# TIME–INTEGRATION METHODS IN COMPUTATIONAL AERODYNAMICS

**Antony Jameson**

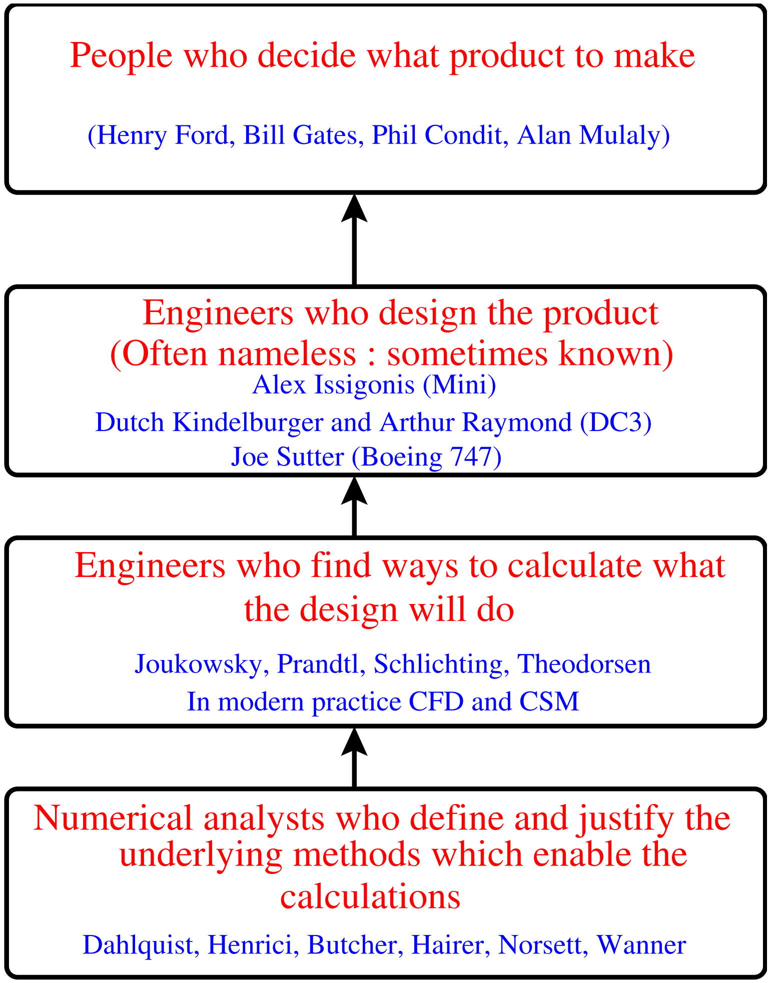
Department of Aeronautics and Astronautics
Stanford University, CA

2003 AFOSR Work Shop on
Advances and Challenges in Time–Integration of PDEs.
August 18, 2003

# From the preface to Gear's book

*"The progression of many numerical methods has been that they originate with the engineer (scientist, astronomer, etc.) who solves his problem that was previously unmanageable, progress through the mathematician-numerical analyst who proves that they do indeed solves the problem under some conditions, and finish with the numerical analyst-computer scientist who concerns himself with their effective implementation and utilization on the computer."*

# Food Chain



People who decide what product to make

(Henry Ford, Bill Gates, Phil Condit, Alan Mulaly)

Engineers who design the product
(Often nameless : sometimes known)

Alex Issigonis (Mini)

Dutch Kindelburger and Arthur Raymond (DC3)

Joe Sutter (Boeing 747)

Engineers who find ways to calculate what the design will do

Joukowsky, Prandtl, Schlichting, Theodorsen

In modern practice CFD and CSM

Numerical analysts who define and justify the underlying methods which enable the calculations

Dahlquist, Henrici, Butcher, Hairer, Norsett, Wanner

# Outline

1. **Introduction**:

   Requirements for the numerical simulation of steady and unsteady flows

2. **Time integration for steady-state problems**

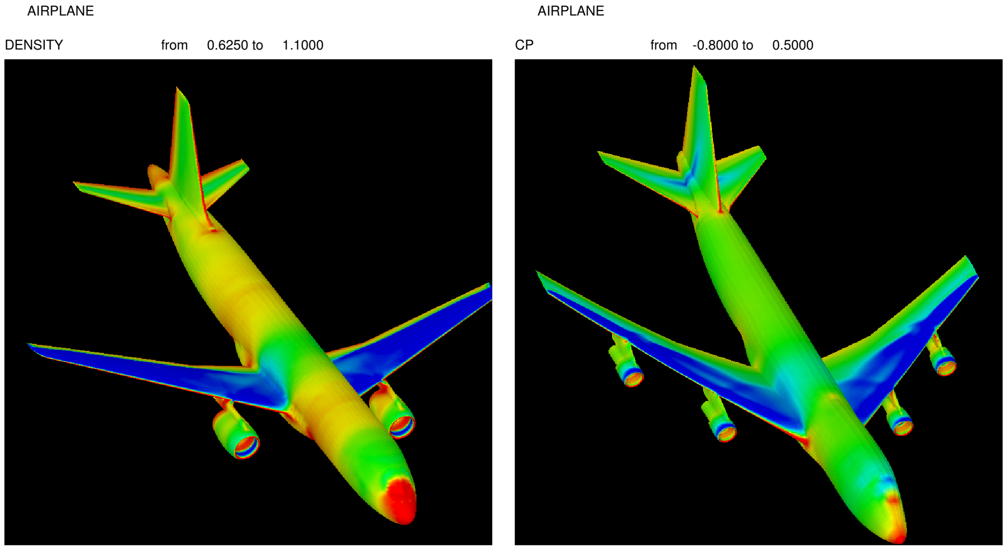
3. **Dual time stepping methods for unsteady problems**

4. **Conclusions**

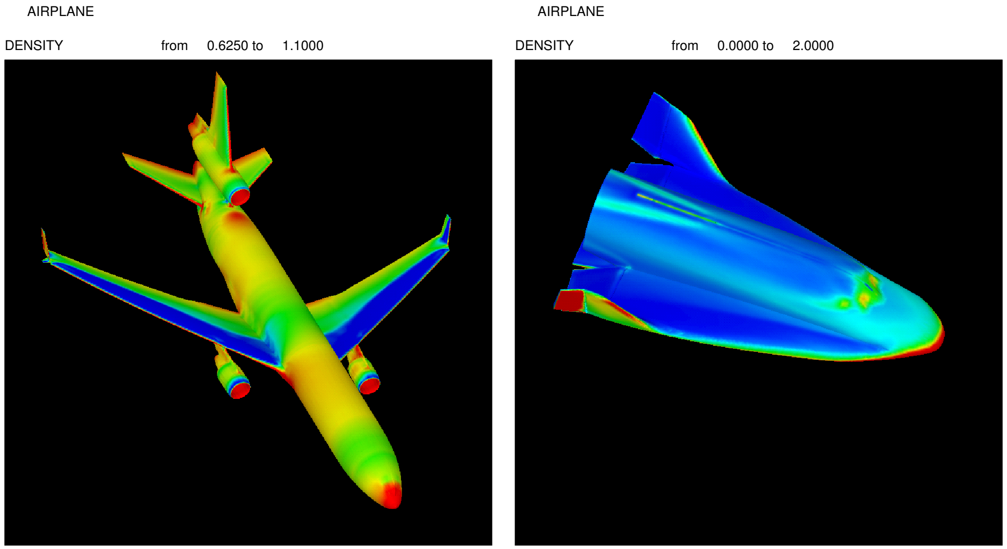
# Some Examples of Aerodynamic Calculations

- Aircraft
  - Steady inviscid flow
  - Viscous flow - possibly unsteady
- Turbomachinery - unsteady turbulent flow

# Aerodynamic Flow computations

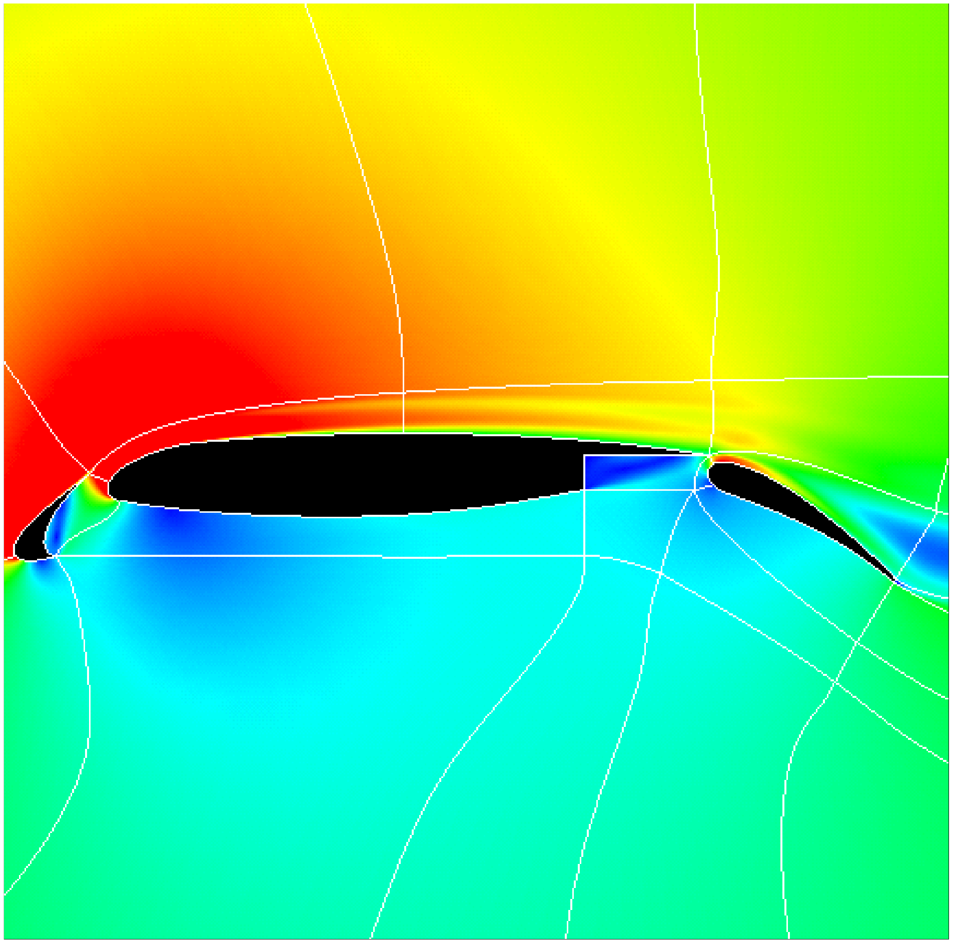


Density and pressure distributions for transonic flow over the A-320 and Boeing 747

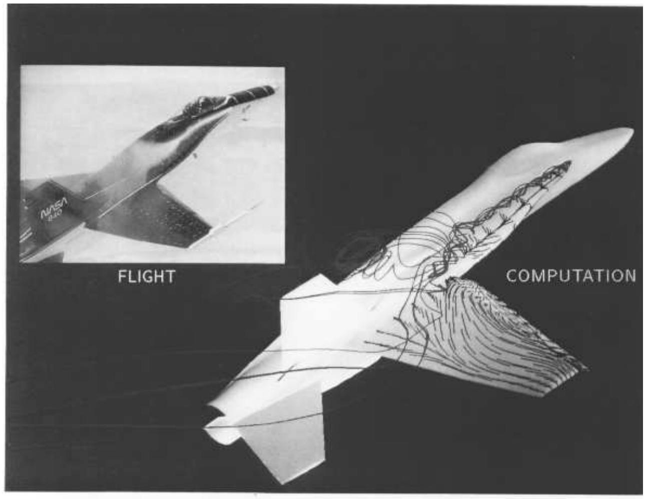


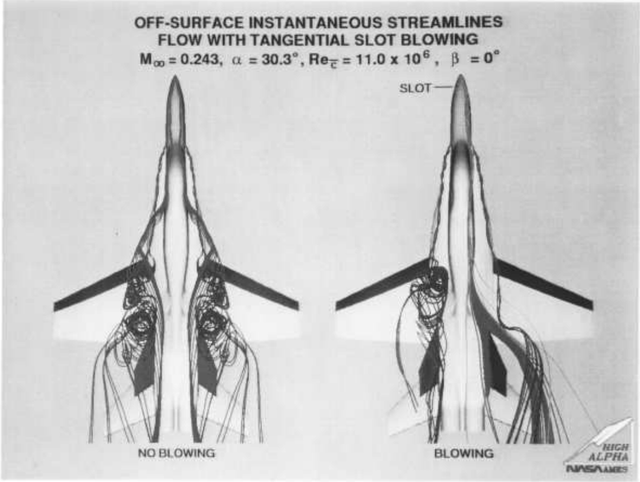
Density contours for transonic and supersonic flow over the MD-11 and Hermes Space Shuttle

# Mach Number Contours for flow around a high-lift Configuration



# Vortical Flow around a F18




OFF-SURFACE INSTANTANEOUS STREAMLINES
FLOW WITH TANGENTIAL SLOT BLOWING
$M_\infty = 0.243,\ \alpha = 30.3^\circ,\ Re_{\bar{c}} = 11.0 \times 10^6,\ \beta = 0^\circ$

# Steady-State Solvers based on Time Evolution

Most existing methods for calculating steady state solutions of the flow equations integrate the time dependent equations

$$\frac{dw}{dt} + R(w) = 0$$

until they reach a steady state. The true time dependent equations reach a steady state very slowly (requiring at least 10000 time steps). Consequently the equations are modified in order to accelerate the evolution to the steady state.

# Acceleration Methods (1)

- Locally varying time steps at the stability limit of the scheme (if it is explicit)
- Preconditioning

$$\frac{dw}{dt} + PR(w) = 0$$

where the preconditioner $P$ is typically chosen to equalize the wave speeds

| | |
|---|---|
| Turkel | 1987 |
| Van Leer, Lee, Roe | 1991 |
| Pierce and Giles | 1996 |

Local time stepping may be regarded as a scalar preconditioner

# Acceleration Methods (2)

- Residual Averaging (Jameson and Baker 1981)

  Replace $R$ by a smoothed residual $\bar{R}$ where (in one dimension)

$$\left(1 - \epsilon \frac{\partial^2}{\partial x^2}\right) \bar{R} = R$$

  This allows the time step of an explicit scheme to be increased (typically by a factor of two)

- Multigrid

| | |
|---|---|
| Federenko | 1964 |
| Brandt | 1977 |
| Ni | 1981 |
| Jameson | 1983 |

  Calculate corrections of the fine grid solution on a sequence of coarser grids. In order to solve

$$L_h v_h - f_h \quad = \quad R_h(v_h) \quad = \quad 0$$

  suppose $v_h$ is an estimate for which $R_h(v_h) \neq 0$. Solve

$$L_{2h} \delta v_{2h} + I_{2h}^{h} R_h(v_h) \quad = \quad 0$$

  and set

$$v_h^{new} \quad = \quad v_h + I_h^{2h} \delta v_{2h}$$

# Steady state Solutions and Implicit Schemes

Consider the semi-discrete system

$$\frac{dw}{dt} + R(w) = 0 \tag{1}$$

where $R(w)$ is the space residual which results from spatial discretization of the flow equations.
Any implicit scheme, for example the backward Euler scheme

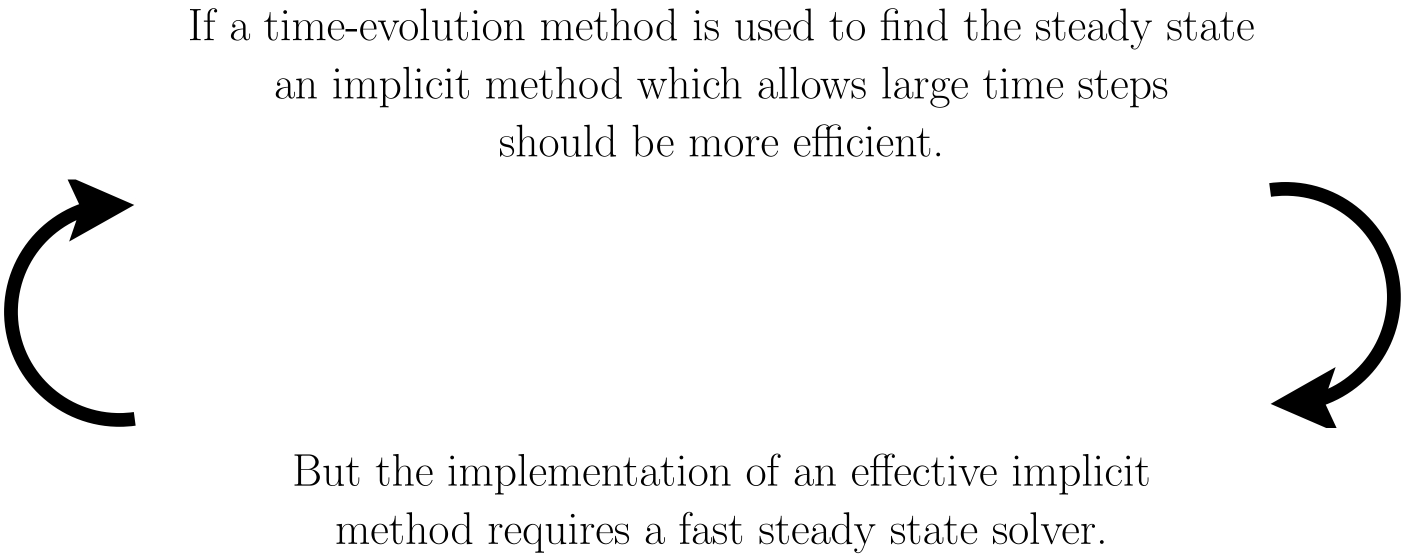
$$w^{n+1} = w^n - \Delta t R(w^{n+1}) \tag{2}$$

requires the solution of a very large number of coupled nonlinear equations which have the same complexity as the steady state problem

$$R(w) = 0 \tag{3}$$

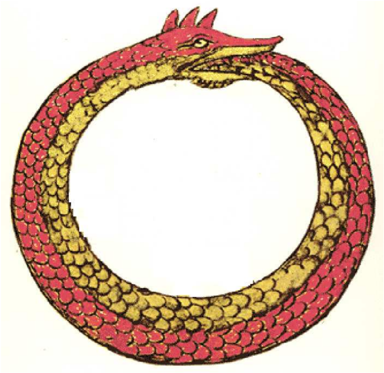
Accordingly a fast steady state solver is an essential building block for an implicit scheme.

