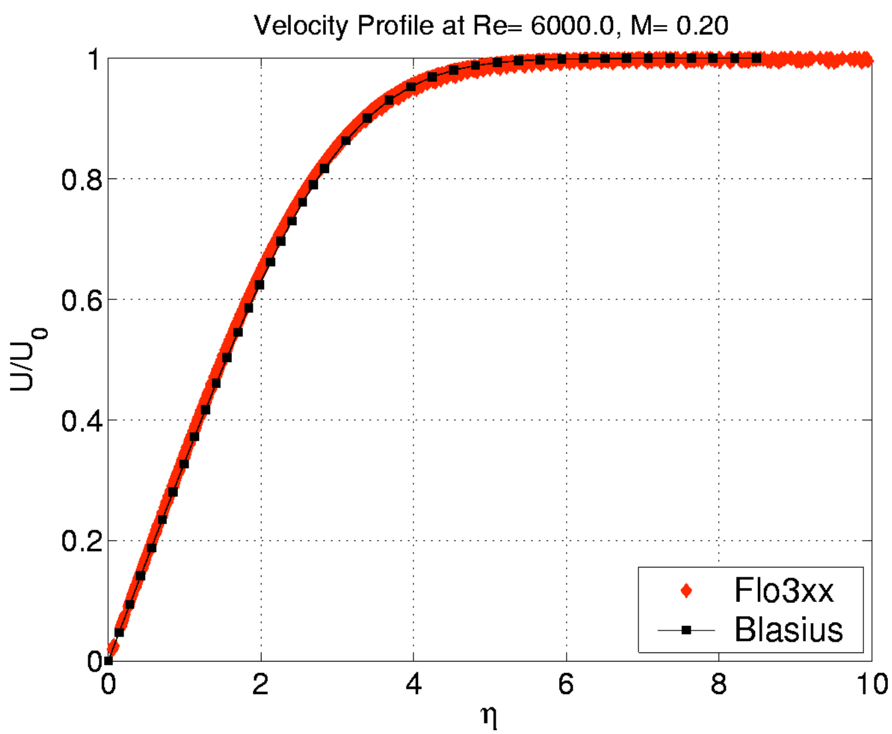
- By setting the collision time τ appropriately, Navier-Stokes fluxes can be computed directly from the distribution function

Payoff

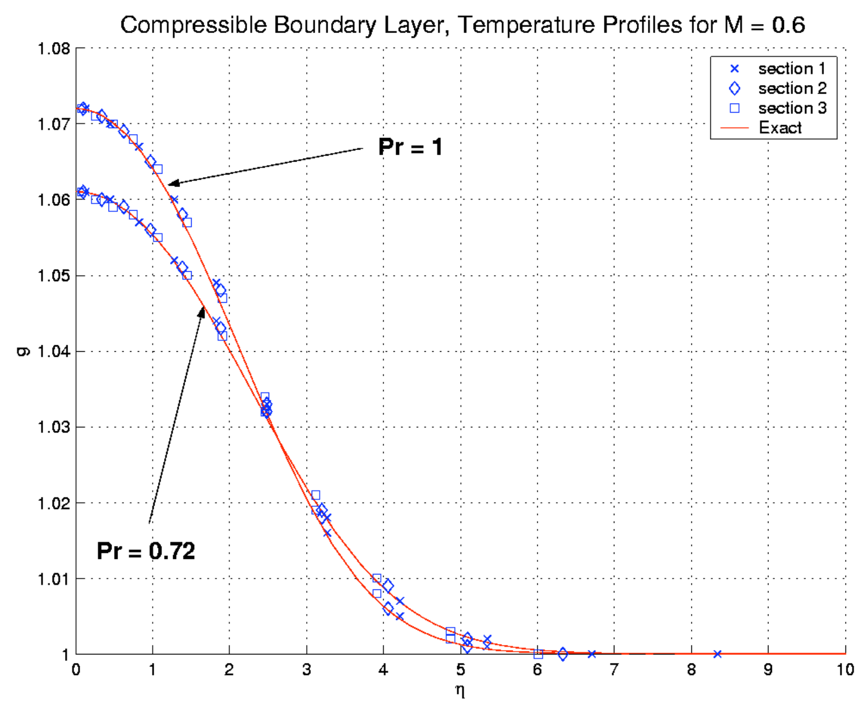
- It is not necessary to compute the rate of strain tensor in order to calculate viscous fluxes
- This eliminates the need to perform two levels of numerical differentiation, which is difficult on arbitrary meshes
- Improved accuracy and reduced sensitivity to the quality of the mesh
- Automatic upwinding via the kinetic model, with no need for explicit artificial diffusion, thus reduced computational complexity

Viscous Validation of the BGK Scheme: Zero-Pressure-Gradient Boundary Layer

Velocity Profile (Incompressible)

