

NUMERICAL SIMULATION METHODS
Part 2 - Applied Concepts

Lecture 12: Jameson's Scheme



University of
BRISTOL

Applied Concepts

- Introduction to meshing
- The finite volume method
- **Today: Jameson's scheme**
 - Evaluating fluxes at cell faces
 - Artificial dissipation
 - Runge-Kutta time-stepping scheme
- Next Lecture: Solution storage approaches and their implications
- Advanced implicit methods
- Introduction to computer hardware and high performance computing
- Parallel decomposition and efficiency

Recap: The Finite-Volume Method

We have derived the general finite-volume scheme, which gives us a numerical updating scheme of:

$$\underline{\mathbf{U}}_i^{n+1} = \underline{\mathbf{U}}_i^n - \Delta t \underline{\mathbf{R}}_i \quad \text{where} \quad \underline{\mathbf{R}}_i = \frac{1}{A} \int_c \underline{\mathbf{F}} dy - \underline{\mathbf{G}} dx \quad (1)$$

Recap: The Finite-Volume Method

We have derived the general finite-volume scheme, which gives us a numerical updating scheme of:

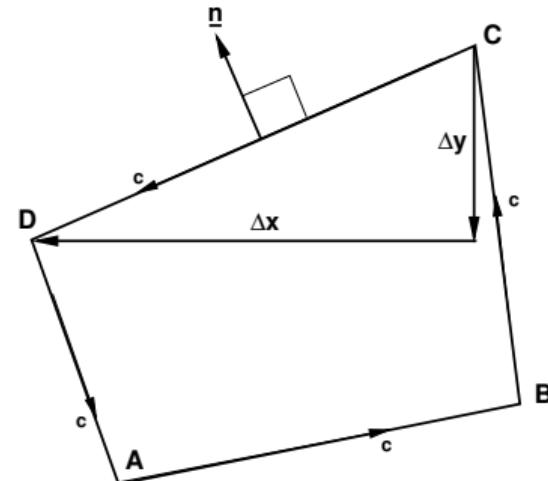
$$\underline{\mathbf{U}}_i^{n+1} = \underline{\mathbf{U}}_i^n - \Delta t \underline{\mathbf{R}}_i \quad \text{where} \quad \underline{\mathbf{R}}_i = \frac{1}{A} \int_c \underline{\mathbf{F}} dy - \underline{\mathbf{G}} dx \quad (1)$$

For a general discrete cell we have

$$\underline{\mathbf{R}}_i = \frac{1}{A} \sum_{k=1}^{N_{\text{face}}} \underline{\mathbf{F}}_k \Delta y_k - \underline{\mathbf{G}}_k \Delta x_k \quad (2)$$

(i is cell counter). Hence we just need to loop over cell faces to compute residual.

But how do we obtain the flux values, $\underline{\mathbf{F}}_k, \underline{\mathbf{G}}_k$, at the cell faces?



Today: Jameson's Scheme

How do we obtain the flux values, $\underline{F}_k, \underline{G}_k$, at the cell faces?

- Today we will look at one method for obtaining face flux values
- A **central discretisation** devised by *Antony Jameson*

Antony Jameson

A ‘big name’ in the field of computational fluid dynamics having made a significant number of contributions to the field.

Today: Jameson's Scheme

How do we obtain the flux values, \underline{F}_k , \underline{G}_k , at the cell faces?

- Today we will look at one method for obtaining face flux values
- A **central discretisation** devised by *Antony Jameson*

Antony Jameson

A ‘big name’ in the field of computational fluid dynamics having made a significant number of contributions to the field.

His 60th Birthday, in 1994, was recognised with an invitation only special conference on CFD methods, with each paper being presented by a world-renowned CFD/aerodynamics researcher. Also, at an international review conference in 1995, a paper was presented on the current status of CFD and past achievements. Of the 200 references, 58 were Jameson’s. Since then he has made further significant contributions, most notably the adjoint approach to optimisation codes.

Contributions to Computational Fluid Dynamics

David A. Caughey¹ and Mohamed M. Hafez²

1.1 Introduction

The contributions of Antony Jameson to computational fluid dynamics (CFD) go far beyond the technical papers and computer codes he has written. In the following, only a brief summary of some of his technical papers will be given. A complete list of his papers (through November 1994) covering CFD as well as other areas is included at the end of this chapter. The list is impressive indeed: 160 items including a publication of his Ph.D. thesis, at Cambridge University, on the generation of standing waves in plasmas in the *Journal of Fluid Mechanics* in 1964, several Hawker-Siddeley technical memoranda, Grumman and NASA reports, as well as his work on control theory and singular perturbations.