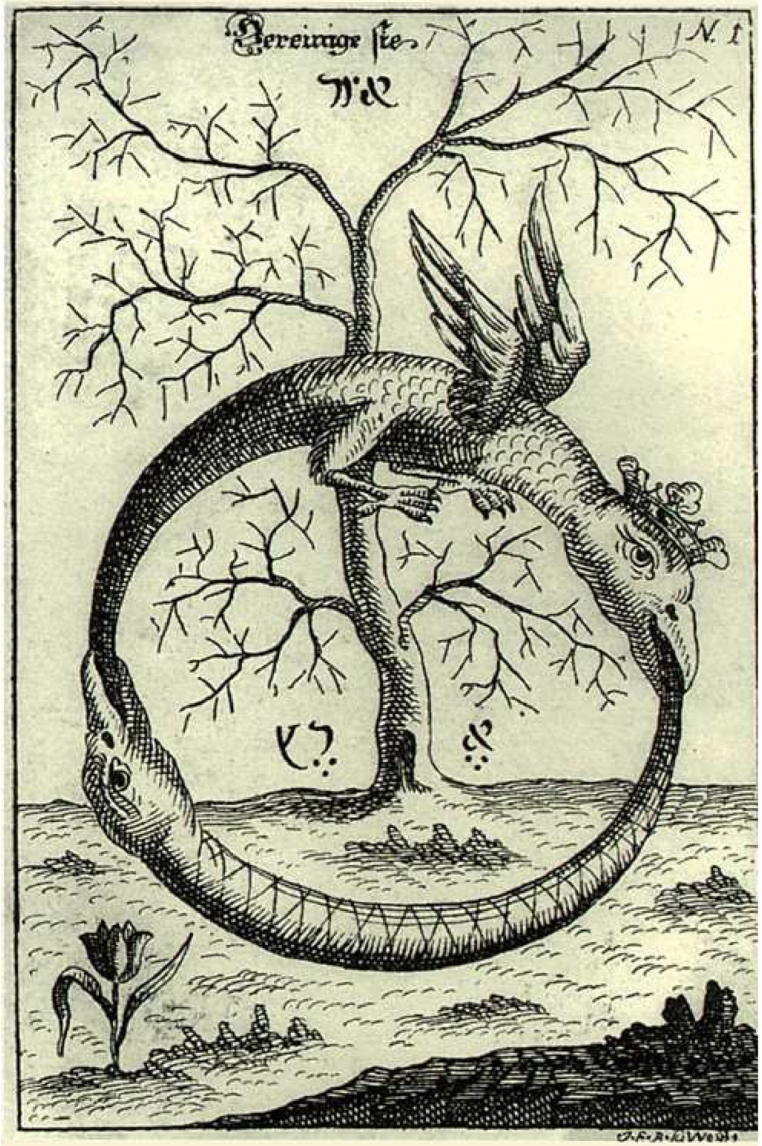
## Paradox



If a time-evolution method is used to find the steady state an implicit method which allows large time steps should be more efficient.

But the implementation of an effective implicit method requires a fast steady state solver.




Vereinige ſte
N. 1

# Dual time Stepping (Jameson 1991)

The idea is to use an implicit scheme with a large stability region (A-stable or stiffly stable) and to solver the implicit equations at each time step by inner iterations using an accelerated time evolution scheme in artificial time. The second order BDF is

$$\frac{3}{2\Delta t}w^{n+1} - \frac{2}{\Delta t}w^{n} + \frac{}{2\Delta t}w^{n-1} + R(w^{n+1}) = 0 \tag{4}$$

With dual time stepping solve

$$\frac{dw}{dt^*} + \frac{3}{2\Delta t}w - \frac{2}{\Delta t}w^{n} + \frac{1}{2\Delta t}w^{n-1} + R(w) = 0 \tag{5}$$

in pseudo time $t^*$ to reach a steady state satisfying equation (4).

# Time Integration for Steady State Problems

1. Multi–stage single step schemes
2. Multigrid time–stepping schemes
3. Implicit schemes
4. Symmetric Gauss–Seidel (SGS) schemes

# Multi-stage Schemes for Steady State Calculations

When the time stepping scheme is used to reach a steady state its order of accuracy is immaterial (it could even be inconsistent).
Accordingly it can be optimized

- to minimize the computational complexity and memory requirements
- to maximize the stability region

# Model Problem for Stability Analysis

Consider a semi-discretization of the linear advection equation

$$\frac{\partial v}{\partial t} + a\frac{\partial v}{\partial x} = 0$$

with central differences and third order artificial diffusion $\sim \Delta t^3 \frac{\partial^3 u}{\partial x^3}$

$$\Delta t \frac{du_j}{dt} = \frac{\lambda}{2}\left(u_{j+1} - u_{j-1}\right) + \lambda\left(u_{j+2} - 4u_{j+1} + 6u_j - 4u_{j-1} + u_{j-2}\right)$$

where $\lambda$ is the CFL number

$$\lambda = \frac{a\Delta t}{\Delta x}$$

For a Fourier mode $u(x,t) = \hat{u}(t)e^{ipx}$

$$\Delta t \frac{du}{dt} = z\hat{u}$$

where $z$ is the Fourier symbol of the space discretization. With $\xi = p\Delta x$

$$z = -\lambda\left\{isin\xi + 4\mu(1 - cos\xi)^2\right\}$$

The permissible CFL number thus depends on the stability interval along the imaginary axis as well as the negative real axis.

## Simplified Multi-Stage scheme (minimum complexity and memory)

$$\begin{aligned}
\mathbf{w^{(0)}} &= \mathbf{w^n} && (6)\\
\mathbf{w^{(1)}} &= \mathbf{w^{(0)}} - \alpha_1 \Delta t \mathbf{R}(\mathbf{w^{(0)}})\\
\ldots\\
\mathbf{w^{(m-1)}} &= \mathbf{w^{(0)}} - \alpha_{m-1} \Delta t \mathbf{R}(\mathbf{w^{(m-2)}})\\
\mathbf{w^{(m)}} &= \mathbf{w^{(0)}} - \Delta t \mathbf{R}(\mathbf{w^{(m-1)}}) && (7)\\
\mathbf{w^{n+1}} &= \mathbf{w^{(m)}}
\end{aligned}$$

The coefficients

$$\begin{aligned}
\alpha_1 &= \frac{1}{m}\\
\alpha_2 &= \frac{1}{m-1}\\
...\\
\alpha_{m-1} &= \frac{1}{2}\\
\alpha_m &= 1
\end{aligned}$$

give $m^{th}$ order accuracy for a linear problem.

They do not maximize the stability interval along the imaginary axis (CFL number).

# Schemes with maximal intervals of stability along the imaginary axis

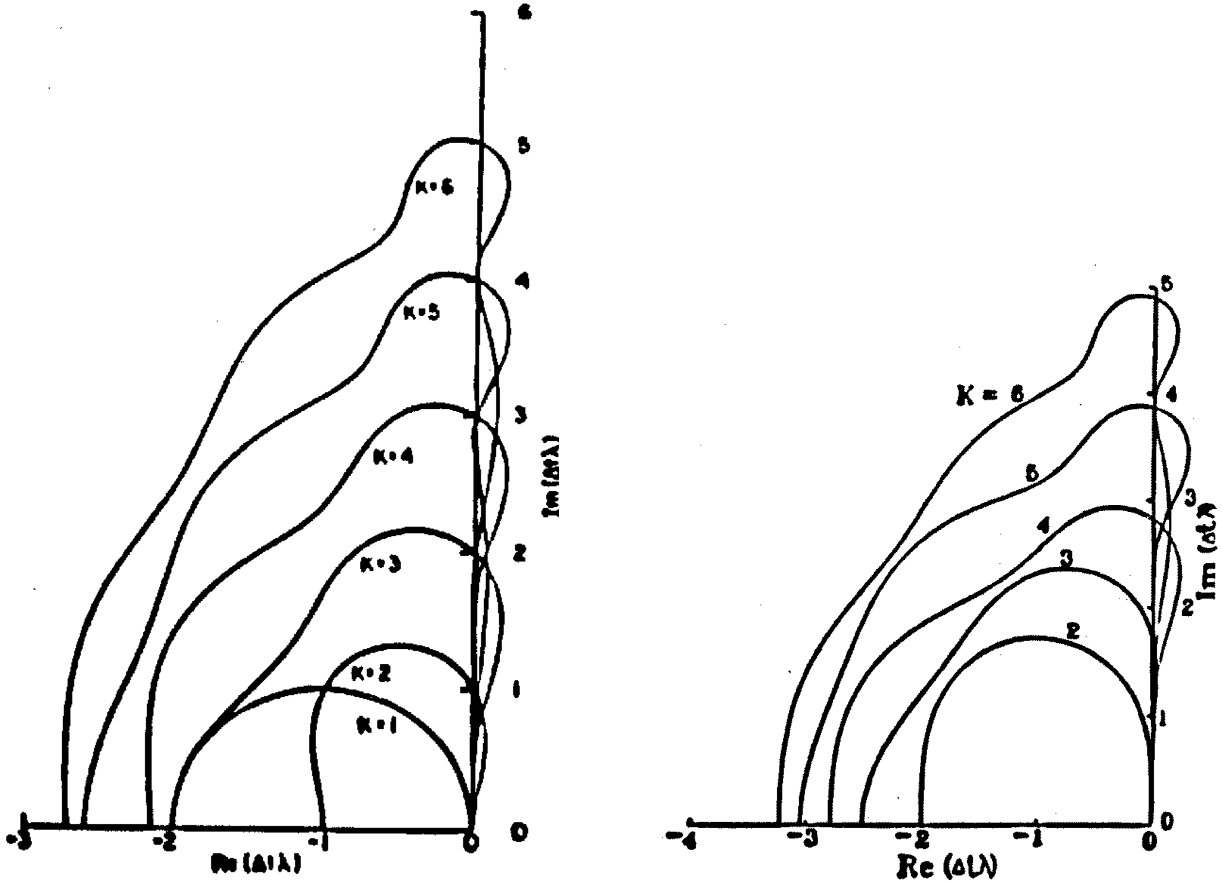


Table 1: I.P.E. Kinmark, *One Step Integration with Large Stability Limits for Hyperbolic Partial Differential Equations*

# Modified Runge Kutta schemes with enhanced stability region

To achieve large stability intervals along both axes it pays to treat the convective and dissipative terms in a distinct fashion (Jameson 1985, 1986, Martinelli 1987).

Accordingly the residual is split as

$$\mathbf{R}(\mathbf{w}) = \mathbf{Q}(\mathbf{w}) + \mathbf{D}(\mathbf{w}),$$

where $Q(w)$ is the convective part and $D(w)$ the dissipative part. Denote the time level $n\Delta t$ by a superscript $n$. Then the multistage time stepping scheme is formulated as

$$\begin{aligned}
\mathbf{w}^{(\mathbf{0})} &= \mathbf{w}^{\mathbf{n}} \\
\mathbf{w}^{(\mathbf{1})} &= \mathbf{w}^{\mathbf{0}} - \alpha_1 \Delta t \left(\mathbf{Q}^{(\mathbf{0})} + \mathbf{D}^{(\mathbf{0})}\right) \\
\mathbf{w}^{(\mathbf{2})} &= \mathbf{w}^{\mathbf{0}} - \alpha_2 \Delta t \left(\mathbf{Q}^{(\mathbf{1})} + \mathbf{D}^{(\mathbf{1})}\right) \\
&\dots \\
\mathbf{w}^{(\mathbf{k})} &= \mathbf{w}^{\mathbf{0}} - \alpha_k \Delta t \left(\mathbf{Q}^{(\mathbf{k-1})} + \mathbf{D}^{(\mathbf{k-1})}\right) \\
&\dots \\
\mathbf{w}^{\mathbf{n+1}} &= \mathbf{w}^{(\mathbf{m})},
\end{aligned}$$

where the superscript $k$ denotes the $k$-th stage, $\alpha_m = 1$, and

$$\begin{aligned}
\mathbf{Q}^{(\mathbf{0})} &= \mathbf{Q}\left(\mathbf{w}^{\mathbf{0}}\right), \quad \mathbf{D}^{(\mathbf{0})} = \beta_1 \mathbf{D}\left(\mathbf{w}^{\mathbf{0}}\right) \\
&\dots \\
\mathbf{Q}^{(\mathbf{k})} &= \mathbf{Q}\left(\mathbf{w}^{(\mathbf{k})}\right) \\
\mathbf{D}^{(\mathbf{k})} &= \beta_{k+1} \mathbf{D}\left(\mathbf{w}^{(\mathbf{k})}\right) + (1 - \beta_{k+1}) \mathbf{D}^{(k-1)}.
\end{aligned}$$

- The coefficients $\alpha_k$ are chosen to maximize the stability interval along the imaginary axis, and the coefficients $\beta_k$ are chosen to increase the stability interval along the negative real axis.
- These schemes do not fall within the standard framework of Runge-Kutta schemes, and they have much larger stability regions.
- Two particularly effective schemes are:

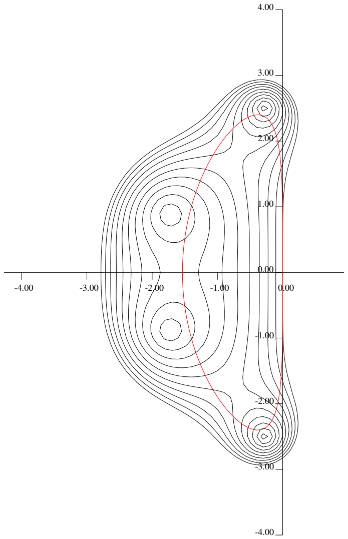
**4-2 scheme**

$$\begin{array}{ll} \alpha_1 = \frac{1}{3} & \beta_1 = 1.00 \\ \alpha_2 = \frac{4}{15} & \beta_2 = 0.50 \\ \alpha_3 = \frac{5}{9} & \beta_3 = 0.00 \\ \alpha_4 = 1 & \beta_4 = 0.00 \end{array} \tag{8}$$

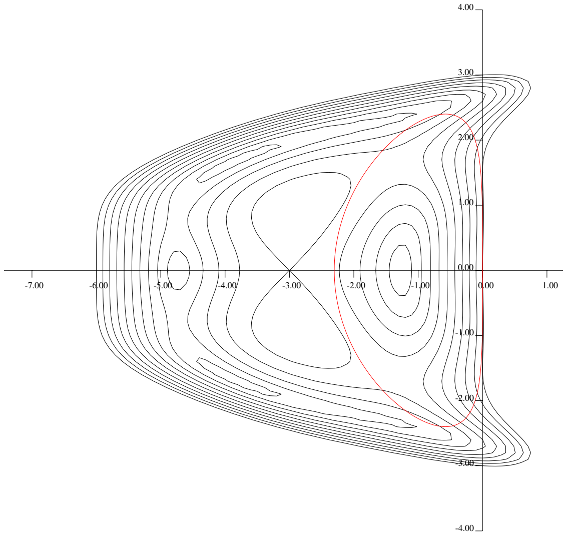
**5-3 scheme**

$$\begin{array}{ll} \alpha_1 = \frac{1}{4} & \beta_1 = 1.00 \\ \alpha_2 = \frac{1}{6} & \beta_2 = 0.00 \\ \alpha_3 = \frac{3}{8} & \beta_3 = 0.56 \\ \alpha_4 = \frac{1}{2} & \beta_4 = 0.00 \\ \alpha_5 = 1 & \beta_5 = 0.44 \end{array} \tag{9}$$

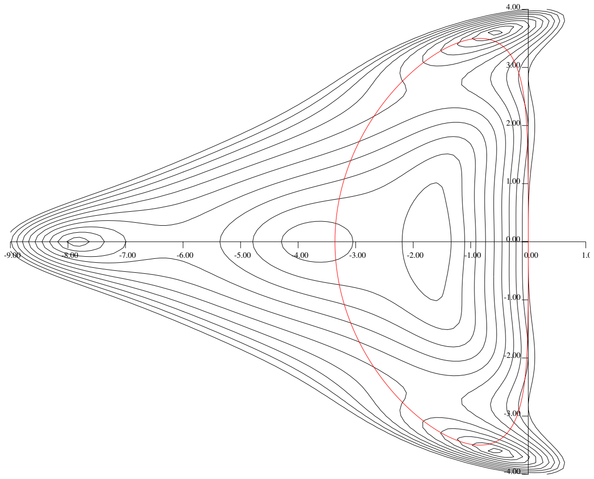
- The figures on the next slide display the stability regions for the standard fourth order RK4 scheme and the 4-2 and 5-3 schemes. The expansion of the stability region is apparent.
- The modified schemes have proved to be particularly effective in conjunction with multigrid.