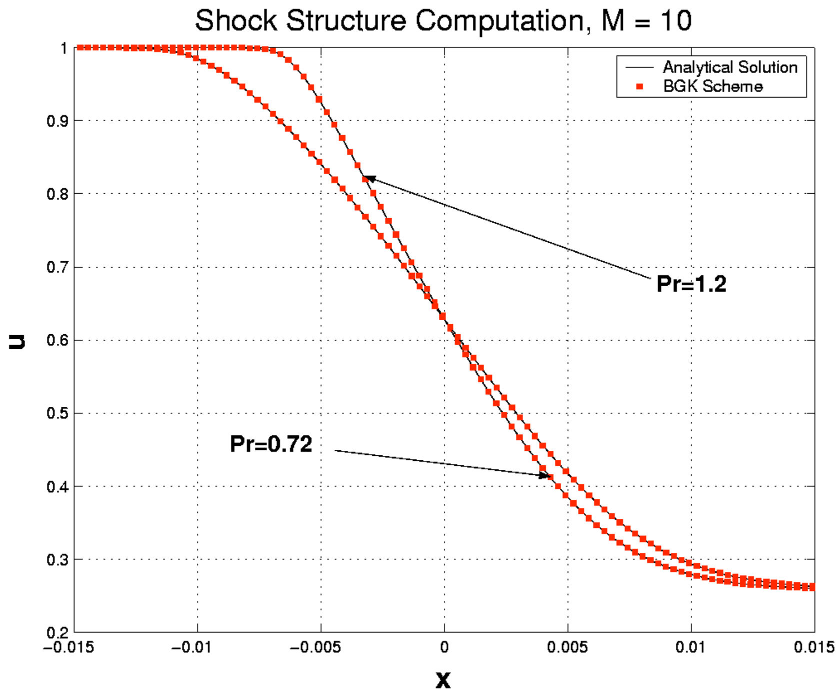


Temperature Profile (Compressible)

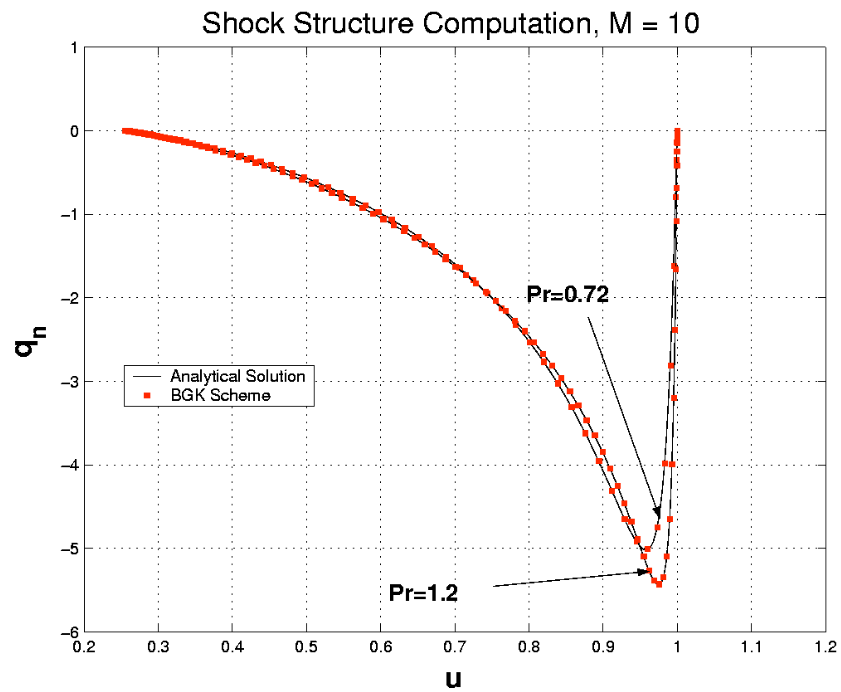


Viscous Validation of the BGK Scheme: 1D Shock Structure (M=10)

Velocity

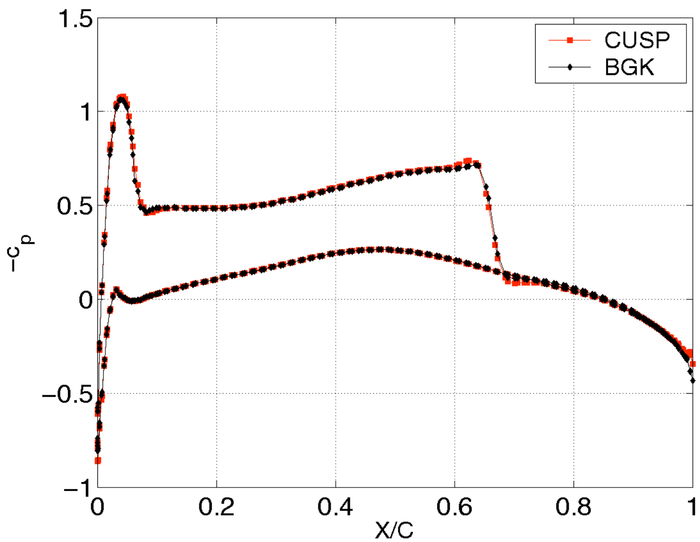


Heat Flux

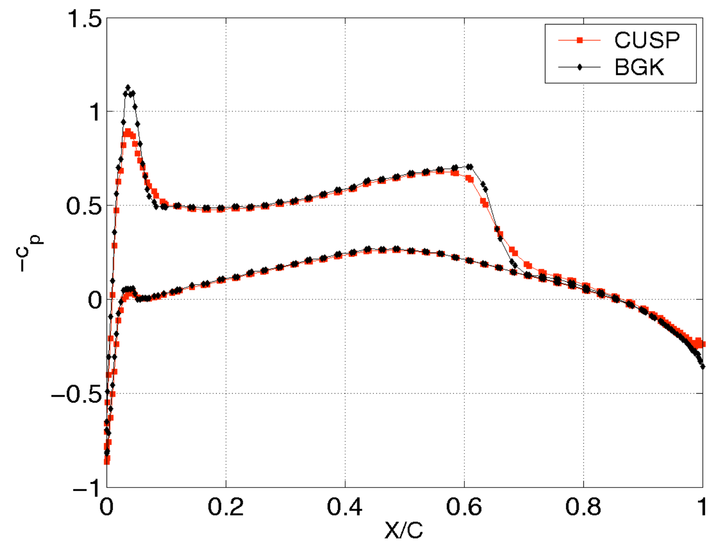


Validation of the BGK Scheme: 3D Inviscid Transonic Flow

Finer Mesh (316k Nodes)