Jameson, himself, has documented his original contributions to CFD in several articles; there are at least ten general review papers written by Jameson as a single author. His early work on potential flow was summarized in Von Karman Institute Lecture Series 87, "The Evolution of Computational Methods in Aerodynamics" was published in the 50th Anniversary Issue of the *Journal of Applied Mechanics* in 1983, followed by "Transonic Flow

The Boeing Airplanes that have Benefited from Antony Jameson's CFD Technology

Paul E. Rubbert¹

Over the span of Professor Jameson's career, computational fluid dynamics (CFD) has advanced from a very primitive technology to one that has emerged as a dominant tool in aerodynamic design. It has changed the way by which airplanes are designed. Many of those advances were propelled by the understanding, the algorithm technology, and the codes that were created by Antony Jameson. Today the knowledge and the technology that he has created is known and utilized throughout the world.

One key element that underlies his continuing position of eminence is his outstanding ability to work problems right, to solve the challenging problems in ways that are most practical, useful, and highly valued. The right word is *elegant!* But within the broad perspective of the airplane design community, that is the only ante that lets one into the game.

The other key element that allows one to be a winner at the game is the ability to "work the right problems," to have the perception and understanding that guides one to work on those problems whose solutions will have the largest impact. In this respect, Jameson's track record is truly uncanny. He developed full potential solvers at the time when hardware advances and the perspective of airplane companies had advanced to the point where such technology would be accepted and used. The same is true of his development of Euler solvers, of Navier-Stokes technology, and of his

¹ Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, New York 14853-7501.

² Department of Mechanical and Aerospace Engineering, University of California, Davis, Davis, California 95616.

Frontiers of Computational Fluid Dynamics - 1994
Editors David A. Caughey and Mohamed M. Hafez

©1994 John Wiley & Sons Ltd.

¹ Boeing Commercial Airplane Group, Seattle, Washington 98124-2207.

Frontiers of Computational Fluid Dynamics - 1994
Editors David A. Caughey and Mohamed M. Hafez

©1994 John Wiley & Sons Ltd.

Antony Jameson

RUBBERT

BOEING AIRPLANES

45

cent work on optimization methodology. Along the way he also made many, any algorithm improvements that increased the accuracy and reliability of curacy of CFD, and which reduced the cost and cycle time of performing FD-based airplane design work. Each of those developments served to crease the effectiveness of CFD and to enlarge its contribution to the process designing airplanes.

Antony Jameson has become a legend within the Boeing company, not only thin the ranks of research and design engineering, but also within the ranks management up through the hallways and offices of the executive suite. He s made his mark with them through his many contributions to the processes designing airplanes, and through exposure at countless quarterly reviews er the years wherein Jameson's technology and codes are regularly presented the core technology that underlies many of the new CFD capabilities that e reported in those reviews.

It is in recognition of those many contributions that The Boeing Company, the occasion of his 60th birthday, presents to Antony Jameson a model splaying the many Boeing airplanes whose aerodynamics design was carried it with the aid of CFD technology and codes developed by him. The list of ose airplanes begins with the Boeing 767, designed in the late 1970s. That is followed by the 757, the 747-400, the new Boeing 777, and the recently-nounced Boeing 737-700 which embodies a new wing and other advanced tures. Each of those airplanes is displayed on the model.

Within the spirit of modern airplane design practice, the model also contains om for growth. There is one model position reserved for a future Boeing 787 plane, and another for a 797. Those airplanes are presently only a gleam our eye, but when they are designed and built, they undoubtedly also will ntain the imprint of Jameson's computational methodology.

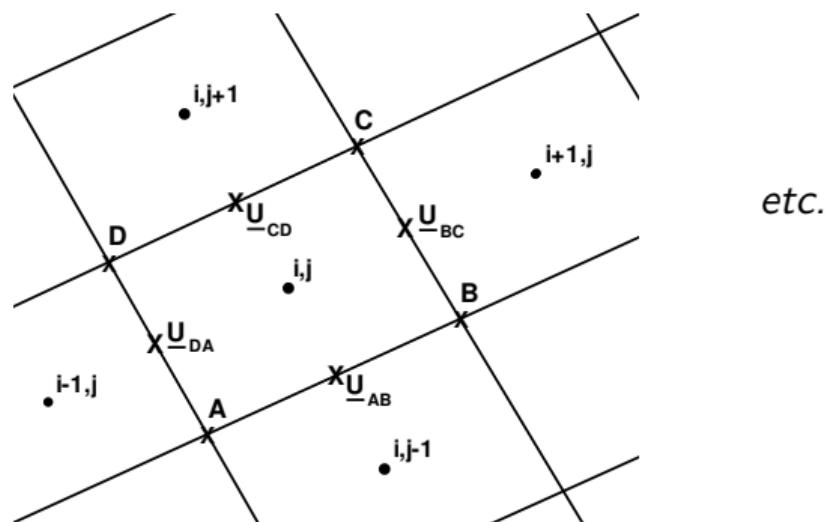
The plaque on the model reads:

Presented to
Antony Jameson
by



Jameson's Central Discretisation

Jameson chose the most simple approximation. He evaluated the solution at each cell face using a simple average of the values either side.