CFL 2.4   DIS .04
C1 .25    C2 .3333  C3 .5     C4 1.
NSTAGE 4  NDISSP 4



CFL 2.4   DIS .06
C1 1./3.  C2 4./15. C3 5./9.  C4 1.
NSTAGE 4  NDISSP 2



CFL 3.5   DIS .06
C1 .25    C2 1/6    C3 .375   C4 0.5    C5 1.
NSTAGE 5  NDISSP 3

# Multigrid Time Stepping Schemes (Jameson 1983)

The underlying idea of a multigrid time stepping scheme is to transfer some of the task of tracking the evolution of the system to a sequence of successively coarser meshes.

This has two advantages.

1. The computational effort per time step is reduced on a coarser mesh.
2. The use of larger control volumes on the coarser grids tracks the evolution on a larger scale, with the consequence that global equilibrium can be more rapidly attained.

In the case of an explicit time stepping scheme, this manifests itself through the possibility of using successively **larger time steps** as the process passes to the **coarser grids**, without violating the **stability bound**.

# Multigrid Time-Stepping

In the case of a cell-centered scheme on an structured mesh, coarse grids can be formed by agglomerating fine grid cells in groups of four (eight in three dimensions).

Suppose that successively coarser auxiliary grids are introduced, with the grids numbered from 1 to $m$, where grid 1 is the original mesh.

Several transfer operations need to be defined. First the solution vector on grid $k$ must be initialized as

$$\mathbf{w_k^{(0)}} = T_{k,k-1}\mathbf{w_{k-1}} \tag{10}$$

where $\mathbf{w_{k-1}}$ is the current value on grid $k-1$, and $T_{k,k-1}$ is a transfer operator. Next it is necessary to transfer a residual forcing function such that the solution on grid $k$ is driven by the residuals calculated on grid $k-1$.

This can be accomplished by setting

$$\mathbf{P_k} = Q_{k,k-1}\mathbf{R_{k-1}}(\mathbf{w_{k-1}}) - \mathbf{R_k}(\mathbf{w_k^{(0)}}) \tag{11}$$

where $Q_{k,k-1}$ is another transfer operator. Then $\mathbf{R_k}(\mathbf{w_k})$ is replaced by $\mathbf{R_k}(\mathbf{w_k}) + \mathbf{P_k}$ in the time stepping scheme.

## Multigrid Time-Stepping(contd.)

For example, the multi-stage scheme defined by equation (6) is reformulated as

$$\begin{aligned}
\mathbf{w_k^{(1)}} &= \mathbf{w_k^{(0)}} - \alpha_1 \Delta t_k \left(\mathbf{R_k}(\mathbf{w_k^{(0)}}) + \mathbf{P_k}\right) && (12)\\
\cdots & && (13)\\
\mathbf{w_k^{(q+1)}} &= \mathbf{w_k^{(0)}} - \alpha_{q+1} \Delta t_k \left(\mathbf{R_k}(\mathbf{w_k^{(q)}}) + \mathbf{P_k}\right) && (14)\\
\cdots & && (15)
\end{aligned}$$

The result $\mathbf{w_k^{(m)}}$ then provides the initial data for grid $k+1$. Finally the accumulated correction on grid $k$ has to be transferred back to grid $k-1$. Let $w_k^+$ be the final value of $w_k$ resulting from both the correction calculated in the time step on grid $k$ and the correction transferred from grid $k+1$. Then one sets

$$\mathbf{w_{k-1}^+} = \mathbf{w_{k-1}} + I_{k-1,k}\left(\mathbf{w_k^+} - \mathbf{w_k^0}\right) \tag{16}$$

where $w_{k-1}$ is the solution on grid $k-1$ after the time step on grid $k-1$ and before the transfer from grid $k$, and $I_{k-1,k}$ is an interpolation operator.

- In this formulation the residuals on each mesh should be re-evaluated after the time step to provide a proper estimate of the current value $\mathbf{R_k}(\mathbf{w_k^+})$ for transfer to the next mesh $k+1$ in the sequence.
- In the first stage $\mathbf{R_k}(\mathbf{w_k^{(0)}}) + \mathbf{P_k} = \mathbf{Q_{k,k-1}}\mathbf{R_{k-1}}(\mathbf{w_{k-1}})$ so the coarser grid is driven by the fine grid error

# Agglomeration Multigrid

The solution transfer operator $T_{k,k-1}$ is defined by the rule

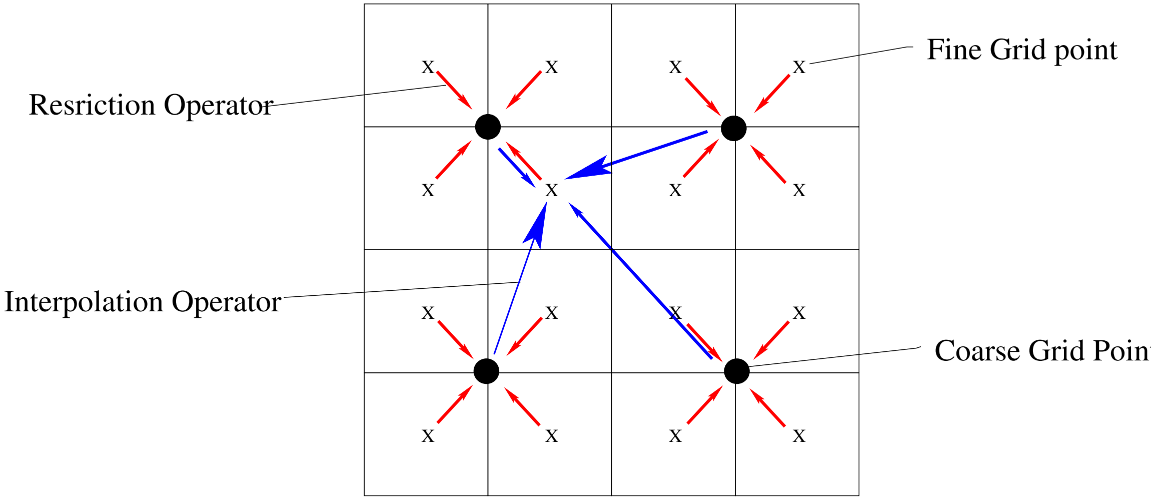
$$T_{k,k-1}\mathbf{w_{k-1}} = \frac{\sum V_{k-1}\mathbf{w_{k-1}}}{V_k} \tag{17}$$

where the sum is over the constituent cells on grid $k-1$, and $V$ is the cell area or volume. This rule conserves mass, momentum and energy.

The residual transferred to grid $k$ is the sum of the residuals in the constituent cells

$$Q_{k,k-1}\mathbf{R_{k-1}} = \sum \mathbf{R_{k-1}} \tag{18}$$

The corrections are transferred up using either bilinear or trilinear interpolation for the operator $I_{k-1,k}$.



## Multigrid V (sawtooth) cycle

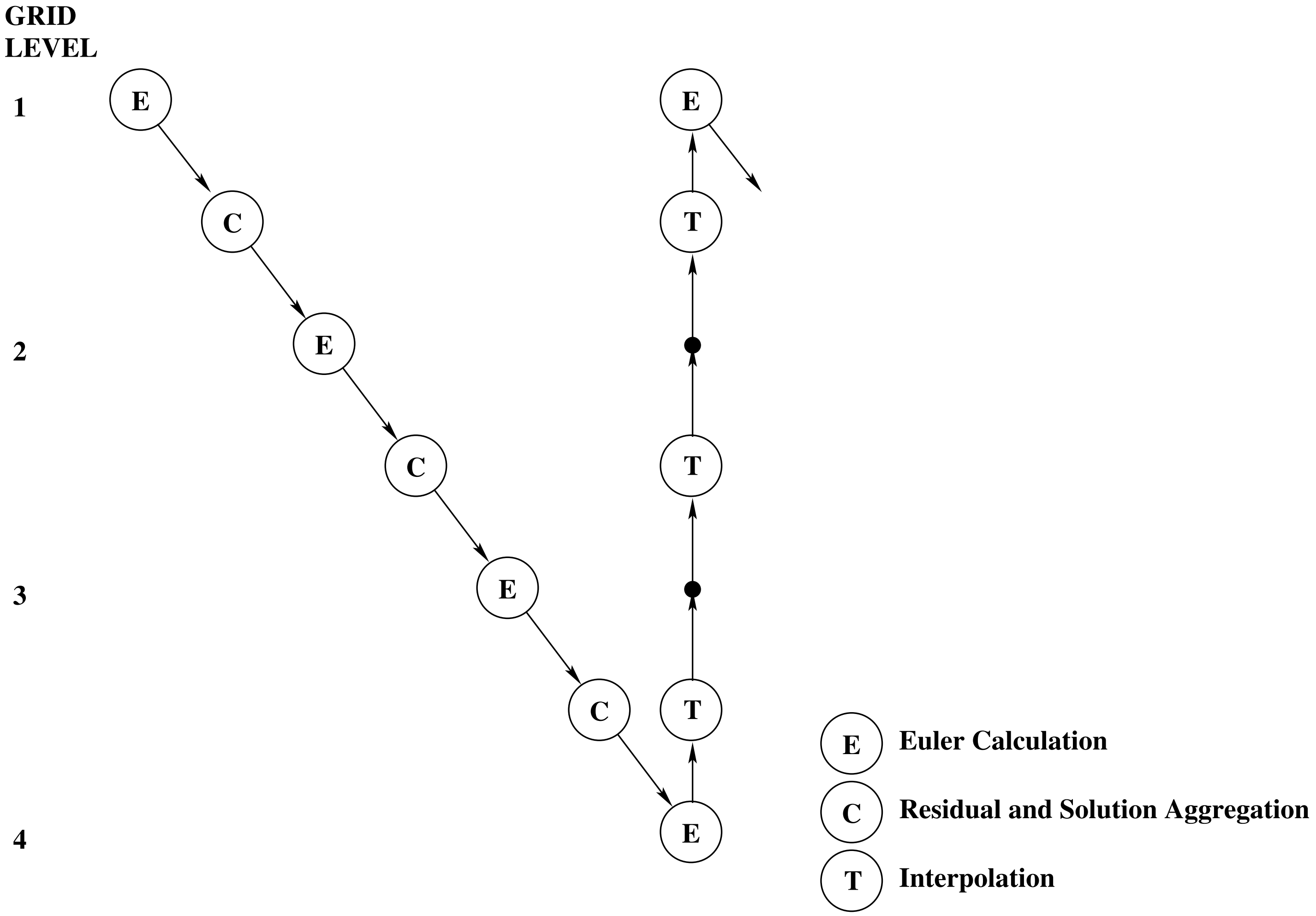
The simplest multigrid cycle is a $V$ (saw-tooth) cycle in which each time step is performed at each grid level on the way down, and the corrections are interpolated with no further steps. Ignoring the overhead cost of the transfer operators the computational effort is

$$\begin{aligned} 1+\frac{1}{4}+\frac{1}{16}+... \ &\leq \ \frac{4}{3} \quad \text{in two dimensions} \\ 1+\frac{1}{8}+\frac{1}{64}+... \ &\leq \ \frac{8}{7} \quad \text{in three dimensions} \end{aligned} \tag{19}$$

With an explicit scheme the time step is doubled on each coarser mesh. In an ideal case, the effective step of the cycle is

$$\Delta t + 2\Delta t + 4\Delta t + ....$$

# Multigrid V (sawtooth) cycle



E Euler Calculation

C Residual and Solution Aggregation

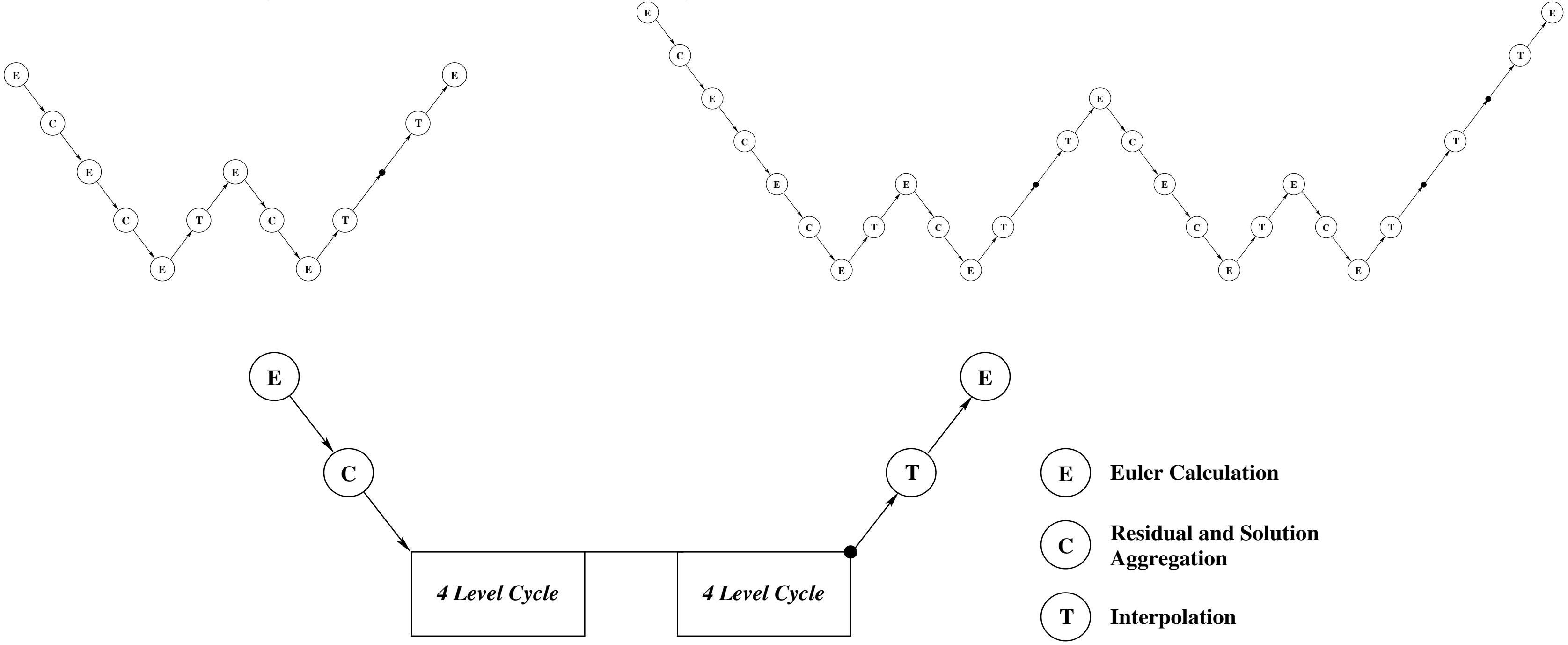
T Interpolation

# Multigrid W Cycle

A $W$-cycle of the type illustrated below proves to be a particularly effective strategy for managing the work split between the meshes. In a three-dimensional case the number of cells is reduced by a factor of eight on each coarser grid. On examination of the figure, it can therefore be seen that the work measured in units corresponding to a step on the fine grid is of the order of

$$1 + 2/8 + 4/64 + \ldots < 4/3, \qquad \text{in three dimension}$$

and consequently the very large effective time step of the complete cycle costs only slightly more than a single time step in the fine grid.