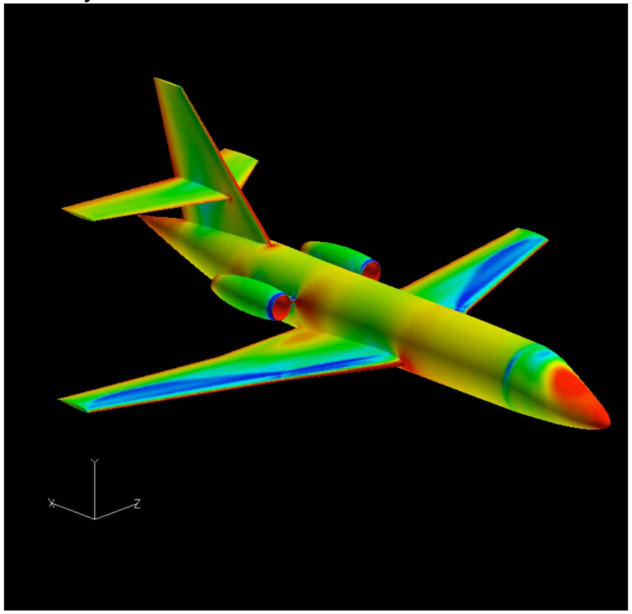
Coarser Mesh (94k Nodes)

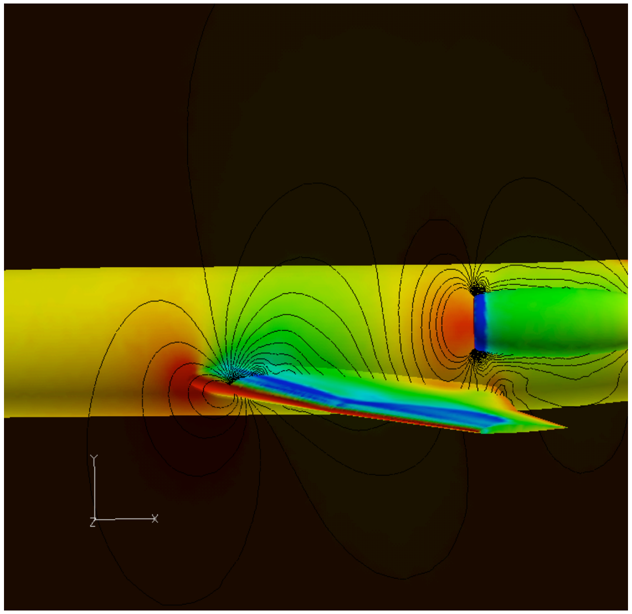


- Onera M6 Wing at M=0.84, / / = 3.06 degrees
- With sufficient resolution CUSP and BGK give similar results
- BGK seems to handle lower-resolution meshes better
- This might allow a reduction in the number of mesh points

Validation of the BGK Scheme using Flo3xx: 3D Inviscid Transonic Flow

Density from 0.625 to 1.1





- Falcon Business Jet
- M = 0.8
- Angle of Attack: 2 degrees

Fast Time Integration Methods for Unsteady Problems

Arathi Gopinath
Matt McMullen
Antony Jameson

Potential Applications

- Flutter Analysis,
- Flow past Helicopter blades,
- Rotor-Stator Combinations in Turbomachinery,
- Zero-Mass Synthetic Jets for Flow Control

Dual Time Stepping BDF

The kth-order accurate backward difference formula (BDF) is of the form

$$D_t = \frac{1}{\Delta t}\sum_{q=1}^{k}\frac{1}{q}(\Delta^{-})^{q} \qquad \text{where} \quad \Delta^{-}w^{n+1} = w^{n+1} - w^{n}$$

The non-linear BDF is solved by inner iterations which advance in pseudo-time t*

The second-order BDF solves

$$\frac{dw}{dt^{*}} + \left[\frac{3w - 4w^{n} + w^{n-1}}{2\Delta t} + R(w)\right] = 0$$

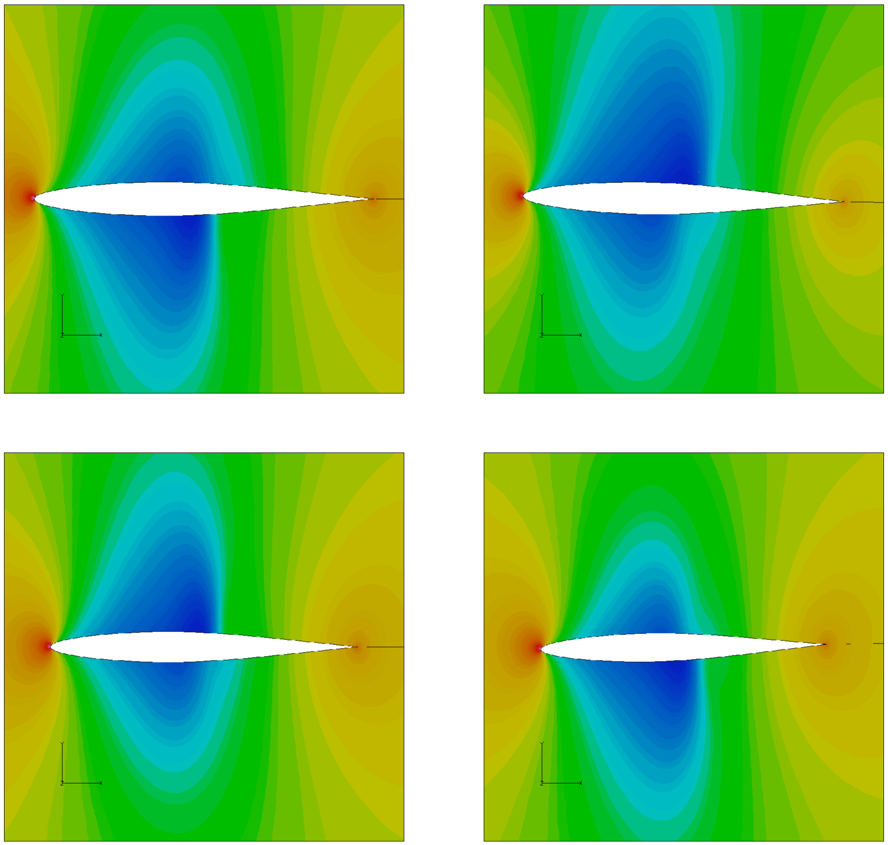
Implementation via

- RK “dual time stepping” scheme with variable local Δt^{*} (RK-BDF)
- Nonlinear SGS “dual time stepping” scheme (SGS-BDF)

