



INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad -500 043

AERONAUTICAL ENGINEERING

IV B. Tech I semester (JNTUH-R15)

ADVANCED COMPUTATIONAL

AERODYNAMICS

Prepared by

D. Anitha,

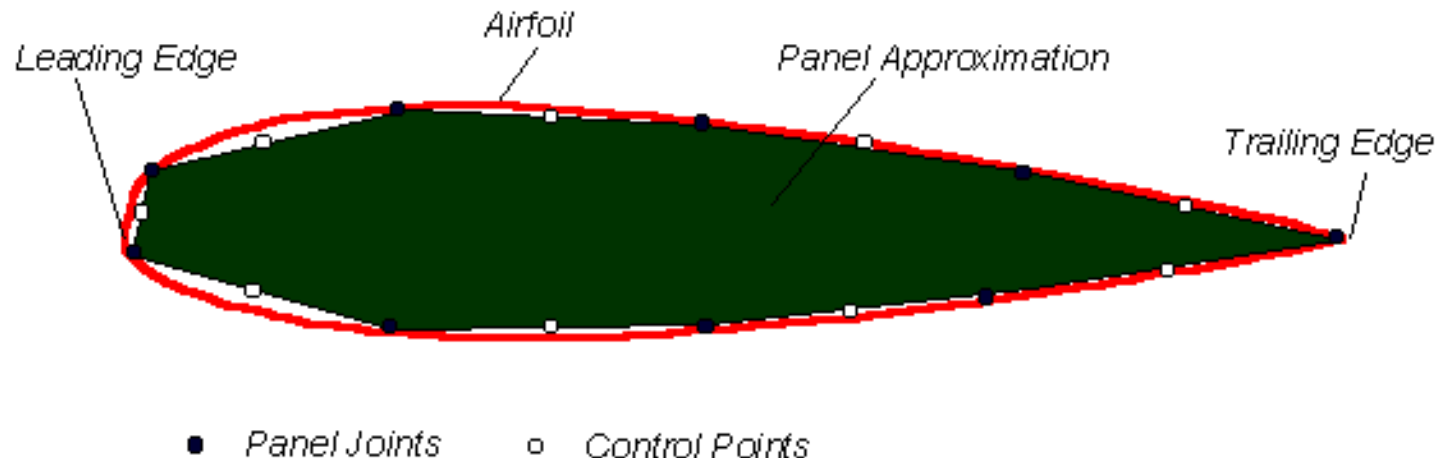
Assistant Professor, AE

UNIT - I

PANEL METHODS

What are panel methods?

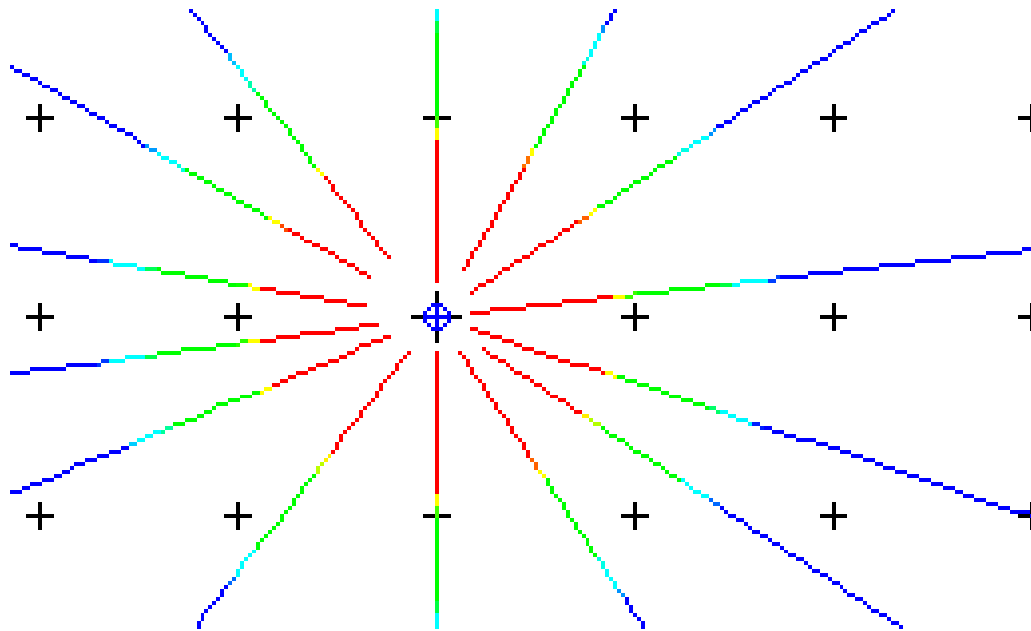
- Panel methods are techniques for solving incompressible potential flow over thick 2-D and 3-D geometries.
- In 2-D, the airfoil surface is divided into piecewise straight line segments.
- These line segments are called as panels.
- Either source sheets of strength m or vortex sheets of strength γ are placed on each panel.
- Vortex sheets mimic the boundary layer around airfoils.



The Point Source

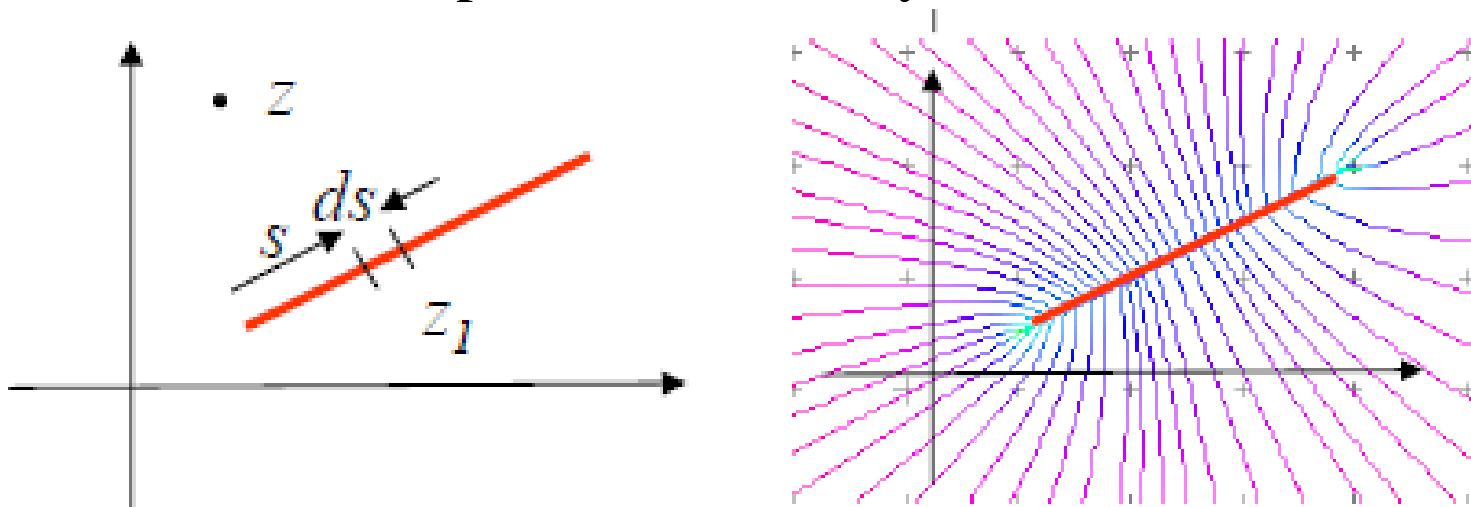
Consider a point source

$$W = \phi + i\psi = \frac{m}{2\pi} \ln z$$



The Source Panel

- Imagine spreading the source along a line.
- Distribute the source as certain strength per unit length $q(s)$ that could vary with distance s along the line.
- Each elemental length of the line ds would behave like a *miniature source* and so would produce a velocity field:



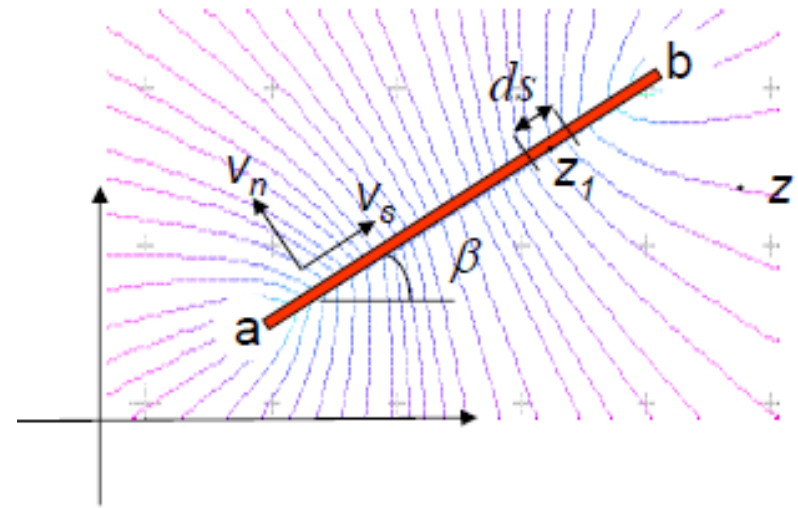
Where $z_I(s)$, gives the coordinate of the line source at s .
The total flow field produced by the line is the strength of the panel.

Uniform Strength Source Panel

$$W(z) = \frac{1}{2\pi} \int_{\text{panel}} \frac{q(s)ds}{z - z_1(s)}$$

If strength q is constant with s then

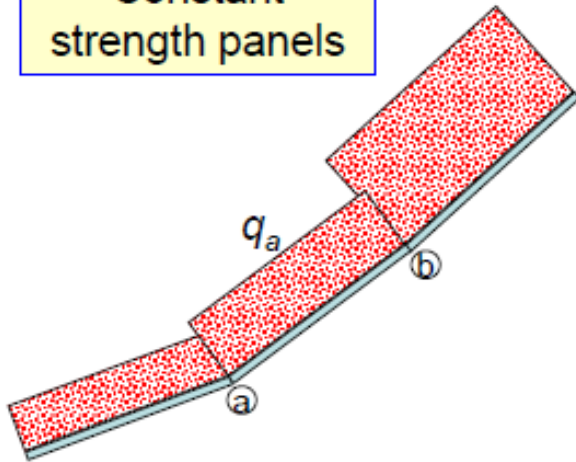
$$W(z) = \frac{q}{2\pi} \int_{\text{panel}} \frac{ds}{z - z_1(s)}$$



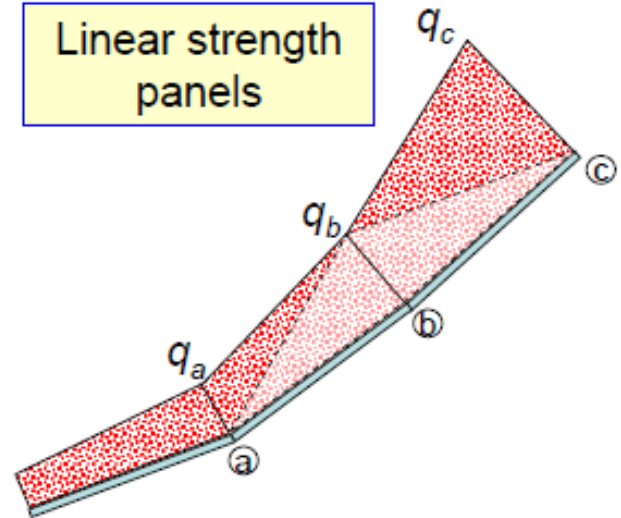
- The velocity in terms of components aligned with the panel:
- The panel is not a solid boundary.
 - To make it behave like a solid boundary in a flow you would have to set the strength q so that the total V_n (due to the panel and the flow) is zero
- $$V_s - iV_n = \frac{q}{2\pi} \ln \left[\frac{z - z_a}{z - z_b} \right]$$

Linear Source Panels

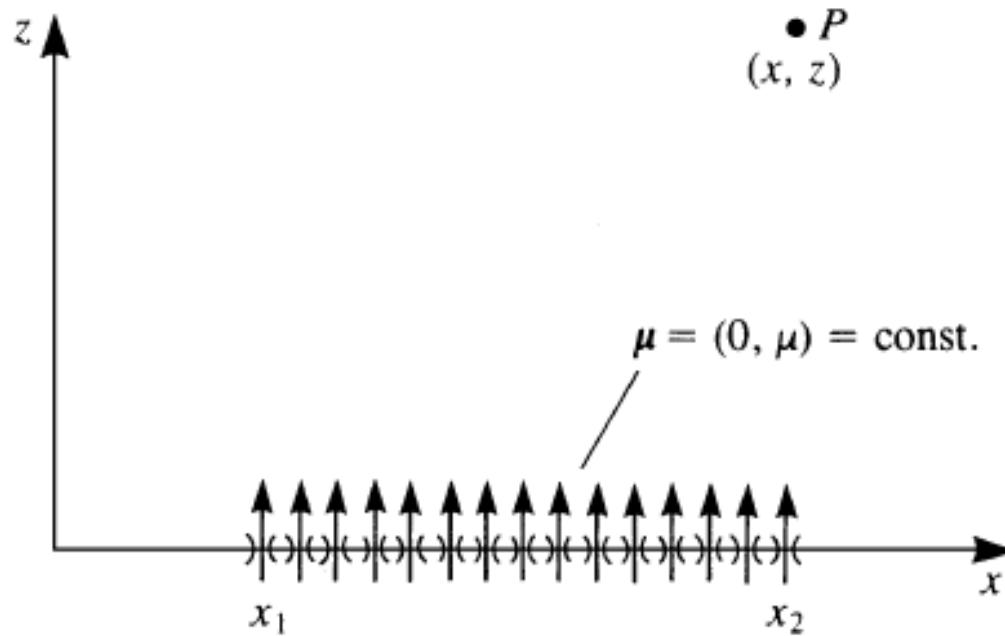
Constant
strength panels



Linear strength
panels



Constant-Strength Doublet Distribution



Uniform Strength Vortex Panel

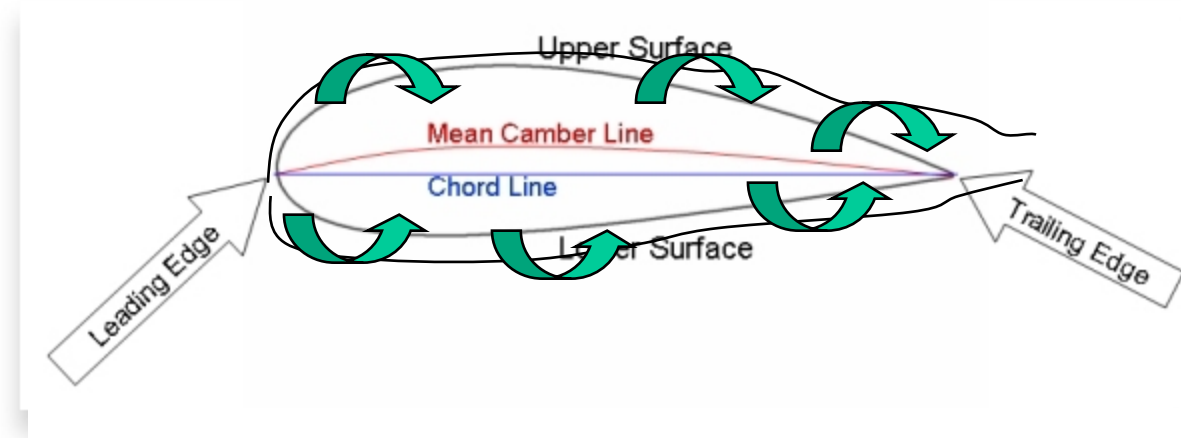
- The integral vortex panel is used to perturb the uniform flow to create a possible lifting body.

$$\psi = Uz - \oint \frac{\gamma_0}{2\pi} \ln(r - r_0) ds_0$$

- This is the stream function at any point in 2-D space.
- Panel methods are evaluated only at the closed streamline.
- This may be the surface of the would be airfoil as the body surface is a streamline.
- Therefore, the stream function value will be some constant value C .

$$\psi = Uz - \oint \frac{\gamma_0}{2\pi} \ln(r - r_0) ds_0 - C = 0$$

Analogy between boundary layer and vortices



Upper surface boundary layer contains, in general, clockwise rotating vorticity.

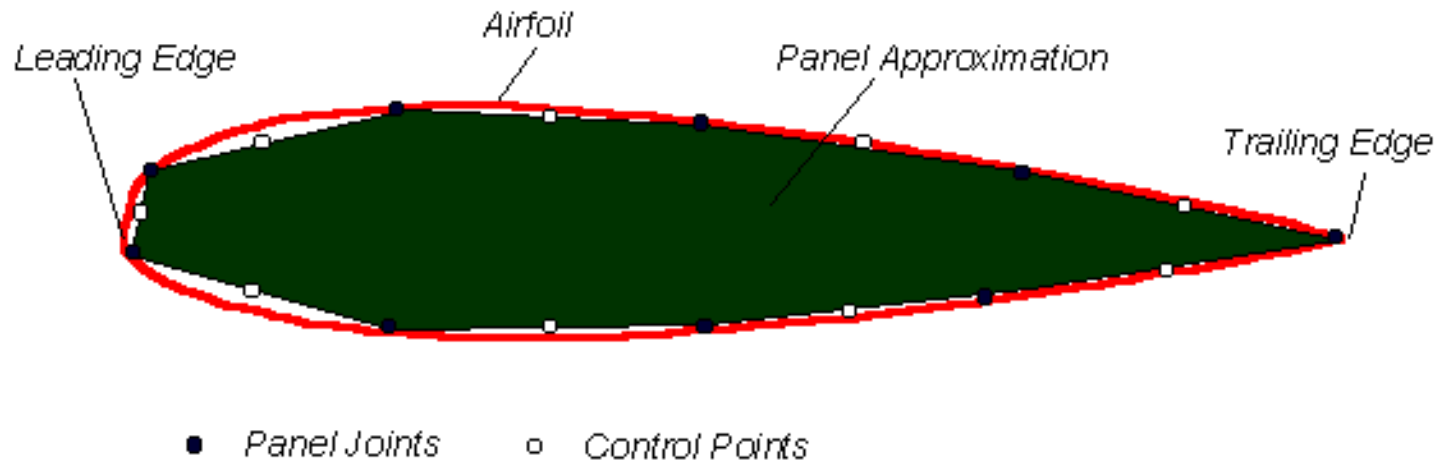
Lower surface boundary layer contains, in general, counter clockwise vorticity.

Because there is more clockwise vorticity than counter clockwise

Vorticity, there is net clockwise circulation around the airfoil.

In panel methods, this boundary layer, which has a small but finite thickness is replaced with a thin sheet of vorticity placed just outside the airfoil.

Panel method treats the airfoil as a series of line segments



On each panel, there is vortex sheet of strength $\Delta\Gamma = \gamma_0 ds_0$.

Where ds_0 is the panel length.

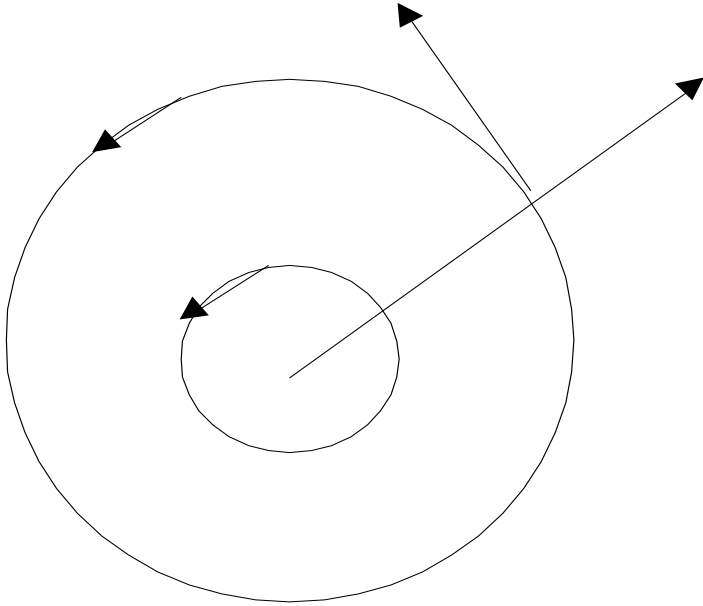
Each panel is defined by its two end points (panel joints) and by the control point, located at the panel center, where we will apply the boundary condition $\psi = \text{Constant} = C$.

The more the number of panels, the more accurate the solution, since we are representing a continuous curve by a series of broken straight lines

Boundary Condition

- We treat the airfoil surface as a streamline.
 - This ensures that the velocity is tangential to the airfoil surface, and no fluid can penetrate the surface.
- We require that at all control points (middle points of each panel) $\psi = C$
- The stream function is due to superposition of the effects of the free stream and the effects of the vortices $\gamma_0 ds_0$ on each of the panel.

Stream function due to a Counterclockwise Vortex of Strength Γ



$$v_r = 0$$

$$v_\theta = \frac{\Gamma}{2\pi r}$$

$$\text{Circulation} = \oint v_\theta r d\theta = \Gamma$$

$$\psi = -\frac{\Gamma}{2\pi} \ln(r)$$

$$v_r = \frac{1}{r} \frac{\partial \psi}{\partial \theta} = 0$$

$$v_\theta = -\frac{\partial \psi}{\partial r} = \frac{\Gamma}{2\pi r}$$

Stream function Vortex Panel

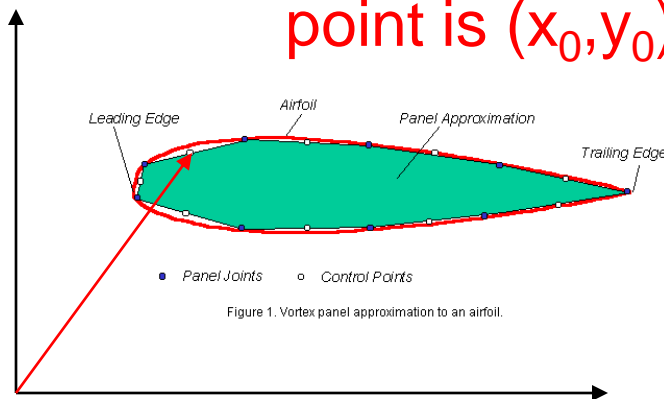
- Pay attention to the signs.
- A counter-clockwise vortex is considered “positive”
- In our case, the vortex of strength $\gamma_0 ds_0$ had been placed on a panel with location $(x_0$ and $y_0)$.
- Then the stream function at a point (x, y) will be

$$\psi = - \frac{\gamma_0 ds_0}{2\pi} \ln \left(\left| \bar{r} - \bar{r}_o \right| \right)$$

where

$$\left(\left| \bar{r} - \bar{r}_o \right| \right) = \sqrt{(x - x_0)^2 + (y - y_0)^2}$$

Panel whose center point is (x_0, y_0)



Superposition of All Vortices on all Panels

- In the panel method we use here, ds_0 is the length of a small segment of the airfoil, and γ_0 is the vortex strength per unit length.
- Then, the stream function due to all such infinitesimal vortices at the control point (located in the middle of each panel) may be written as the interval below, where the integral is done over all the vortex elements on the airfoil surface.

$$- \oint \frac{\gamma_0}{2\pi} \ln (\bar{r} - r_0) ds_0$$

Adding the free stream and vortex effects..

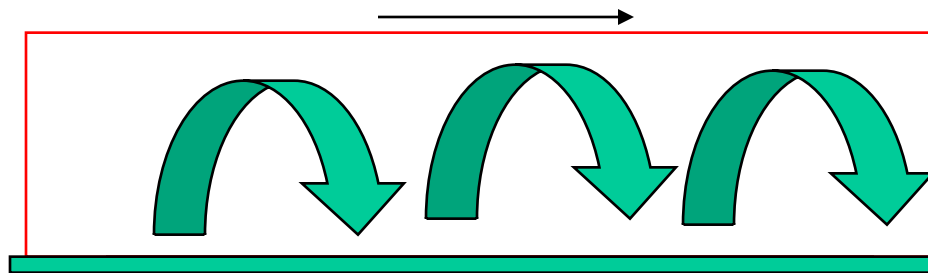
$$u_{\infty} y - v_{\infty} x - \frac{1}{2\pi} \oint \gamma_0 \ln \left(\left| \bar{r} - \bar{r}_0 \right| \right) ds_o = C$$

The unknowns are the vortex strength γ_0 on each panel, and the value of the Stream function C.