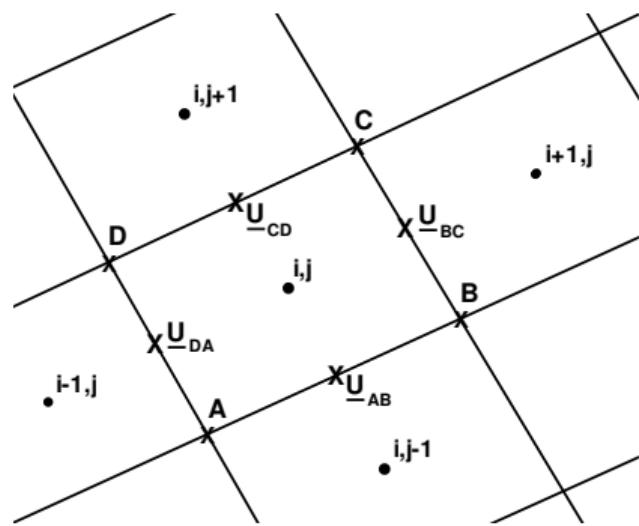
$$\underline{\mathbf{U}}_{BC} = \frac{1}{2}(\underline{\mathbf{U}}_{i,j} + \underline{\mathbf{U}}_{i+1,j}) \quad (3)$$

$$\underline{\mathbf{U}}_{DA} = \frac{1}{2}(\underline{\mathbf{U}}_{i-1,j} + \underline{\mathbf{U}}_{i,j}) \quad (4)$$

etc.

Jameson's Central Discretisation

Jameson chose the most simple approximation. He evaluated the solution at each cell face using a simple average of the values either side.



$$\underline{\mathbf{U}}_{BC} = \frac{1}{2}(\underline{\mathbf{U}}_{i,j} + \underline{\mathbf{U}}_{i+1,j}) \quad (3)$$

$$\underline{\mathbf{U}}_{DA} = \frac{1}{2}(\underline{\mathbf{U}}_{i-1,j} + \underline{\mathbf{U}}_{i,j}) \quad (4)$$

etc.

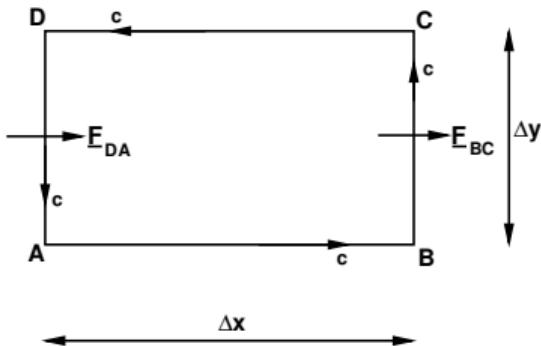
Important

Here we must be careful. The flux functions $\underline{\mathbf{F}}$ and $\underline{\mathbf{G}}$ are non-linear, and so:

$$\underline{\mathbf{F}}\left(\frac{1}{2}(\underline{\mathbf{U}}_{i,j} + \underline{\mathbf{U}}_{i+1,j})\right) \neq \frac{1}{2}(\underline{\mathbf{F}}(\underline{\mathbf{U}}_{i,j}) + \underline{\mathbf{F}}(\underline{\mathbf{U}}_{i+1,j}))$$

Central Discretisation - Cartesian Cell Case

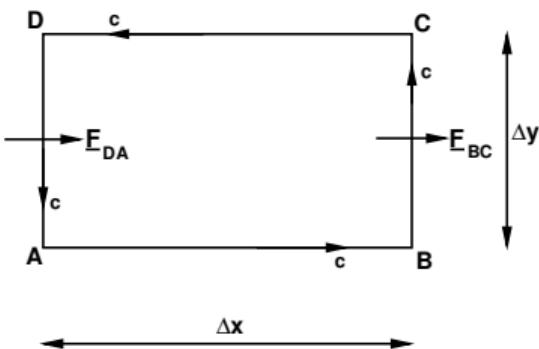
Consider a structured cartesian cell, length Δx , height Δy , with a scalar equation.