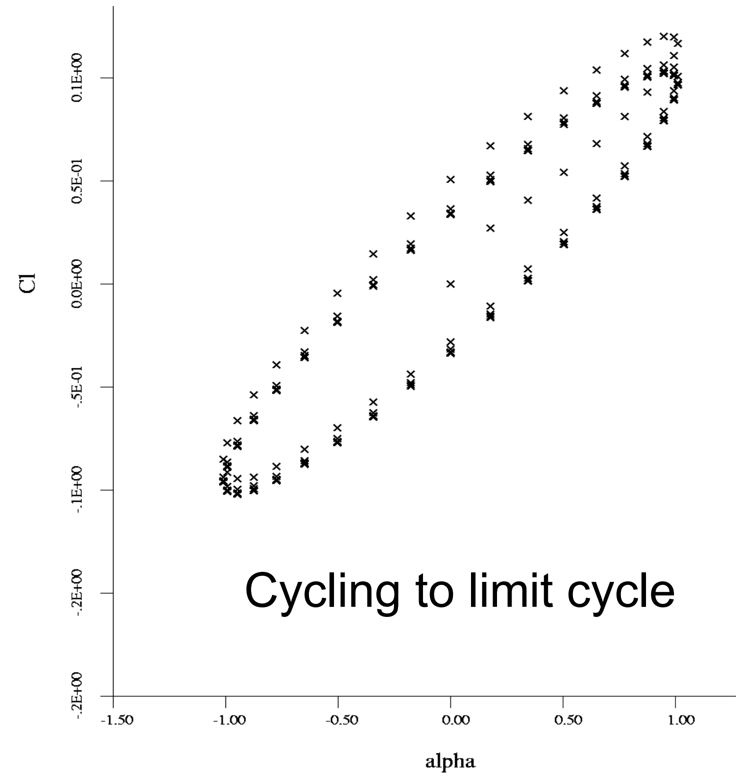
with Multigrid

Test Case: NACA64A010 pitching airfoil (CT6 Case)

Mach Number	0.796
Pitching amplitude	+/- 1.01deg.
Reduced Freq.	0.202
Reynolds Number	12.36 million



Pressure Contours at Various Time Instances (AGARD 702)



Results of SGS-BDF Scheme

(36 time steps per pitching cycle, 3 iterations per time step)

Payoff of Dual-time Stepping BDF Schemes

- Accurate simulations with an order of magnitude reduction in time steps.

- For the pitching airfoil:
  from ~ 1000 to 36 time steps per pitching cycle
  with three sub-iterations in each step.

Frequency Domain and Global Space-Time Multigrid Spectral Methods

Application : Time-periodic flows

Using a Fourier representation in time, the time period T is divided into N steps.

Then,

$$\hat{w}_k = \frac{1}{N} \sum_{n=0}^{N-1} w^n e^{-ikn\Delta t}$$

The discretization operator is given by

$$D_t w^n = \frac{2\pi}{T} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} ik\hat{w}_k e^{ikn\Delta t}$$

Method 1 (McMullen et.al.) : Transform the equations into frequency domain and solve them in pseudo-time t*