# Prototype Implicit Scheme

In order to discretize

$$\frac{dw}{dt} + R(w) = 0$$

estimate $\frac{dw}{dt}$ at $t + \mu\Delta t$ to obtain

$$w^{n+1} = w^n - \Delta t \left\{ (1-\mu)R(w^n) + \mu R(w^{n+1}) \right\} \tag{20}$$

Usually

$$\begin{aligned} \mu &= 1 && \textbf{(Backward Euler)} \text{ for steady state calculations, and} \\ \mu &= 1/2 && \textbf{(Trapezoidal rule)} \text{ for second order accuracy} \end{aligned}$$

The main possibilities to solve equation (20) at each time step are

- **linearization**
- **iterative solution**

# Linearized Implicit Scheme

Consider the **two dimensional conservation law**

$$\frac{\partial w}{\partial t} + \frac{\partial}{\partial x} f(w) + \frac{\partial}{\partial y} g(w) = 0 \tag{21}$$

Here the **space residual** is

$$R(w) = D_x f(w) + D_y g(w) \tag{22}$$

where $D_x$ and $D_y$ are the discrete operators approximating $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$.
Defining the correction vector

$$\delta w = w^{n+1} - w^n$$

the implicit scheme can be linearized by setting

$$\begin{aligned} f(w^{n+1}) &= f(w^n) + A\Delta w^n + \mathcal{O}\,||\Delta w||^2 \\ g(w^{n+1}) &= g(w^n) + B\Delta w^n + \mathcal{O}\,||\Delta w||^2 \end{aligned}$$

where $A$ and $B$ are the **Jacobian matrices**

$$A = \frac{\partial f(w)}{\partial w}, B = \frac{\partial g(w)}{\partial w}$$

This leads to the **linearized implicit scheme**

$$\left\{ I + \mu \Delta t (D_x A + D_y B) \right\} \delta \mathbf{w} + \Delta t R(w^n) = 0 \tag{23}$$

## Advantages of the linearized Implicit Scheme

- We can recognize $D_x A + D_y B = \frac{\partial R}{\partial w}$.
  With $\mu = 1$ and $\Delta t = \longrightarrow \infty$ the scheme reduces to a **Newton iteration**
- The linearization error is
  $$\mathcal{O} \left|\left|\Delta w\right|\right|^2 = \mathcal{O}(\Delta t)^2$$
  Hence with $\mu = 1/2$ both the discretization and linearization error are $\mathcal{O}((\Delta t)^2$

## Disadvantages of the linearized Implicit Scheme

- In the three dimensional case with, say, an $N \times N \times N$ mesh, the bandwidth is $B = N^2$, with a corresponding inversion cost of the order $N^3 B^2 = N^7$.

  This is prohibitive.
- The linearization prevents the extension to a scheme with third or higher order accuracy.

# Factorized Implicit Schemes

The main possibilities for approximate factorization are the alternating direction implicit (ADI) method (Mitchell and Gourlay, 1966, Beam and Warming, 1976) and the LU implicit scheme (Jameson and Turkel, 1981).

The ADI method replaces the linearized implicit scheme by

$$(I + \mu \Delta t D_x A)(I + \mu \Delta t D_y B)\delta w + \Delta t R(w) = 0 \tag{24}$$

This may be solved in two steps:

$$\begin{aligned}(I + \mu \Delta t D_x A)\delta w^* &= -\Delta t R(w) && (25)\\ (I + \mu \Delta t D_y B)\delta w &= \delta w^* && (26)\end{aligned}$$

## Advantages of the ADI scheme

- Nominally second order accurate in time with 3 sources of error:
  1. the discretization error of the BDF
  2. the linearization error
  3. the factorization error
- Low computational complexity (order $N^3$ on a $N \times N \times N$ mesh)

## Disadvantages of the ADI scheme

- The factorization error dominates at high frequencies
- In three dimensions the simple versions is unstable
- Not easily amenable to parallelization

# LU Implicit Scheme

This uses two factors with backward and forward differencing for any number of space dimensions.
In the one dimensional case it takes the form

$$\left(I + \mu \Delta t D_x^- A^+\right) + \left(I + \mu \Delta t D_x^+ A^-\right) \delta w + \Delta t R(w^n) = 0 \tag{27}$$

Here $A^+$ and $A^-$ are splitting such that

$$A = A^+ + A^-$$

where

$$A^\pm \quad \text{have positive and negative eigenvalues}$$

while

$$D_x^\pm \quad \text{are forward and backward difference operators}$$

We may take

$$A^\pm = \frac{1}{2}(A \pm \epsilon I)$$

where

$$\epsilon \geq max|\lambda(A)|$$

# LU-Symmetric Gauss Seidel (LUSGS) Scheme

Yoon and Jameson 1988, Rieger and Jameson, 1988

Consider a flux split semi-discretization of the one dimensional conservation law

$$\frac{dw}{dt} + D_x^- f^+(w) + D_x^+ f^-(w) = 0$$

where $D_x^\pm$ are forward and backward difference operators and

$$A^\pm = \frac{\partial f^\pm}{\partial w} \quad \text{have positive and negative eigenvalues} \tag{28}$$

The **linearized implicit scheme** becomes

$$\left\{ I + \lambda \left( \delta_x^+ A^- + \delta_x^- A^+ \right) \right\} \delta w + \Delta t R = 0, \tag{29}$$

where $\lambda = \frac{\Delta t}{\Delta x}$ and $R$ is the residual, or

$$\delta w_i + \lambda \left( A_{i+1}^- \delta w_{i+1} - A_i^- \delta w_i + A_i^+ \delta w_i - A_{i-1}^+ \delta w_{i-1} \right) + \Delta t R_i = 0 \tag{30}$$

A **symmetric Gauss-Seidel scheme** is

$$\delta w_i^{(1)} + \lambda \left( A_i^+ - A_i^- \right) \delta w_i^{(1)} - \lambda A_{i-1}^+ \delta w_{i-1}^{(1)} + \Delta t R_i = 0 \tag{31}$$

$$\delta w_i^{(2)} + \lambda \left( A_i^+ - A_i^- \right) \delta w_i^{(2)} + \lambda A_{i+1}^- \delta w_{i+1}^{(2)} - \lambda A_{i-1}^+ \delta w_{i-1}^{(1)} + \Delta t R_i = 0 \tag{32}$$

Subtract (31) from (32) to obtain

$$\left\{ I + \lambda \left( A_i^+ - A_i^- \right) \right\} \delta w_i^{(2)} + \lambda A_{i+1}^- \delta w_{i+1}^{(2)} = \left\{ I + \lambda \left( A_i^+ - A_i^- \right) \right\} \delta w_i^{(1)} \tag{33}$$

# LUSGS scheme (contd.)

The scheme can now be written as

$$LD^{-1}U\delta w = -\Delta t R, \tag{34}$$

where

$$L \equiv I - \lambda A^{-} + \lambda \delta_x^{-} A^{+}, U \equiv I + \lambda A^{+} + \lambda \delta_x^{+} A^{-}. \tag{35}$$

$$D = I + \lambda \left( A^{+} - A^{-} \right). \tag{36}$$

If one takes

$$A^{+} = \frac{1}{2} \left( A + \epsilon I \right), A^{-} = \frac{1}{2} \left( A - \epsilon I \right), \tag{37}$$

where $\epsilon = \max \left| \lambda \left( A \right) \right|$ then $D$ reduces to a scalar factor and this is a variation of the $LU$ scheme.

## Advantages of the LUSGS scheme

- With the simplified form of $A^{\pm}$ no matrix inversions are required, so the complexity is minimal
- Easily extendable to treat flows with chemical reaction and many species
- Vectorizable along diagonals (diagonal planes in three dimensions)

## Disadvantages of the LUSGS scheme

- Not time accurate
- Hard to parallelize

# Nonlinear SGS scheme (Jameson and Caughey 2001)

The terms $\Delta t R_i - \frac{\Delta t}{\Delta x} A^{+}_{i-1} \delta w^{(1)}_{i-1}$ are a linearization of $\Delta t R_i$ evaluated with $w^{(1)}_{i-1} = w_{i-1} + \delta w^{(1)}_{i-1}$. Following this line, the **LU-SGS scheme** can be recast as

$$1) \left\{ I + \frac{\Delta t}{\Delta x} \left( A^{+}_{i} - A^{-}_{i} \right) \right\} \delta w^{(1)}_{i} + \Delta t R^{(01)}_{i} = 0; \tag{38}$$

$$2) \left\{ I + \frac{\Delta t}{\Delta x} \left( A^{+}_{i} - A^{-}_{i} \right) \right\} \delta w^{(2)}_{i} + \Delta t R^{(12)}_{i} = 0, \tag{39}$$

where

$$\begin{array}{ll} w^{(1)}_{i} = w^{(0)}_{i} + \delta w^{(1)}_{i}; & f^{\pm(1)}_{i} = f^{\pm}(w^{(1)}_{i}); \\ w^{n+1}_{i} = w^{(2)}_{i} = w^{(0)}_{i} + \delta w^{(2)}_{i}; & f^{\pm(2)}_{i} = f^{\pm}(w^{(2)}_{i}); \end{array}$$

and

$$\begin{array}{l} R^{(01)}_{i} = \frac{1}{\Delta x} \left( f^{-}_{i+1}{}^{(0)} - f^{-(0)}_{i} + f^{+(0)}_{i} - f^{+}_{i-1}{}^{(1)} \right), \\ R^{(12)}_{i} = \frac{1}{\Delta x} \left( f^{-}_{i+1}{}^{(2)} - f^{-(1)}_{i} + f^{+(1)}_{i} - f^{+}_{i-1}{}^{(1)} \right). \end{array}$$

# Motivation for the SGS scheme

## Solution of the Advection Equation by Relaxation

Consider the advection equation

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \quad a > 0 \tag{40}$$

$u(x, 0)$, $u(0, t)$ given.

With $\lambda = a\frac{\Delta t}{\Delta x}$ the **forward Euler (FE) scheme** is

$$v_j^{n+1} = v_j^n - \lambda \left( v_j^n - v_{j-1}^n \right) \tag{41}$$

stable for $0 \leq \lambda \leq 1$.

# Solution of the Advection Equation (contd.)

The implicit **backward Euler (BE) scheme** is

$$v_j^{n+1} = v_j^n - \lambda \left( v_j^{n+1} - v_{j-1}^{n+1} \right) \tag{42}$$

which can be solved to give

$$v_j^{n+1} = \frac{1}{1+\lambda} v_j^n + \frac{\lambda}{1+\lambda} v_{j-1}^{n+1} \tag{43}$$

stable for $0 \leq \lambda < \infty$.

The **Gauss-Seidel (GS)** ”blow out” scheme is

$$v_j^{n+1} = v_j^n - \bar{\lambda} \left( v_j^n - v_{j-1}^{n+1} \right) \tag{44}$$

- This is $L_\infty$ stable for $0 \leq \bar{\lambda} \leq 1$ (easily proved by induction).
- With $\bar{\lambda} = 1$, GS blows out the solution in 1 step. It is equivalent to BE with $\bar{\lambda} = \frac{\lambda}{1+\lambda}$.

# SGS scheme for Burger's Equation: Upwind Discretization

Consider

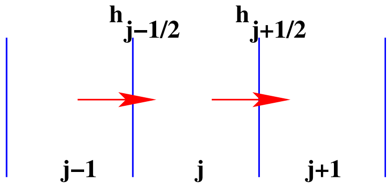
$$\frac{\partial v}{\partial t} + \frac{\partial}{\partial x} f\left(v\right) = 0, a > 0 \tag{45}$$

$v(x, 0)$ given

$$f\left(v\right) = \frac{v^2}{2} \tag{46}$$

wave speed

$$a\left(v\right) = \frac{\partial f}{\partial v} = v \tag{47}$$



A forward Euler scheme is

$$v_j^{n+1} = v_j^n - \frac{\Delta t}{\Delta x}\left(h_{j+\frac{1}{2}} - h_{j-\frac{1}{2}}\right) \tag{48}$$

where $h_{j+\frac{1}{2}}$ is the numerical flux from $j$ to $j + 1$.

# SGS scheme for Burger's Equation (contd.)

Define the wave speed at the interface as

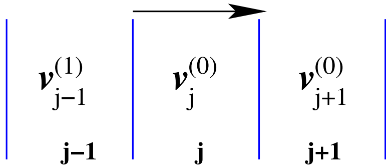
$$v^n_{j+\frac{1}{2}} = \frac{1}{2}\left(v^n_{j+1} + v^n_j\right) \tag{49}$$

Then the upwind scheme is

$$h_{j+\frac{1}{2}} = h\left(v^n_{j+1}, v^n_j\right) = \begin{cases} f^n_j = \frac{1}{2}v^{n2}_j & : \quad v^n_{j+\frac{1}{2}} > 0 \\ f^n_{j+1} = \frac{1}{2}{v^n_{j+1}}^2 & : \quad v^n_{j+\frac{1}{2}} < 0 \\ f^n_j = \frac{1}{2}\left(f^n_{j+1} + f^n_j\right) & : \quad v^n_{j+\frac{1}{2}} = 0 \end{cases} \tag{50}$$

# Solution of Burger's equation by Symmetric Relaxation

Sweep left to right and right to left re-evaluating the flux through each interface with the latest available values of $v$.



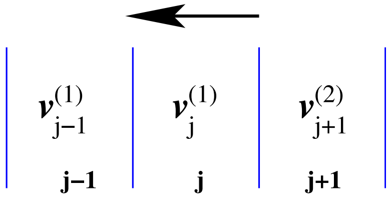
Set $\quad v_j^{(0)} = v_j^n$
Forward pass

$$v_j^{(1)} = v_j^{(0)} - \frac{\Delta t}{\Delta x}\left(h^{(00)}_{j+\frac{1}{2}} - h^{(01)}_{j-\frac{1}{2}}\right) \tag{51}$$

where

$$h^{(00)}_{j+\frac{1}{2}} = h\left(v^{(0)}_{j+1}, v^{(0)}_{j}\right), h^{(01)}_{j-\frac{1}{2}} = h\left(v^{(0)}_{j}, v^{(1)}_{j-1}\right), \tag{52}$$

# Solution of Burger's equation (contd.)



Backward pass

$$v_j^{(2)} = v_j^{(1)} - \frac{\Delta t}{\Delta x}\left(h^{(12)}_{j+\frac{1}{2}} - h^{(11)}_{j-\frac{1}{2}}\right) \tag{53}$$

where

$$h^{(12)}_{j+\frac{1}{2}} = h\left(v^{(2)}_{j+1}, v^{(1)}_{j}\right), h^{(11)}_{j-\frac{1}{2}} = h\left(v^{(1)}_{j}, v^{(1)}_{j-1}\right), \tag{54}$$

Finally

$$v_j^{n+1} = v_j^{(2)} \tag{55}$$

For stability take

$$\Delta t = \frac{\Delta x}{\max\left(v_{j+1}, v_j, v_{j-1}\right)} \tag{56}$$

Refined variants:

1. The solution for $v$ may be iterated more than once in each cell during each sweep
2. One can use multiple double sweeps

# Code to solve Burger's Equation

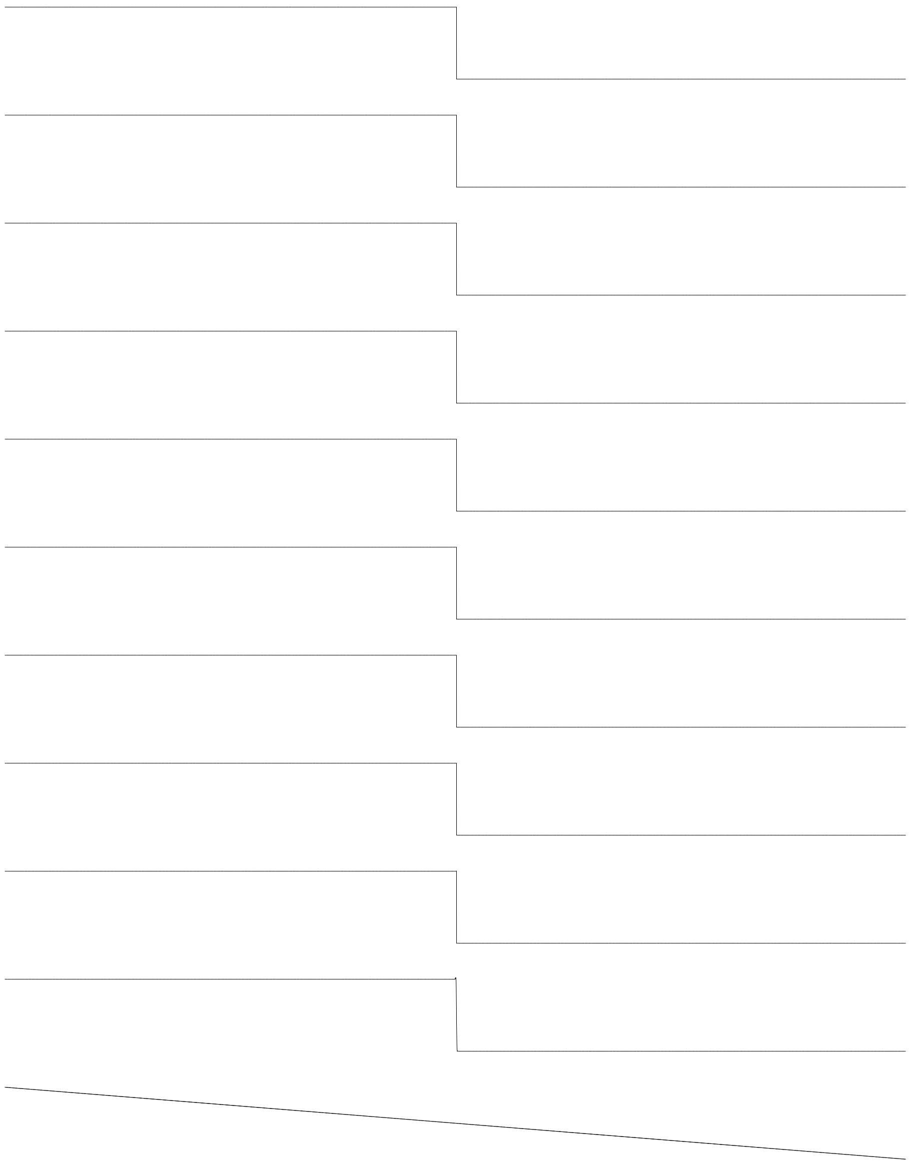
```
    DO 100 NSWEEP=1,2

    IF (NSWEEP.EQ.1) THEN
       I1        = 2
       I2        = IL
       INC       = 1
    ELSE
       I2        = 2
       I1        = IL
       INC       = -1
    END IF

    DO I=I1,I2,INC
    DO NUPD=1,2

    UM        = .5*(U(I)  +U(I-1))
    IF (UM.GE.0.) THEN
       FM        = .5*U(I-1)*U(I-1)
    ELSE
       FM        = .5*U(I)*U(I)
    END IF

    UP        = .5*(U(I+1)  +U(I))
    IF (UP.GE.0.) THEN
       FP        = .5*U(I)*U(I)
    ELSE
       FP        = .5*U(I+1)*U(I+1)
    END IF

    RES       = FP  -FM
    IF (ABS(RES).GT.ABS(RMAX)) THEN
       RMAX      = RES
       IMAX      = I
    END IF

    A         = MAX(ABS(U(I+1)),ABS(U(I)),ABS(U(I-1)))
    U(I)      = U(I)  -CFL*RES/A

    END DO
    END DO

100 CONTINUE
```