Assume $\underline{\mathbf{U}} = u$, $\underline{\mathbf{F}}(\underline{\mathbf{U}}) = cu$
where c is a constant, and
 $\underline{\mathbf{G}}(\underline{\mathbf{U}}) = 0$.

Central Discretisation - Cartesian Cell Case

Consider a structured cartesian cell, length Δx , height Δy , with a scalar equation.



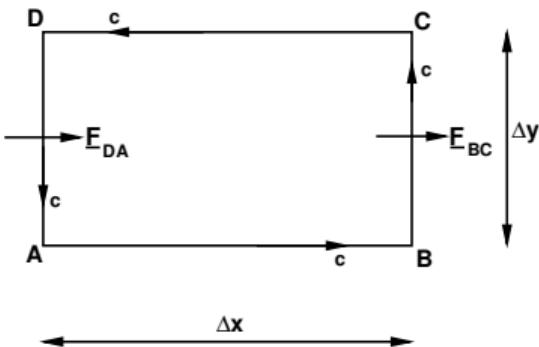
For cartesian cell $A = \Delta x \cdot \Delta y$, so the flux integral then becomes simply:

$$\frac{1}{A} \int_c \underline{\mathbf{F}} dy - \underline{\mathbf{G}} dx = \frac{1}{A} (\underline{\mathbf{F}}_{BC} \Delta y - \underline{\mathbf{F}}_{DA} \Delta y)$$

Assume $\underline{\mathbf{U}} = u$, $\underline{\mathbf{F}}(\underline{\mathbf{U}}) = cu$
where c is a constant, and
 $\underline{\mathbf{G}}(\underline{\mathbf{U}}) = 0$.

Central Discretisation - Cartesian Cell Case

Consider a structured cartesian cell, length Δx , height Δy , with a scalar equation.



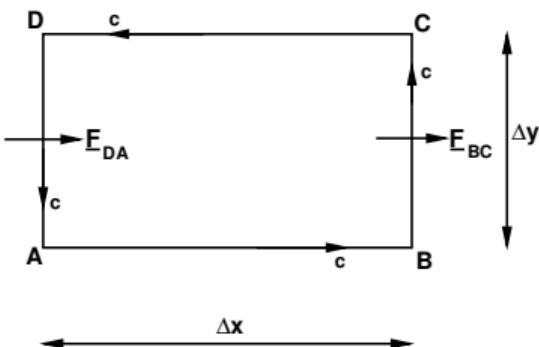
For cartesian cell $A = \Delta x \cdot \Delta y$, so the flux integral then becomes simply:

$$\frac{1}{A} \int_c \underline{\mathbf{F}} dy - \underline{\mathbf{G}} dx = \frac{1}{A} (\underline{\mathbf{F}}_{BC} \Delta y - \underline{\mathbf{F}}_{DA} \Delta y) = \frac{\underline{\mathbf{F}}_{BC} - \underline{\mathbf{F}}_{DA}}{\Delta x}$$

Assume $\underline{\mathbf{U}} = u$, $\underline{\mathbf{F}}(\underline{\mathbf{U}}) = cu$
where c is a constant, and
 $\underline{\mathbf{G}}(\underline{\mathbf{U}}) = 0$.

Central Discretisation - Cartesian Cell Case

Consider a structured cartesian cell, length Δx , height Δy , with a scalar equation.



For cartesian cell $A = \Delta x \cdot \Delta y$, so the flux integral then becomes simply:

$$\frac{1}{A} \int_c \underline{\mathbf{F}} dy - \underline{\mathbf{G}} dx = \frac{1}{A} (\underline{\mathbf{F}}_{BC} \Delta y - \underline{\mathbf{F}}_{DA} \Delta y) = \frac{\underline{\mathbf{F}}_{BC} - \underline{\mathbf{F}}_{DA}}{\Delta x}$$

Evaluating the fluxes at the faces:

$$= \frac{\underline{\mathbf{F}}_{BC} - \underline{\mathbf{F}}_{DA}}{\Delta x} = \frac{\frac{1}{2}(cu_{i+1,j}^n + cu_{i,j}^n) - \frac{1}{2}(cu_{i,j}^n + cu_{i-1,j}^n)}{\Delta x}$$

(Consider an explicit scheme, so residual evaluated at n .)

Assume $\underline{\mathbf{U}} = u$, $\underline{\mathbf{F}}(\underline{\mathbf{U}}) = cu$ where c is a constant, and $\underline{\mathbf{G}}(\underline{\mathbf{U}}) = 0$.