$$\frac{d\hat{w}_k}{dt^*} + \frac{2\pi}{T} ik\hat{w}_k + \hat{R}_k = 0$$

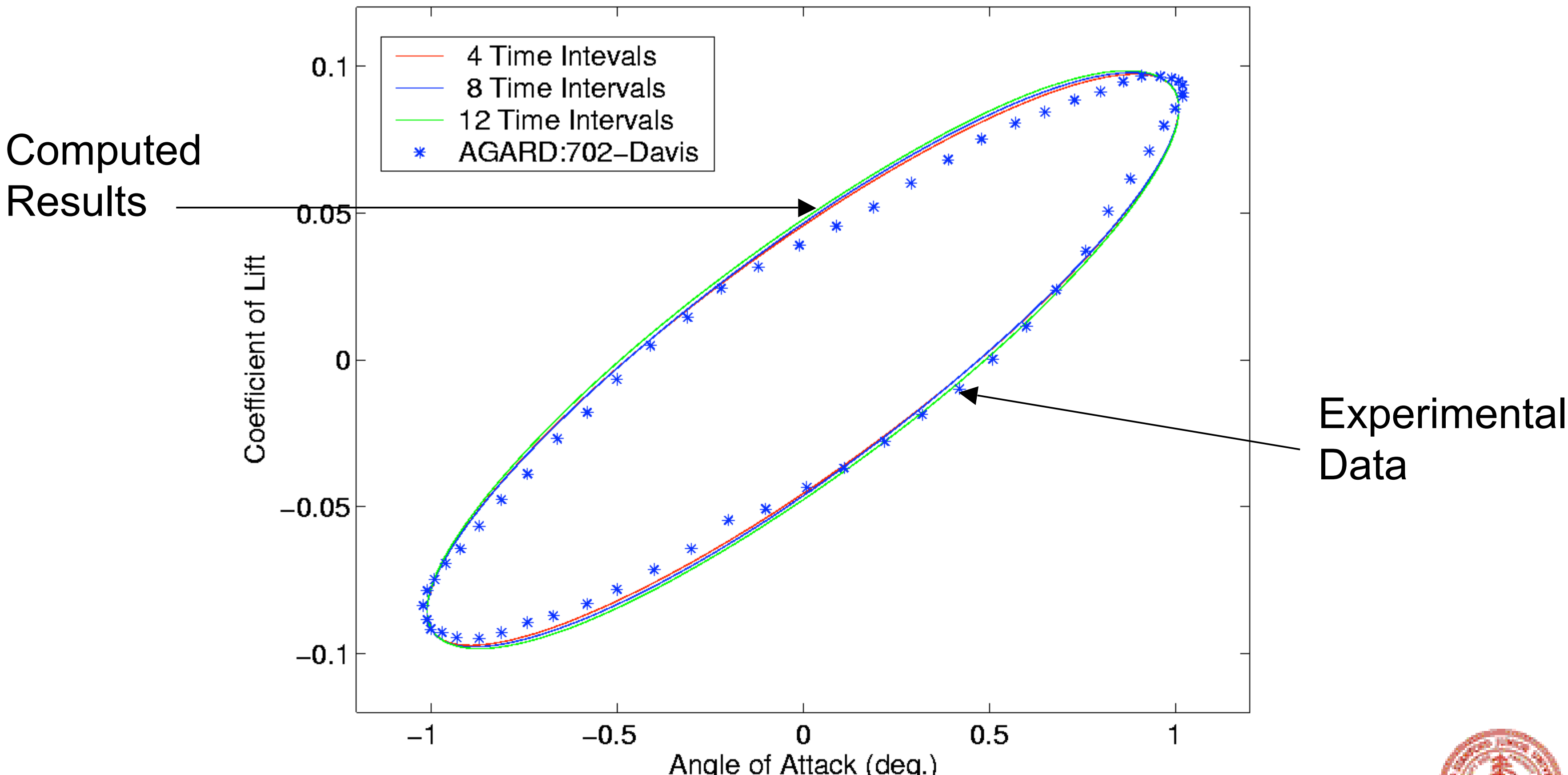
Method 2 (Gopinath et.al.) : Solve the equations in the time-domain. The space-time spectral discretization operator is

$$D_t w^n = \sum_{m=-\frac{N}{2}+1}^{\frac{N}{2}-1} d_m w^{n+m}, \qquad d_m = \frac{2\pi}{T}\frac{1}{2}(-1)^{m+1}\cot\left(\frac{\pi m}{N}\right), m \neq 0$$

This is a central difference operator connecting all time levels, yielding an integrated space-time formulation which requires simultaneous solution of the equations at all time levels.

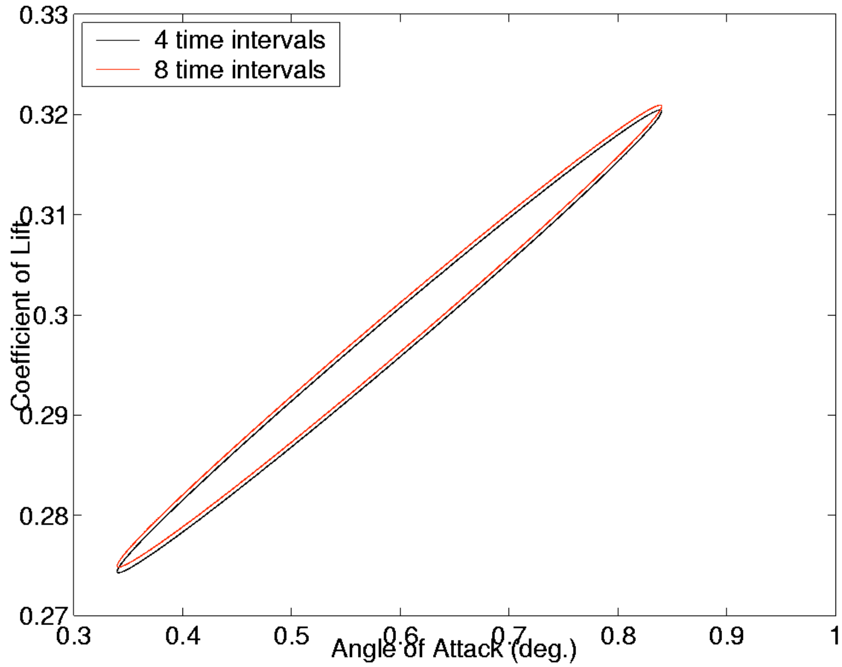
Comparison with Experimental Data - C_L vs. □ (CT6 Case)