SOLUTION OF BURGERS EQUATION BY SYMMETRIC RELAXATION

131072  CELLS     15 LEVELS     10 CYCLES

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  0.1144E-03 -0.3284E+00 65653
    2  0.6239E-05 -0.4331E+00 65718
    3  0.6249E-10  0.1808E-02 65720
    4  0.1129E-23 -0.2807E-09 65718
    5  0.0000E+00  0.0000E+00     0
    6  0.0000E+00  0.0000E+00     0
    7  0.0000E+00  0.0000E+00     0
    8  0.0000E+00  0.0000E+00     0
    9  0.0000E+00  0.0000E+00     0
   10  0.0000E+00  0.0000E+00     0
```

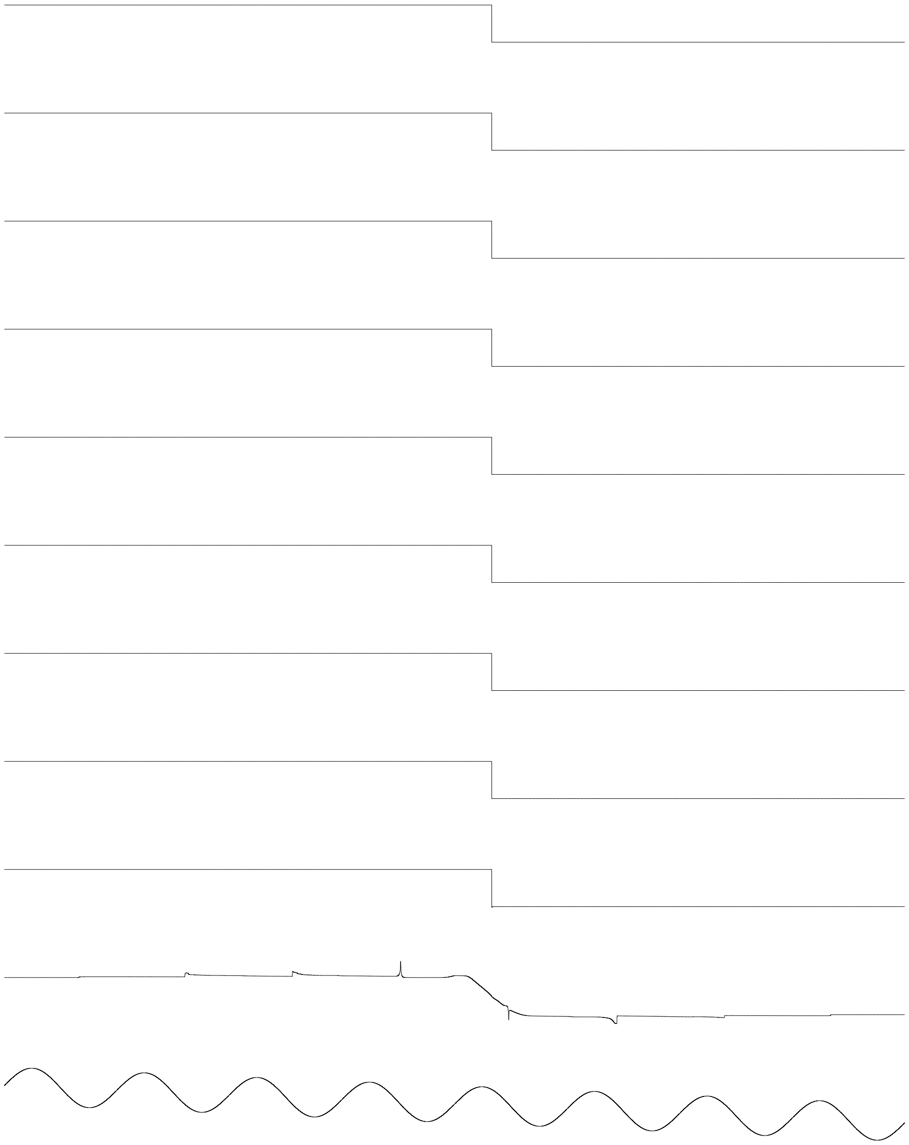


SOLUTION OF BURGERS EQUATION BY SYMMETRIC RELAXATION

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 BURGERS EQUATION BY SYMMETRIC RELAXATION

131072 CELLS 15 LEVELS

CFL 1.000 RAVG 0.0

# Preconditioning of the nonlinear SGS scheme

In the limit $\Delta t \rightarrow \infty$ the sweeps reduce to

$$|A|\delta w^{(1)} + \sigma R^{(01)} = 0|A|\delta w^{(2)} + \sigma R^{(12)} = 0$$

where $|A|$ is the **absolute Jacobian matrix**

$$|A| = A^{+} - A^{-}$$

and $\sigma$ is a relaxation factor $\sim 1$.

In two dimensional calculations each sweep has the form

$$D\delta w + \sigma R = 0$$

where $D$ is the sum of the **absolute Jacobian matrices**

$$D = |A| + |B|$$

Numerical experiments confirm that it pays to use the **true absolute Jacobians** obtained by replacing the eigenvalues by their absolute values

$$A = T\Lambda T^{-1}, \qquad\qquad |A| = T|\Lambda|T^{-1}$$

This is like the **block Jacobi preconditioner** of Pierce 1996, but it emerges naturally as a consequence of the formulation of the SGS scheme

# Symmetrization of the Jacobian Matrices

The Jacobians can be symmetrized by a transformation of dependent variables. For example, the quasi-linear form of the Euler equations

$$\frac{\partial w}{\partial t} + A\frac{\partial w}{\partial \xi} + B\frac{\partial w}{\partial \eta} = 0 \tag{57}$$

can be transformed to

$$\frac{\partial \overline{w}}{\partial t} + S^{-1}AS\frac{\partial \overline{w}}{\partial \xi} + S^{-1}BS\frac{\partial \overline{w}}{\partial \eta} = 0 \tag{58}$$

where

$$\overline{A} = S^{-1}AS = \left\{ \begin{array}{cccc} Q & cS_x & cS_y & 0 \\ cS_x & Q & 0 & 0 \\ cS_y & 0 & Q & 0 \\ 0 & 0 & 0 & Q \end{array} \right\} \tag{59}$$

where

$$Q = S_x u + S_y v \tag{60}$$

and $S_x$ and $S_y$ are the face areas projected parallel to the $x$ and $y$ axes, respectively. The Jacobian $\overline{B}$ has a similar form.

# Symmetrization of the Jacobian Matrices (contd.)

Each factor of the split scheme can then be written

$$\left\{I + \Delta t \left(|\overline{A}| + |\overline{B}|\right)\right\} \delta \overline{w}_{i,j} = \Delta t S^{-1} Res_{i,j} \tag{61}$$

where

$$|\overline{A}| = Q|\Lambda_A|Q^{-1}\,, \quad |\overline{B}| = Q|\Lambda_B|Q^{-1} \tag{62}$$

and

$$\delta w_{i,j} = S \delta \overline{w}_{i,j} \tag{63}$$

The “absolute” Jacobians have the (symmetric) form

$$|\overline{A}| = \begin{pmatrix} R_1 & \overline{S}_x R_2 & \overline{S}_y R_2 & 0 \\ \overline{S}_x R_2 & \overline{S}_y^2 Q_1 + \overline{S}_x^2 R_1 & \overline{S}_x \overline{S}_y \left[R_1 - Q_1\right] & 0 \\ \overline{S}_y R_2 & \overline{S}_x \overline{S}_y \left[R_1 - Q_1\right] & \overline{S}_x^2 Q_1 + \overline{S}_y^2 R_1 & 0 \\ 0 & 0 & 0 & Q_1 \end{pmatrix} \tag{64}$$

where

$$R_1 = \frac{Q_2 + Q_3}{2} \quad \text{and} \quad \mathrm{R}_2 = \frac{\mathrm{Q}_2 - \mathrm{Q}_3}{2}, \tag{65}$$

$$\overline{S}_x = \frac{S_x}{\sqrt{S_x^2 + S_y^2}} \quad \text{and} \quad \overline{\mathrm{S}}_\mathrm{y} = \frac{\mathrm{S}_\mathrm{y}}{\sqrt{\mathrm{S}_\mathrm{x}^2 + \mathrm{S}_\mathrm{y}^2}}, \tag{66}$$

and $Q_1$, $Q_2$, $Q_3$ are the absolute values of the eigenvalues of $\overline{A}$.

# Preconditioner for 1D Flows : Characteristic Time Stepping

In the one dimensional case the residual is

$$R(w) = \frac{\partial}{\partial x} f(w) = A \frac{\partial w}{\partial x} \tag{67}$$

The preconditioner simulates

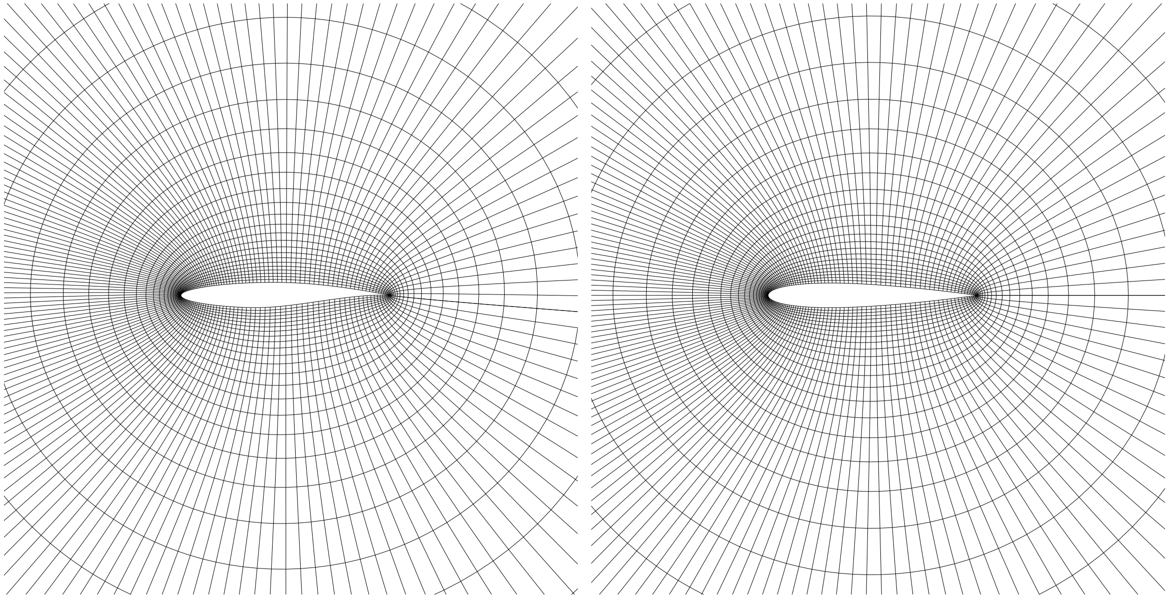
$$\frac{\partial w}{\partial t} + |A|^{-1} A \frac{\partial w}{\partial x} = 0 \tag{68}$$

where

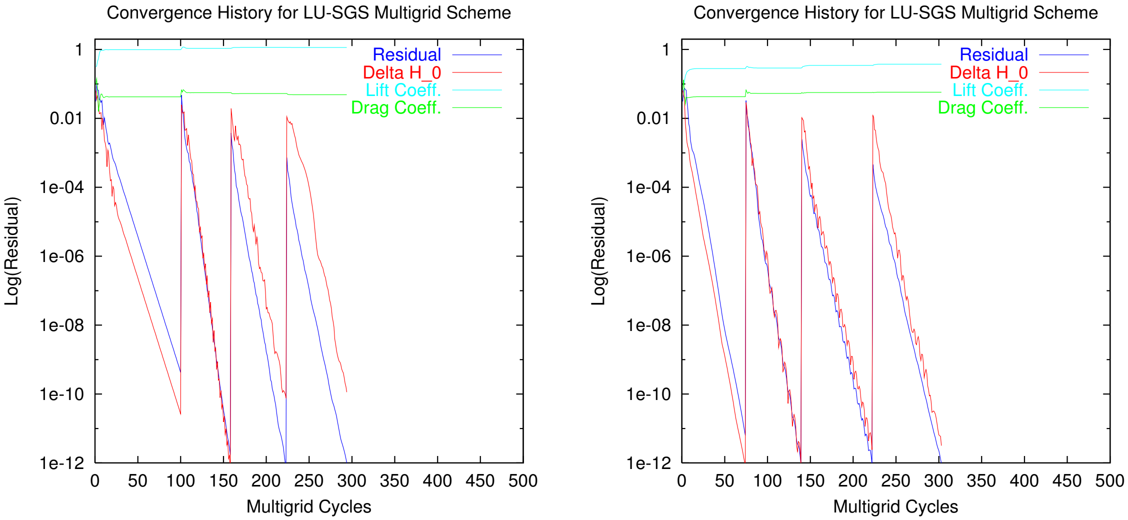
$$|A^{-1}|A = V|\Lambda|^{-1}V^{-1}V\Lambda V^{-1} = V \mathrm{diag}\left(\frac{\lambda}{|\lambda|}\right) V^{-1} \tag{69}$$

The **wave speeds** are **equalized** at ±1.

# Results from the nonlinear SGS scheme

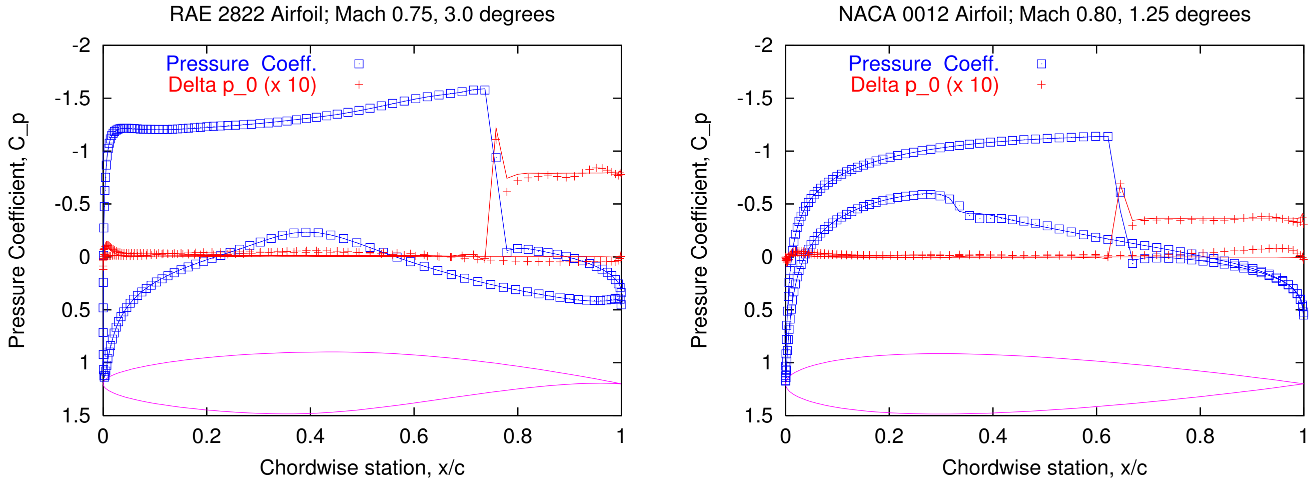


Grid around RAE 2822 and NACA 0012 airfoil

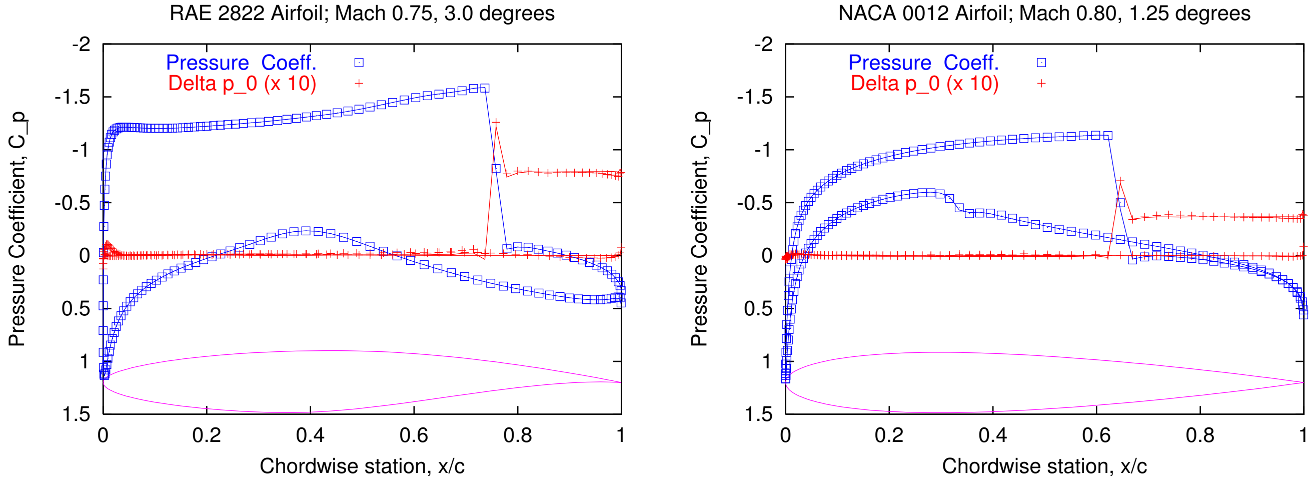


Convergence history for flow past the RAE2822 at $M_\infty$=0.75, $\alpha$ = 3 degrees and NACA 0012 airfoil at $M_\infty$= 0.8, $\alpha$ = 1.25 degrees

# Results from the nonlinear SGS scheme



**Pressure distribution after 3 cycles for flow past the RAE 2822 and NACA 0012 airfoil**



**Pressure distribution after 5 cycles for flow past the RAE 2822 and NACA 0012 airfoil**

**solid —— and —— lines : fully converged results**

# Results from the nonlinear SGS scheme

| Case | Fig. | MG Cycles | $C_L$ | $C_D$ |
|---|---|---|---|---|
| RAE 2822; $M_\infty = 0.75; \quad \alpha = 3.00$ | – | 100 | 1.1417 | 0.04851 |
| | 10(b) | 5 | 1.1429 | 0.04851 |
| | 7(b) | 3 | 1.1451 | 0.04886 |
| NACA 0012; $M_\infty = 0.80; \quad \alpha = 1.25$ | – | 100 | 0.3725 | 0.02377 |
| | 11(b) | 5 | 0.3746 | 0.02391 |
| | 8(b) | 3 | 0.3770 | 0.02387 |

Table 2: Force coefficients for the fast, preconditioned multigrid solutions using CUSP spatial discretization

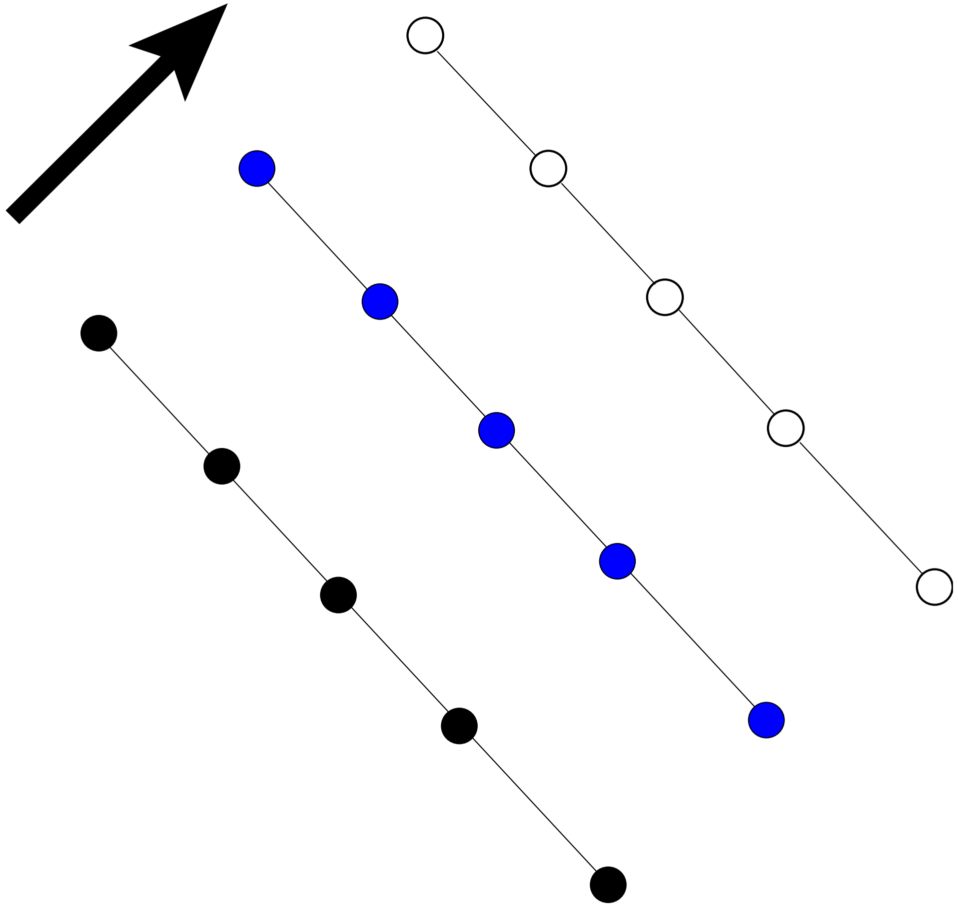
# SGS Scheme Versus Modified RK Scheme

The SGS scheme converges four to five times faster than the modified RK scheme.
Why not abandon the RK scheme?

- The RK scheme is fully parallelizable
- It can be used on arbitrary unstructured grids without modification
- While the SGS scheme could be vectorized by solving along diagonals, it is hard to parallelize.
- On an arbitrary grid there is no obvious ordering of the cells for the SGS scheme

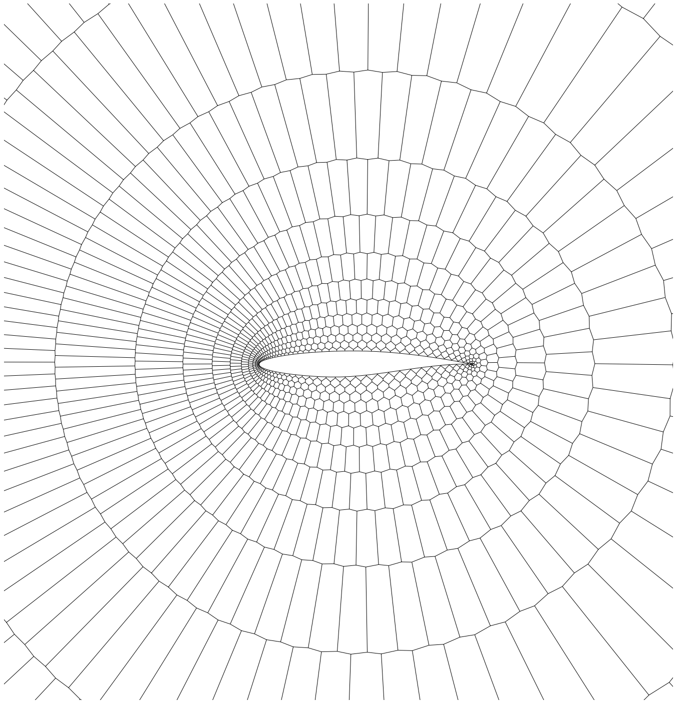
# Vectorization of SGS

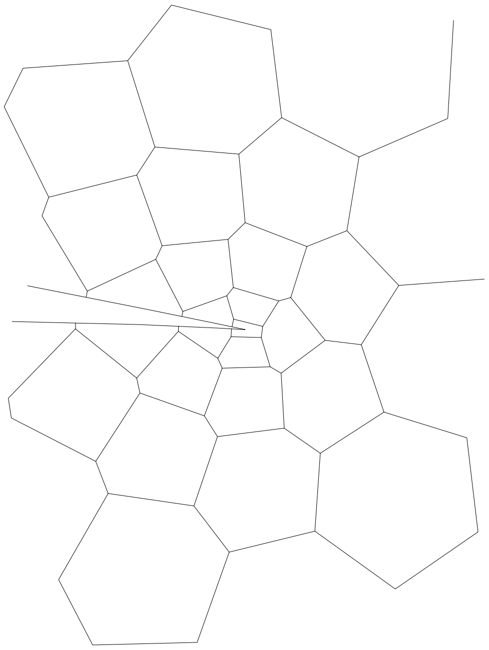
Yoon 1988



# Face–based Gauss Seidel (FBGS) Scheme

On an arbitrary grid, instead of looping over the cells and evaluating the fluxes of all the faces of each cell, one can loop over the faces, and after re–evaluating the flux across each face, update the solution in the two cells separated by that face. Each cell will be repeatedly updated during the sweep.