Jameson's Central Discretisation - Cartesian Cell Case

From our flux integration, we have:

$$\begin{aligned} R_{i,j} &= \frac{\frac{1}{2}(cu_{i+1,j}^n + cu_{i,j}^n) - \frac{1}{2}(cu_{i,j}^n + cu_{i-1,j}^n)}{\Delta x} \\ &= c \frac{(u_{i+1,j}^n - u_{i-1,j}^n)}{2\Delta x} \end{aligned} \tag{5}$$

Jameson's Central Discretisation - Cartesian Cell Case

From our flux integration, we have:

$$\begin{aligned} R_{i,j} &= \frac{\frac{1}{2}(cu_{i+1,j}^n + cu_{i,j}^n) - \frac{1}{2}(cu_{i,j}^n + cu_{i-1,j}^n)}{\Delta x} \\ &= c \frac{(u_{i+1,j}^n - u_{i-1,j}^n)}{2\Delta x} \end{aligned} \tag{5}$$

Now using a first- order explicit approximation for the temporal derivative (as usual) would give:

$$u_{i,j}^{n+1} = u_{i,j}^n - \frac{c\Delta t}{2\Delta x} (u_{i+1,j}^n - u_{i-1,j}^n) \tag{6}$$

Jameson's Central Discretisation - Cartesian Cell Case

From our flux integration, we have:

$$\begin{aligned} R_{i,j} &= \frac{\frac{1}{2}(cu_{i+1,j}^n + cu_{i,j}^n) - \frac{1}{2}(cu_{i,j}^n + cu_{i-1,j}^n)}{\Delta x} \\ &= c \frac{(u_{i+1,j}^n - u_{i-1,j}^n)}{2\Delta x} \end{aligned} \tag{5}$$

Now using a first-order explicit approximation for the temporal derivative (as usual) would give:

$$u_{i,j}^{n+1} = u_{i,j}^n - \frac{c\Delta t}{2\Delta x} (u_{i+1,j}^n - u_{i-1,j}^n) \tag{6}$$

But this is the FTCS finite-difference analogue for

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0$$

and this is unconditionally unstable.

Jameson's Central Discretisation - Artificial Dissipation

The basic Jameson method (cell-centred, explicit, finite-volume) is now a very famous classical finite-volume method.

- The idea is simple but the basic method **is unstable**, and so to make it work in practise requires the **addition of artificial dissipation** (damping).
- Jameson spent a huge amount of time perfecting the dissipation. He added two types of dissipation:

Jameson's Central Discretisation - Artificial Dissipation

The basic Jameson method (cell-centred, explicit, finite-volume) is now a very famous classical finite-volume method.

- The idea is simple but the basic method **is unstable**, and so to make it work in practise requires the **addition of artificial dissipation** (damping).
- Jameson spent a huge amount of time perfecting the dissipation. He added two types of dissipation:

(1) Background 4th order dissipation: used always at all points in the flow. ($\epsilon_4 = \frac{1}{256}$.)

$$\epsilon_4 \Delta x^4 \frac{\partial^4 u}{\partial x^4} \quad (7)$$

Jameson's Central Discretisation - Artificial Dissipation

The basic Jameson method (cell-centred, explicit, finite-volume) is now a very famous classical finite-volume method.

- The idea is simple but the basic method **is unstable**, and so to make it work in practise requires the **addition of artificial dissipation** (damping).
- Jameson spent a huge amount of time perfecting the dissipation. He added two types of dissipation:

(1) Background 4th order dissipation: used always at all points in the flow. ($\epsilon_4 = \frac{1}{256}$.)

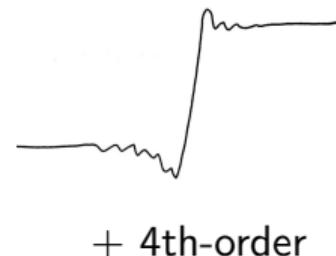
$$\epsilon_4 \Delta x^4 \frac{\partial^4 u}{\partial x^4} \quad (7)$$

(2) A 2nd-order term: used only at large gradients

$$\epsilon_2 \Delta x^2 \frac{\partial^2 u}{\partial x^2} \quad (8)$$

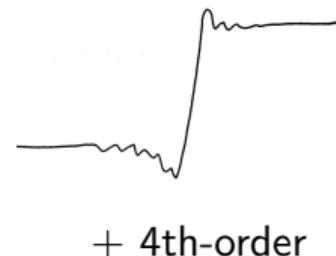
Jameson's Central Discretisation - Artificial Dissipation

(1) Background 4th order dissipation: used always at all points in the flow. This is sufficient to stabilise the flow, however large gradients (shocks) cause 'wiggles' in the solution.