Before we go to the trouble of solving for γ_0 , we ask what is the purpose..

Physical meaning of γ_0

V = Velocity of the flow just outside the boundary layer



Panel of length ds_0 on the airfoil.

Sides of our contour have zero height.
Bottom side has zero Tangential velocity
Because of viscosity

$$\text{Circulation } \oint_{\text{Contour}} \vec{V} \cdot d\vec{s} = \gamma_0 ds_0 = -V ds_0$$

$$\text{Or , } V = -\gamma_0$$

If we know γ_0 on each panel, then we know the velocity of the flow outside the boundary layer for that panel, and hence pressure over that panel.

Pressure distribution and Loads

Bernoulli says : $p + \frac{1}{2} \rho (u^2 + v^2) = p_{\infty} + \frac{1}{2} \rho_{\infty} V_{\infty}^2$

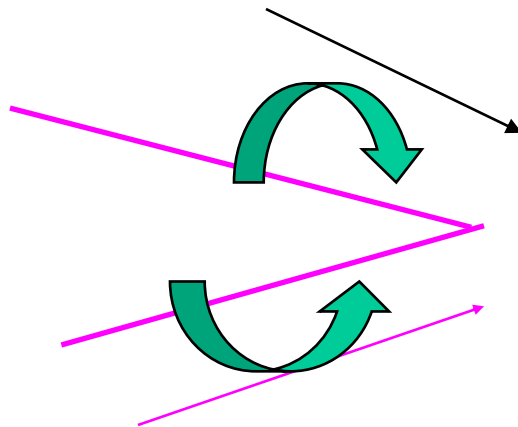
$$C_p = \frac{p - p_{\infty}}{\frac{1}{2} \rho_{\infty} V_{\infty}^2} = 1 - \frac{u^2 + v^2}{V_{\infty}^2} = 1 - \frac{V^2}{V_{\infty}^2}$$

Since $V = -\gamma_0$

$$C_p = 1 - \frac{\gamma_0^2}{V_{\infty}^2}$$

Kutta Condition

- Kutta condition states that the pressure above and below the airfoil trailing edge must be equal, and that the flow must smoothly leave the trailing edge in the same direction at the upper and lower edge.



$$\begin{aligned}\gamma_{\text{upper}}^2 &= V_{\text{upper}}^2 \\ \gamma_{\text{lower}}^2 &= V_{\text{lower}}^2\end{aligned}$$

$$\gamma_{\text{Upper}} = -\gamma_{\text{lower}}$$

From this sketch above, we see that pressure will be equal, and the flow will leave the trailing edge smoothly, only if the vorticity on each panel is equal in magnitude above and below, but spinning in opposite directions relative to each other.

The Closure

- We need to solve the integral equation derived earlier
- And, satisfy Kutta condition.

$$u_{\infty} y - v_{\infty} x - \frac{1}{2\pi} \oint \gamma_0 \ln \left(\left| \bar{r} - \bar{r}_0 \right| \right) ds_o = C$$

$$\gamma_{Upper} = -\gamma_{lower}$$

Numerical Procedure

- We divide the airfoil into N panels. A typical panel is given the number j, where J varies from 1 to N.
- On each panel, we assume that γ_0 is a piecewise constant. Thus, on a panel numbered j, the unknown strength is $\gamma_{0,j}$
- We number the control points at the centers of each panel as well. Each control point is given the symbol “i”, where i varies from 1 to N.
- The integral equation becomes

$$u_{\infty} y_i - v_{\infty} x_i - \sum_{j=1..N} \frac{\gamma_{0,j}}{2\pi} \int_j \ln(|\bar{r}_i - \bar{r}_0|) ds_o - C = 0$$

Numerical procedure

- Notice that we use two indices ‘i’ and ‘j’. The index ‘i’ refers to the control point where equation is applied.
- The index ‘j’ refers to the panel over which the line integral is evaluated.
- The integrals over the individual panels depends only on the panel shape (straight line segment), its end points and the control point i’.
- Therefore this integral may be computed analytically.
- We refer to the resulting quantity as

$$u_{\infty} y_i - v_{\infty} x_i - \sum_{j=1}^N A_{i,j} \gamma_{0j} - C = 0$$

where ,

$$A_{i,j} = \text{Influence of Panel } j \text{ on index } i = \frac{1}{2\pi} \int \ln \left(\left| \bar{r}_i - \bar{r}_0 \right| \right) ds_0 \quad 22$$

Numerical procedure

- We thus have $N+1$ equations for the unknowns $\gamma_{0,j}$ ($j=1 \dots N$) and C .
- We assume that the first panel ($j=1$) and last panel ($j=N$) are on the lower and upper surface trailing edges.

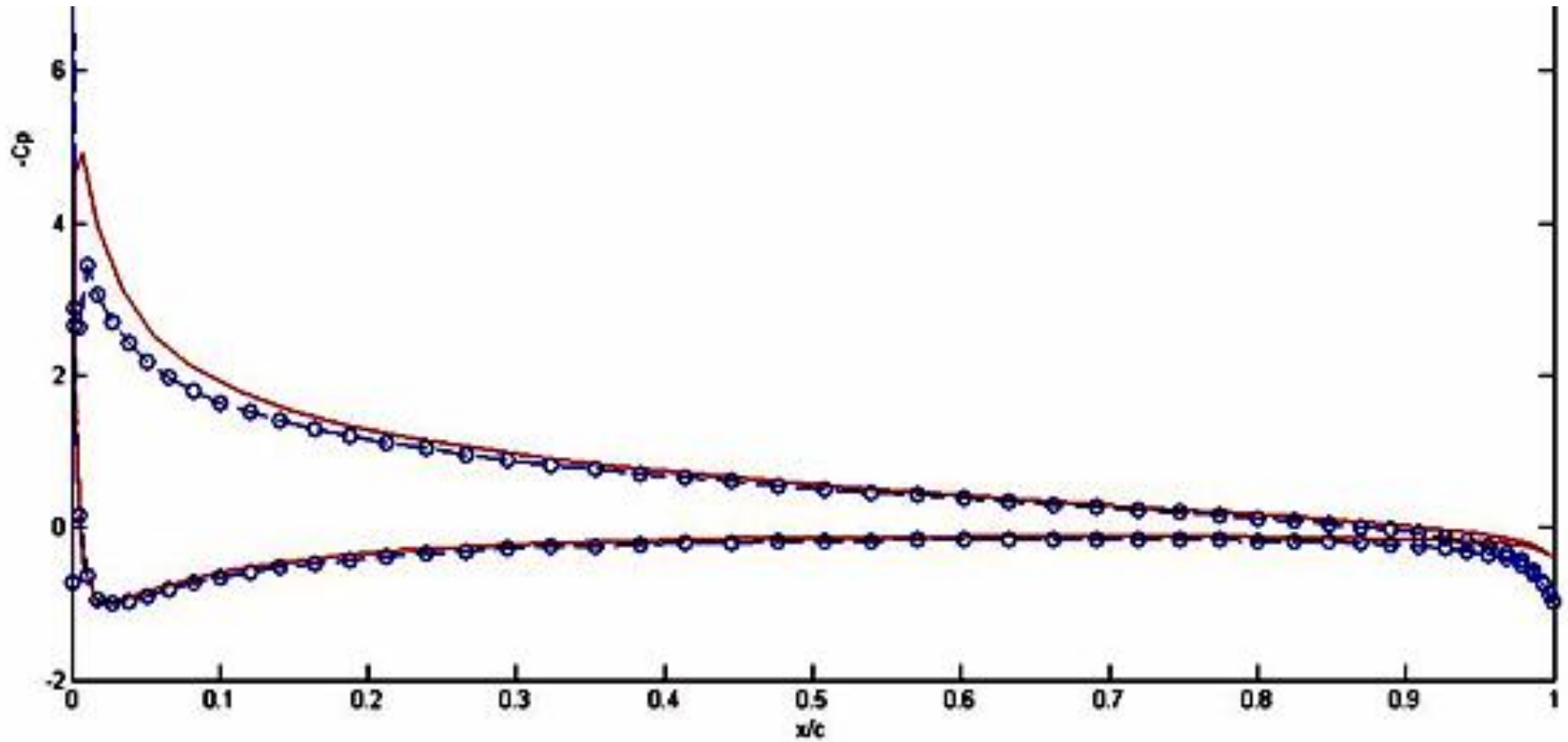
$$u_{\infty} y_i - v_{\infty} x_i - \sum_{j=1}^N A_{i,j} \gamma_{0,j} - C = 0$$

$$\gamma_{0,1} = -\gamma_{0,N}$$

This linear set of equations may be solved easily, and γ_0 found.

Once γ_0 is known, we can find pressure, and pressure coefficient C_p .

An Useful Aerofoil



Lifting Bodies in Subsonic Compressible Flows

- The Velocity Potential function is also valid for compressible isentropic subsonic flows.

$$\vec{V} = \nabla \phi$$

From this velocity potential we can find the velocity components

$$u = \frac{\partial \phi}{\partial x} \quad \& \quad v = \frac{\partial \phi}{\partial y}$$

Physical Boundary Conditions

Boundary Conditions

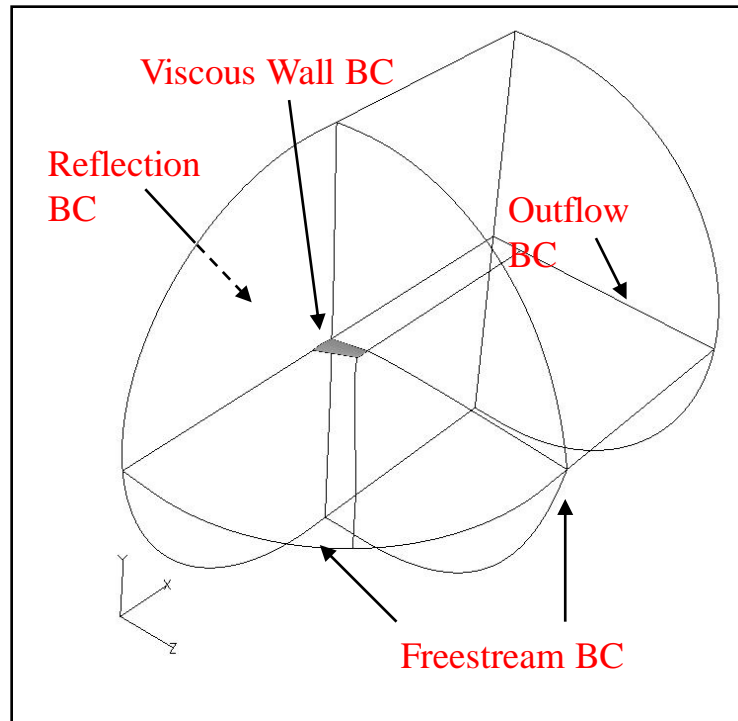
The governing equations are solved within a finite flow domain and zone.

A solution of the equations requires accurate specification of conditions at the boundaries of the domain and zones \Rightarrow **Boundary Conditions**.

Distinguish between two types:

Physical Boundary Conditions at the boundary of the flow domain.

Zonal Boundary Conditions at the boundary between two zones.



Physical Boundary Conditions

Types of Physical Boundary Conditions

Physical boundary conditions can be grouped:

Flow Interface Boundary Conditions

Boundaries at which the flow crosses freely:

Frozen

Freestream

Arbitrary Inflow

Outflow

Boundary conditions for
turbulence, chemistry, and
MFD equations also.

Wall Boundary Conditions

Boundaries with a solid surface (small amounts of flow may cross):

Viscous Wall

Inviscid Wall

Reflection

Bleed

Blowing

Moving Wall

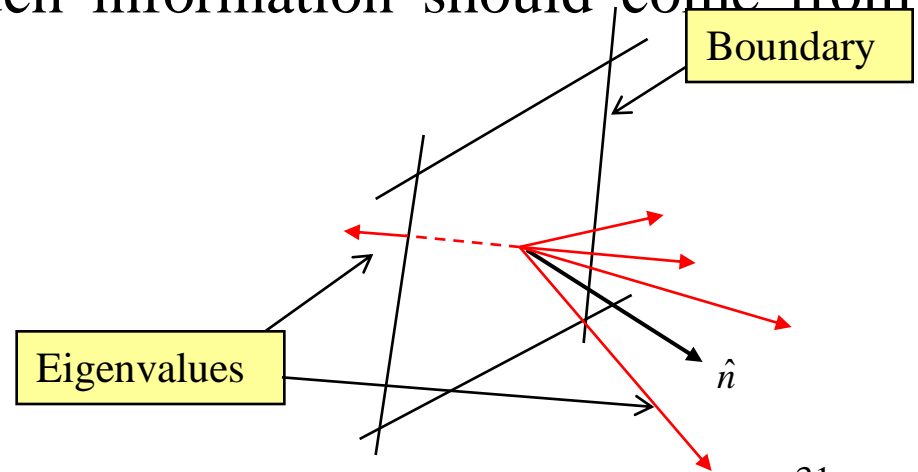
Characteristic Nature at Boundaries

The *characteristic* (*acoustic wave*) nature of the flow at the boundaries provides critical information for specifying boundary conditions.

Eigenvalues of the flow equations at the boundaries indicate the speed of the acoustic waves and whether they are entering (positive) or leaving (negative) the flow domain at the boundary.

This indicates how much information should come from outside the domain (*physical*) and how much information should come from inside the domain (*numerical*).

Boundary normal \hat{n} is positive directed into the flow domain



Eigenvalues

There are 5 eigenvalues λ for the Navier-Stokes equations.

The first 3 are associated with *convective waves* while last 2 are associated with *acoustic waves* (c is speed of sound):

$$\lambda_1 = \lambda_2 = \lambda_3 = v_n'$$

$$\lambda_4 = v_n' + c$$

$$\lambda_5 = v_n' - c$$

$$v_n' = (\vec{v} - \vec{g}) \cdot \hat{n}$$

Positive when the flow enters the flow domain.

Fluid speed relative to the motion of the boundary normal to the boundary.

Sign of λ_4 and λ_5 depend on sign and magnitude of normal velocity.

$\lambda > 0$ indicates that a wave enters the domain and *physical information* (i.e. pressure) must be specified as part of the boundary condition.

$\lambda < 0$ indicates that wave exits the domain and *numerical information* (i.e. an extrapolation) must be specified as part of the boundary condition.

Flows and Eigenvalues

The flow at the boundary is:

Inflow when $v_n' > 0$ and so $\lambda_1, \lambda_2, \lambda_3 > 0$

Outflow when $v_n' < 0$ and so $\lambda_1, \lambda_2, \lambda_3 < 0$

Subsonic when $|v_n'| < c$

Supersonic when $|v_n'| > c$

Combinations are:

Physical

Numerical

Subsonic Inflow $\lambda_1, \lambda_2, \lambda_3, \lambda_4 > 0$ and $\lambda_5 < 0$ 4 1

Subsonic Outflow $\lambda_4 > 0$ and $\lambda_1, \lambda_2, \lambda_3, \lambda_5 < 0$ 1 4

Supersonic Inflow $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5 > 0$ 5 0

Supersonic Outflow $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5 < 0$ 0 5

Subsonic Inflow

The **Subsonic Inflow** condition has eigenvalues which indicates that 4 physical and 1 numerical conditions need to be imposed at the boundary. These conditions can be imposed when using the **Freestream** or **Arbitrary Inflow** BC types at flow interfaces or the **Bleed** or **Blowing** BC types at walls.

$$\lambda_1, \lambda_2, \lambda_3, \lambda_4 > 0 \qquad \lambda_5 < 0$$

Subsonic Outflow

The **Subsonic Outflow** condition has eigenvalues which indicates that 1 physical and 4 numerical conditions need to be imposed at the boundary. These conditions can be imposed when using the **Freestream** or **Outflow** BC types at flow interfaces or the **Bleed** or **Blowing** BC types at walls.

$$\lambda_1, \lambda_2, \lambda_3, \lambda_5 < 0 \qquad \lambda_4 > 0$$

Supersonic Inflow / Frozen BC

The **Supersonic Inflow** condition has all positive eigenvalues which indicates that all information should come from outside the flow domain. The **Frozen** BC type is appropriate for supersonic inflow because the boundary condition simply keeps the flow conditions at the boundary unchanged. These conditions can be also imposed when using the **Freestream** or **Arbitrary Inflow** BC types at flow interfaces.

$$\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5 > 0$$

Supersonic Outflow

The **Supersonic Outflow** condition has all negative eigenvalues which indicates that all information should come from inside the flow domain. An extrapolation of the solution to the boundary is the typical numerical method. The **Freestream** or **Outflow** BC types are both capable of imposing the correct conditions.

$$\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5 < 0$$

Flow BCs: Freestream

The **Freestream** BC type is used at boundaries where flow conditions are expected to be fairly uniform and close to freestream conditions. The BC is flexible in that it checks whether the flow is inflow or outflow and subsonic or supersonic at the boundary and imposes the correct conditions.

For **supersonic inflow** or **supersonic outflow** the previous two slides indicate the conditions that are imposed.

For **subsonic inflow** and **subsonic outflow**, a characteristic-based method is used to determine conditions. Assuming uniform conditions, two Riemann invariants R^+ and R^- can each be evaluated either numerically or from freestream conditions as appropriate and then used to compute the other flow properties.

$$R^{\pm} = v_n' \pm \frac{2c}{\gamma - 1} \quad v_n' = \frac{1}{2} (R^+ + R^-) \quad c = \frac{\gamma - 1}{4} (R^+ + R^-)$$

Flow BCs: Arbitrary Inflow

The **Arbitrary Inflow** BC type is used mainly to impose a subsonic inflow condition, especially for internal flows (nozzle plenum). Four physical conditions need to be specified while the one numerical condition is usually extrapolated from the interior flow solution. The BC has the capability to:

- Hold total pressure and total temperature at the inflow.
- Specify the flow angle at the inflow.
- Hold total pressure, temperature, and flow angles of input solution.
- Specify an inflow profile.
- Specify inflow turbulence levels.
- Specify inflow chemistry composition.
- Specify inflows with a solid-body rotation.
- Specify time-varying inflows.

Flow BCs: Outflow

The **Outflow** BC type is used mainly to impose a *subsonic outflow* condition, especially for internal flows. The four numerical conditions are imposed through extrapolation. The one physical condition is imposed by determining the static pressure at the boundary in some way. Options include:

Uniform Static Pressure (constant or time-varying)

Variable Static Pressure (reference pressure with spatial variation over the boundary)

Mass Flow (for boundary and then compute the local static pressures)

Average Mach Number (for boundary and then compute the local static pressures)

Compressor Face (reflection coefficients for wave interactions with compressors)

Non-Reflecting (reference Mach number with non-reflecting constraints)

Wall Boundary Conditions

Wall boundary conditions involve a solid surface which may be porous. The fundamental relation at a wall boundary is

$$\rho (\vec{v} - \vec{g}) \cdot \hat{n} S_B = \rho v_n' S_B = \dot{m}$$

SB is the area of the cell face over which the BC is applied.

The **Bleed** or **Blowing** BC will have $\dot{m} \neq 0$, and so, $v_n' \neq 0$.

The **Moving Wall** BC will have $\vec{g} \neq 0$.

For all the wall boundary conditions, the value of v_n' is based on the presence of bleed or blowing. If $\dot{m} = 0$ then $v_n' = 0$.

Wall BCs: Viscous Wall

The **Viscous Wall** BC type is used at solid boundaries for which a velocity no-slip condition is required. A boundary layer forms along the wall. The no-slip condition is that the tangential flow relative to the boundary is zero:

$$\begin{aligned} v'_{t_1} &= (\vec{v} - \vec{g}) \cdot \hat{t}_1 = 0 & v'_{t_2} &= (\vec{v} - \vec{g}) \cdot \hat{t}_2 = 0 \\ \hat{t}_1 \cdot \hat{t}_2 &= 0. \end{aligned}$$

where the two tangential vectors are perpendicular, Classical boundary layer theory indicates that the pressure gradient in the direction normal to the wall is approximately zero, this is used as a physical condition at the viscous wall

$$\frac{\partial p}{\partial n} = 0$$

Another physical condition used at the wall assumes an adiabatic condition for which $\frac{\partial T}{\partial n} = 0$ or a constant temperature at the wall

$$\frac{\partial T}{\partial n} = 0$$

$$T_B = T_{wall}.$$

Wall BCs: Inviscid Wall / Reflection

The **Inviscid Wall** BC type is used at solid boundaries for which a velocity slip condition is desired. Thus the tangential velocity components v'_{t_1} and w'_{t_2} must now be computed in some manner. One approach is to simply extrapolate the tangential velocity components from the interior cells to the boundary.

The **Reflection** BC type uses the same methods as the inviscid wall BC type; however, the intent of the reflection BC is to apply it at a planar boundary of flow symmetry.

Wall BCs: Bleed / Blowing

The **Bleed** and **Blowing** BC types are used at solid boundaries for which there is a small amount of flow across the boundary. Bleed and blowing are often used in aerospace systems for flow control. The relative normal velocity component is computed as

$$v'_n = - \frac{\dot{m}_{bleed}}{\rho A_{region}}$$

where A_{region} is the area of the boundary over which bleed / blowing is specified.

One model for computing the bleed mass flow is

$$\dot{m}_{bleed} = Q_{sonic} p_t \Phi A_{region} \left(\frac{\gamma}{R T_t} \right)^{1/2} \left(1 + \frac{\gamma - 1}{2} \right)^{-[(\gamma + 1)/2(\gamma - 1)]}$$

Which requires the specification of the bleed region porosity Φ and a table or function for the bleed sonic flow function Q_{sonic} .

Wall BCs: Moving Wall

The **Moving Wall** boundary condition is used at solid boundaries that have uniform translation or rotation. An example of a uniform translation is an airplane going down a runway. An example of a uniform rotation is the airplane spinning about its x -axis. This BC type can be used with the viscous or inviscid wall BC types.

Boundary Condition Numerical Methods

- For a cell-vertex finite-volume cell, the boundary condition is applied at the solution point on the boundary (approach of WIND).
- For a cell-centered finite-volume cell, the boundary condition is applied at the cell face on the boundary as a flux.
- Boundary conditions can be imposed *explicitly* at boundary solution points (this is the default method in WIND).
- Boundary conditions can be imposed *implicitly* by including the conditions into the implicit matrices of the numerical method for the inner flow field. Wall BCs can be solved implicitly in WIND.
- Boundary conditions are imposed at the end of each iteration of the flow equations.

Flow Property Extrapolations

Most of the boundary conditions use an extrapolation of flow field properties to the boundary as part of imposing a numerical condition. A simple zero-order extrapolation of a property ϕ (which can represent static pressure or density) is

$$\phi_B = \phi_A$$

Where ϕ_B is the boundary value and ϕ_A is the value at a solution point adjacent to the boundary. A first-order extrapolation has the form

$$\phi_B = (1 + \alpha) \phi_A - \alpha \phi_{AA} \quad \alpha = \delta s_A / \delta s_{AA}$$

Where ϕ_{AA} is the value at a solution point adjacent to the adjacent point to the boundary. The “adjacent” direction is usually approximately normal to the boundary and along the grid line from the boundary. The δs are the spacings between the solution points.

Neumann and Dirichlet boundary conditions

- When using a Dirichlet boundary condition, one prescribes the value of a variable at the boundary, e.g. $u(x) = \textit{constant}$.
- When using a Neumann boundary condition, one prescribes the gradient normal to the boundary of a variable at the boundary, e.g. $\partial_n u(x) = \textit{constant}$.
- When using a mixed boundary condition a function of the form $au(x) + b\partial_n u(x) = \textit{constant}$ is applied.
- Note that at a given boundary, different types of boundary conditions can be used for different variables.