# Time Accurate Schemes

1. LED schemes

2. Implicit schemes based on the backward difference formula (BDF)

3. Time-spectral schemes

# Local Extremum Diminishing (LED) schemes

Numerical simulations of flows with shockwaves and contact discontinuities are prone to spurious oscillations which can be prevented by schemes which control the growth of the extrema.

Consider the general discrete scheme

$$v_i^{n+1} = v_i^n + \Delta t \sum_j b_{ij} v_j^n = \sum_j a_{ij} v_j^n \tag{70}$$

If equation (70) is consistent with a differential equation with no source term

$$\sum_j b_{ij} = 0 \qquad \sum_j a_{ij} = 1$$

For $L_\infty$ stability

$$\sum_j |a_{ij} \leq 1$$

Hence it is necessary that

$$a_{ij} \geq 0$$

The scheme is local extremum diminishing (LED) if it has a compact stencil. It is then also total variation diminishing (TVD)

# LED schemes (contd.)

- Given a first-order accurate (forward Euler) discretization with positive coefficients, higher-order RK schemes do not necessarily preserve positivity
- For steady flow simulations it is generally sufficient to use an LED semi-discretization

  $$\frac{dv_i}{dt} = \sum a_{ij}(v_j - v_i)$$

  with

  $$a_{ij} \geq 0.$$

# TVD Runge Kutta schemes

Shu and Osher have developed higher order Runge-Kutta schemes which preserve the LED or TVD property, provided that it is satisfied by the semi-discretization. Their third order scheme is

$$\begin{aligned}
\mathbf{w}^{(\mathbf{1})} &= \mathbf{w}^{(\mathbf{0})} - \Delta t R(\mathbf{w}^{(\mathbf{0})}) \\
\mathbf{w}^{(\mathbf{2})} &= \frac{3}{4}\mathbf{w}^{(\mathbf{0})} + \frac{1}{4}\mathbf{w}^{(\mathbf{1})} - \frac{1}{4}\Delta t R(\mathbf{w}^{(\mathbf{1})}) \\
\mathbf{w}^{(\mathbf{3})} &= \frac{1}{3}\mathbf{w}^{(\mathbf{0})} + \frac{2}{3}\mathbf{w}^{(\mathbf{2})} + \frac{2}{3}\Delta t R(\mathbf{w}^{(\mathbf{2})}) \qquad (71)
\end{aligned}$$

which is LED if the CFL number has an absolute value $\leq 1$. This scheme has demonstrated to give good results for shock tube problems (Jiang and Shu, 1996).

The search for TVD Runge Kutta schemes with maximal CFL number and minimum complexity is a subject of ongoing research.

## Multi-stage linear schemes with positive coefficients

Consider the discrete scheme in matrix-vector notation

$$v^{n+1} = (I + \Delta t B)v^n = Av^n \tag{72}$$

where, as before,

$$\sum_j b_{ij} = 0, \quad \sum_j a_{ij} = 1, \quad a_{ij} \geq 1$$

Suppose equation (72) is consistent with the linear differential equation

$$\frac{\partial u}{\partial t} = \mathcal{L}u$$

Then

$$u(t + \Delta t) = u(t) + \Delta t u_t + \frac{\Delta t^2}{2!} u_{tt} + ...$$

where

$$u_{tt} = \mathcal{L}u_t = \mathcal{L}^2 u$$

$$...$$

Accordingly the scheme

$$v^{n+1} = (I + \Delta B + \frac{\Delta t^2}{2!}B^2 + .... + \frac{\Delta t^n}{n!}B^n)v^n \tag{73}$$

which can be written as an $n$-stage Runge-Kutta scheme, is $n^{th}$ order accurate.

# Proof of Positivity

Let $C = A^r$. Then

$$c_{ij} \geq 0 \tag{74}$$

and

$$\sum_j c_{ij} = 1 \tag{75}$$

The proof is by induction. Suppose that conditions in equations (74 and 75) are satisfied by $A^r$. Then

$$A_{ij}^{r+1} = \sum_k a_{ik} c_{kj} \geq 0$$

and

$$\begin{aligned}
\sum_j A_{ij}^{r+1} &= \sum_j \sum_k a_{ik} c_{kj} \\
&= \sum_k a_{ik} \left( \sum_j c_{kj} \right) \\
&= \sum_k a_{ik} \\
&= 1
\end{aligned} \tag{76}$$

Since $\Delta t B = A - T$, equation (73) can be written as

$$v^{n+1} = P_n(A)v^n \tag{77}$$

where $P_n(x)$ is a polynomial consiting of the first $n+1$ terms of the series representing $exp(x-1)$.

$$P_n(x) = 1 + x - 1 + \frac{(x-1)^2}{2!} + ... + \frac{(x-1)^n}{n!} \tag{78}$$

Then

$$P_n(x) = \sum_{r=0}^{n} b_{n,r} x^r \tag{79}$$

where

$$b_{n,r} \geq 0 \tag{80}$$

Proof: This is true when $n = 0, 1$.
Suppose that it is true for $n = k-1$. Since

$$P_k^{'}(x) = \sum_{r=1}^{k} r b_{k,r} x^{r-1} = P_{k-1}(x) = \sum_{r=0}^{k-1} b_{k-1,r} x^{r-1}$$

it follows that

$$b_{k,r} = \frac{1}{r} b_{k-1,r} \geq 0, r = 1, 2...k-2$$

while

$$b_{k,k-1} = b_{k-1,k-1} - \frac{1}{(k-1)!} = 0$$

Also

$$b_{k,k} = \frac{1}{k!} > 0$$

and

$$b_{k,0} = P_k(0) = \frac{1}{2!} - \frac{1}{3!} + \frac{1}{4!} - \frac{1}{5!} .... + \frac{-1^k}{k!}$$

When $k$ is odd this is the sum of positive terms of the form $\frac{1}{r!} - \frac{1}{(r+1)!}$. When $k$ is even, it is larger than $b_{k-1,0}$ by $\frac{1}{k!}$. Thus $b_{k,r} \geq 0$ for all $r$, and the result in equation (80) follows by induction.
Moreover

$$\sum_{r=0}^{n} b_{n,r} = P_n(1) = 1 \tag{81}$$

Now equation (77) can be written as

$$v^{n+1} = Dv^n$$

where

$$D = \sum_{r=0}^{n} b_{n,r} A^r$$

It follows from equations (75,77,80 and 81) that $d_{ij} \geq 0$ and $\sum_j d_{ij} = 1$.

Thus the $n^{th}$ **order linear RK scheme** inherits the **positivity** of **the first-order scheme**.

# Implicit Schemes Based on the Backward Difference Formula (BDF)

Suppose that the semi-discretization

$$\frac{dw}{dt} + R(w) = 0$$

is fully discretized as

$$D_t w^{n+1} + R(w^{n+1}) = 0.$$

Here $D_t$ is a $k^{th}$ order accurate **Backward Difference Formula (BDF)** of the form

$$D_t = \frac{1}{\Delta t} \sum_{q=1}^{k} \frac{1}{q} (\Delta^-)^q,$$

where

$$\Delta^- w^{n+1} = w^{n+1} - w^n.$$

The schemes with $k = 1, 2$ are A-stable.

- Dahlquist has shown that A-stable linear multi-step schemes are at best second order accurate.
- Gear, however, has shown that the schemes with $k \leq 6$ are stiffly stable, and one of the higher order schemes may offer a better compromise between accuracy and stability.

# ADI Scheme with the Backward Difference Formula

- Replace the left hand side of the linearized BDF by an approximate factorization, giving the ADI-BDF scheme
$$\left(I+\frac{2\Delta t}{3V}D_xA\right)\left(I+\frac{2\Delta t}{3V}D_yB\right)\Delta w^n=\frac{1}{3}\Delta w^{n-1}-\frac{2\Delta t}{3V}R(w^n) \tag{82}$$
where $R(w^n)=D_xf(w^n)+D_yg(w^n)$.

- **Advantages:**

  1. Nominally second order accurate in time with 3 sources of error:
     - the discretization error of the BDF
     - the linearization error