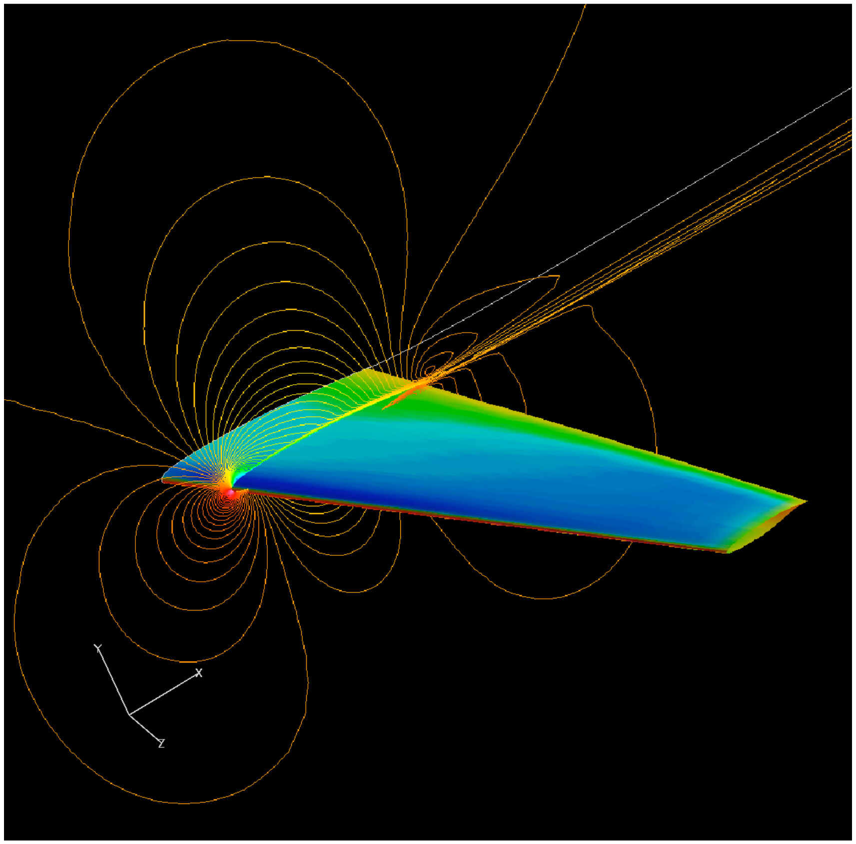
RANS Time-Spectral Solution with 4, 8 and 12 intervals per pitching cycle



3D Test Case
NLR LANN Pitching Wing - RANS

Mach number	0.621
Mean Alpha	0.59deg
Pitching Amplitude	0.25deg
Pitching axis	62% RChord
Red. Frequency	0.133
Reynolds Number	6.28 million



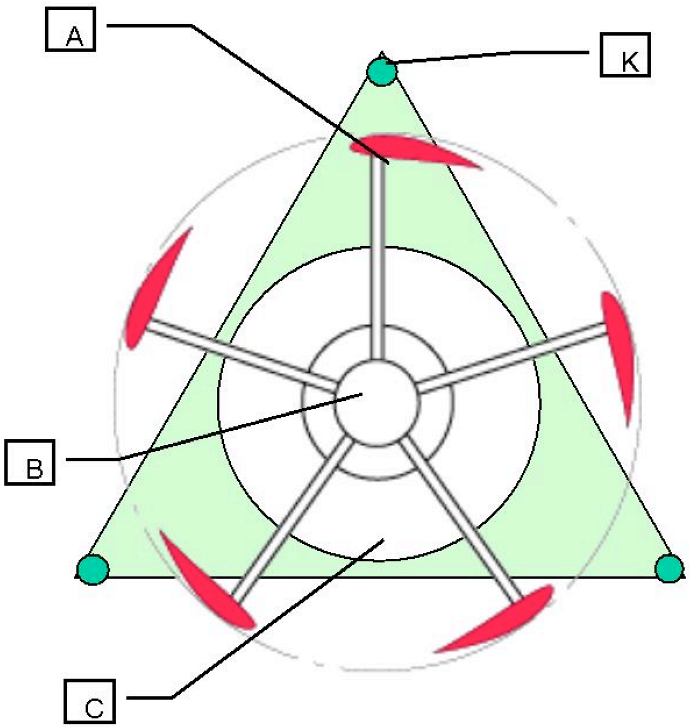


Pressure Contours on the Wing

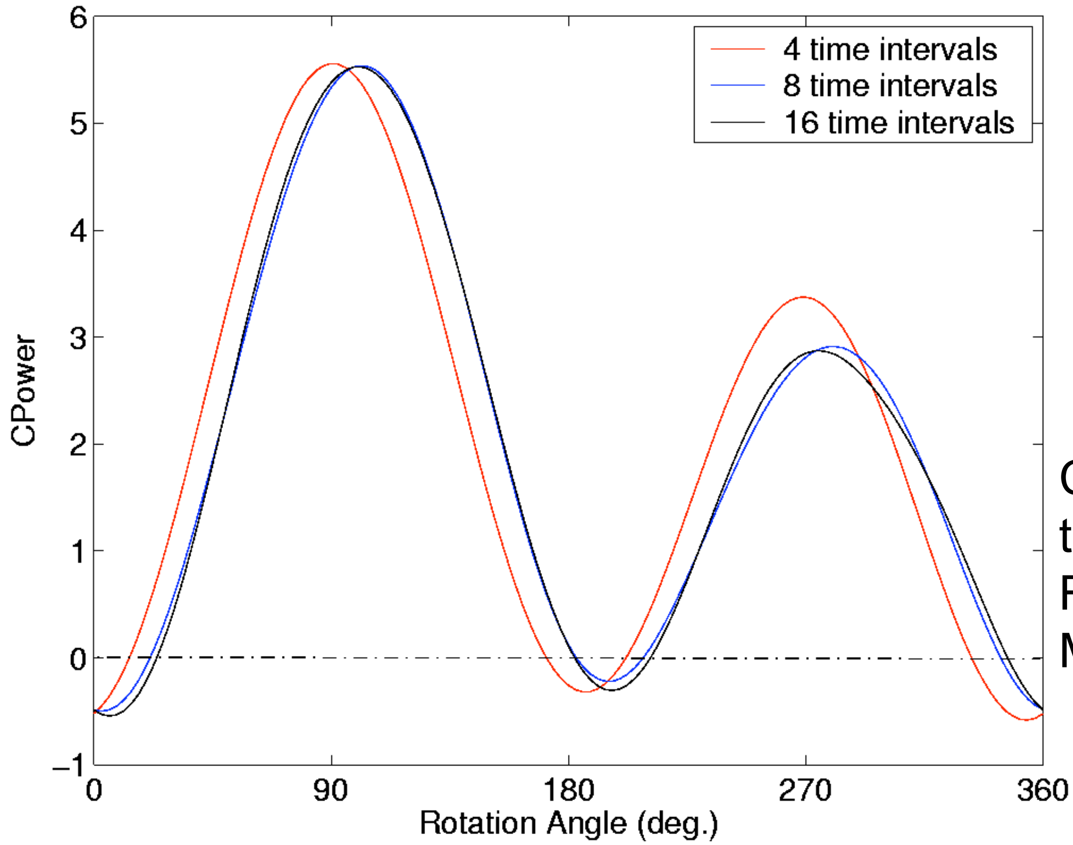
C_L vs. □ plot with 4 and 8 time intervals