UNIT – II

METHOD OF CHARACTERISTICS, BOUNDARY CONDITIONS

METHOD OF CHARACTERISTICS

- Universal way to examine supersonic flow
- Name comes from looking at characteristic lines (Mach waves) propagating downstream
- Isentropic (no shocks)
- Seek rates of change along physical waves

$$\nabla \cdot (\rho \vec{u}) = 0$$

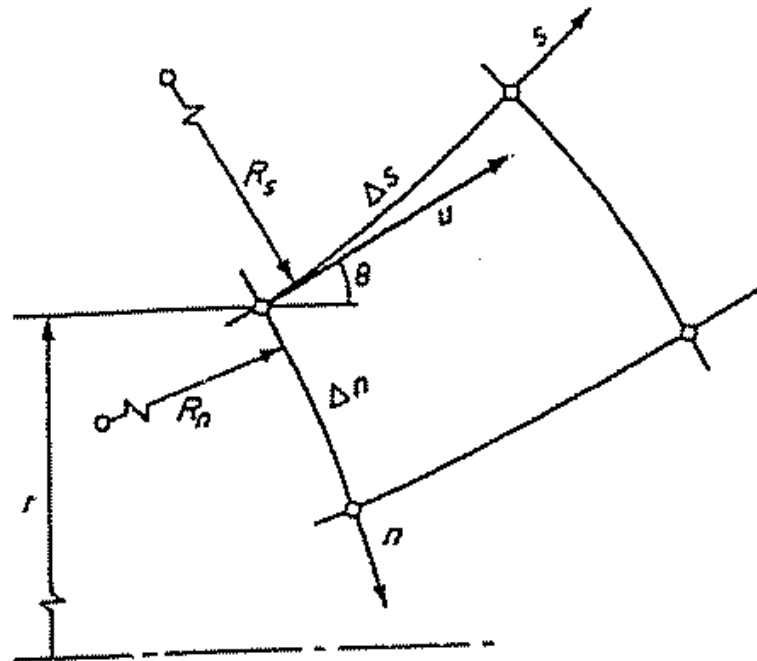
Continuity

$$\nabla \left(\frac{u^2}{2} \right) + \frac{1}{\rho} \nabla P = 0$$

Momentum

$$\nabla \times \vec{u} = 0$$

Irrotational



METHOD OF CHARACTERISTICS

ABREVIATED DERIVATION

Geometrical Statement

$$\frac{\partial u}{\partial n} + u \frac{\partial \theta}{\partial s} = 0$$

Continuity + Momentum

$$\frac{M^2 - 1}{u} \frac{\partial u}{\partial s} + \frac{\partial \theta}{\partial n} = \frac{\sin \theta}{r}$$

Characteristic Equations

$$\left(\frac{\partial}{\partial s} + \tan \mu \frac{\partial}{\partial n} \right) (\theta + \omega) = \frac{\tan \mu \sin \theta}{r}$$

$$\left(\frac{\partial}{\partial s} - \tan \mu \frac{\partial}{\partial n} \right) (\theta - \omega) = \frac{-\tan \mu \sin \theta}{r}$$

Introduction of m^+ and m^-

$$\frac{\partial (\theta + \omega)}{\partial m^+} = \frac{\sin \mu \sin \theta}{r}$$

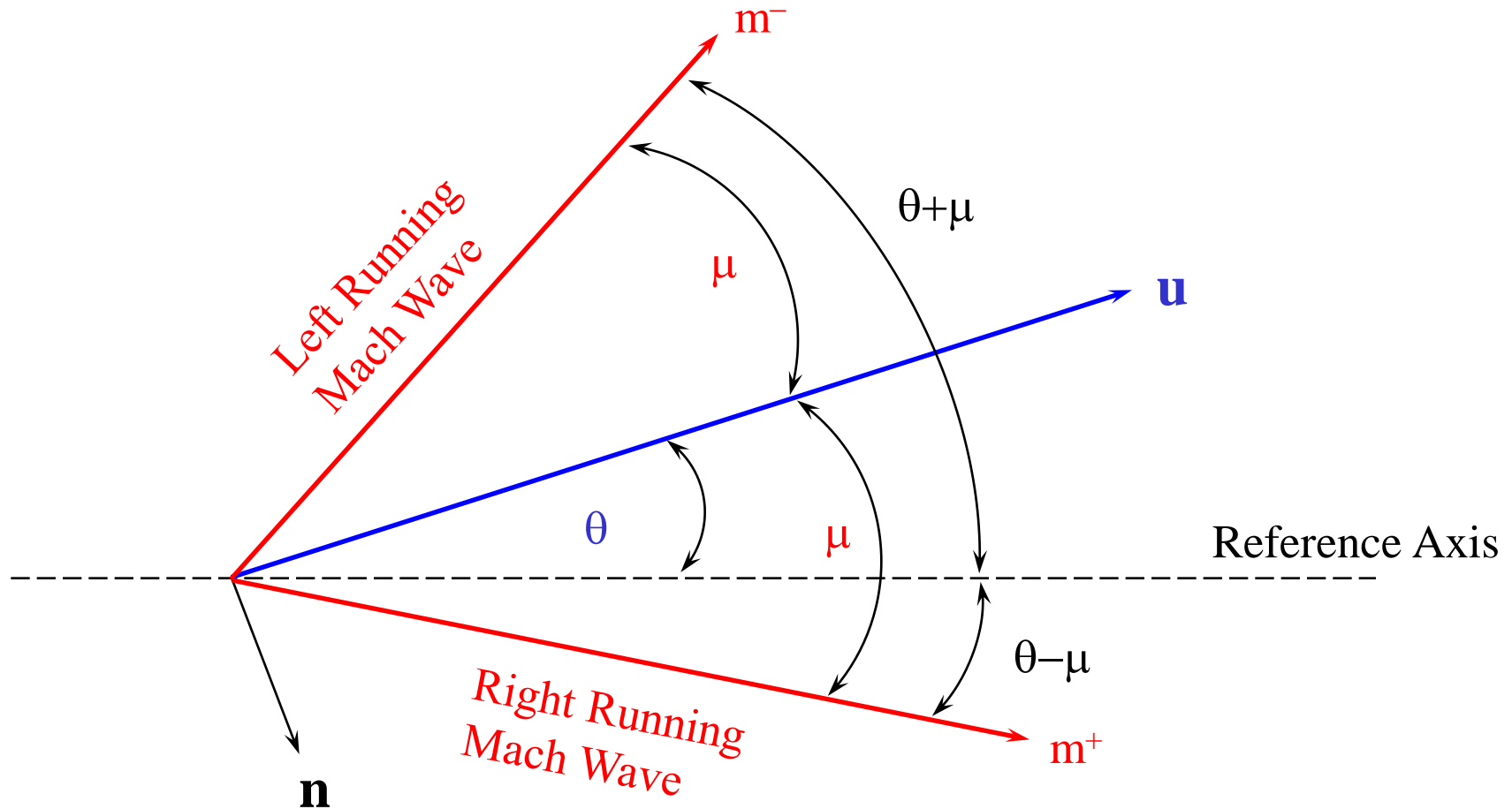
$$\frac{\partial (\theta - \omega)}{\partial m^-} = \frac{-\sin \mu \sin \theta}{r}$$

For 2D, $r \rightarrow \infty$

$\theta + \omega = \text{constant}$ on characteristic $m^+ = I^+$

$\theta - \omega = \text{constant}$ on characteristic $m^- = I^-$

NOMENCLATURE



θ : Angle between \mathbf{u} and Reference Axis

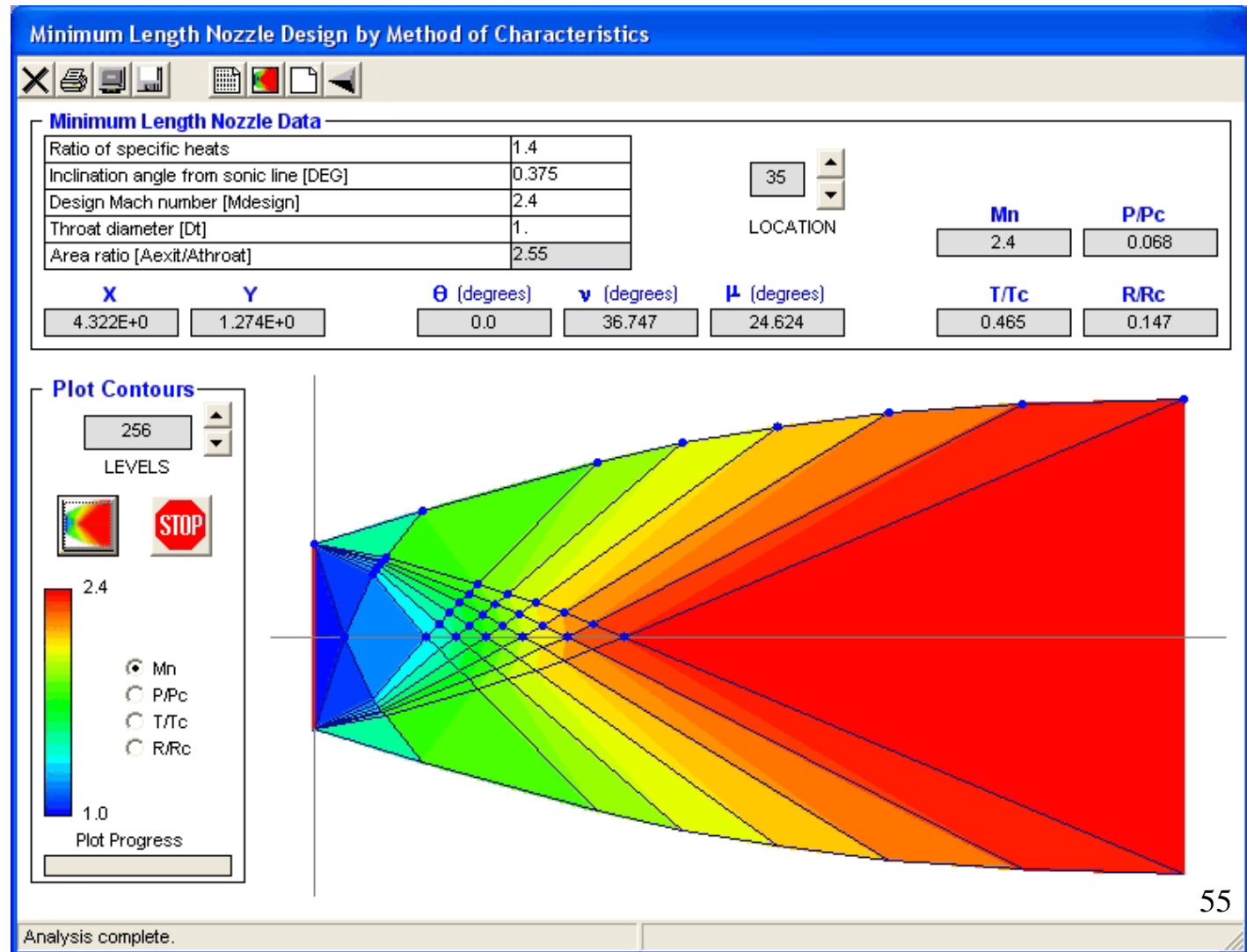
μ : Angle between \mathbf{u} and \pm Mach Waves

\mathbf{n} : reference normal vector, perpendicular to \mathbf{u} (or streamline)

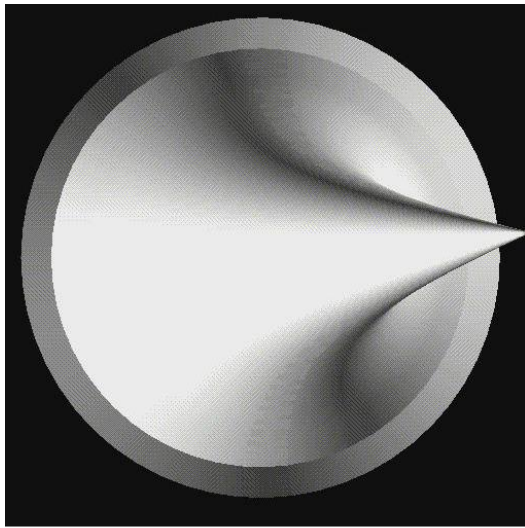
$q + w = \text{constant}$ on characteristic $\mathbf{m}^+ = \mathbf{I}^+$

$q - w = \text{constant}$ on characteristic $\mathbf{m}^- = \mathbf{I}^-$

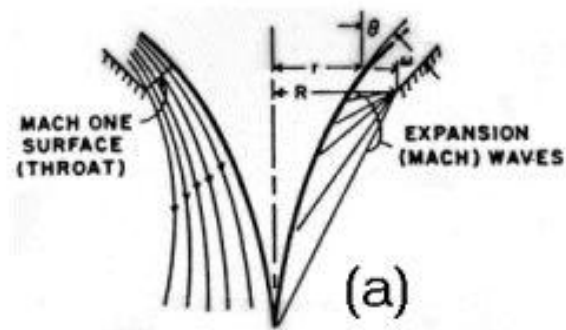
METHOD OF CHARACTERISTICS



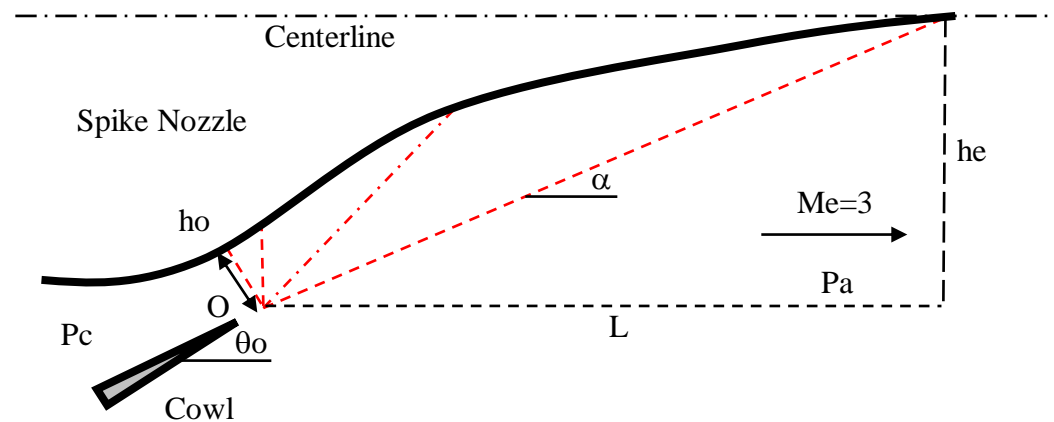
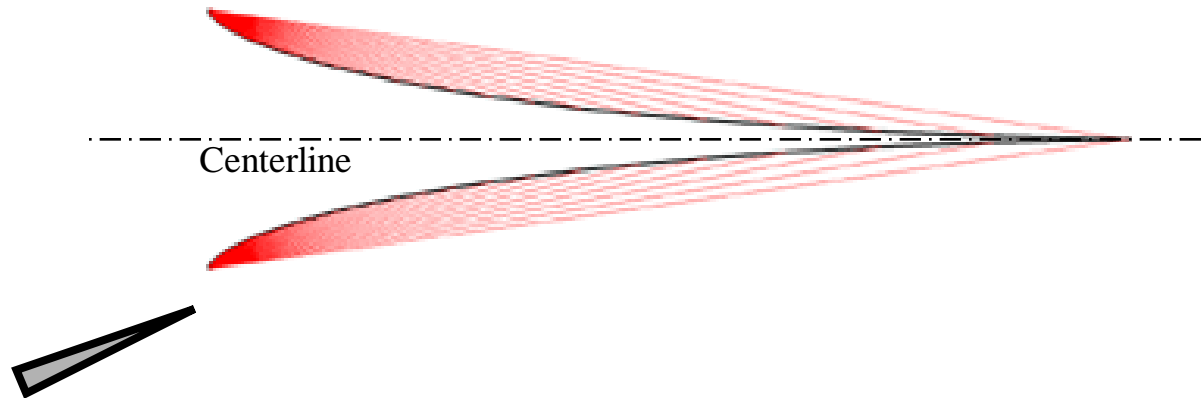
SPIKE EXAMPLE



The ideal 100% length aerospike nozzle



Ideal Aerospike Contour

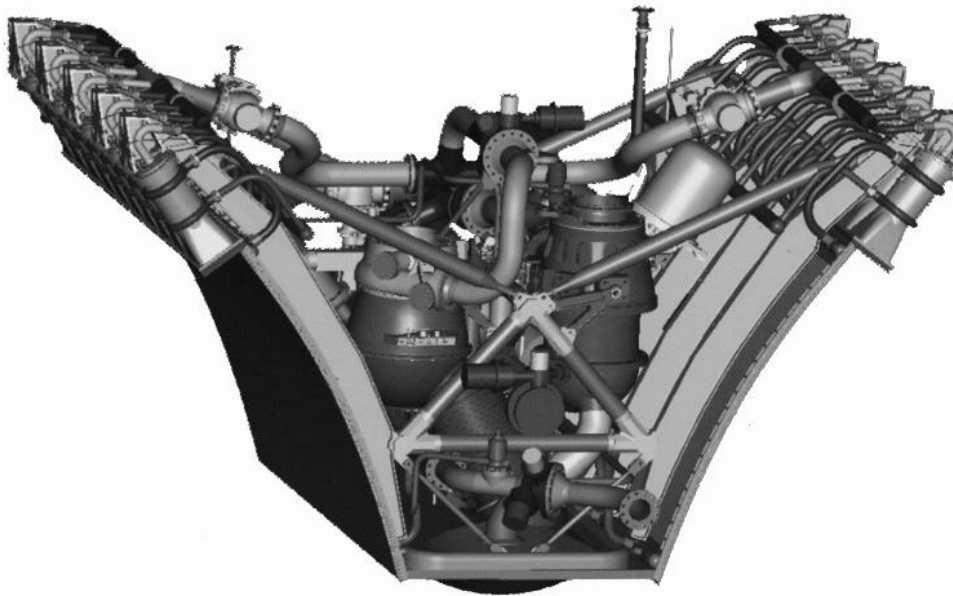


AIAA 2nd Responsive Space Conference 2004-7003
A STATUS REPORT ON THE DEVELOPMENT OF A NANOSAT LAUNCH VEHICLE AND
ASSOCIATED LAUNCH VEHICLE TECHNOLOGIES
by: John M. Garvey and Eric Besnard



Figure 6: Static Fire Test of Aerospike Nozzle. From Ignition to Continuous Operation 57

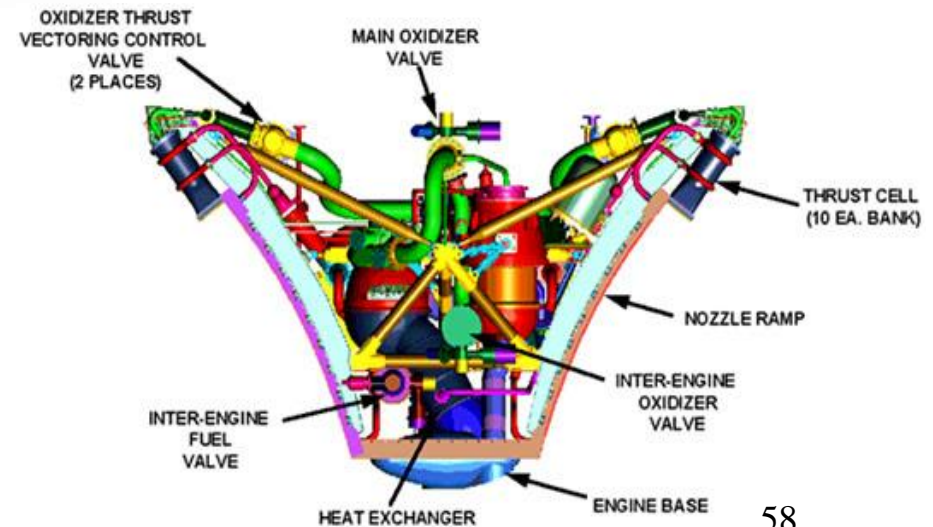
LINEAR AEROSPIKE ENGINE (X-33)



<http://www.msfc.nasa.gov/news/news/photos/1998/photos98-204.htm>

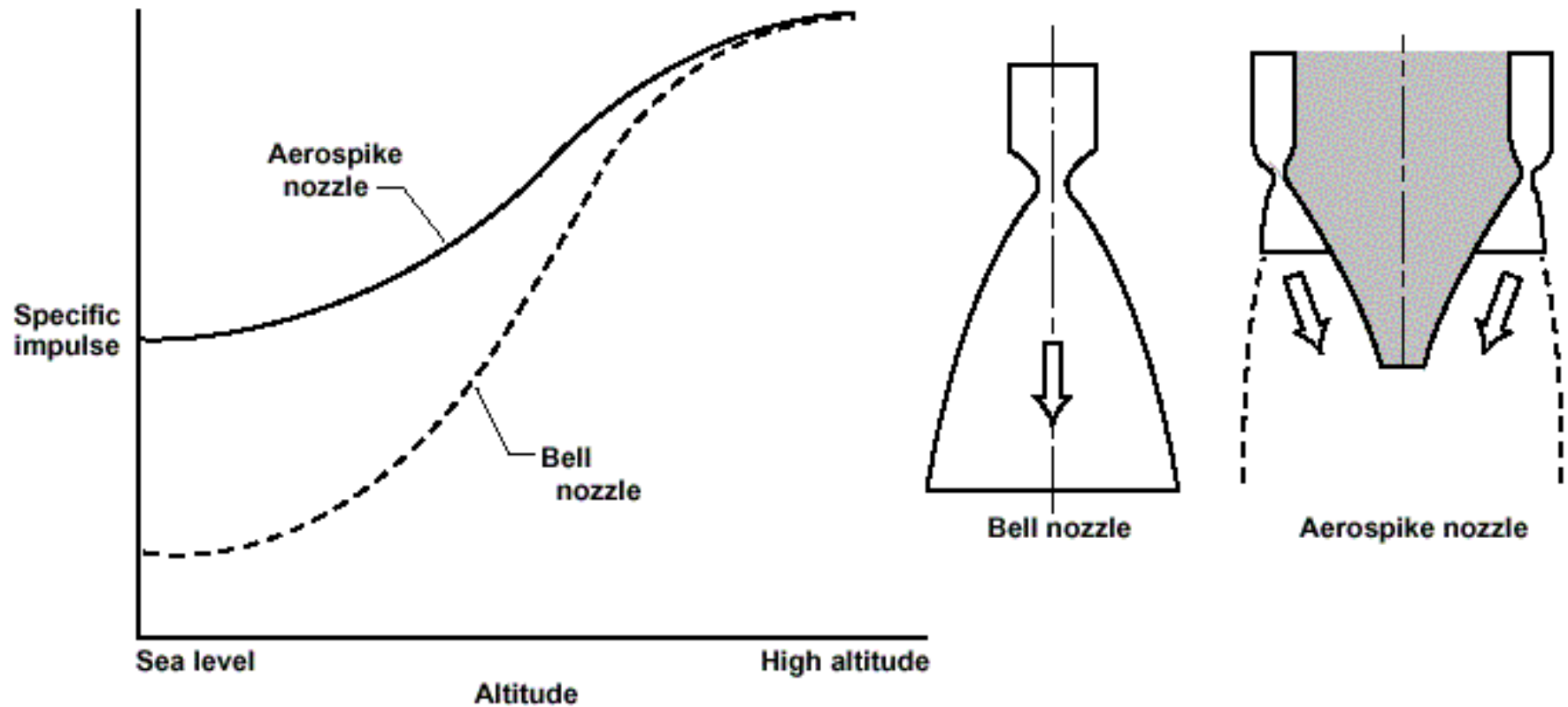


http://www.nasaexplores.com/show2_article.php?id=01-001



http://observe.arc.nasa.gov/nasa/ootw/1997/ootw_971001/x33.html

AEROSPIKE NOZZLE PERFORMANCE



Comparison of flow through conventional bell nozzle and aerospike nozzle.

INTRODUCTION

Designing planes that will fly quickly, efficiently and economically is all about making air flow smoothly over their wings and past their tube like bodies. Once a plane is up in the air, there is no easy way to see how air is moving past it. That's why every modern spacecraft and airplane is tested on the ground first in a wind tunnel.