     - the factorization error
  2. can be solved at low computational cost in two steps.

- **Disadvantages:**

  1. The factorization error dominates at large CFL numbers
  2. The scheme isn't amenable to parallel processing: it may lose its stability if applied separately in each of a large numbers of blocks.

# Dual Time Stepping BDF Scheme

- Solve the full nonlinear BDF by inner iterations which advance in pseudo time $t^*$.

  For the second order BDF solve

  $$\frac{dw}{dt^*} + R^*(w) = 0$$

  where

  $$R^*(w) = \left[\frac{3w - 4w^n + w^{n-1}}{2\Delta t} + R(w)\right] \tag{83}$$

- On convergence to a steady state, $\frac{\partial w}{\partial t^*} = 0$, and the solution of the BDF is recovered.
- We solve (83) using either the **RK–BDF scheme:**
  - explicit multistage scheme with variable local $\Delta t^*$
  - implicit residual averaging
  - multigrid

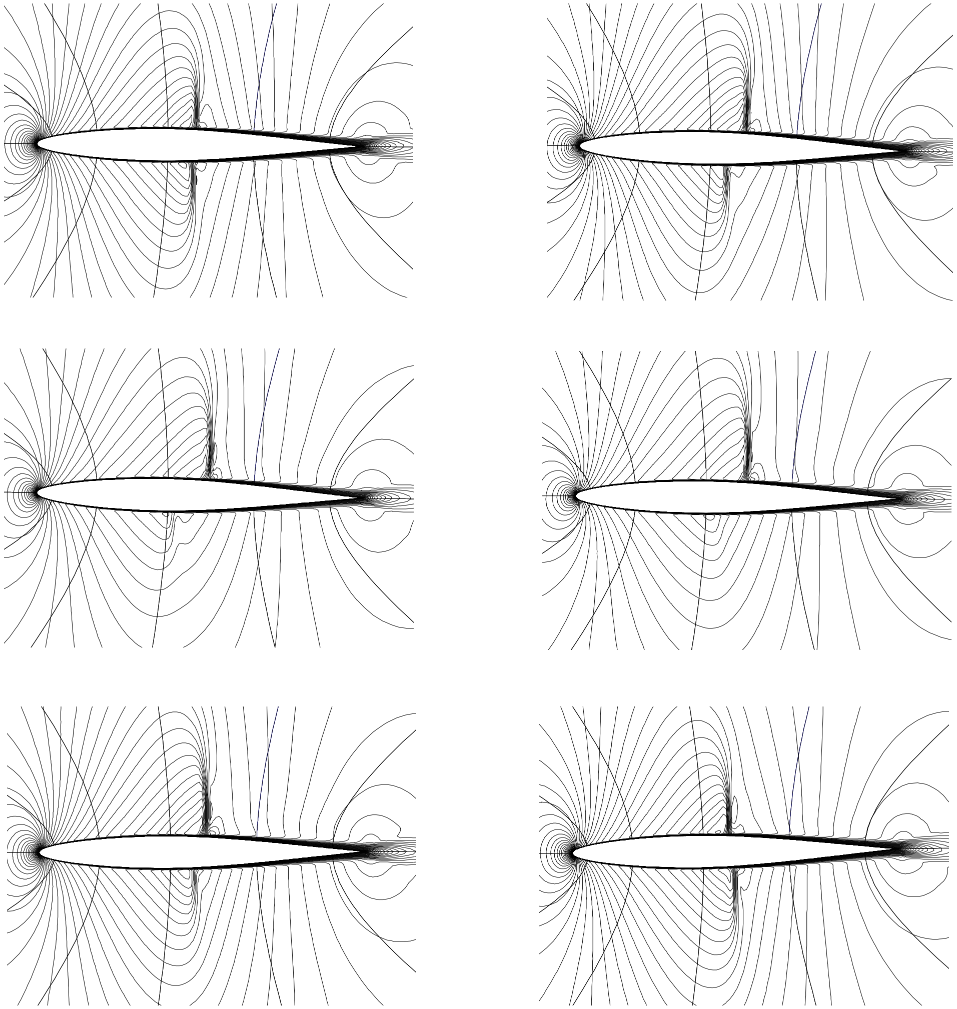
  or using the **SGS–BDF scheme:**
  - nonlinear SGS scheme
  - multigrid

# Test case for Unsteady flow Simulations

The unsteady flow past the pitching airfoil provides a useful test case for time-integration methods. Results are presented in the following slides for the case defined in the table below, for which experimental data is also available (AGARD case CT6).

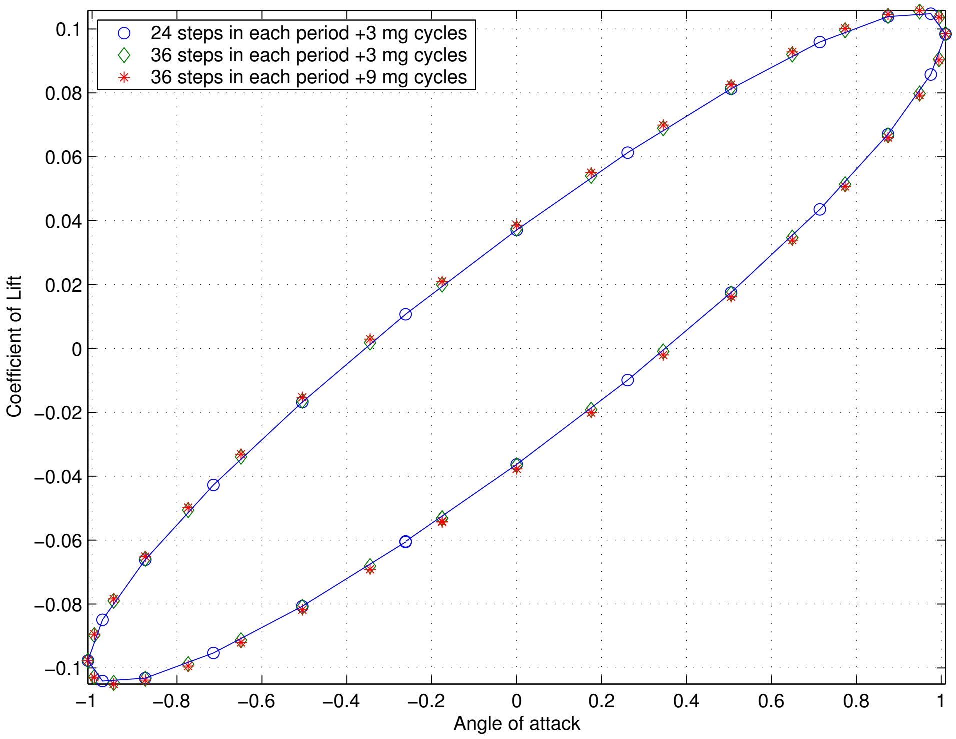
| Flow Conditions for pitching airfoil (NACA64A010) |
|---|
| $M_\infty$ =0.796<br>$\alpha = \pm$ 1.01 degrees<br>Reduced Frequency $\omega_r = \frac{\omega chord}{2u_\infty} = 0.202$ |

# Pressure Contours at various time instances



# Results of the SGS-BDF Method

The figure below shows the oval curve traced by the lift coefficient due to its phase lag as the angle of attack varies sinusoidally. These results were obtained using the third order accurate backward difference formula. It can be seen that the results over-plot when 24 or 36 time steps are used in each pitching cycle. In the case of 24 steps the maximum CFL number is 4153 in the vicinity of the trailing edge. It can also be seen that the results using 3 multigrid cycles in each time step are almost identical to those using 9 multigrid cycles in each time step.



# Dual Time Stepping BDF Scheme (contd.)

- **Advantage:**
  - If the inner iterations converge fast enough, we solve the fully nonlinear BDF, giving an efficient A-stable or stiffly stable scheme which allows very large $\Delta t$.
- **Disadvantages:**
  - Hard to assess accuracy unless the inner iterations are fully converged.
  - If a large number of inner iterations are required, the scheme becomes expensive.

# Consequences of Inexact Solution of the BDF

With the nonlinear system it is impossible to solve the BDF exactly.

Instead we obtain perturbed values $\hat{w}^n$.

This affects both

- **stability**
- **accuracy**

Bertil Gustafsson has analyzed the affect on the stability region (Stanford, ASCI Report, 2001). It seems that the perturbation of the stability region is small but possibly unfavorable.

# Accuracy of the BDF with inexact Solution

The error

$$e^n = \hat{w}^n - w^n$$

may be treated in the same way as round off errors. Assuming that

$$||e^n|| \leq \mathcal{K}\Delta t^{q+1}$$

Henrici (1962) derives a bound of order of order $\Delta t^q$.

With a $p^{th}$ order BDF this suggests that the dual time stepping scheme should be continued until $||e^n||$ is of order $\Delta t^{p+1}$.

# Accuracy of the BDF with inexact Solution (contd.)

Only the residual $R^*(w)$ can be measured during the iterations. To derive a bound on $||e||$ assume that the operator $\hat{R}(w) = \frac{3}{2\Delta t}w + R(w)$
satisfies the condition

$$||\hat{R}(w) - \hat{R}(\hat{w})|| \geq \mathcal{K}||w - \hat{w}||$$

For the exact solution $R^*(w^n) = 0$. Then

$$R^*(\hat{w}^n) = R^*(w^n) = \hat{R}(\hat{w}^n) - \hat{R}(w^n) - \frac{2}{\Delta t}e^{n-1} + \frac{1}{2\Delta t}e^{n-2}$$

Hence the error may be bounded as

$$||e^n|| \leq \frac{1}{\mathcal{K}}||R^*(\hat{w}^n) + \frac{2}{\Delta t}e^{n-1} + \frac{1}{2\Delta t}e^{n-2}||$$

In this case the iterations should be continued at least until $||R^*(\hat{w})|| = \mathcal{O}(\Delta t^{p+1})$.

# Accuracy of the BDF with inexact Solution (contd.)

In the case of the diffusion equation

$$\frac{\partial w}{\partial t} = \sigma \Delta w$$

the second order BDF has the modified residual

$$R^*(w) = \frac{3w - 4w^n + w^{n-1}}{2\Delta t} - \mathcal{L}w$$

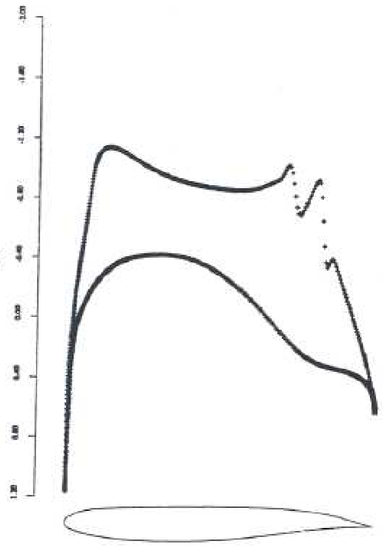
where $\mathcal{L}$ is the discrete Laplacian and is negative definite. Hence the accuracy condition is satisfied with an additional margin supplied by the leading term $\frac{3}{2\Delta t}w$.

In the case of the Euler equations there exist shapes admitting non-unique transonic solutions (Jameson, 1991, Hafez 2002).

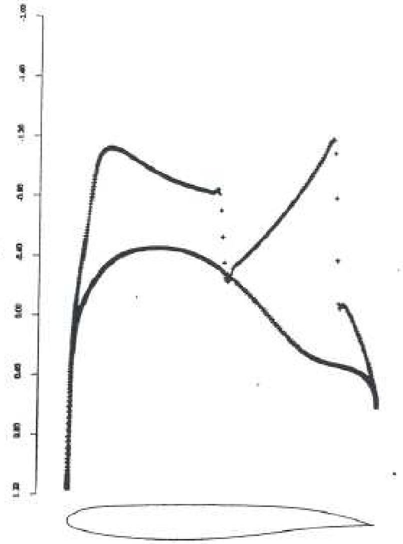
Thus we have

$$||R(w_1) - R(w_2)|| = 0 \qquad \text{with} \quad ||w_1 - w_2|| > 0$$

violating the accuracy assumption.



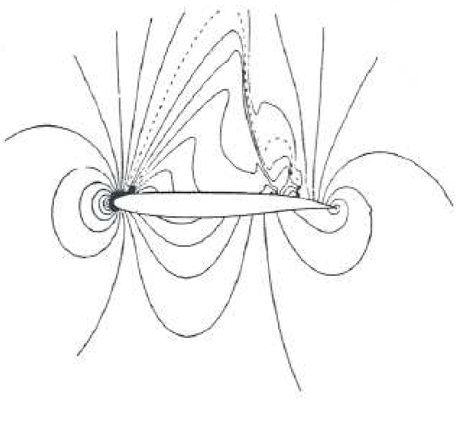
J-78 AIRFOIL
MACH 0.780 ALPHA -0.600
CL 0.6590 CD 0.0014 CM -0.1710
GRID 640X 128 NCYC 800 RES 0.158E-13



J-78 AIRFOIL
MACH 0.780 ALPHA -0.600
CL 0.5653 CD 0.0067 CM -0.1563
GRID 640X 128 NCYC 800 RES 0.163E-11

MACH NUMBER

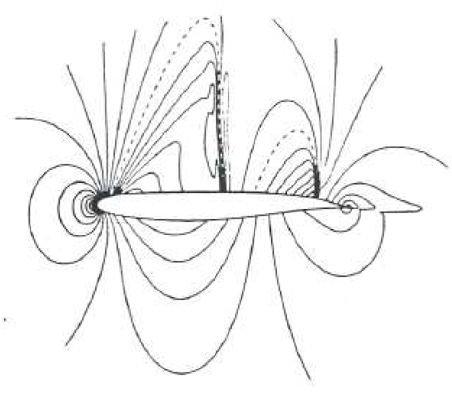
CONTOUR LEVELS INCREMENT BY 0.050



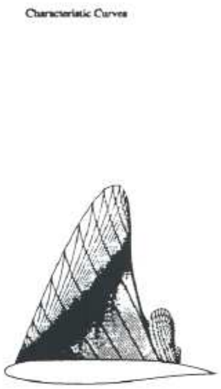
J-78 AIRFOIL
MACH 0.780 ALPHA -0.600
CL 0.6590 CD 0.0014 CM -0.1710
GRID 640X 128 NCYC 800 RES 0.158E-13

MACH NUMBER

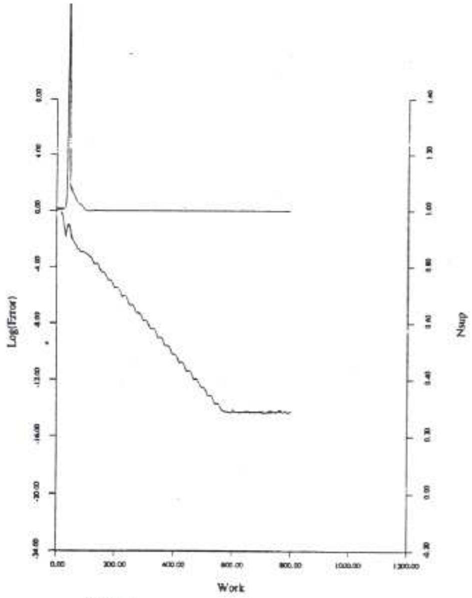
CONTOUR LEVELS INCREMENT BY 0.050



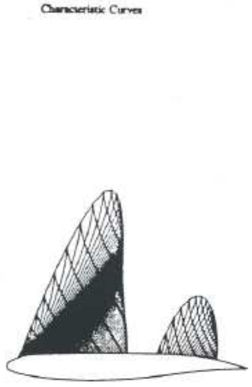
J-78 AIRFOIL
MACH 0.780 ALPHA -0.600
CL 0.5653 CD 0.0067 CM -0.1563
GRID 640X 128 NCYC 800 RES 0.163E-11



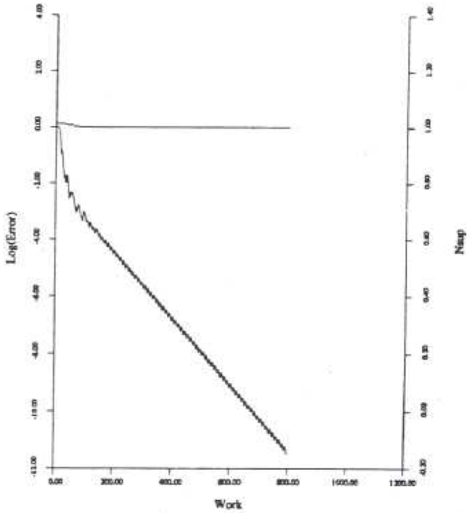
J-78 AIRFOIL
MACH 0.780 ALPHA -0.600
CL 0.6590 CD 0.0014 CM -0.1710
GRID 640X 128 NCYC 800 RES 0.158E-13



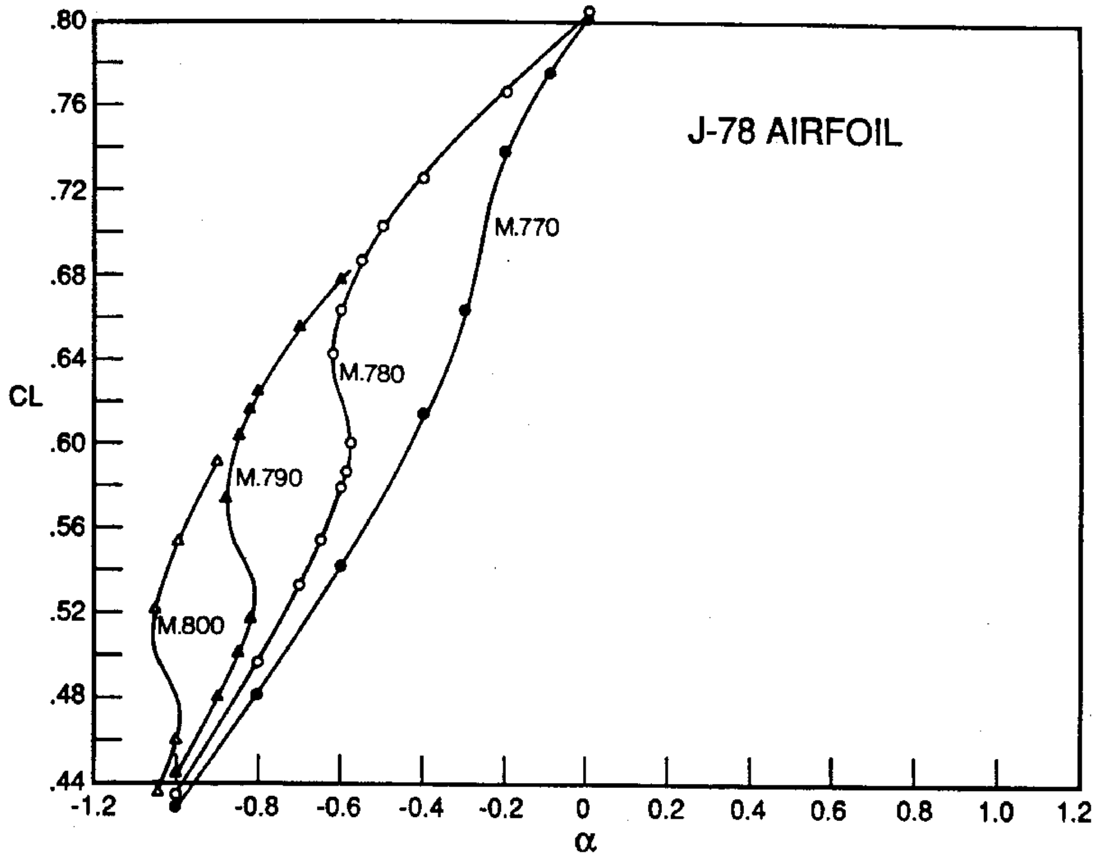
J-78 AIRFOIL
MACH 0.780 ALPHA -0.600
RESID1 0.322E+01 RESID2 0.196E-13
WORK 794.00 RATE 0.9398
GRID 640X 128



J-78 AIRFOIL
MACH 0.780 ALPHA -0.800
CL 0.5653 CD 0.0067 CM -0.1563
GRID 640X 128 NCYC 800 RES 0.163E-11



J-78 AIRFOIL
MACH 0.780 ALPHA -0.800
RESID1 0.410E+00 RESID2 0.143E-11
WORK 798.00 RATE 0.9675
GRID 640X 128