Application : Vertical-Axis Wind Turbine(VAWT)

Objective : To maximize power output of the VAWT by turbine blade redesign and various parametric studies.



VAWT : NACA0015 Airfoil
Single-Blade Inviscid 3D model



Free-stream Mach Number	0.1
Blade Tip-Speed / V_inf	5
Turbine Radius / Blade Chord	8

Coefficient of Power generated by the VAWT as a function of Rotation Angle - Time Spectral Method with 4,8 and 16 time intervals

Payoff of Time Spectral Schemes

- Engineering accuracy with very small number of time intervals and same rate of convergence as the BDF.

- Spectral accuracy for sufficiently smooth solutions.

- Periodic solutions directly without the need to evolve through 5-10 cycles, yielding an order of magnitude reduction in computing cost beyond the reduction already achieved with the BDF, for a total of two orders of magnitude.

Filtering the Navier-Stokes Equations with an Invertible Filter

Consider the incompressible Navier--Stokes equations

$$\square\frac{\partial u_i}{\partial t} + \square u_j \frac{\partial u_i}{\partial x_j} + \square\frac{\partial p}{\partial x_i} = \square\frac{\partial^2 u_i}{\partial x_i \partial x_j} \qquad (1)$$

where

$$\frac{\partial u_i}{\partial x_i} = 0$$

In large eddy simulation (LES) the solution is filtered to remove the small scales. Typically one sets

$$\overline{u}_i(x) = \int G(x - x')u(x')dx' \qquad (2)$$

where the kernel *G* is concentrated in a band defined by the filter width. Then the filtered equations contain the extra virtual stress

$$\tau_{ij} = \overline{u_i u_j} - \overline{u}_i \overline{u}_j \qquad (3)$$

because the filtered value of a product is not equal to the product of the filtered values. This stress has to be modeled.

A filter which completely cuts off the small scales or the high frequency components is not invertible. The use, on the other hand, of an invertible filter would allow equation (1) to be directly expressed in terms of the filtered quantities. Thus one can identify desirable properties of a filter as

1. Attenuation of small scales
2. Commutativity with the differential operator
3. Invertibility

Suppose the filter has the form

$$\bar{u}_i = P u_i \qquad (4)$$

which can be inverted as

$$Q \bar{u}_i = u_i \qquad (5)$$

where $Q = P^{-1}$. Moreover Q should be coercieve, so that

$$\|Qu\| > c\|u\| \qquad (6)$$

for some positive constant c.

Note that if Q commutes with $\frac{\partial}{\partial x_i}$ then so does Q^{-1}, since for any quantity f which is sufficiently differentiable

$$\frac{\partial}{\partial x_i}(Q^{-1} f) = Q^{-1} Q \frac{\partial}{\partial x_i}(Q^{-1} f)$$

$$= Q^{-1} \frac{\partial}{\partial x_i}(Q Q^{-1} f)$$

$$= Q^{-1} \frac{\partial}{\partial x_i}(f)$$

Also since Q commutes with $\frac{\partial}{\partial x_i}$,

$$\frac{\partial \bar{u}}{\partial x_i} = 0 \qquad (7)$$

As an example P can be the inverse Helmholtz operator, so that one can write

$$Q\bar{u}_i = \left(1 - \Delta^2 \frac{\partial^2}{\partial x_k \partial x_k}\right)\bar{u}_i = u_i \qquad (8)$$

where Δ is a length scale proportional to the largest scales to be retained. One may also introduce a filtered pressure p, satisfying the equation

$$Q\bar{p} = \left(1 - \Delta^2 \frac{\partial^2}{\partial x_k \partial x_k}\right)\bar{p} = p \qquad (9)$$

Now one can substitute equation (8) and(9) for u_i and p in equation (1) to get

$$\square\frac{\partial}{\partial t}\left[\left(1-\alpha^2\frac{\partial^2}{\partial x_k\partial x_k}\right)\bar{u}_i\right]+\square\left[\left(1-\alpha^2\frac{\partial^2}{\partial x_k\partial x_k}\right)\bar{u}_j\frac{\partial}{\partial x_j}\left(1-\alpha^2\frac{\partial^2}{\partial x_l\partial x_l}\right)\bar{u}_i\right]+\frac{\partial}{\partial x_i}\left(1-\alpha^2\frac{\partial^2}{\partial x_k\partial x_k}\right)\bar{p}$$

$$=\square\frac{\partial^2}{\partial x_j\partial x_j}\left(1-\alpha^2\frac{\partial^2}{\partial x_k\partial x_k}\right)\bar{u}_i$$