SO, WHAT IS WIND TUNNEL?

A wind tunnel is a tool used in aerodynamic research to study the effects of air moving past solid objects. Actually, the basic idea of a wind tunnel is: if you can't move the plane through the air, why not move the air past the plane instead.

OBJECTIVE

Our objective will be to obtain relations that indicate the variation of fluid properties with area changes and mach number. In this manner, we can distinguish the important differences in subsonic and supersonic behaviour in a supersonic wind tunnel.

Types of Wind Tunnel

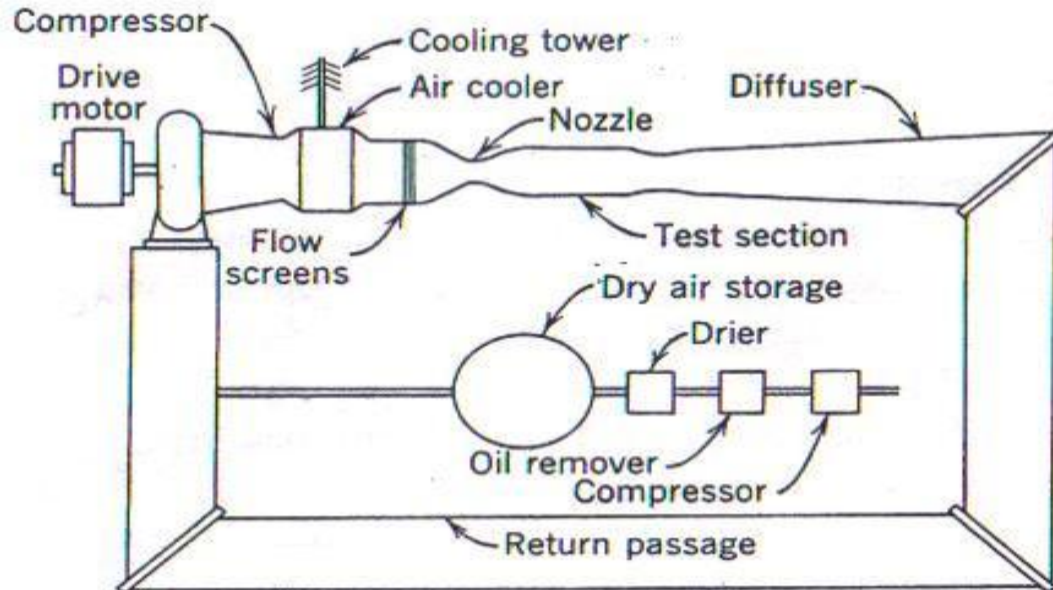
- Subsonic wind tunnel $(M < 1)$
- Transonic wind tunnel $(0.8 < M < 1.0)$
- Supersonic wind tunnel $(1.0 < M < 5)$
- Hypersonic wind tunnel $(M > 5)$

Types of Supersonic Wind Tunnel

- Continuous wind tunnels
- Blowdown tunnels
- Intermittent indraft wind tunnels

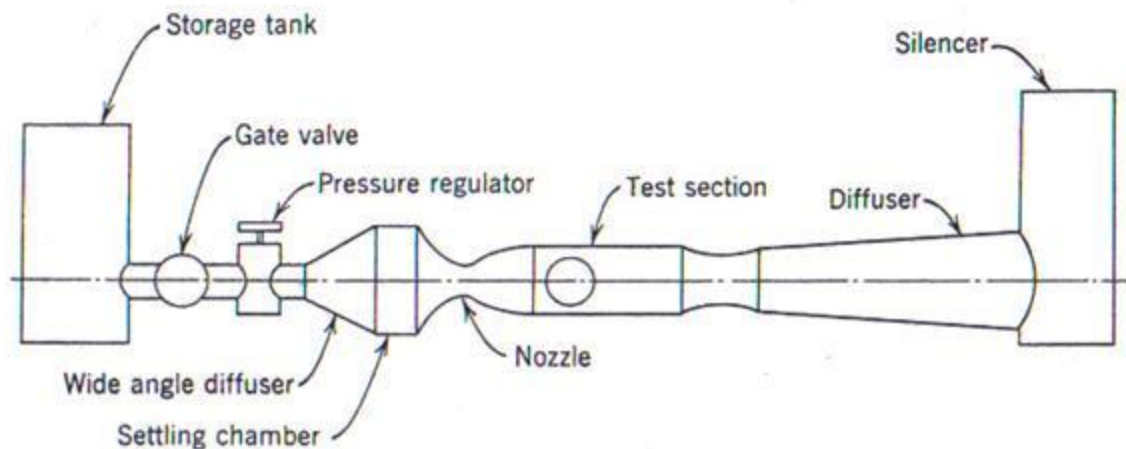
Continuous wind tunnels

- They are designed so that the air that passes through the tunnel does not exhaust to the atmosphere; instead, it enters through a return passage and is cycled through the test section repeatedly



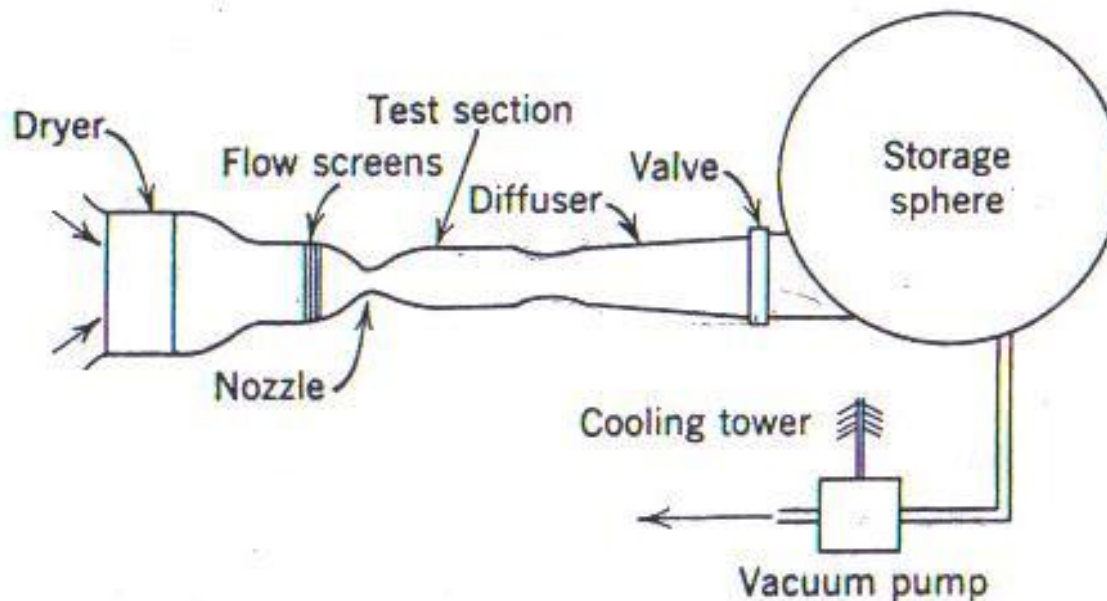
Blowdown tunnels

- Blowdown tunnels uses the difference between a pressurized tank and the atmosphere to attain supersonic speed. Pressure in the tank is greater than pressure in the environment in order to create a flow in the environment



Indraft wind tunnel

- Indraft wind tunnel uses the differences between a low pressure tank and the atmosphere to create a flow



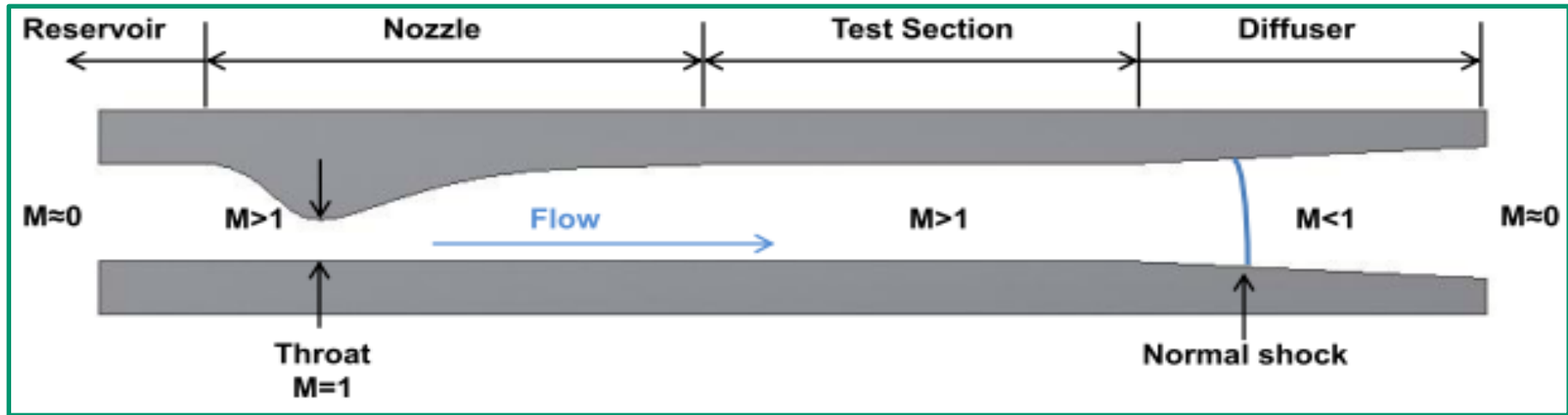
SUPERSONIC WIND TUNNEL

- **What is a Supersonic Wind Tunnel?**

A supersonic wind tunnel is a test bed for examining the fluid mechanics and associated fluid phenomena for air travelling faster than the speed of sound. ($Ma > 1$)

How is supersonic achieved?

This question leads us to the elements working principle of supersonic wind tunnel



Elements of Supersonic Wind Tunnel: The major elements of a supersonic wind tunnels are;

- (a) Reservoir
- (b) Nozzle
- (c) Test Section
- (d) Diffuser

WORKING PRINCIPLE OF THE WIND TUNNEL

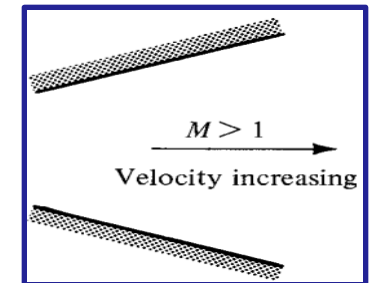
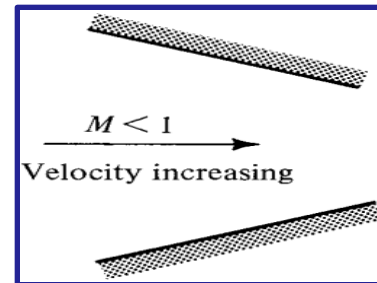
In order for the air inside the tunnel to reach supersonic conditions, the flow must be accelerated from rest through a Delaval nozzle.

To operate economically, the nozzle–test-section combination must be followed by a diffusing section which also must be converging–diverging.

What is a nozzle? A device is called a nozzle because of what it does, not what it looks like. It is a device that converts enthalpy into kinetic energy.

Types of a nozzle:

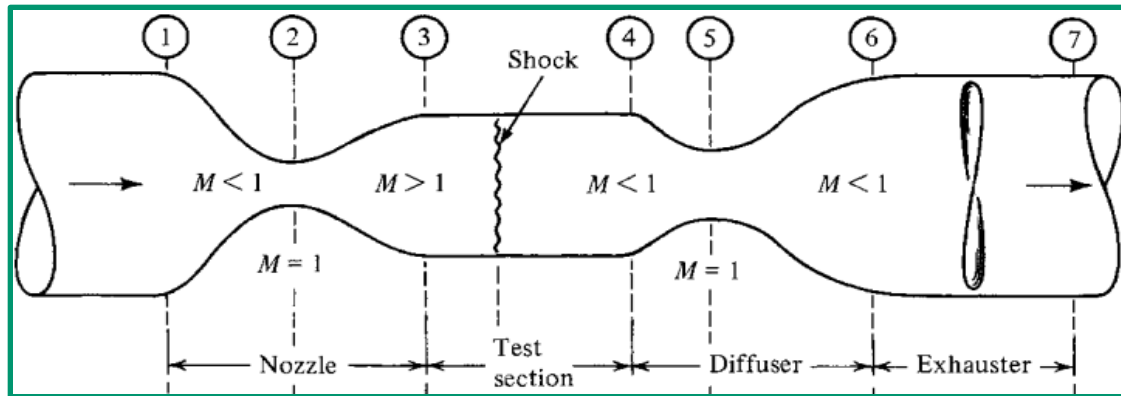
- A converging/diverging-only nozzle
- A converging-diverging nozzle.



Starting Condition

- **How air flow is generated into the tunnel?** The flow inside the tunnel is naturally quiescent if the pressures are equal. However, once the pressure ratio increases, the air starts moving, and the wind tunnel progresses through stages until it is fully started.
- **How is this pressure difference achieved?** There are three main categories of supersonic wind tunnels, differentiated by how the pressure ratio between the inlet and exhaust is achieved and controlled.

- i. An indraft tunnel: In an indraft tunnel, the inlet pressure is at atmospheric pressure, while the exhaust is at a lower vacuum pressure.
- ii. The blowdown tunnel: it has the opposite pressure setup; high pressure is stored at the inlet, and the exhaust is at atmospheric pressure.
- iii. Pressure-vacuum: The pressure-vacuum tunnel, combines both a high pressure inlet and a lower pressure vacuum at the exhaust.



Starting condition

Now that we are able to achieve fluid flow into the tunnel, how is supersonic speed achieved?

- **How is the supersonic speed achieved?** This leads us to the wave formation from the entrance of the deLaval nozzle, through test section and the diffuser.

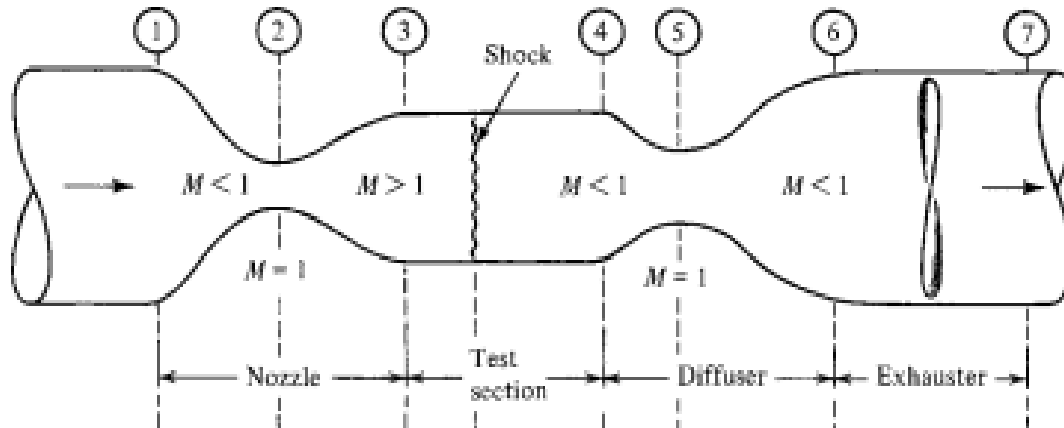
- The speed of the wave in the tunnel beforehand i.e. before startup is subsonic.
- Therefore, pressure variation increase and area changes of the nozzle determine our Mach number. i.e. the kind of speed that we achieve.
- it should be noted that, no matter what, the only flow that can be achieved in the throat of converging-only nozzle will never be greater than Mach 1
- At this Mach number, choking occurs at the throat of the nozzle. The speed is sonic ($Ma=1$).
- Further change in the pressure ratio, and the area between the test section and the throat cause the speed to increase the more.
- Basically into supersonic speed

SHOCK WAVE GENERATION

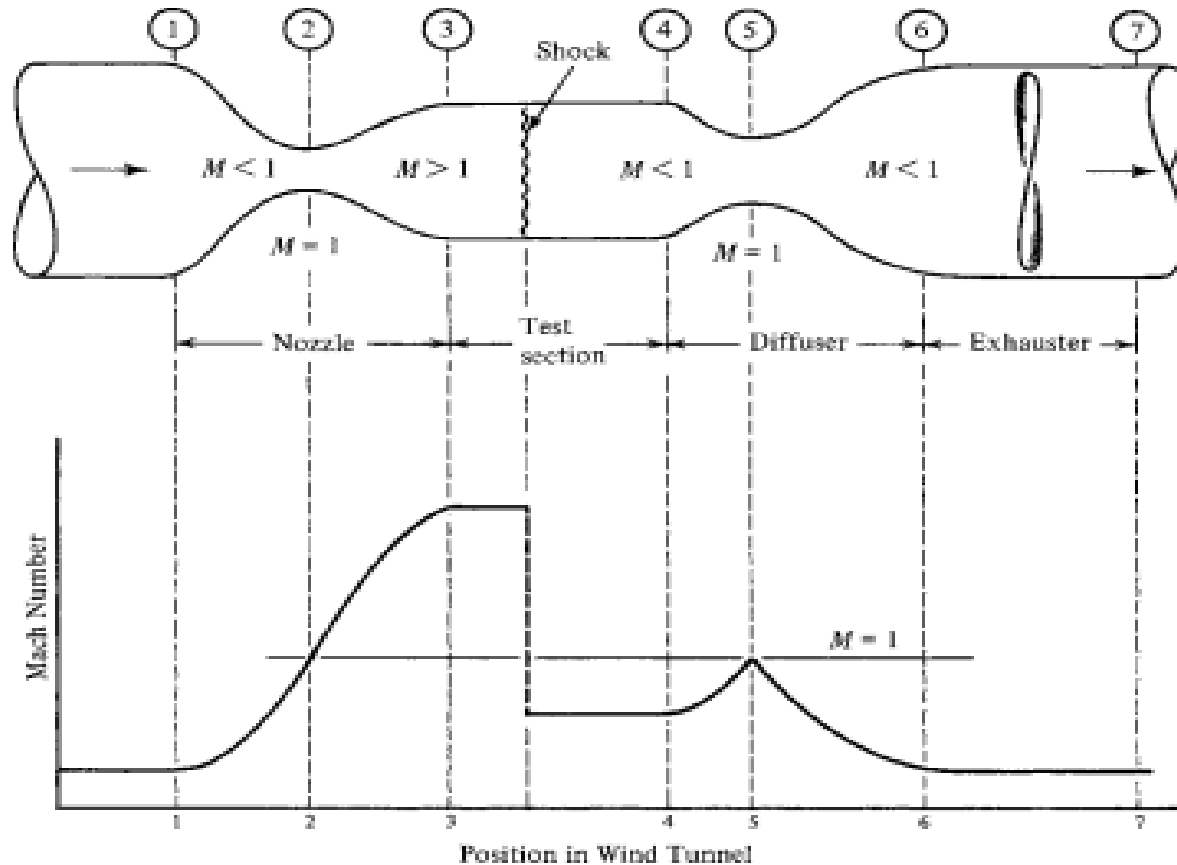
Now what happens if we decrease the nozzle exit pressure even further?

A sharp discontinuity is observed in the Test section.

This is otherwise called a SHOCK WAVE

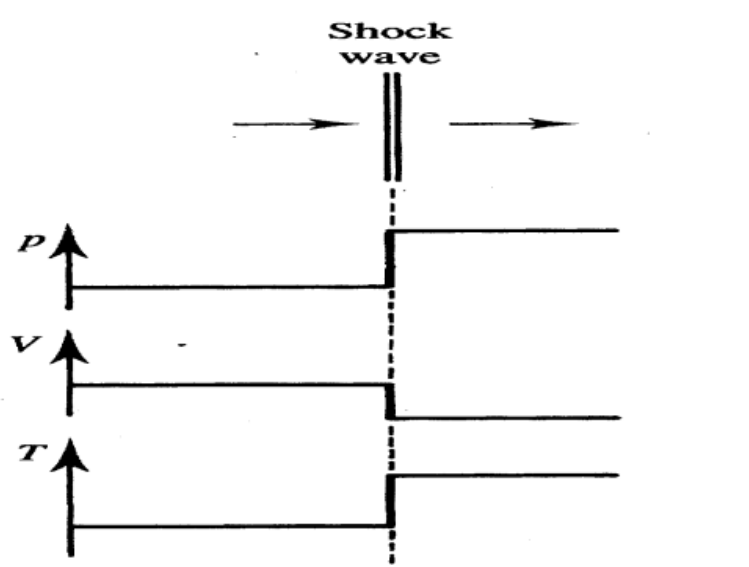


SHOCK WAVE IN A TUNNEL



Supersonic tunnel at start up with variable Mach number variation

UNDESIRED EFFECT OF SHOCK WAVE.



In some sense a shock wave can also be visualized as a very sharp and thin wave front across which pressure, temperature, and *velocity* of flow change abruptly

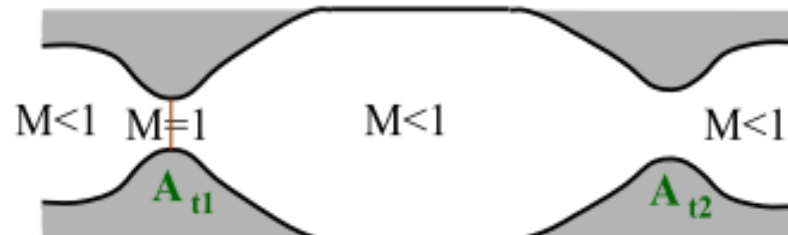
So, what do we DO?

As the pressure difference in the tunnel keeps decreasing, the shock wave moves from the test section and towards the diffusing end.

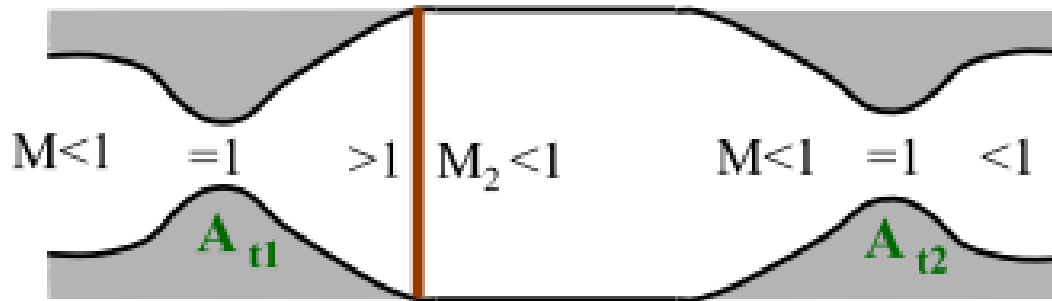
The throat of the diffuser must be configured to allow shock waves to pass through it.

As a rule of thumb, diffuser throat area must be larger than that of the nozzle to allow the throat to swallow shock waves

i.e $A_{\text{nozzle}} \leq A_{\text{diffuser}}$



WHY?



Knowing fully well that sonic velocity is achieved both at the nozzle and diffuser throats, there exist a similarity between the flow properties at the two ends.

$$P_t A^* = \text{constant}$$

Therefore,

$$P_t \text{ nozzle} \times A \text{ nozzle} = P_t \text{ diffuser} \times A \text{ diffuser} \dots \dots \dots$$

(*)

Due to the friction losses in the tunnel (not an ideal situation), there is a pressure loss across the sections.