J-78 AIRFOIL
CL
.80
.76
.72
.68
.64
.60
.56
.52
.48
.44
α
-1.2
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1.0
1.2
M.770
M.780
M.790
M.800

# Hybrid Scheme

- The proposed hybrid scheme will take an initial ADI step in real time $\Delta t$:

$$\left(I + \frac{2\Delta t}{3} D_x A\right)\left(I + \frac{2\Delta t}{3} D_y B\right)\Delta w^{(1)} + \frac{2\Delta t}{3} R(w^n) - \frac{1}{3}\Delta w^{n-1} = 0 \qquad (84)$$

  yielding a nominal second order accuracy without iterations.

- Then follow it with the iterative multistage time stepping scheme augmented by multigrid to drive the solution in the steady state limit towards the fully nonlinear BDF.

$$\Delta w^{(k)} - \Delta w^{(k-1)} + \beta_k \left[\frac{3}{2\Delta t}\left(\Delta w^{(k)} - \frac{1}{3}\Delta w^{n-1}\right) + R(w^{(k-1)})\right] = 0 \qquad (85)$$

# Hybrid Scheme

- **The advantages of this scheme are that:**

  1. We should retain formal second order accuracy with any number of iterations, and it should not be necessary to iterate to convergence within each implicit time step, in contrast to existing dual-time stepping schemes which are only second order accurate if the inner iterations are fully converged.
  2. The additional iterations with multigrid should provide information exchange between processors which is needed to stabilize the ADI scheme run separately in each processor.

# Time-spectral Method

(Hall 2000, McMullen, Jameson and Alonso 2002)

- There are many unsteady flows in engineering devices such as turbomachinery or helicopter rotors in which the flow is periodic.
- In this situation there is the opportunity to gain spectral accuracy by using a Fourier representation in time.
- Suppose the period $T$ is divided into $N$ time steps $\Delta t = T/N$.

Let $\hat{w}_k$ be the discrete Fourier transform of $w^n$,

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

# Time-spectral Method (contd.)

- Then the semi-discretization
$$\frac{dw}{dt} + R(w) = 0 \tag{86}$$
is discretized as the pseudo-spectral scheme
$$D_t w^n + R(w^n) = 0 \tag{87}$$
where
$$D_t w^n = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} ik\hat{w}_k e^{ikn\Delta t}$$

- Here $D_t$ is a central difference formula connecting all the time levels so equation (87) is an **integrated space-time formulation** which requires the simultaneous solution of the equations for all the time levels.

- Provided, however, that the solution is sufficiently smooth, equation (87) should yield **spectral accuracy** (exponential convergence with increasing $N$).

# Solution of the time-spectral method

The **time spectral equation** (87) may be solved by **dual time-stepping** as

$$\frac{dw^n}{dt^*} + D_t w^n + R(w^n) = 0 \tag{88}$$

in **pseudo-time** $t^*$, as in the case of the BDF.
Alternatively it may be solved in the **frequency domain**. In this case we represent equation (87) as

$$\hat{R}_k^* = ik\hat{w}_k + \hat{R}_k = 0 \tag{89}$$

where $\hat{R}_k$ is the Fourier transform of $R(w(t))$. Because $R(w)$ is **nonlinear**, $\hat{R}_k$ depends on **all** the **modes** $\hat{w}_k$. We now solve equation (89) by time evolution in **pseudo-time**

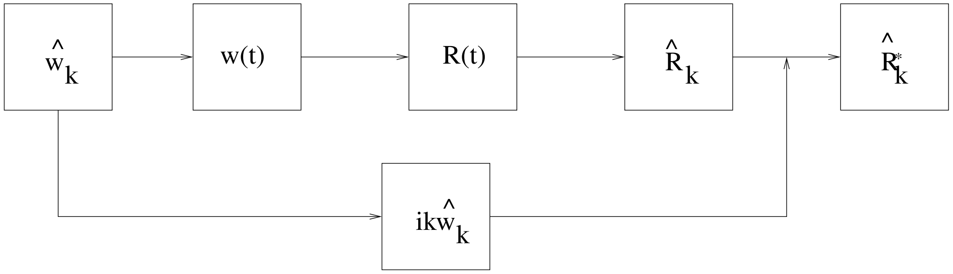
$$\frac{dw_k}{dt^*} + \hat{R}_k^* = 0 \tag{90}$$

# Solution of the time-spectral method (contd.)

At each iteration in pseudotime $\hat{R}_k$ is evaluated indirectly. First $w(t)$ is obtained as the reverse transform of $\hat{w}_k$. Then we calculate the corresponding time history of the residual

$$R(t) = R(w(t))$$

and obtain $\hat{R}_k$ as the Fourier transform of $R(t)$, as shown in the diagram



# Multigrid Solution

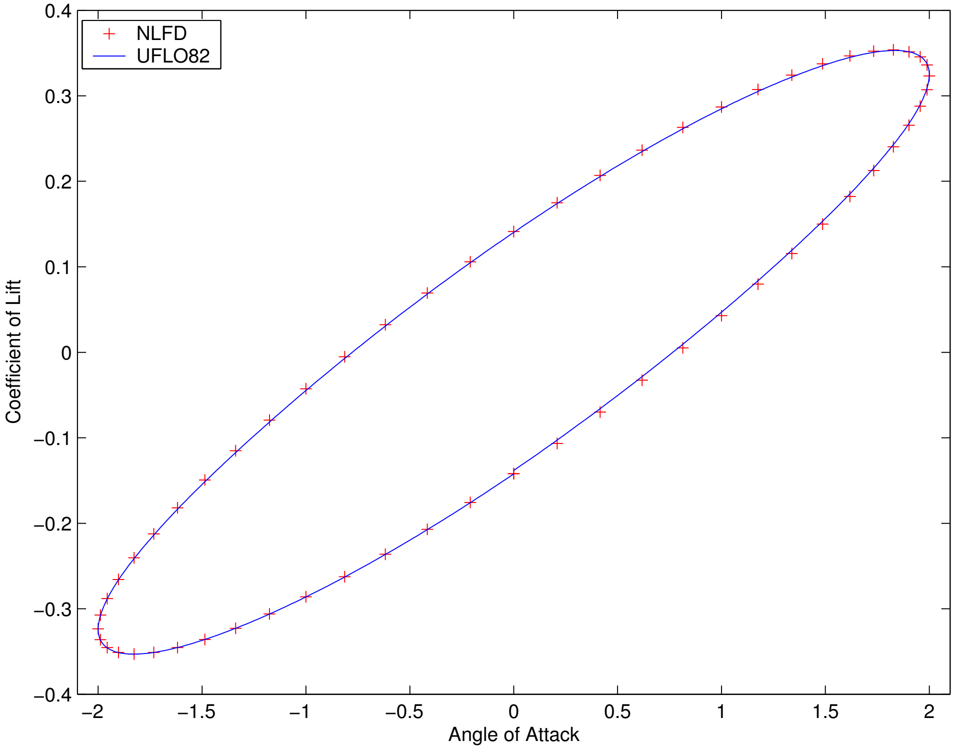
- McMullen has shown that the **modified RK method** with **multigrid acceleration** achieves essentially the same rate of convergence for the solution of the **frequency-domain equation** (89) as it does for the **BDF**.
- An alternative under investigation is to solve equation (89) by using a **nonlinear SGS method**.
- Further research is also needed to determine whether it is more efficient to solve the time-spectral equations in the time domain or the frequency domain.

# Results of the Time-Spectral Method

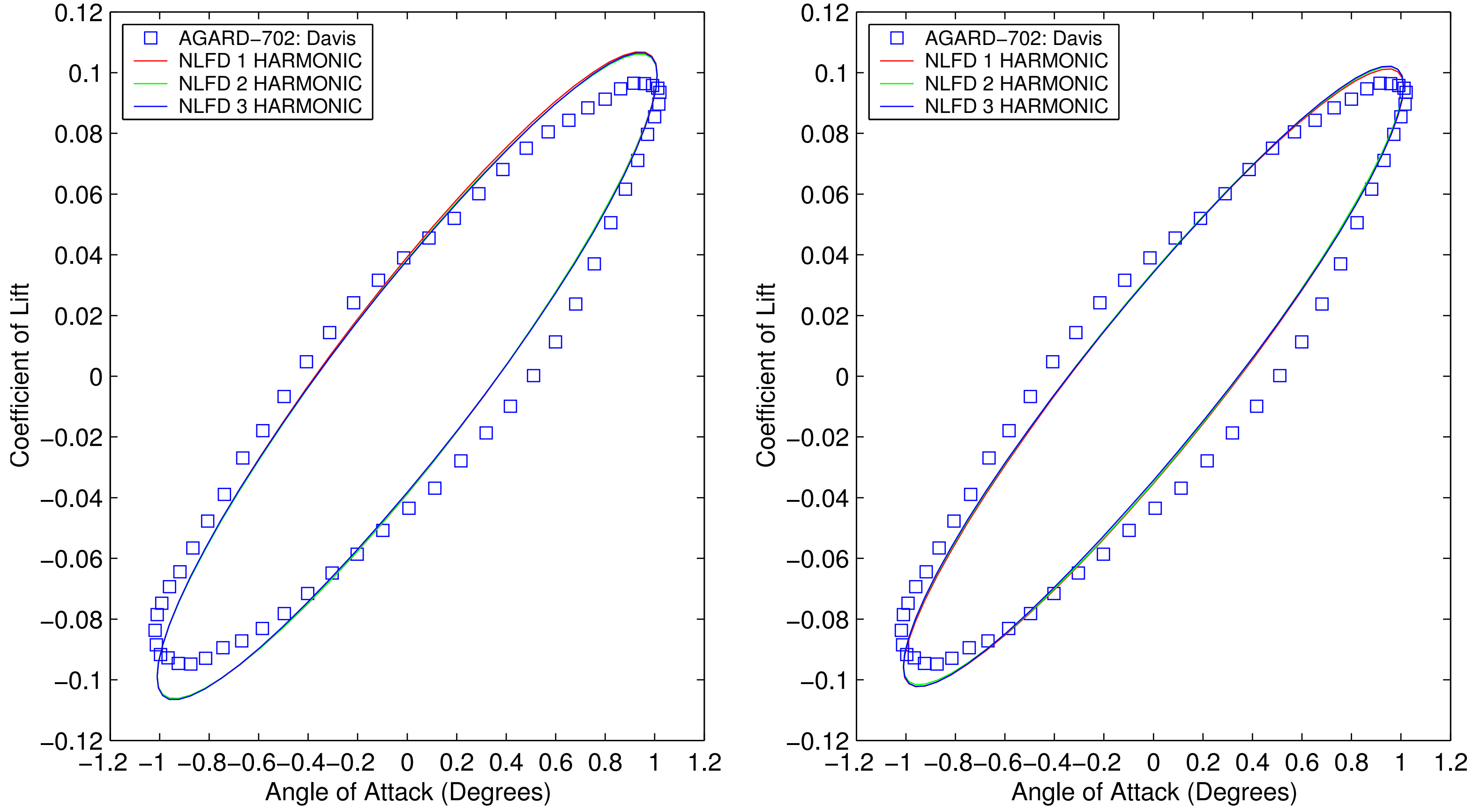
While the time-spectral method should make it possible to achieve **spectral accuracy**, numerical tests have shown that it can give the accuracy required for practical applications (**‘engineering accuracy’**) with very small numbers of modes. This is illustrated below for the pitching airfoil.

| Flow Conditions for pitching airfoil (NACA64A010) |
|---|
| $M_\infty$ =0.796<br>$\alpha = \pm$ 1.01 degrees<br>Reduced Frequency $\omega_r = 0.202$ |

# Comparison of the time-spectral and dual time-stepping schemes



# $C_l$ over the pitching cycle



Inviscid

Viscous

# Conclusions (A)

## For Steady State Problems

- The **modified RK method** with multigrid acceleration is an efficient time-integration method which is easily adaptable to **unstructured grids** and **parallel computers**.
- The **nonlinear SGS method** approaches **'textbook' multigrid performance**, but is **hard** to **parallelize**.
- In the solution of large scale nonlinear problems, a **fast steady state solver** is an **essential building block** for the realization of an **implicit scheme**.

# Conclusions (B)

## For Unsteady Problems

- For problems **dominated by small time scales** and propagation of **discontinuities** the most reliable choice is an **explicit TVD time–stepping scheme**.
- For problems where the **dominant time scales** are much larger than the **acoustic scale implicit schemes** are likely to be more effective.

- The **dual time stepping BDF scheme** is appropriate for **general unsteady problems**.
- The **RK-BDF scheme** is easily **parallelizable** and adaptable to **unstructured grids**. The **SGS-BDF scheme** offers greater **efficiency** but is hard to parallelize.

- These techniques could also be combined with **implicit RK methods**.
- More analysis is needed of the effects of **inexact solution** on both **stability** and **accuracy**.

- The **time-spectral method** offers **optimal efficiency** for **periodic problems**.