Because the order of the differentiations can be interchanged and the Helmholtz operator satisfies condition(6), it can be removed. The product term can be written as

$$\square\frac{\partial}{\partial x_j}\left[\left(1-\alpha^2\frac{\partial^2}{\partial x_k\partial x_k}\right)\bar{u}_i\left(1-\alpha^2\frac{\partial^2}{\partial x_l\partial x_l}\right)\bar{u}_j\right]$$

$$=\square\frac{\partial}{\partial x_j}\left[\bar{u}_i\bar{u}_j-\alpha^2\bar{u}_i\frac{\partial^2\bar{u}_j}{\partial x_k\partial x_k}-\alpha^2\bar{u}_j\frac{\partial^2\bar{u}_i}{\partial x_k\partial x_k}+\alpha^4\frac{\partial^2\bar{u}_i}{\partial x_k\partial x_k}\frac{\partial^2\bar{u}_j}{\partial x_l\partial x_l}\right]$$

$$=\square\frac{\partial}{\partial x_j}\left[\bar{u}_i\bar{u}_j-\alpha^2\frac{\partial^2}{\partial x_k\partial x_k}\left(\bar{u}_i\bar{u}_j\right)+2\alpha^2\frac{\partial\bar{u}_i}{\partial x_k}\frac{\partial\bar{u}_j}{\partial x_k}+\alpha^4\frac{\partial^2\bar{u}_i}{\partial x_k\partial x_k}\frac{\partial^2\bar{u}_j}{\partial x_l\partial x_l}\right]$$

$$=\square Q\frac{\partial}{\partial x_j}\left[\bar{u}_i\bar{u}_j+\alpha^2Q^{-1}\left(2\frac{\partial\bar{u}_i}{\partial x_k}\frac{\partial\bar{u}_j}{\partial x_k}+\alpha^2\frac{\partial^2\bar{u}_i}{\partial x_k\partial x_k}\frac{\partial^2\bar{u}_j}{\partial x_l\partial x_l}\right)\right]$$

According to condition (6), if $Qf=0$ for any sufficiently differentiable quantity f, then $f=0$.

Thus the filtered equation finally reduces to

$$\square\frac{\partial \bar{u}_i}{\partial t} + \square\frac{\partial}{\partial x_j}\left(\bar{u}_i \bar{u}_j\right) + \frac{\partial \bar{p}}{\partial x_i} = \square\frac{\partial^2 \bar{u}_i}{\partial x_k \partial x_k} \square \square\frac{\partial}{\partial x_j}\square_{ij} \tag{10}$$

with the virtual stress

$$\square_{ij} = \square^2 Q^{\square 1}\square 2\frac{\partial \bar{u}_i}{\partial x_k}\frac{\partial \bar{u}_j}{\partial x_k} + \square^2 \frac{\partial^2 \bar{u}_i}{\partial x_k \partial x_k}\frac{\partial^2 \bar{u}_j}{\partial x_l \partial x_l}\square \tag{11}$$

The virtual stress may be calculated by solving

$$\square 1 \square \square^2 \frac{\partial^2}{\partial x_k \partial x_k}\square \square_{ij} = \square^2 \square 2\frac{\partial \bar{u}_i}{\partial x_k}\frac{\partial \bar{u}_j}{\partial x_k} + \square^2 \frac{\partial^2 \bar{u}_i}{\partial x_k \partial x_k}\frac{\partial^2 \bar{u}_j}{\partial x_l \partial x_l}\square \tag{12}$$

Taking the divergence of equation (10), it also follows that $\bar{p}$ satisfies the Poisson equation

$$\frac{\partial^2 \bar{p}}{\partial x_i \partial x_i} + \square\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}\left(\bar{u}_i \bar{u}_j\right) + \square\frac{\partial^2}{\partial x_i \partial x_j}\square_{ij} = 0 \tag{13}$$

In a discrete solution scales smaller than the mesh width would not be resolved, amounting to an implicit cut off. There is the possibility of introducing an explicit cut off off in τ_{ij}. Also one could use equation (8) to restore an estimate of the unfiltered velocity.

In order to avoid solving the Helmholtz equation (12), the inverse Helmholtz operator could be expanded formally as

$$\left(1 - \alpha^2 \Delta\right)^{-1} = 1 + \alpha^2 \Delta + \alpha^4 \Delta^2 + \ldots$$

where Δ denotes the Laplacian $\frac{\partial^2}{\partial x_k \partial x_k}$. Now retaining terms up to the fourth power of α, the approximate virtual stress tensor assumes the form

$$\tau_{ij} = 2\alpha^2 \frac{\partial \bar{u}_i}{\partial x_k} \frac{\partial \bar{u}_j}{\partial x_k} + \alpha^4 \left[2\Delta \left(\frac{\partial \bar{u}_i}{\partial x_k} \frac{\partial \bar{u}_j}{\partial x_k} \right) + \Delta \bar{u}_i \Delta \bar{u}_j \right] \qquad (14)$$

One may regard the forms (11) or (14) as prototypes for subgrid scale (SGS) models.

The inverse Helmholtz operator cuts off the smaller scales quite gradually. One could design filters with a sharper cut off by shaping their frequency response. Denote the Fourier transform of f as

$$\hat{f} = Ff$$

where (in one space dimension)

$$\hat{f}(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-ikx}\, dx$$

$$f(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(k) e^{-ikx}\, dk$$

Then the general form of an invertible filter is

$$F\,\tilde{P}f = S(k)\hat{f}(k)$$

$$F\,\tilde{Q}f = \frac{1}{S(k)}\hat{f}(k)$$

where $S(k)$ should decrease rapidly beyond a cut off wave number inversely proportional to a length scale Δ.

In the case of a general filter with inverse *Q*, the virtual stress follows from the relation

$$Q\overline{u_i u_j} = u_i u_j = Q\bar{u}_i Q\bar{u}_j$$

Then

$$\tau_{ij} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j = Q^{-1}(Q\bar{u}_i Q\bar{u}_j - Q(\bar{u}_i \bar{u}_j))$$

This formula provides the form for a family of subgrid-scale models.