Therefore P_t diffuser will always be lower than P_t nozzle
i.e $P_t \text{ diffuser} < P_t \text{ nozzle}$

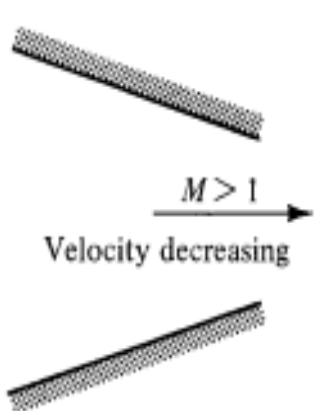
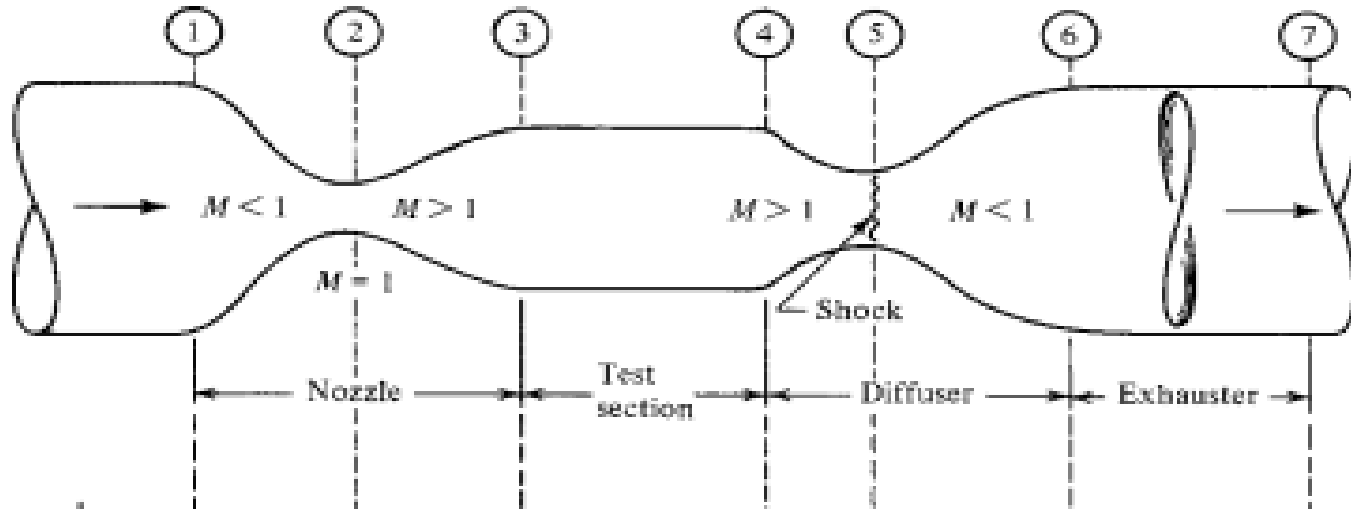
If the continuity equation is not obeyed, the shock will not pass through the diffuser. Hence, the diffuser area is made bigger than the nozzle area.

Therefore, for shock to move through the test section and pass through the diffuser.

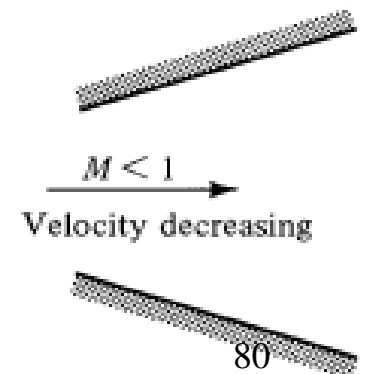
$A \text{ diffuser} > A \text{ nozzle}$ (for running condition)

DIFFUSER SECTION

The supersonic speed achieved is step down in the diffuser to subsonic.



Shocks disappear at the throat where the velocity becomes sonic again



VARIABLE GEOMETRY

STARTING CONDITIION

To ensure chocking takes place at the nozzle throat, the diffuser area must be small enough to quicken the process and big enough to avoid chocking at the diffuser.

RUNING CONDITION

To ensure that shockwaves is passes throughout the test section and the into the diffuser section, the diffuser area must be big enough to accommodate for the losses in the system (eqn *, continuity equation).

Therefore the diffuser throat is made such that it changes according to the time of operation.(variable Geometry)

DIFFICULTIES IN DESIGN AND MEANS OF OVERCOMING THEM

Starting and Unstarting of the Tunnel

Mach – Area ratio and Variable Geometry design

Adequate supply of dry air

The use of drier for closed tunnels and operating at ambient conditions for open tunnel.

Wall interference effect(Formation of Oblique shocks)

Contour of the throat is critically shaped to maintain a smooth flow through the test section, and the blower blade (For blow down tunnels) are made straight and flat to produce a uniform flow.

High quality instrument capable of rapid measurement due to short run time on intermittent tunnel

The use of diffuser to increase working time by producing a minimum pressure gradient to drive the tunnel.

PRESSURE DIFFERENCE RELATIONS

➤ Assuming a source of pressure 200 psai and 1200°R is contained in the reservoir/inlet where stagnation conditions prevailed.

- At $P_e = P_t$ no there will be no flow in the tunnel
- Pressure difference P_e / P_t controls the flow into the tunnel

Assume $P_e < P_t$ i.e. $P_e = 150$

$$P_2/P_{t2} = (P_2/P_{t1}) (P_{t1}/P_{t2})$$

Using our isentropic table,

$M = 0.66$; $M < 1$ subsonic

$$(150/200) = 0.75$$

Assume $P_e = 105.65$

$$(105.65/200) = 0.52825 \quad M = 1.0 \text{ (sonic). At this point,}$$

chocking occurs

P_e = Pressure at the exhaust or receiver

P_2 = Pressure at the throat

P_t = Pressure at the reservoir/supply/inlet or stagnation pressure

CHOKE FLOW RATE

But the choked flow rate can change if, for example the supply pressure or temperature is changed or the size of the throat (exit hole) is changed.

Let's consider this equation

We now look at four distinct possibilities:

$$\frac{\dot{m}}{A^*} = \text{constant} \frac{p_t}{\sqrt{T_t}}$$

1. For a fixed T_t , p_t , and $A^* \Rightarrow \dot{m}_{\max}$ constant.
2. For only p_t increasing $\Rightarrow \dot{m}_{\max}$ increases.
3. For only T_t increasing $\Rightarrow \dot{m}_{\max}$ decreases.
4. For only A^* increasing $\Rightarrow \dot{m}_{\max}$ increases.

For us to achieve a supersonic flow, it means we need an increasing area and increasing pressure P_t . With these criteria, we have our convergent-divergent nozzle through which our supersonic pressure can be achieved.

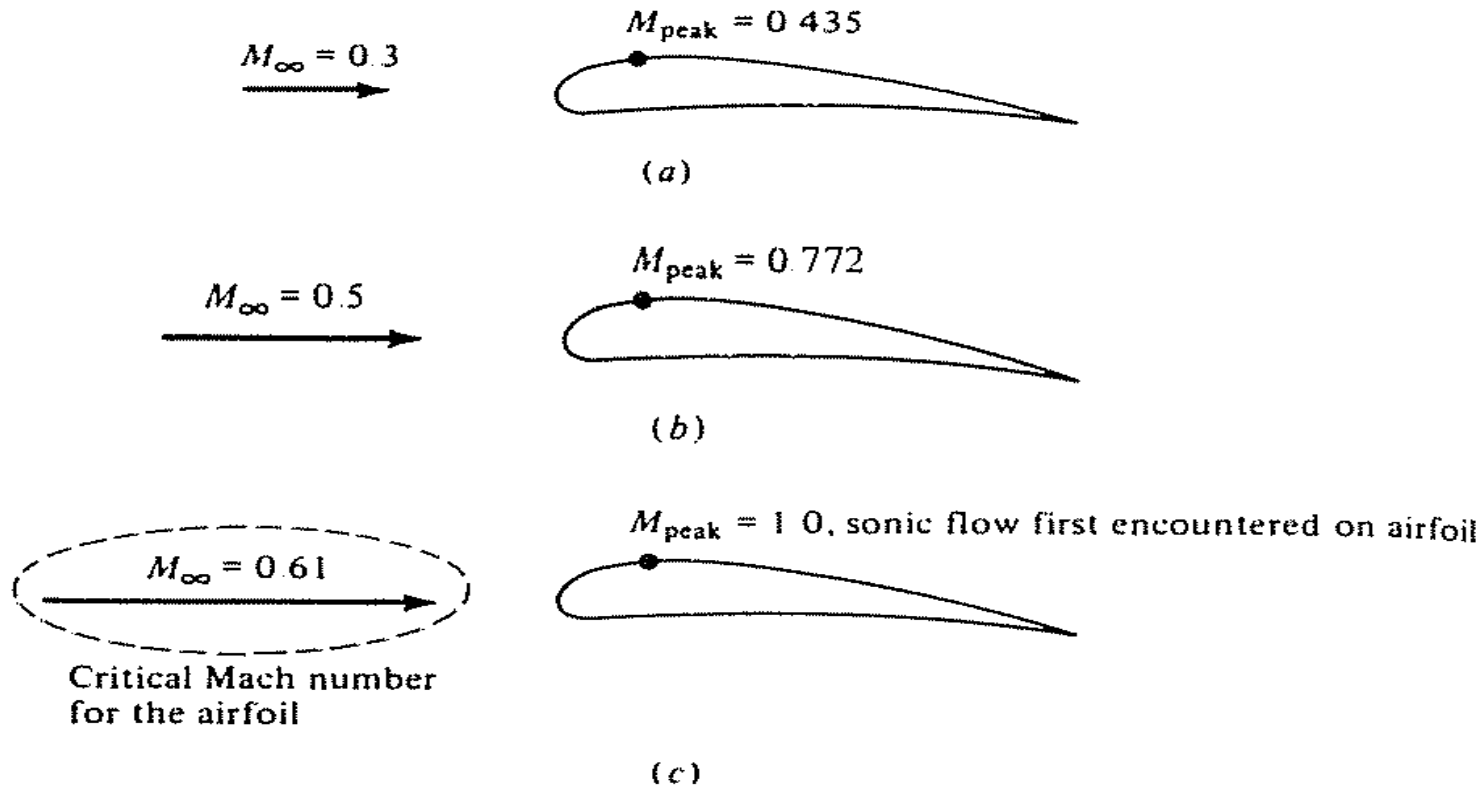
- Although the supersonic wind tunnel is used primarily for aeronautically oriented work, its operation serves to solidify many of the important concepts of **variable-area flow, normal shocks, and their associated flow losses**.
- Shocks are found only in supersonic flow, and the flow is always subsonic after a normal shock. The shock wave is a type of compression process, although a rather inefficient one since relatively large losses are involved in the process. (What has been lost?) Shocks provide a means of flow adjustment to meet imposed pressure conditions in supersonic flow.
- Wind tunnels are used to study the flow of air around an object. An object is placed in a tunnel and subjected to varying wind speeds to determine how it is affected by wind current. Airplanes and rockets are tested in large wind tunnels.
- To operate economically, the nozzle–test-section combination must be followed by a diffusing section which also must be converging–diverging.

UNIT – III

NUMERICAL SOLUTION OF TRANSONIC SMALL DISTURBANCE EQUATION

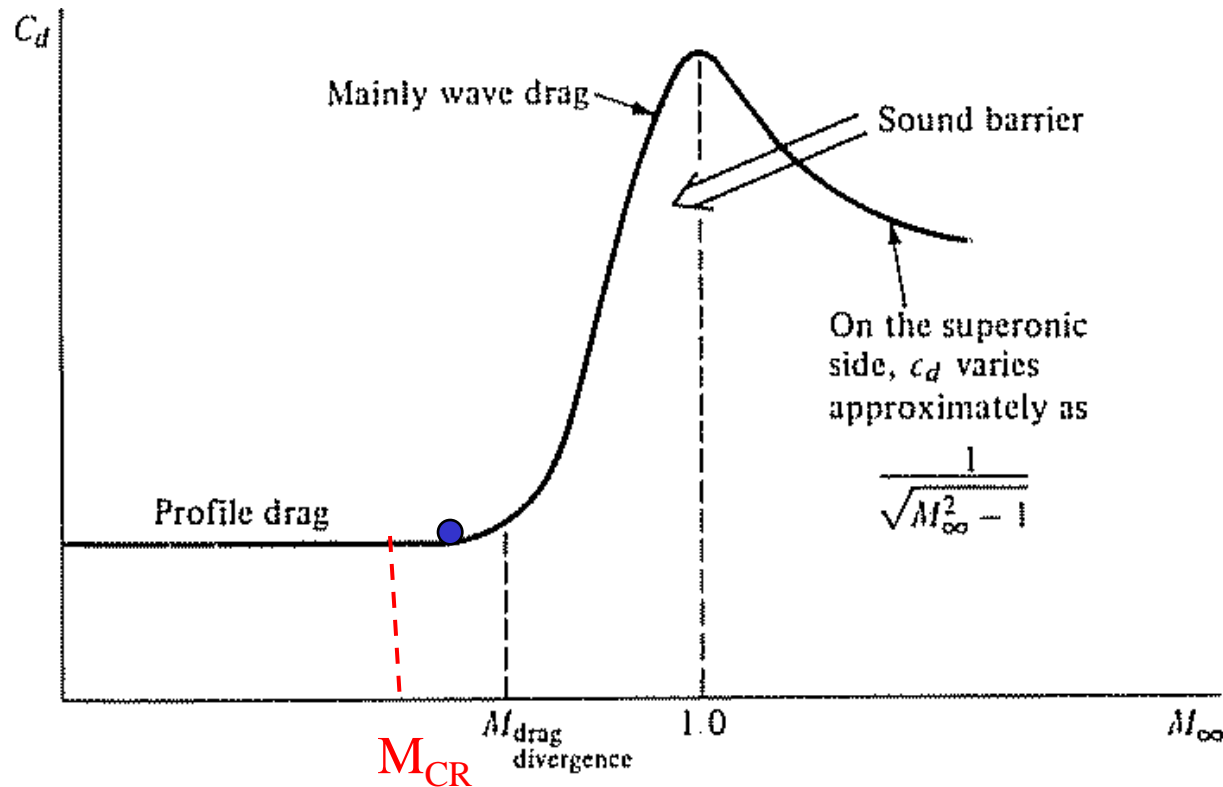
CRITICAL MACH NUMBER, M_{CR}

- As air expands around top surface near leading edge, velocity and M will increase
- Local $M > M_\infty$



Flow over airfoil may have sonic regions even though freestream $M_\infty < 1$ **INCREASED DRAG!**

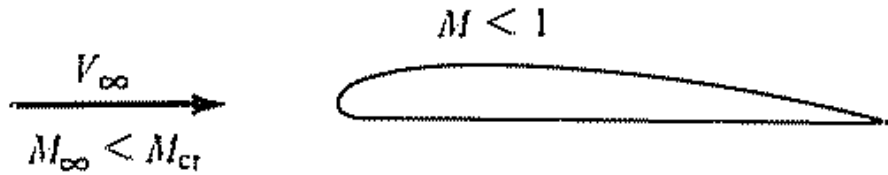
CRITICAL FLOW AND SHOCK WAVES



$$M_{CR} < M_{Drag - Divergence} < 1.0$$

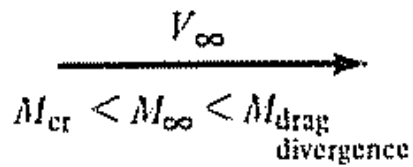
- Sharp increase in c_d is combined effect of shock waves and flow separation
- Freestream Mach number at which c_d begins to increase rapidly called Drag-Divergence Mach number

CRITICAL FLOW AND SHOCK WAVES

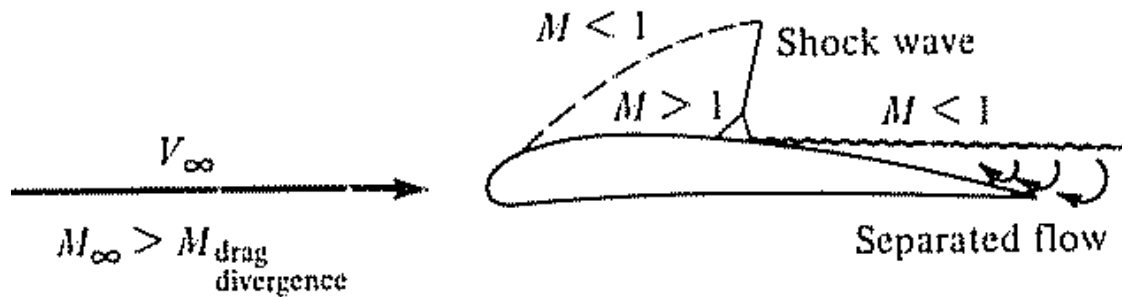
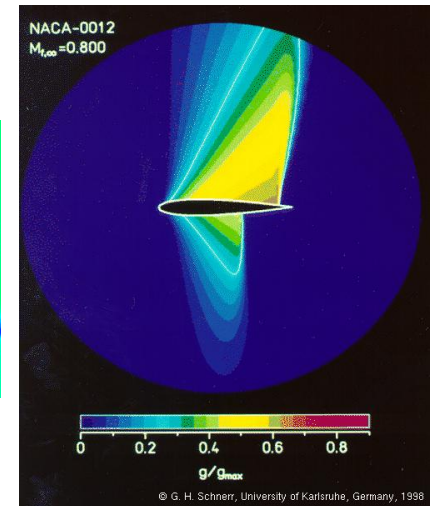
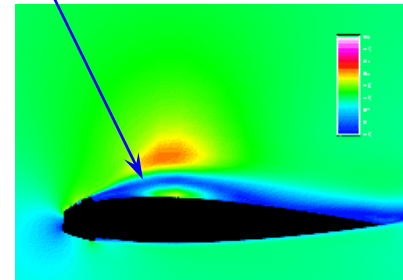


(a)

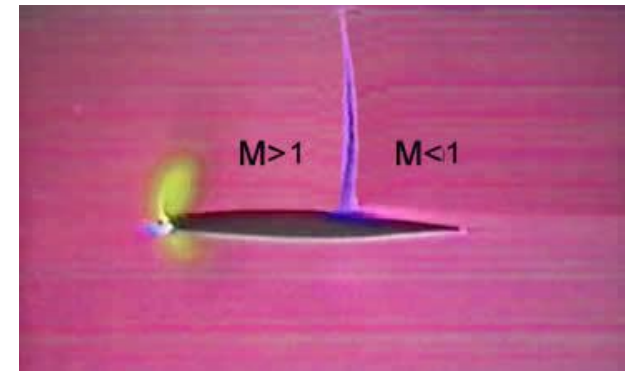
‘bubble’ of supersonic flow



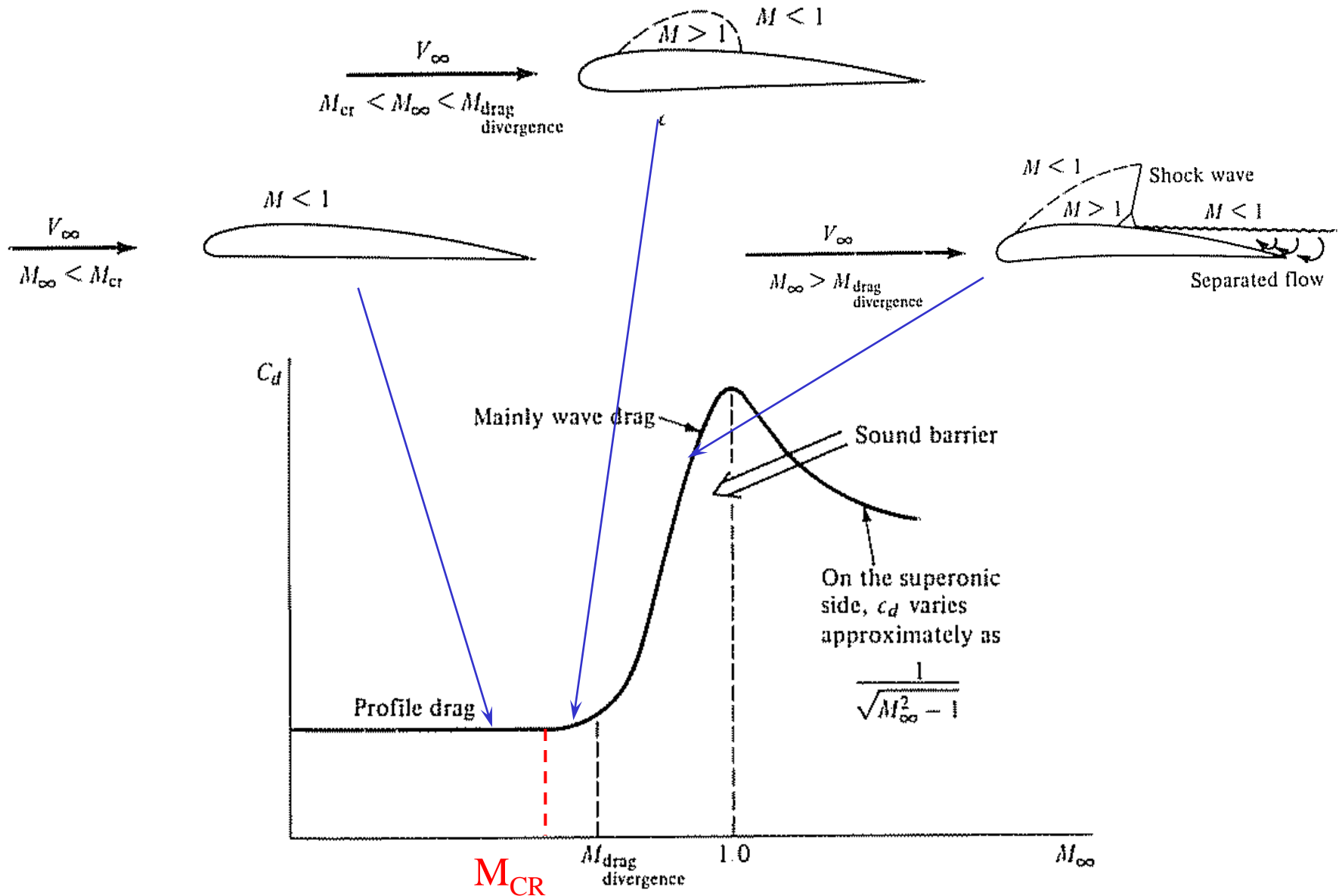
(b)



(c)



CRITICAL FLOW AND SHOCK WAVES



EXAMPLE: IMPACT ON AIRFOIL / WING DRAG

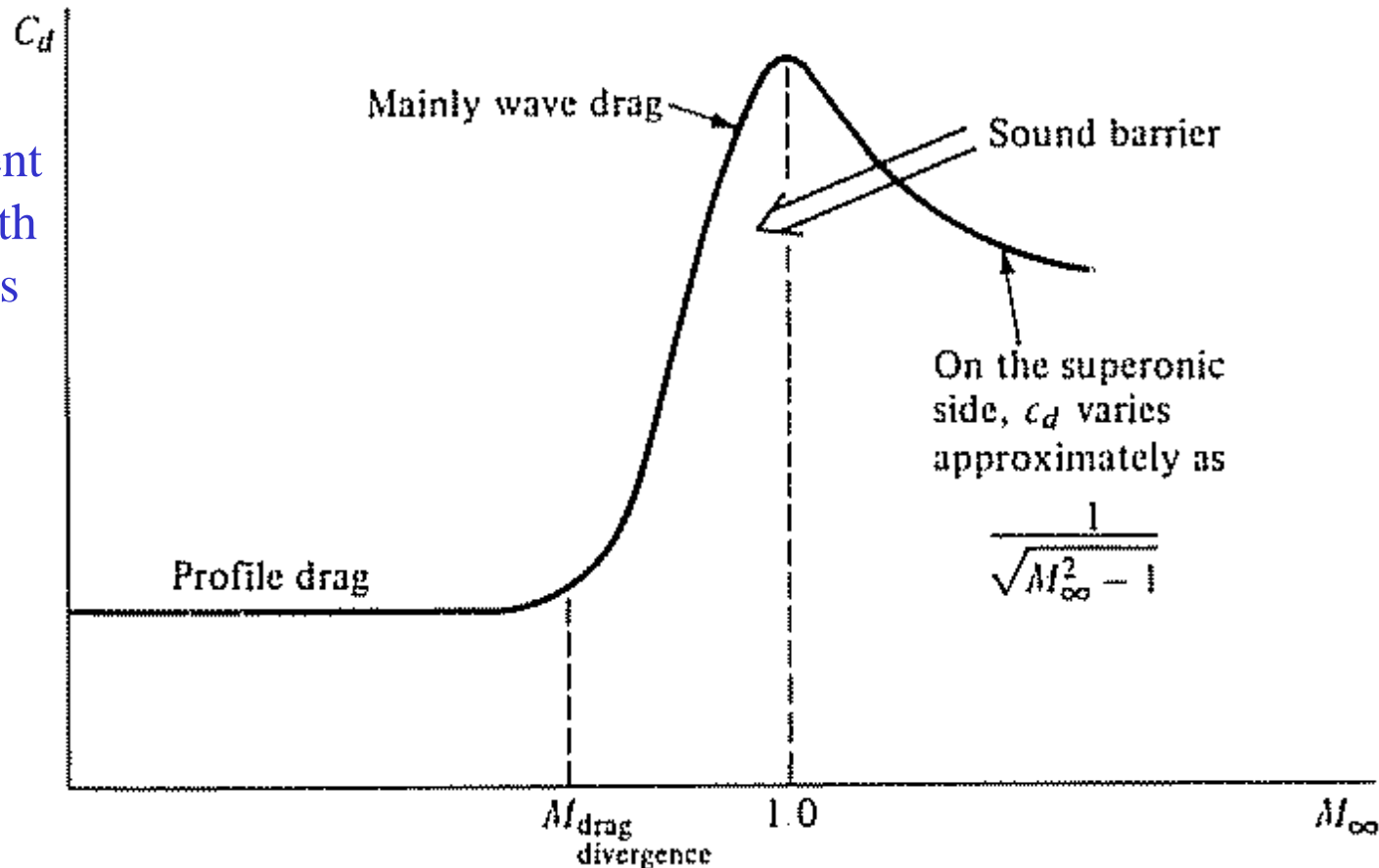
$$D = D_{friction} + D_{pressure} + D_{wave}$$