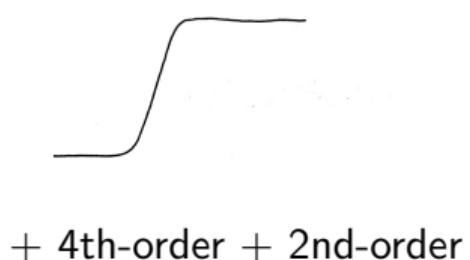


Jameson's Central Discretisation - Artificial Dissipation

(1) Background 4th order dissipation: used always at all points in the flow. This is sufficient to stabilise the flow, however large gradients (shocks) cause 'wiggles' in the solution.



(2) A 2nd-order term: has a strong smoothing effect which is undesirable in most of the flow. Jameson's 'trick' was to scale this with the local pressure gradient. This is large around a shock wave, but small everywhere else, so ϵ_2 acts as a 'switch' to turn on the 2nd-order dissipation where it is needed.



```

!$OMP PARALLEL REDUCTION(+:LAP) PRIVATE(ed,WL,WR)    SHARED(LAP)
do edcnt=1,nedgei
    ed=edgei(edcnt)
    WL=W1(edge(ed,1),:)
    WR=W1(edge(ed,2),:)
    LAP(edge(ed,1),:) = LAP(edge(ed,1),:) + (WR-WL)
    LAP(edge(ed,2),:) = LAP(edge(ed,2),:) - (WR-WL)
enddo
 !$OMP END PARALLEL

```

```

do edcnt=1,nedgei

    ed=edgei(edcnt)

    WL=Wl(edge(ed,1),:)
    WR=Wl(edge(ed,2),:)

    c01=0.5*(sqrt(gam*p(edge(ed,1))/rho(edge(ed,1)))+
& sqrt(gam*p(edge(ed,2))/rho(edge(ed,2)))) 

    u0lx=0.5*(u(edge(ed,1))+u(edge(ed,2)))
    u0ly=0.5*(v(edge(ed,1))+v(edge(ed,2)))
    u0lz=0.5*(w(edge(ed,1))+w(edge(ed,2)))

    vel(1)=u0lx
    vel(2)=u0ly
    vel(3)=u0lz
    norml=norm(ed,:)

    call DOTPROD(dimnum,norml,vel,vnorm)

    lam01=(abs(vnorm)+c01)*cfa(ed)
    psi01=lam(edge(ed,1))/(4*lam01)
    psil1=lam(edge(ed,2))/(4*lam01)
    psi01=4*psil1*psi0/(psi0+psil1)
    if(psi01.le.0.0)then
        print *, 'warning no diss edge',ed
        stop
    endif
    !if(psi01.eq.0.0)then
    !    print *, 'warning no diss edge',ed
    !endif

    s2=3.0*((m(edge(ed,1))+m(edge(ed,2)))/
    & (m(edge(ed,1))*m(edge(ed,2))))
    s4=(s2**2)/4.0

    alp1=k1*0.5*(pdss(edge(ed,1))+pdss(edge(ed,2)))*s2
    alp2=max(0.0,k2-alp1)*s4

    DISSV=((alp1*(WL(:)-WR(:))-alp2*(
    & (LAP(edge(ed,1),:)-LAP(edge(ed,2),:))))*psi01*lam01

    D(edge(ed,1),:)=D(edge(ed,1),:)+DISSV
    D(edge(ed,2),:)=D(edge(ed,2),:)-DISSV

enddo

```

Jameson's Time-Stepping Scheme

Jameson also introduced a different approach to time-stepping. Instead of the simple first-order time-stepping scheme, he used a multi-stage Runge-Kutta scheme.

Multi-stage Runge-Kutta Scheme

One or more intermediate time-steps are performed between time-levels n to $n + 1$. The solution at $n + 1$ is calculated from some weighted average of the intermediate solutions.

Jameson's Time-Stepping Scheme

Jameson also introduced a different approach to time-stepping. Instead of the simple first-order time-stepping scheme, he used a multi-stage Runge-Kutta scheme.

Multi-stage Runge-Kutta Scheme

One or more intermediate time-steps are performed between time-levels n to $n + 1$. The solution at $n + 1$ is calculated from some weighted average of the intermediate solutions.

Jameson chose a four-stage Runge-Kutta scheme (four intermediate time-steps between n to $n + 1$).

- The RK4 scheme **increases the temporal accuracy** of the scheme and **increases the useable CFL number**
- Jameson simplified the RK4 scheme to not require storing all the intermediate solutions (which increases memory requirements)

Jameson's Four-Stage Temporal Scheme

Jameson's simplified temporal scheme only requires the solution to be stored at two levels.

1-D equation example:

$$\underline{\mathbf{U}}_i^{n+\frac{1}{4}} = \underline{\mathbf{U}}_i^n - \frac{\Delta t}{4} \underline{\mathbf{R}}_i(\underline{\mathbf{U}}_i^n)$$

$$\underline{\mathbf{U}}_i^{n+\frac{1}{3}} = \underline{\mathbf{U}}_i^n - \frac{\Delta t}{3} \underline{\mathbf{R}}_i(\underline{\mathbf{U}}_i^{n+\frac{1}{4}})$$

$$\underline{\mathbf{U}}_i^{n+\frac{1}{2}} = \underline{\mathbf{U}}_i^n - \frac{\Delta t}{2} \underline{\mathbf{R}}_i(\underline{\mathbf{U}}_i^{n+\frac{1}{3}})$$

$$\underline{\mathbf{U}}_i^{n+1} = \underline{\mathbf{U}}_i^n - \Delta t \underline{\mathbf{R}}_i(\underline{\mathbf{U}}_i^{n+\frac{1}{2}})$$

Jameson's Four-Stage Temporal Scheme

Jameson's simplified temporal scheme only requires the solution to be stored at two levels.

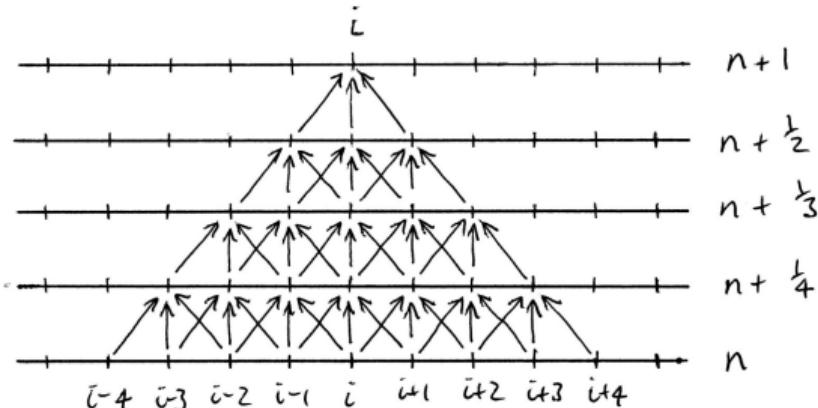
1-D equation example:

$$\underline{\mathbf{U}}_i^{n+\frac{1}{4}} = \underline{\mathbf{U}}_i^n - \frac{\Delta t}{4} \underline{\mathbf{R}}_i(\underline{\mathbf{U}}_i^n)$$

$$\underline{\mathbf{U}}_i^{n+\frac{1}{3}} = \underline{\mathbf{U}}_i^n - \frac{\Delta t}{3} \underline{\mathbf{R}}_i(\underline{\mathbf{U}}_i^{n+\frac{1}{4}})$$

$$\underline{\mathbf{U}}_i^{n+\frac{1}{2}} = \underline{\mathbf{U}}_i^n - \frac{\Delta t}{2} \underline{\mathbf{R}}_i(\underline{\mathbf{U}}_i^{n+\frac{1}{3}})$$

$$\underline{\mathbf{U}}_i^{n+1} = \underline{\mathbf{U}}_i^n - \Delta t \underline{\mathbf{R}}_i(\underline{\mathbf{U}}_i^{n+\frac{1}{2}})$$