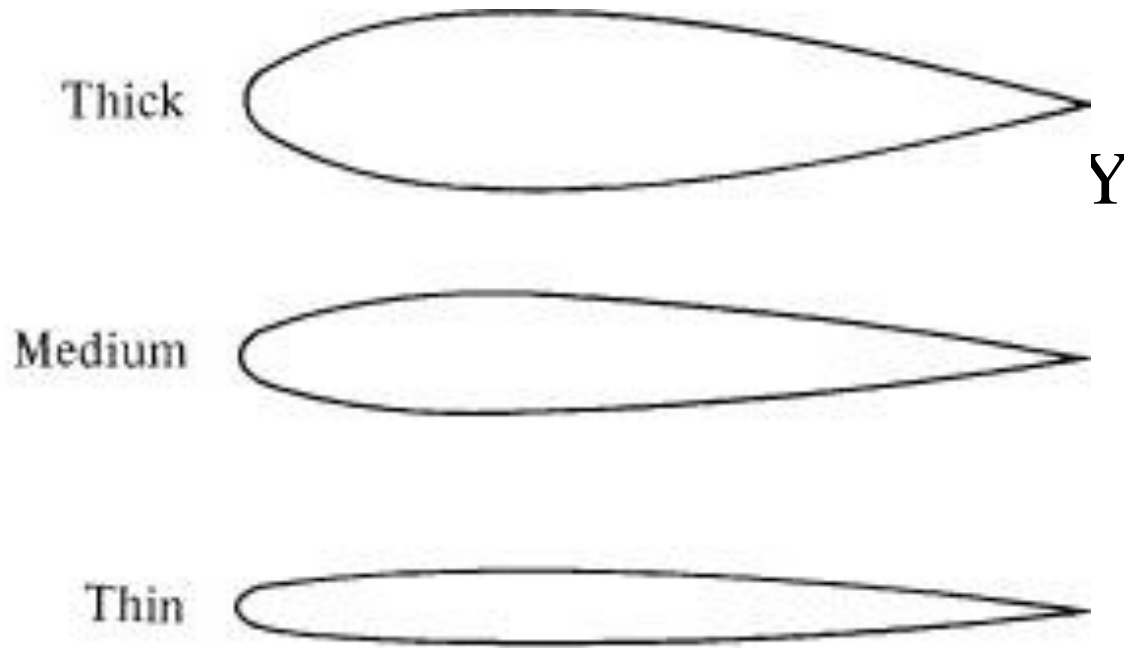
$$C_d = C_{d,f} + C_{d,p} + C_{d,w}$$

Only at transonic and supersonic speeds

$D_{wave} = 0$ for subsonic speeds below $M_{drag-divergence}$

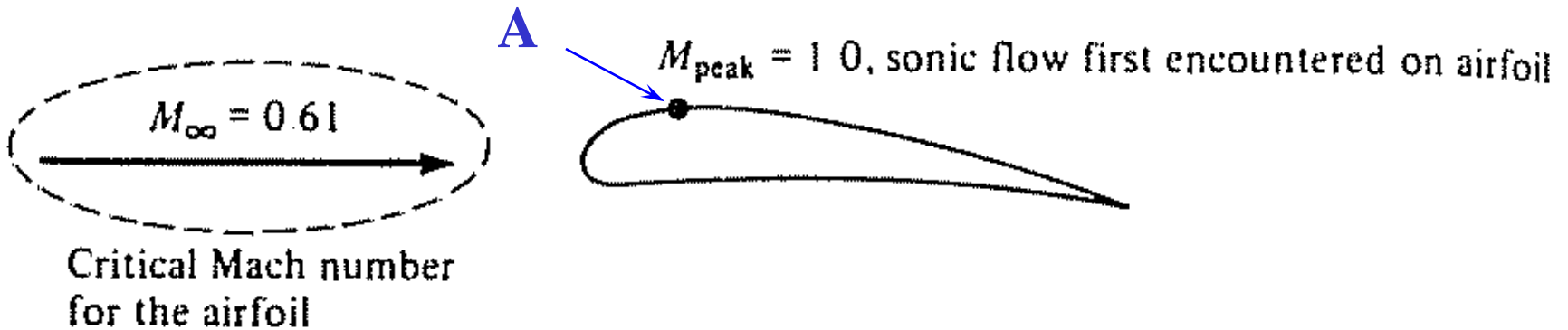
Profile Drag
Profile Drag coefficient relatively constant with M_∞ at subsonic speeds





- Which creates most lift?
 - Thicker airfoil
- Which has higher critical Mach number?
 - Thinner airfoil
- Which is better?
 - Application dependent!

Note: thickness is relative
to chord in all cases
Ex. NACA 0012 \rightarrow 12 %



$$C_{p,A} = \frac{2}{\gamma M_\infty^2} \left(\frac{p_A}{p_\infty} - 1 \right)$$

$$\frac{p_A}{p_\infty} = \frac{p_A/p_0}{p_\infty/p_0} = \left(\frac{1 + \frac{\gamma-1}{2} M_\infty^2}{1 + \frac{\gamma-1}{2} M_A^2} \right)^{\gamma/(\gamma-1)}$$

- Pressure coefficient defined in terms of Mach number (instead of velocity)
- In an isentropic flow total pressure, p_0 , is constant
- May be related to freestream pressure, p_∞ , and static pressure at **A**, p_A

CAN WE PREDICT M_{CR} ?

$$C_{p,A} = \frac{2}{\gamma M_\infty^2} \left[\left(\frac{1 + \frac{\gamma-1}{2} M_\infty^2}{1 + \frac{\gamma-1}{2} M_A^2} \right)^{\gamma/(\gamma-1)} - 1 \right]$$

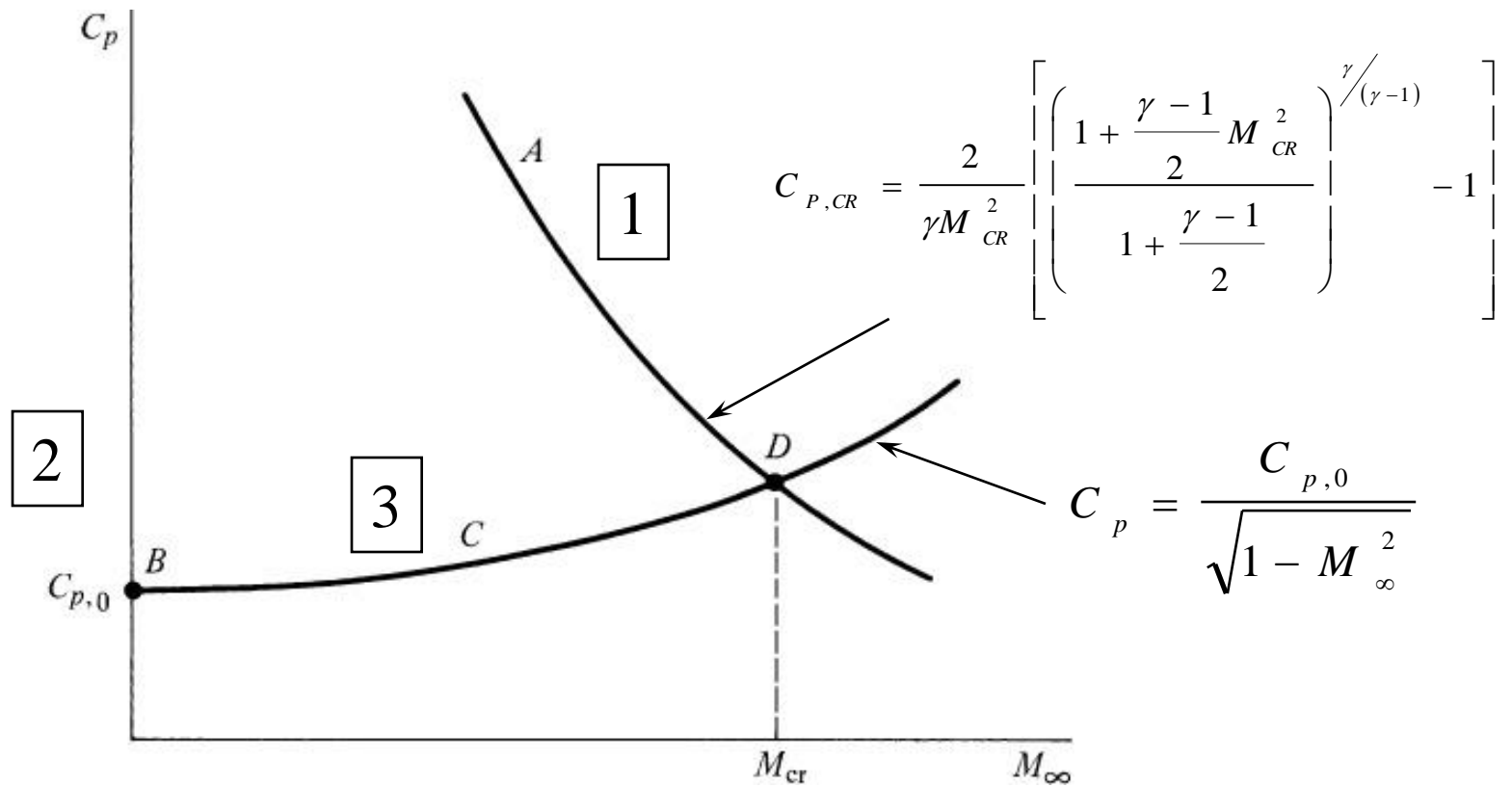
- Combined result
 - Relates local value of C_p to local Mach number
 - Can think of this as compressible flow version of Bernoulli's equation

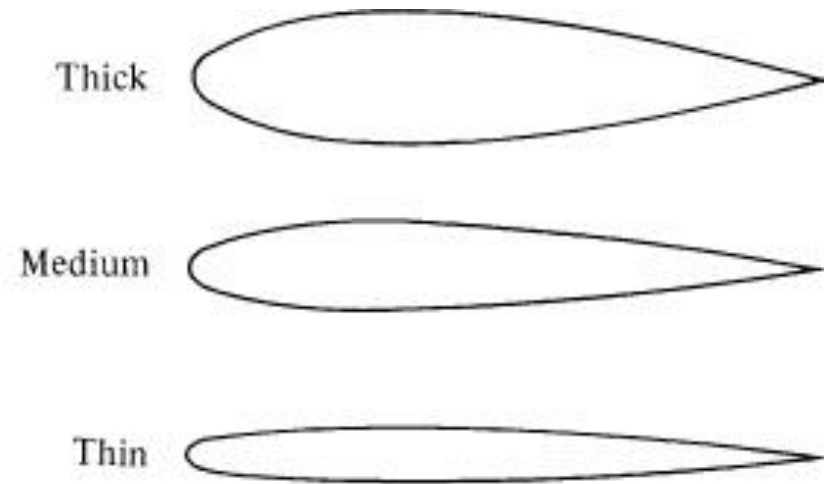
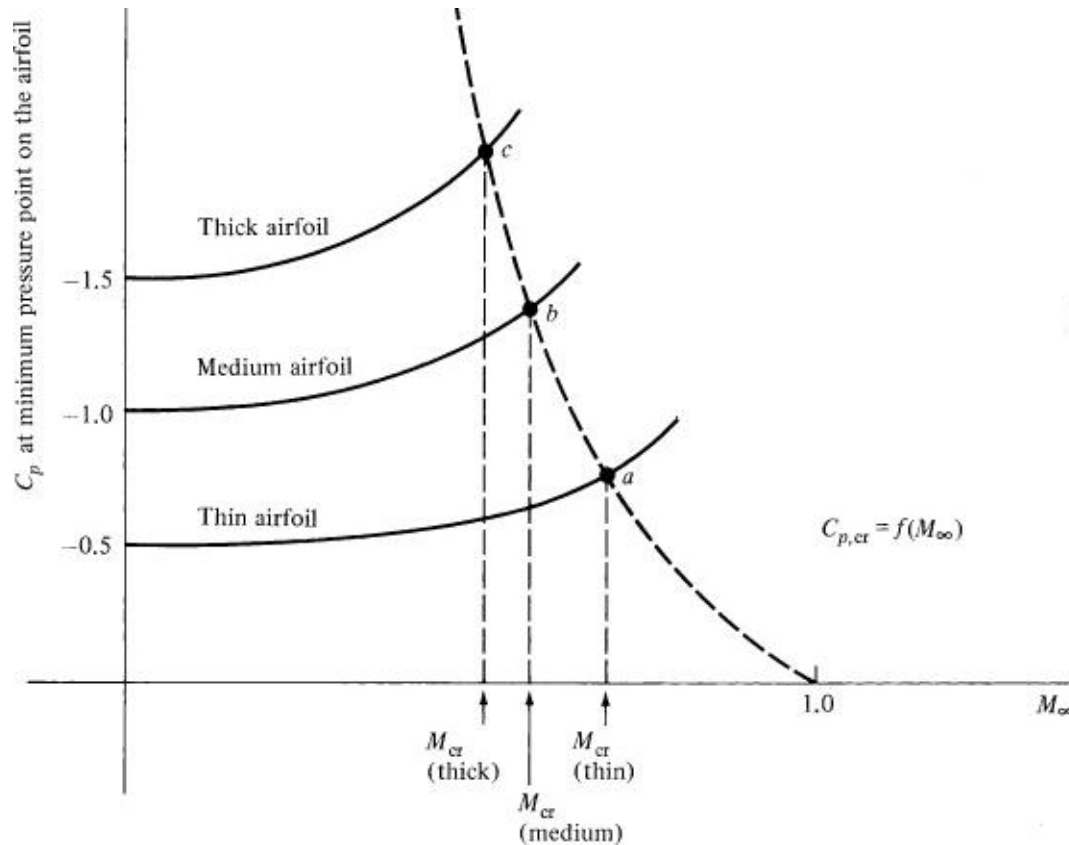
$$C_{p,CR} = \frac{2}{\gamma M_{CR}^2} \left[\left(\frac{1 + \frac{\gamma-1}{2} M_{CR}^2}{1 + \frac{\gamma-1}{2}} \right)^{\gamma/(\gamma-1)} - 1 \right]$$

- Set $M_A = 1$ (onset of supersonic flow)
- Relates $C_{p,CR}$ to M_{CR}

HOW DO WE USE THIS?

1. Plot curve of $C_{P,CR}$ vs. M_∞
2. Obtain incompressible value of C_p at minimum pressure point on given airfoil
3. Use any compressibility correction (such as P-G) and plot C_p vs. M_∞
 - Intersection of these two curves represents point corresponding to sonic flow at minimum pressure location on airfoil
 - Value of M_∞ at this intersection is M_{CR}

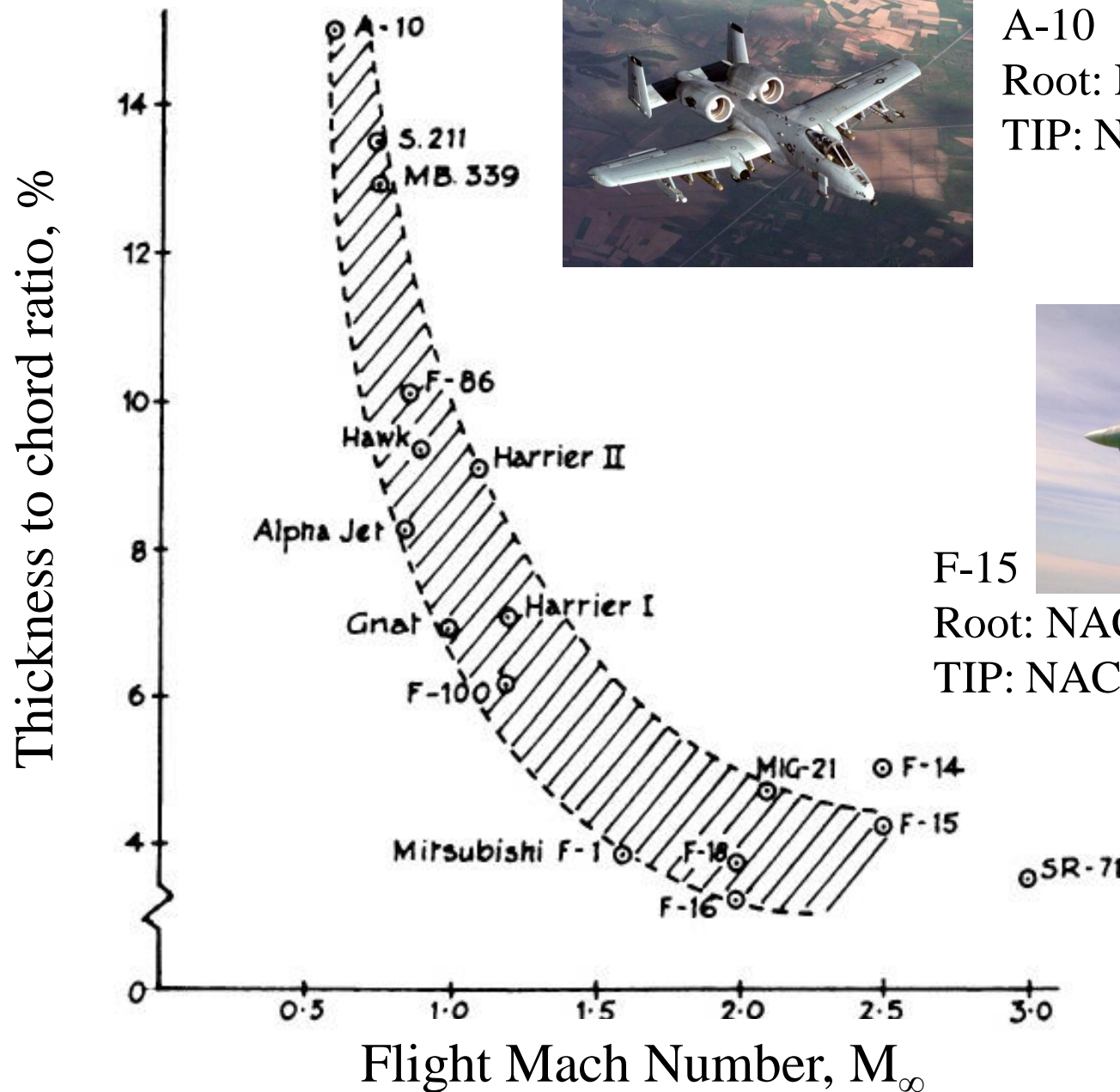




Note: thickness is relative to chord in all cases
Ex. NACA 0012 \rightarrow 12 %

- Thick airfoils have a lower critical Mach number than thin airfoils
- Desirable to have MCR as high as possible
- Implication for design \rightarrow high speed wings usually design with thin airfoils - Supercritical airfoil is somewhat thicker