- The simplified four-stage scheme is **2nd-order accurate** in time and has a maximum CFL number of $2\sqrt{2}$

```

do nstage=4,1,-1
    call FLUX(ncell,nedge,edge,rho,u,v,w,E,p,fvel,R,
&             Runiv,machinf,rhoinf,Tinf,gam,aoa,norm,cfa,nedgei,nedgeb,
&             edgei,edgeb)

    if(dsf(nstage).eq.1)then
        call DISS(ncell,nedge,Wl,edge,rho,u,v,w,p,
&             gam,k1,k2,D,m,lam,nstage,norm,cfa,nedgei,nedgeb,
&             edgei,edgeb)
    endif

    if(iunflag.eq.1)then
        !print *, Wn-Wnml
        !Wn=Wl
        !Wnml=Wl
        call FLUX2(ncell,fnpl,fn,fnml,Wl,Wn,Wnml,
&             vol,voln,volnml,RT)
        !print *, RT
        ! pause
    else
        RT=0.0
    endif

    !print *, "res"
    !print *, R

    !$OMP PARALLEL DO SCHEDULE(STATIC)
    do nc=1,ncell
        ! print *, R(nc,:)
        Wl(nc,:)=W0(nc,:)-(deltat(nc)*
&             (R(nc,:)+D(nc,:)+RT(nc,:)))/(vol(nc)*nstage)
    enddo
    !$OMP END PARALLEL DO

    call PRIM(ncell,Wl,gam,rho,u,v,w,E,p)

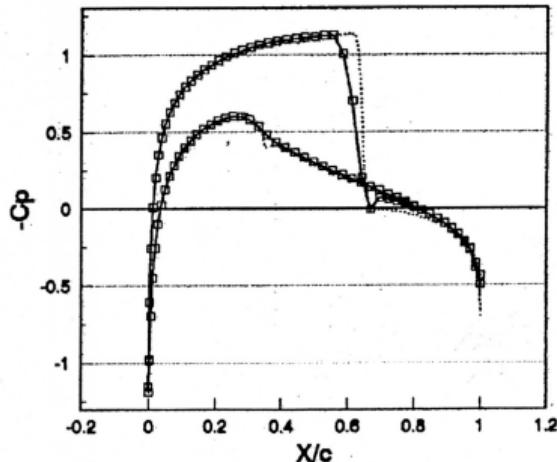
enddo

```

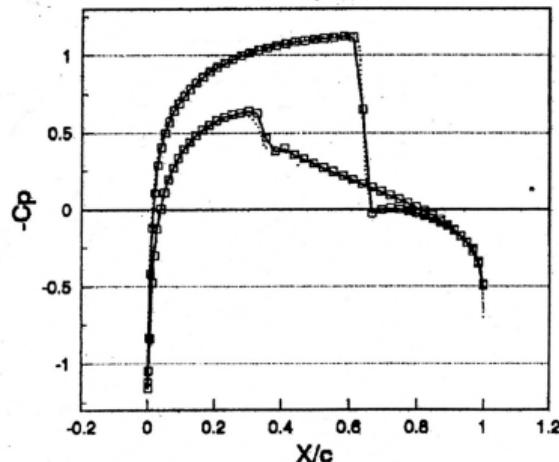
Jameson's Scheme - Comparison

The two difference schemes were first used to compute a steady flow. The standard test case of a NACA 0012 aerofoil at 1.25° incidence in a flow of freestream Mach number 0.8 was used. The grid used consisted of 129×25 points with 99 points on the aero-

the smallest value over the whole computational domain. Also the time-step evaluation is slightly different, requiring the inclusion of the grid speeds. The same grid was used as for the steady computations; 129×25 points with 99 points on the aerofoil, and the farfield boundary being approximately 25 chords from the aerofoil.



Central-Difference Ref. Solution



Upwind-Difference Ref. Solution

Figure 5. Steady pressure distributions over NACA0012 aerofoil at $\alpha = 1.25^\circ$, $M = 0.8$.

Final perspectives

- ‘Jameson’s scheme’ I do not really consider a single scheme, because I don’t think anyone has ever implemented it twice in exactly the same way. Various adjustments are needed on stretched boundary layer cells, and unstructured meshes. People do these differently depending on their needs
- It evolved from his earlier efforts on full potential methods. It didn’t come out of the blue
- Second order upwind methods using Roe or van Leer methods are more consistent between implementations, and probably just as common or more so, and they’re used with higher order methods too
- Jameson’s scheme and upwind methods are linked - as we mentioned when looking at subtracting central and upwinded terms for the 1D equation. They are relatives in a single family
- The RK integration is applicable to any explicit scheme

- Jameson's success was in working out a fairly simple method that worked really well nearly all the time. People are always drawn to simpler, effective ideas. **This is something to remember whatever you do in your future careers**
- Personally I think his dual-time unsteady work is just as or more important (my bias...)
- He also lived at the right time and made excellent career and location choices, including moving to the USA
- What's the next computational frontier?

Summary

- Finite Volume Method
 - A more general approach to solving PDEs (can use unstructured meshes)
 - Integrate fluxes through cell faces
 - Need to evaluate fluxes at the cell faces
- Jameson's Scheme
 - A **central scheme**: face solution is average of neighbouring cell solutions (very simple)
 - Need **artificial dissipation** to stabilise the basic scheme: 4th-order background + 2nd-order at shocks
 - Use low-memory **four-stage Runge-Kutta** scheme for time-stepping: increased accuracy and CFL

Next Lecture: Where is the solution stored? At the cell centres or at the mesh nodes?