THICKNESS-TO-CHORD RATIO TRENDS



A-10

Root: NACA 6716

TIP: NACA 6713



F-15

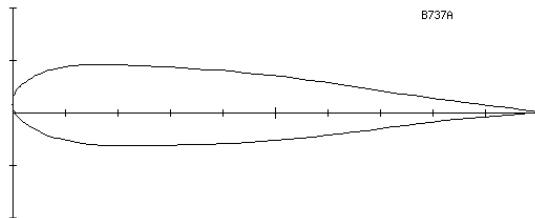
Root: NACA 64A(.055)5.9

TIP: NACA 64A203

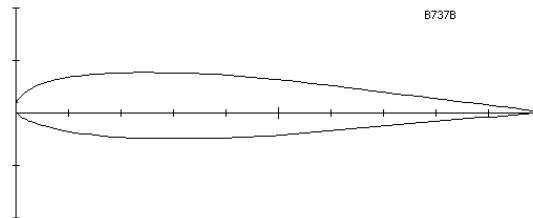
Boeing 737



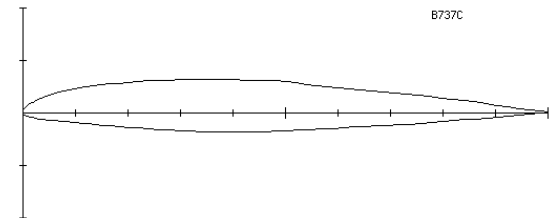
Root



Mid-Span

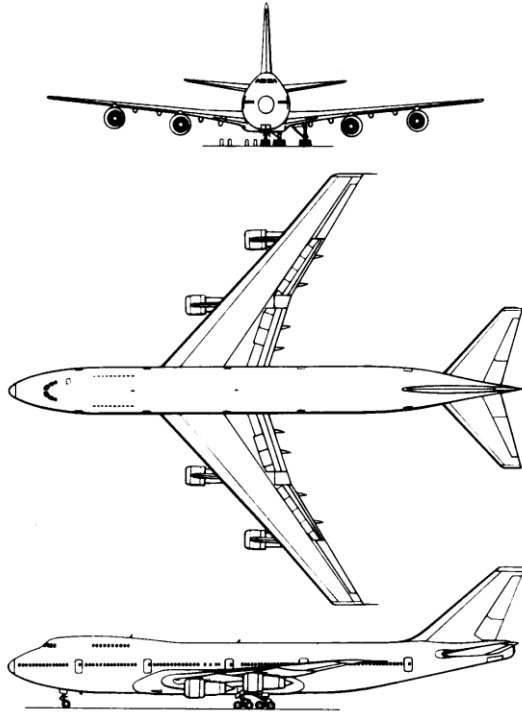


Tip

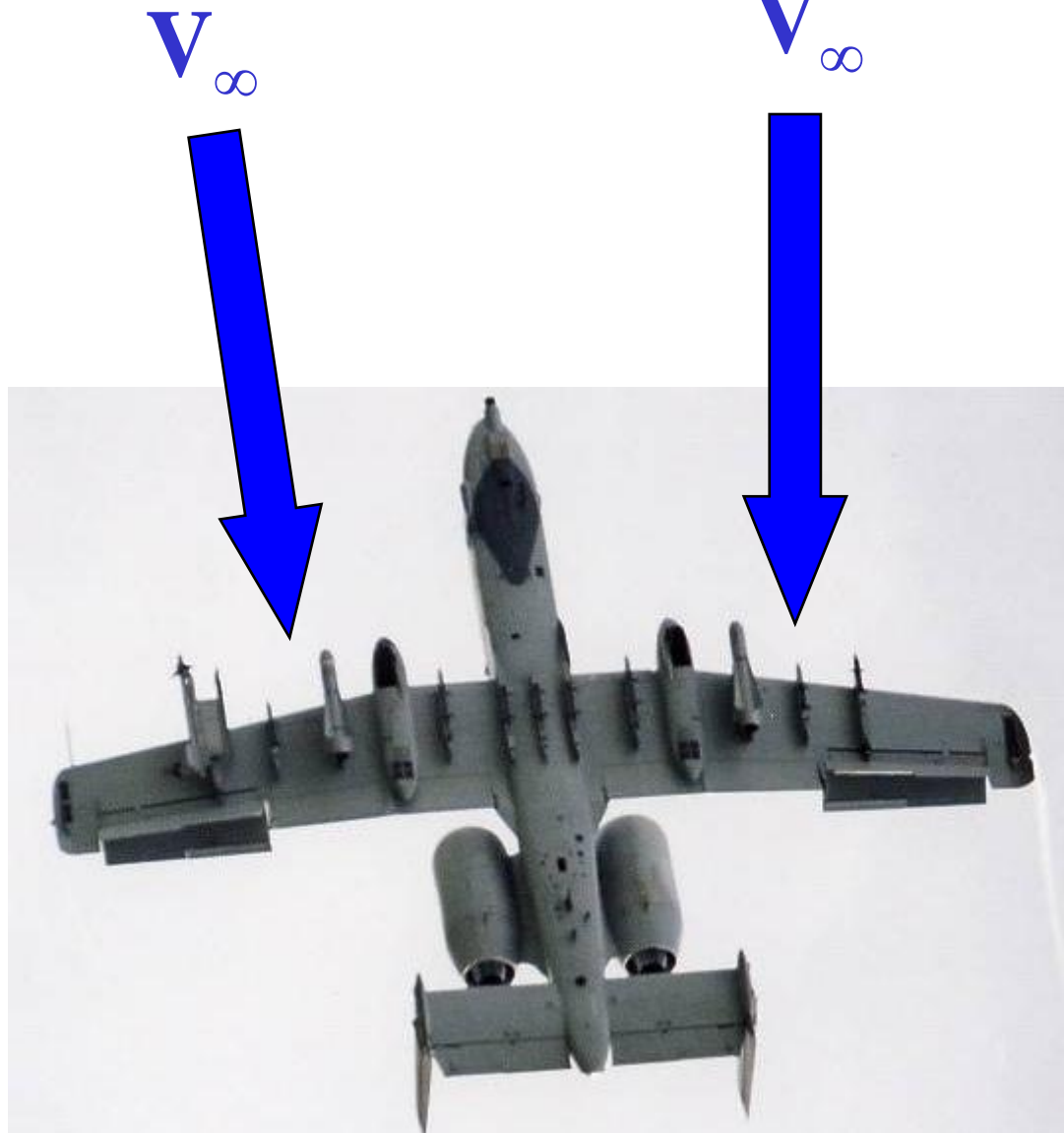


<http://www.nasg.com/afdb/list-airfoil-e.phtml>

- All modern high-speed aircraft have swept wings: WHY?
SWEPT WINGS

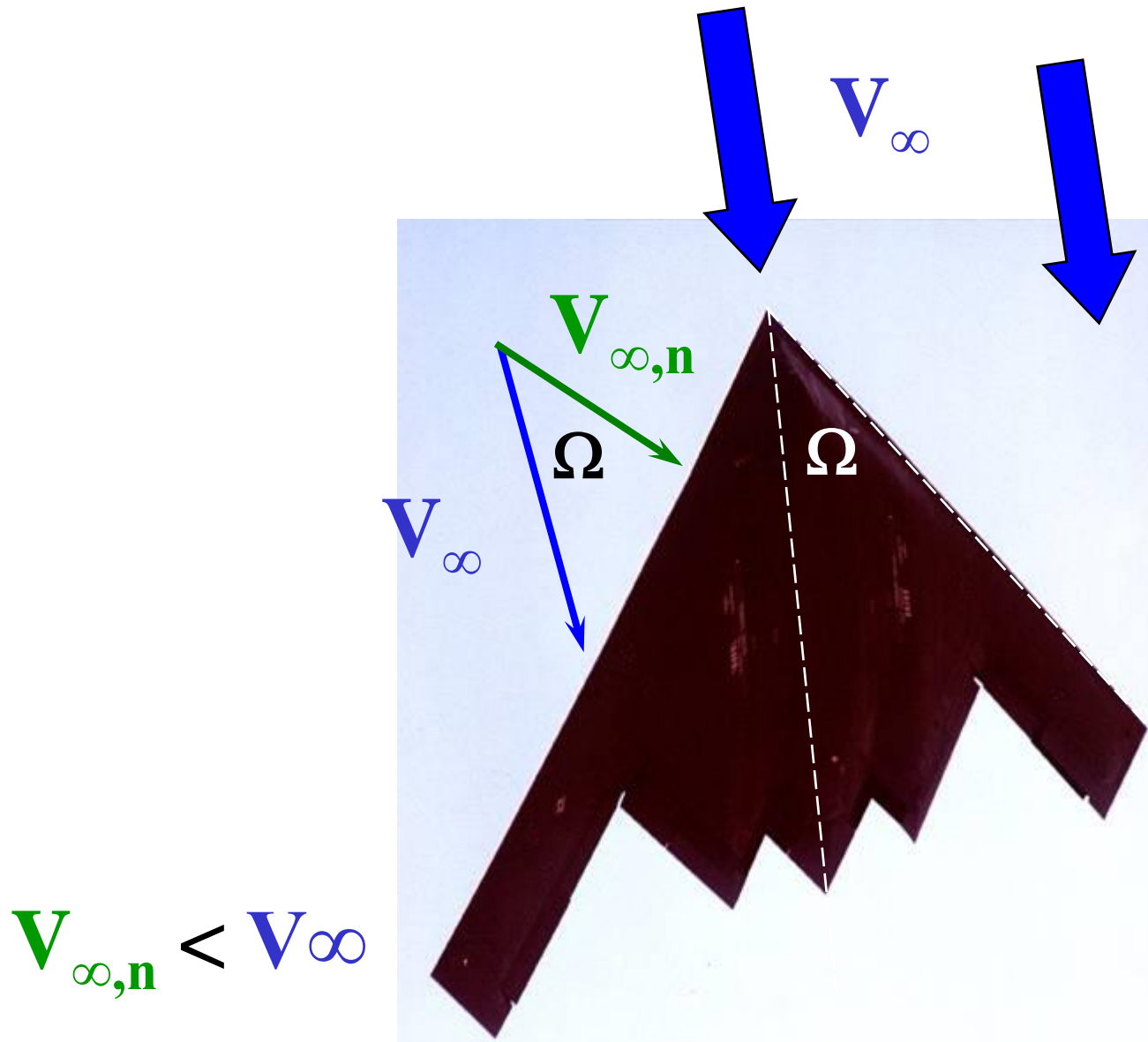


WHY WING SWEEP?



Wing sees component of flow normal to leading edge

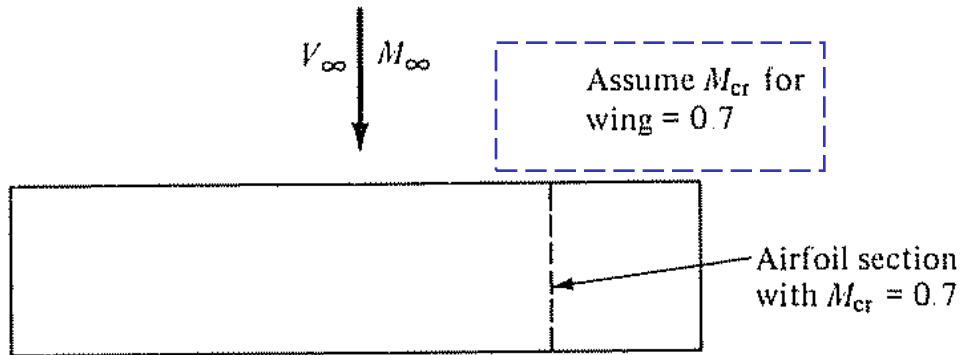
WHY WING SWEEP?



Wing sees component of flow normal to leading edge

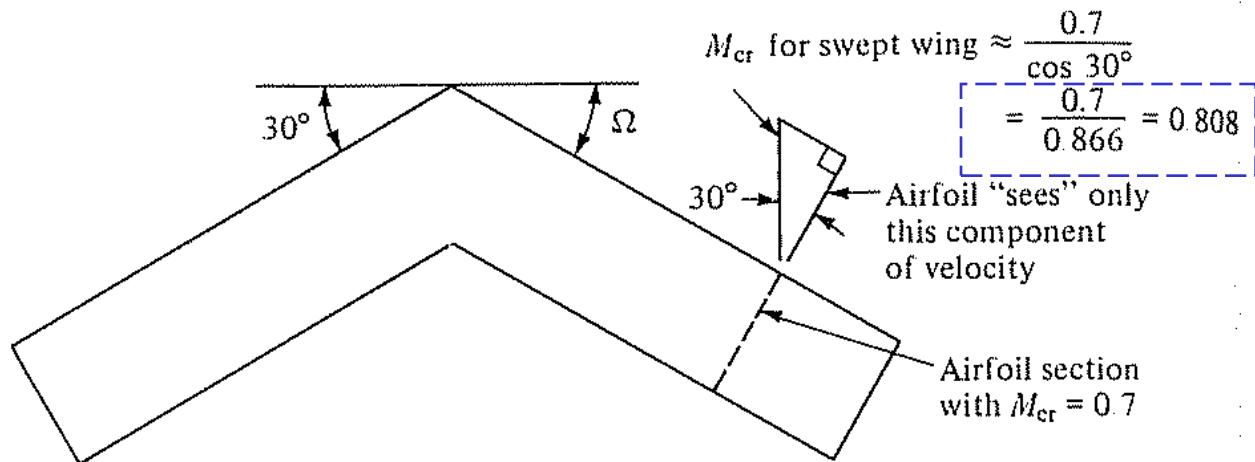
SWEPT WINGS: SUBSONIC FLIGHT

- Recall M_{CR}
- If $M_\infty > M_{CR}$ large increase in drag
- Wing sees component of flow normal to leading edge**
- Can increase M_∞
- By sweeping wings of **subsonic** aircraft, drag divergence is delayed to higher Mach numbers



Now sweep the same wing by 30°

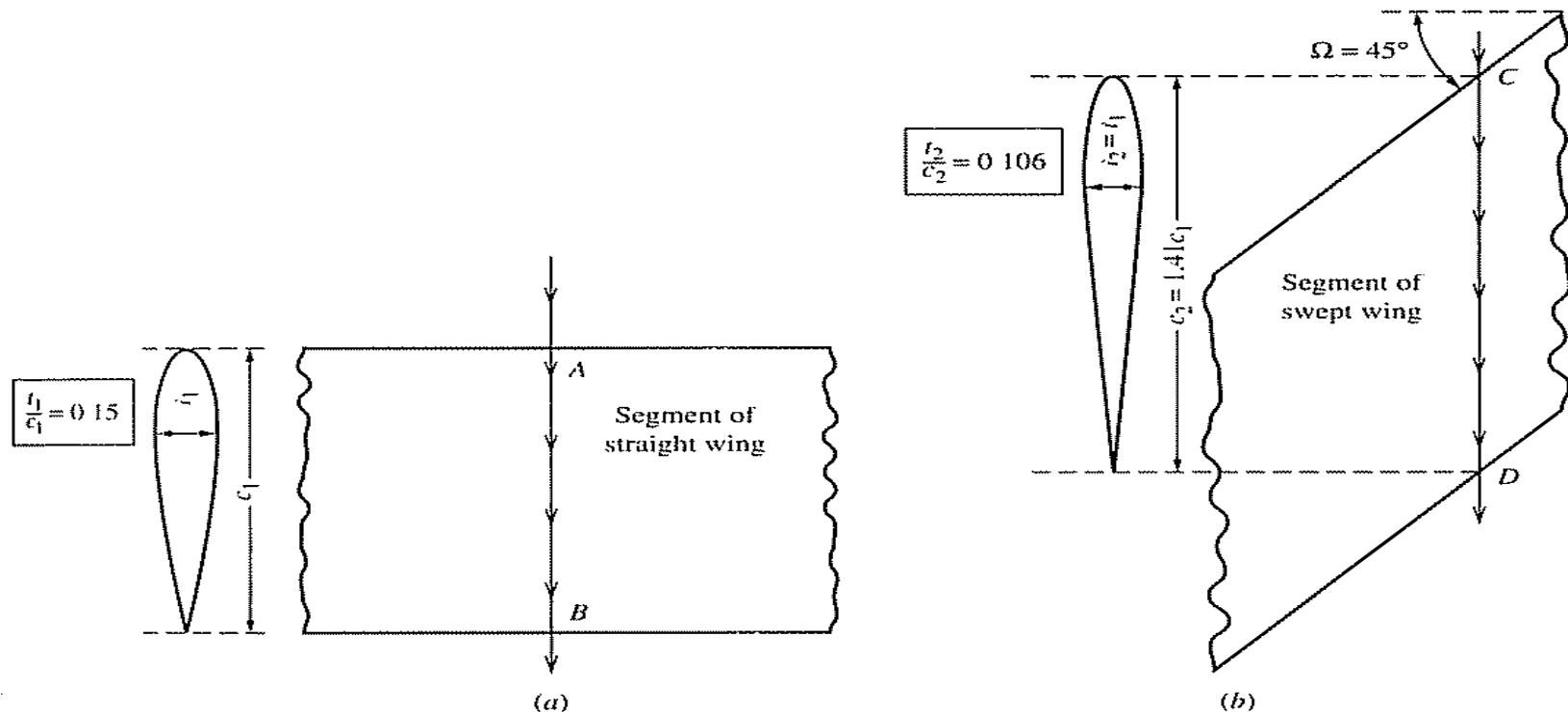
(a)



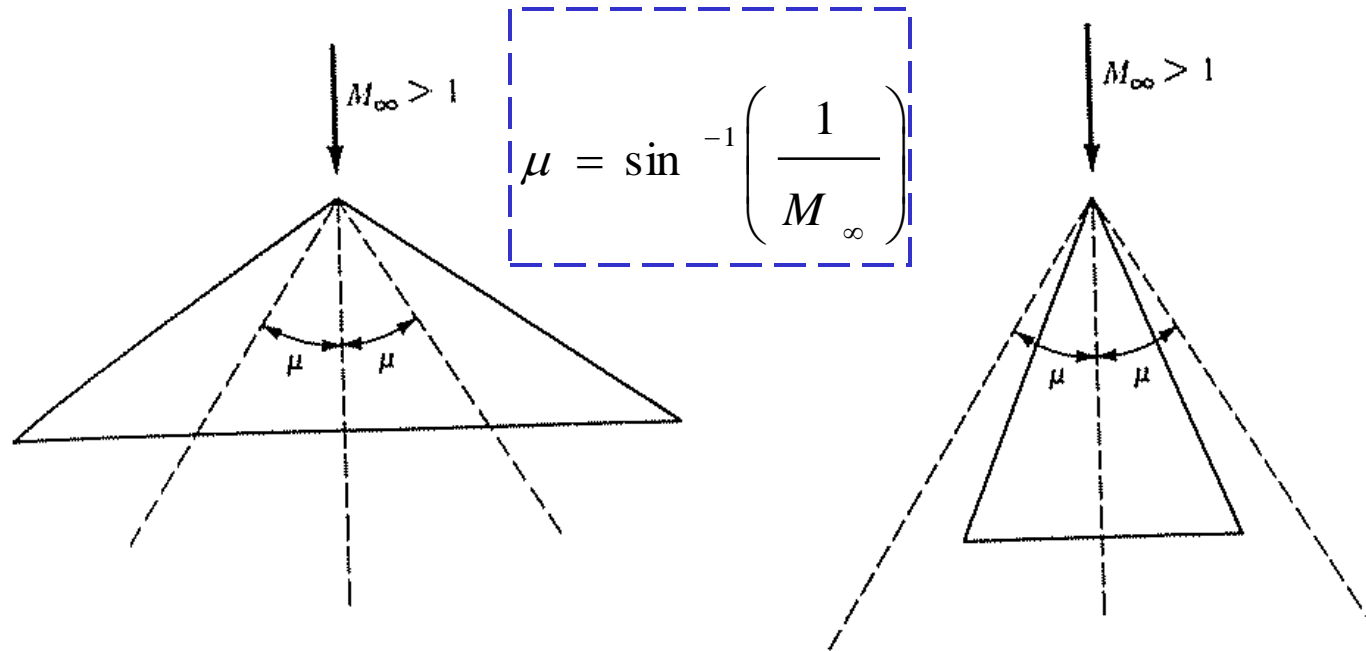
(b)

SWEPT WINGS: SUBSONIC FLIGHT

- Alternate Explanation:
 - Airfoil has same thickness but longer effective chord
 - **Effective airfoil section is thinner**
 - Making airfoil thinner increases critical Mach number
- Sweeping wing usually reduces lift for subsonic flight



SWEEP WINGS: SUPERSONIC FLIGHT



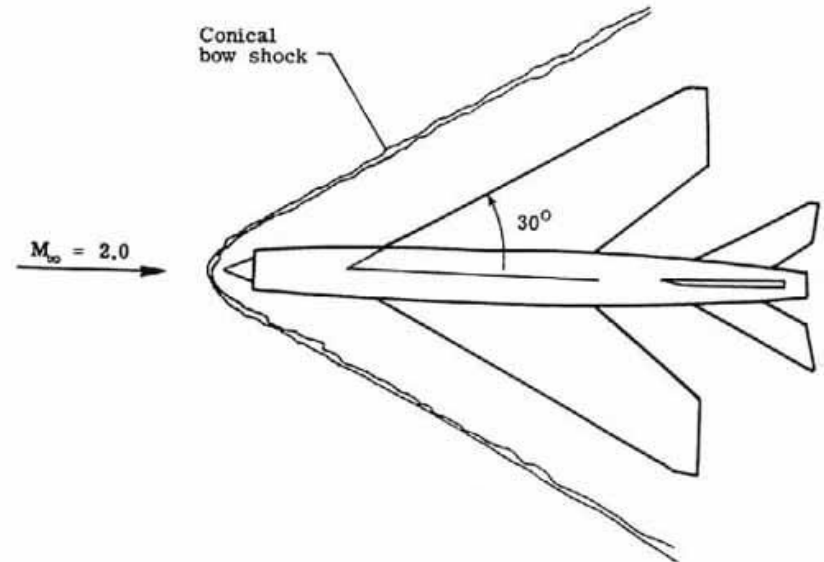
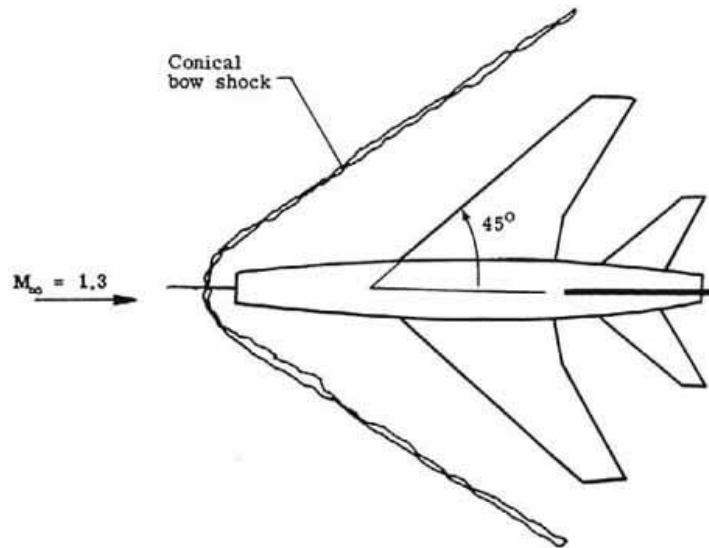
- If leading edge of swept wing is outside Mach cone, component of Mach number normal to leading edge is supersonic → Large Wave Drag
- If leading edge of swept wing is inside Mach cone, component of Mach number normal to leading edge is subsonic → Reduced Wave Drag
- For supersonic flight, swept wings reduce wave drag

WING SWEEP COMPARISON

F-100D



English Lightning



SWEPT WINGS: SUPERSONIC FLIGHT

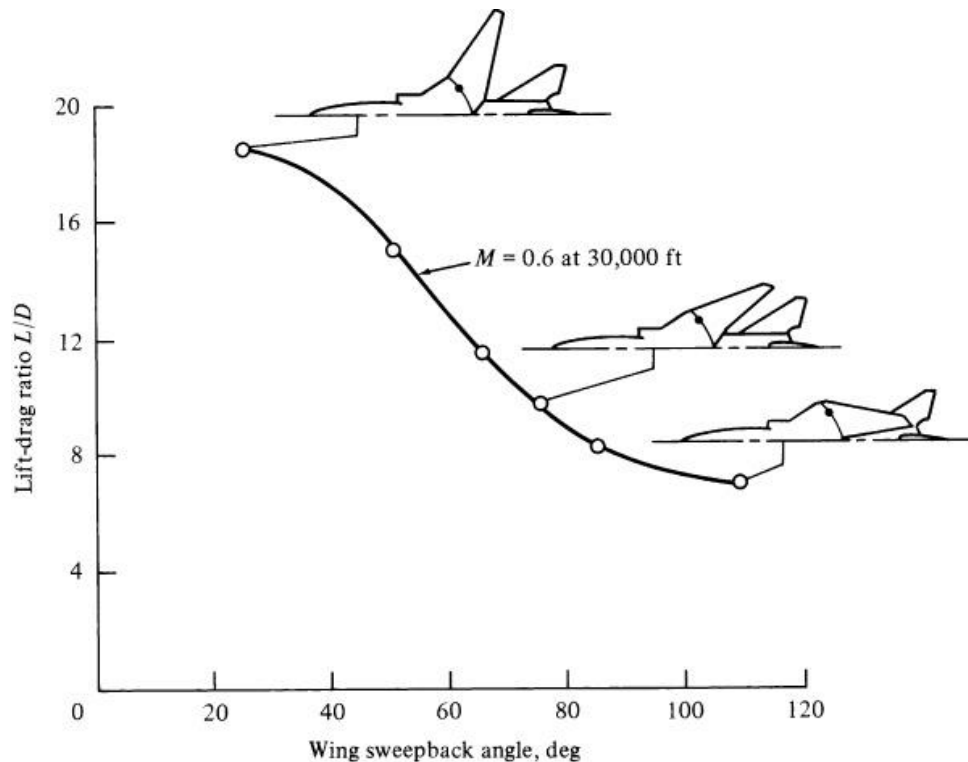


$$\theta \sim 26^{\circ}$$

$$\mu(M=1.2) \sim 56^{\circ}$$

$$\mu(M=2.2) \sim 27^{\circ}$$

WING SWEEP DISADVANTAGE



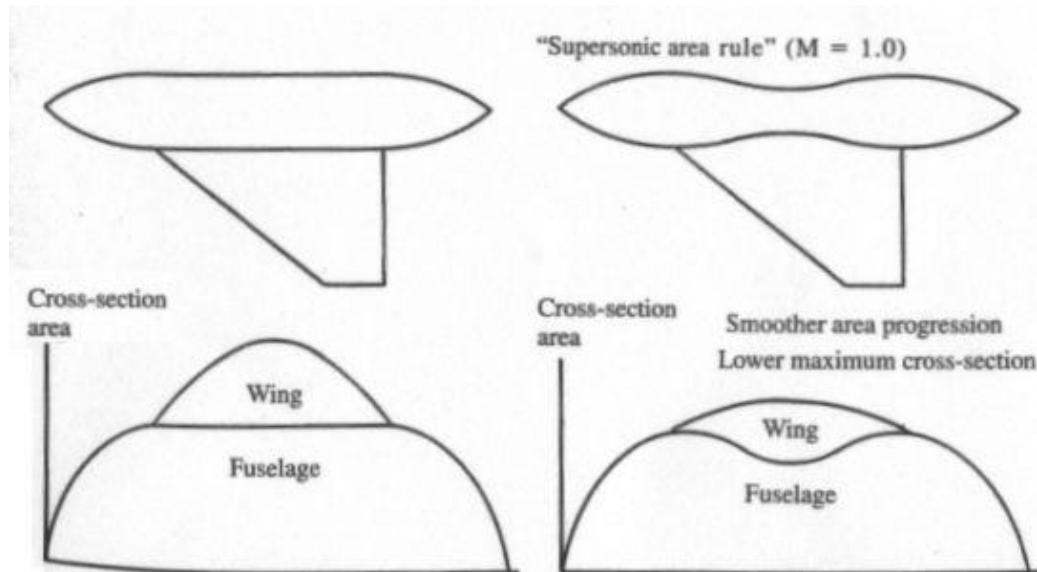
- At $M \sim 0.6$, severely reduced L/D
- Benefit of this design is at $M > 1$, to sweep wings inside Mach cone

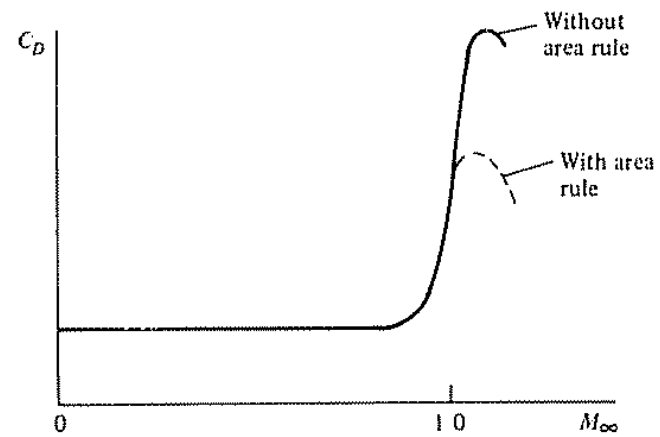
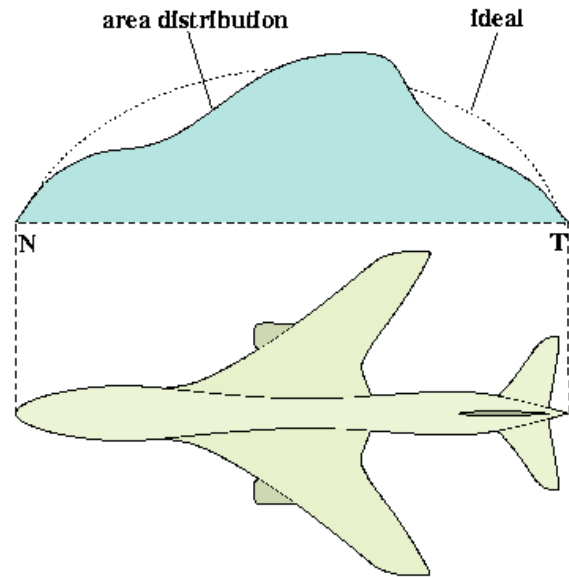


- Wing sweep beneficial in that it increases drag-divergences Mach number
- Increasing wing sweep reduces the lift coefficient

TRANSONIC AREA RULE

- Drag created related to change in cross-sectional area of vehicle from nose to tail
- Shape itself is not as critical in creation of drag, but rate of change in shape
 - Wave drag related to 2nd derivative of volume distribution of vehicle





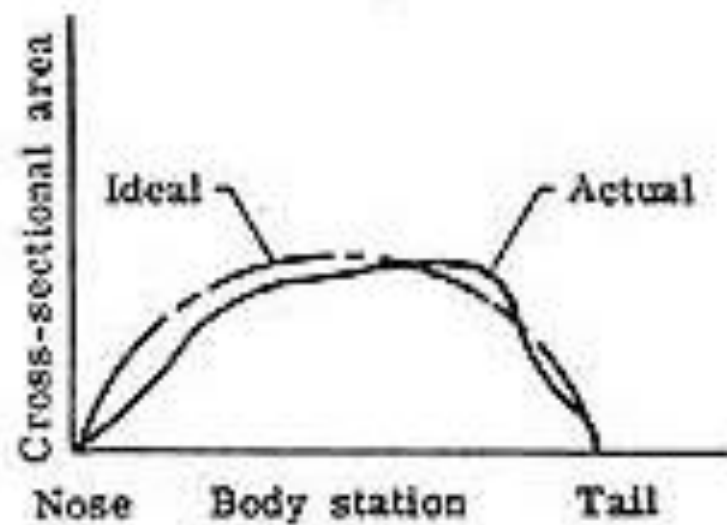
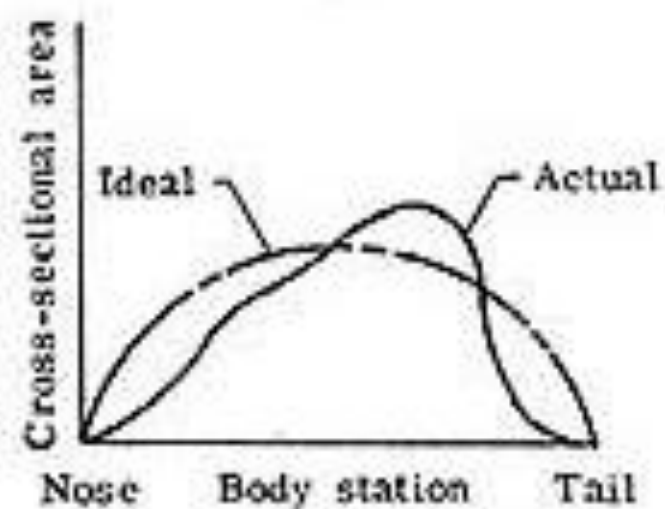
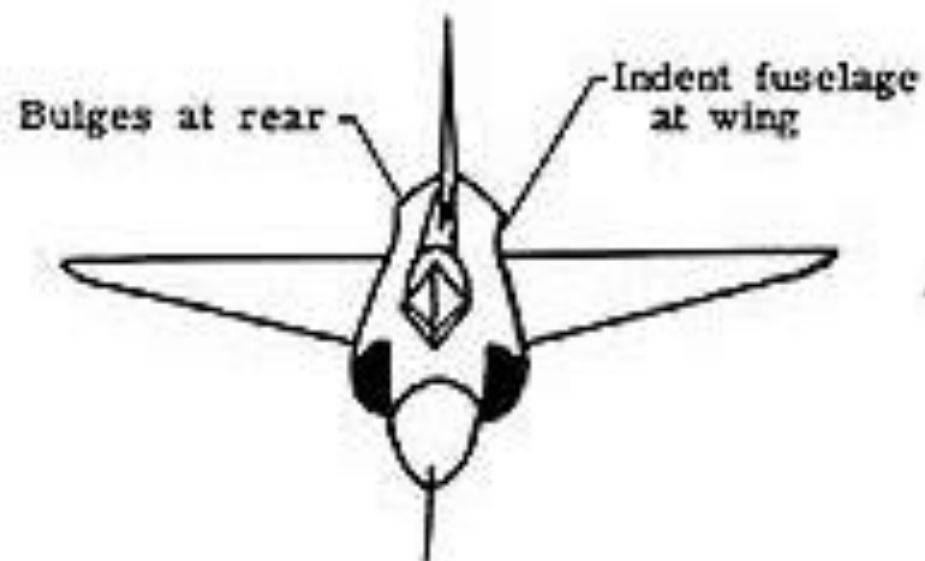
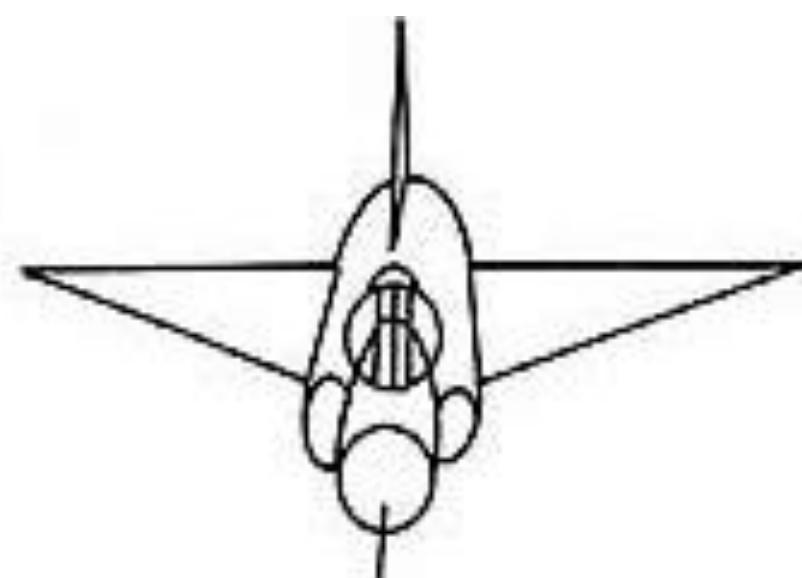
EXAMPLE: YF-102A vs. F-102A

YF-102A



YF-102





(a) YF-102A before area ruling.

(b) F-102A after area ruling.

UNIT – IV

**NUMERICAL METHODS FOR
EULER EQUATIONS, BOUNDARY
LAYER EQUATIONS**

Numerical Solutions of Differential Equations

Euler's Method

Differential Equations

A **differential equation** is a relation between the independent variable (x), the dependent variable (y) and its derivatives ($y', y'', y''', y^{(4)}, \dots$). Some of these variables might be missing from the equation.

There are many known methods to solve differential equations using non-numerical techniques (i.e. by hand). Most of these involve integration methods. As we have previously mentioned it is not always possible to find a closed form antiderivative for a given function made up of functions we commonly use.

Many situations in not only mathematics, but physics, engineering, biology, chemistry, economics as well as many other disciplines can be described using differential equations. Here are some examples:

$$\frac{d^2 s}{dt^2} = -g$$

Free Falling
Body

$$\frac{d^2 \theta}{dt^2} + \frac{g}{l} \theta = 0$$

Harmonic
Oscillator

$$\frac{dT}{dt} = k(T - T_m)$$

Newton's Law
of Cooling

$$\frac{d^2 y}{dx^2} = k \sqrt{1 + \left(\frac{dy}{dx}\right)^2}$$

Shape of a
hanging string

$$\frac{dP}{dt} = kP$$

Population
Growth

$$\frac{dP}{dt} = P(a - bP)$$

Population
Growth (limited
resources)

$$\frac{dx}{dt} = kxy$$

Spread of
Disease

$$\left. \begin{aligned} \frac{dy}{dt} &= y(\alpha - \beta x) \\ \frac{dx}{dt} &= x(-\gamma + \delta y) \end{aligned} \right\}$$

Predator-Prey

$$\frac{dT}{dt} = k(T - T_m)$$

$$\frac{1}{T - T_m} dT = k dt$$

$$\int \frac{1}{T - T_m} dT = \int k dt$$

$$\ln(T - T_m) = kt + C$$

$$e^{\ln(T - T_m)} = e^{kt + C}$$

$$T - T_m = c_0 e^{kt}$$

$$T = T_m + c_0 e^{kt}$$

To the right is an example of an explicit solution of a separable (Newton's Law of Cooling) differential equation by non-numerical methods.

Notice the c_0 term that appears in the solution. This come from not knowing the constant (+C) when you integrate.

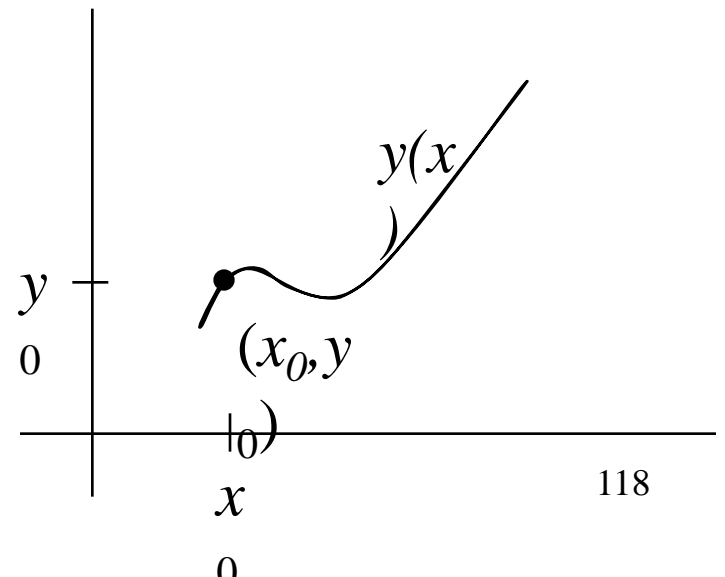
If we knew a certain value for the function say $T(1)=95^\circ$, for example, we could find the value for c_0 and thus have everything we need to solve the differential equation.

This specific piece of information is sometime called a **boundary condition** for the differential equation. This will become an important part of generating a numerical solution for a differential equation.

First Order Degree 1 Differential Equations

The study of differential equation is a subject of its own. The way they are normally studied is much like solving algebraic equations, look to see if the differential equation fits a certain pattern and then apply a certain technique to it.

The methods we will focus on here will all have the same initial problem. We will be given a differential equation with the derivative a function of the dependent and independent variable and an initial condition.



$$\frac{dy}{dx} = f(x, y) \quad \text{and} \quad y(x_0) = y_0$$

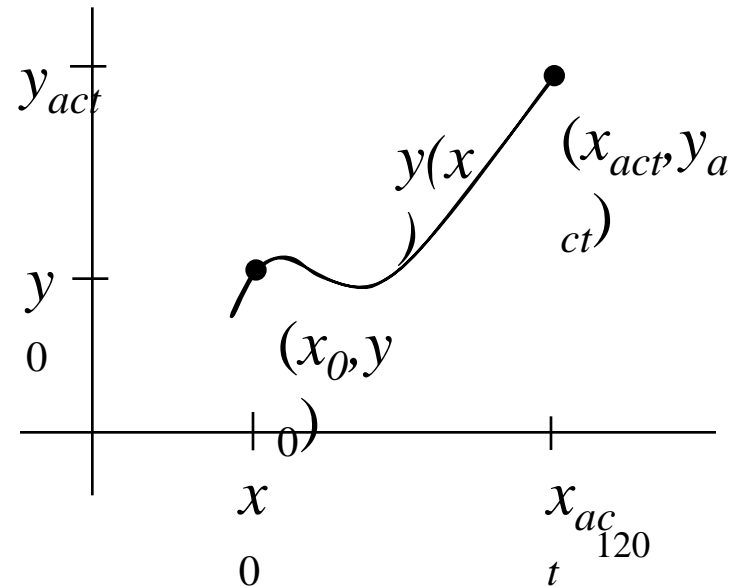
$$\frac{dy}{dx} = f(x, y)$$

$$y'(x) = f(x, y(x))$$

The solution which we call a function $y(x)$ to such an equation can be pictured graphically. The point (x_0, y_0) must be on the graph. The function $y(x)$ would also satisfy the differential equation if you plugged $y(x)$ in for y .

Solutions to Equations

The solution to a differential equation (unlike an algebraic equation) is a function. The problem with this is there are many ways to describe functions. Some can be described in terms of some equation relating the dependent and independent variable, some by a graph and some by a very complicated rule.



The way we will “solve” a differential equation is to use the definition of a function. That is to say if we are given a differential equation $y'(x) = f(x,y)$ with a boundary condition $y(x_0) = y_0$ and another value of x , say x_{act} we can find a number y_{act} so that $y(x_{act}) = y_{act}$.

The problem that exists is that y_n can not be exactly computed only estimated. There are several different ways in which to estimate the value for y_n . We will study a couple of them in this section.

The point (x_0, y_0) is called the **initial point** and the point (x_{act}, y_{act}) is called the **terminal point**.

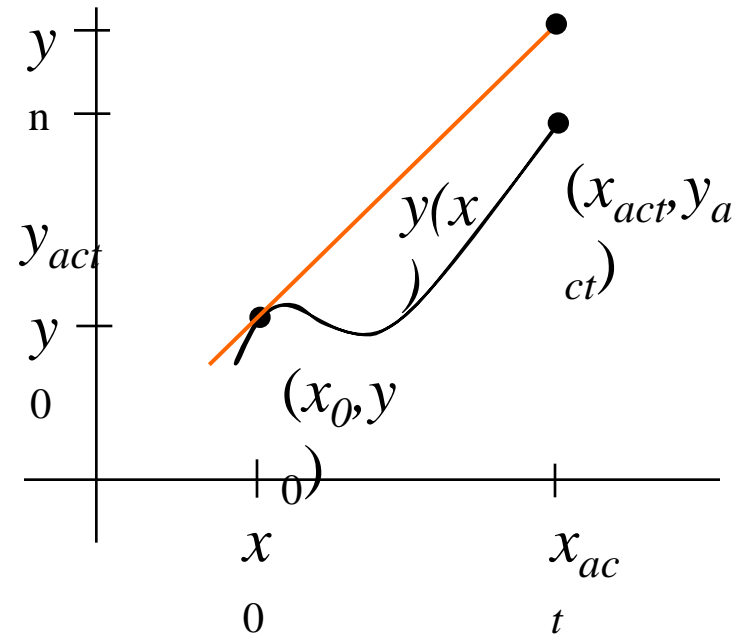
Euler's Method

The method that Euler used to estimate a solution (i.e. the corresponding value of y for a given value of x) of a differential equation was to follow the tangent line from the initial point to the terminal point.

Here we use the value y_n to estimate the value of y_{act} . This can be directly computed from the information given by the following equation

$$y_n - y_0 = f(x_0, y_0)(x_{act} - x_0)$$

$$y_n = y_0 + f(x_0, y_0)(x_{act} - x_0)$$



The insight that Euler had was to see how this estimate could be improved on. The strategy he used was to divide the interval $[x_0, x_{act}]$ (or $[x_{act}, x_0]$ in the case $x_{act} < x_0$) into equal subintervals and recompute the tangent line as you go. This would not allow the tangent line to “drift” far from the function itself. This would hopefully produce a more accurate estimate for y_{act} at the end.

To Apply Euler’s Method

1. Divide the interval n equal subintervals.
2. Compute the width of each subinterval which is
$$\Delta x = h = (x_n - x_0) / n.$$
3. Compute the sequence of points as follows:

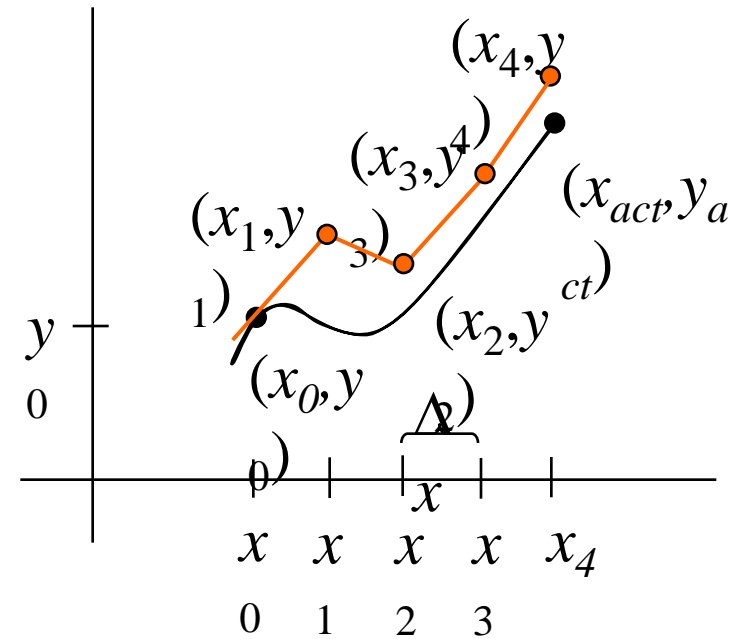
$$(x_1, y_1) = \begin{cases} x_1 = x_0 + \Delta x \\ y_1 = y_0 + f(x_0, y_0) \Delta x \end{cases}$$

$$(x_2, y_2) = \begin{cases} x_2 = x_1 + \Delta x \\ y_2 = y_1 + f(x_1, y_1) \Delta x \end{cases}$$

$$(x_3, y_3) = \begin{cases} x_3 = x_2 + \Delta x \\ y_3 = y_2 + f(x_2, y_2) \Delta x \end{cases}$$

$$(x_4, y_4) = \begin{cases} x_4 = x_3 + \Delta x \\ y_4 = y_3 + f(x_3, y_3) \Delta x \end{cases}$$

$$(x_{n+1}, y_{n+1}) = \begin{cases} x_{n+1} = x_n + \Delta x \\ y_{n+1} = y_n + f(x_n, y_n) \Delta x \end{cases}$$



In general the coordinates of the point (x_{n+1}, y_{n+1}) can be computed from the coordinates of the point (x_n, y_n) as follows:

Example:

Given the differential equation to the right with its boundary conditions find the value of $y(2)$ using Euler's method with 4 iterations.

$$\frac{dy}{dx} = x^2 y$$

$$y(0) = 1$$

$$1. x_0 = 0 \quad \text{and} \quad y_0 = 1 \quad \text{and} \quad x_4 = 2$$

$$2. h = \Delta x = \frac{2-0}{4} = \frac{1}{2}$$

$$3. \begin{cases} x_1 = 0 + \frac{1}{2} = \frac{1}{2} \\ y_1 = 1 + \left(0^2 \cdot 1\right) \frac{1}{2} = 1 \end{cases}$$

$$4. \begin{cases} x_2 = \frac{1}{2} + \frac{1}{2} = 1 \\ y_2 = 1 + \left(\left(\frac{1}{2}\right)^2 \cdot 1\right) \frac{1}{2} = 1 + \frac{1}{8} = 1.125 \end{cases}$$

$$5. \begin{cases} x_3 = 1 + \frac{1}{2} = 1.5 \\ y_3 = 1.125 + \left((1)^2 \cdot 1.125\right) \frac{1}{2} = 1.125 + .5625 = 1.6875 \end{cases}$$

$$6. \begin{cases} x_4 = 1.5 + \frac{1}{2} = 2 \\ y_4 = 1.6875 + \left((1.5)^2 \cdot 1.6875\right) \frac{1}{2} = 1.6875 + 1.89844 = 3.58594 \end{cases}$$

The actual solution is:

$$y = e^{\frac{1}{3}x^3}$$

$$y(2) = e^{\frac{8}{3}} = 14.3919$$

Some explicit integration method for a linear wave equation.

- Some basic schemes $\partial_t f + c \partial_x f = 0$ are presented. $f = f(t, x)$,
 $c = \text{constant}$.

(1) FTCS (Forward in Time and Central difference Scheme).

$$w_j^{n+1} = w_j^n - \frac{1}{2} \nu (w_{j+1}^n - w_{j-1}^n)$$

$\nu = \frac{c \tau}{h}$,
Courant number.

(2) Lax (- Friedrich) scheme.

$$w_j^{n+1} = \frac{1}{2}(w_{j+1}^n + w_{j-1}^n) - \frac{1}{2} \nu (w_{j+1}^n - w_{j-1}^n)$$

(3) Leap-Frog scheme

$$w_j^{n+1} = w_j^{n-1} - \nu (w_{j+1}^n - w_{j-1}^n)$$

(4) Lax-Wendroff scheme

$$w_j^{n+1} = w_j^n - \frac{1}{2} \nu (w_{j+1}^n - w_{j-1}^n) + \frac{1}{2} \nu^2 (w_{j+1}^n - 2w_j^n + w_{j-1}^n)$$

(5) 1st order upwind scheme

$$w_j^{n+1} = w_j^n - \nu (w_j^n - w_{j-1}^n)$$

$c > 0$ constant.

Some explicit integration method for a linear wave equation

Lax, Lax-Wendroff, 1st order upwind schemes can be understood as FTCS scheme + Diffusion term.

(1) Explicit Euler scheme (FTCS : Forward in Time and Central difference). $w_j^{n+1} = w_j^n - \frac{1}{2} \nu (w_{j+1}^n - w_{j-1}^n)$

$$\nu = \frac{c \tau}{h},$$

Courant number.

(2) Lax (- Friedrich) scheme.

$$\begin{aligned} w_j^{n+1} &= \frac{1}{2}(w_{j+1}^n + w_{j-1}^n) - \frac{1}{2} \nu (w_{j+1}^n - w_{j-1}^n) \\ &= w_j^n - \frac{1}{2} \nu (w_{j+1}^n - w_{j-1}^n) + \frac{1}{2} (w_{j+1}^n - 2w_j^n + w_{j-1}^n) \end{aligned}$$

(3) Lax-Wendroff scheme.

$$w_j^{n+1} = w_j^n - \frac{1}{2} \nu (w_{j+1}^n - w_{j-1}^n) + \frac{1}{2} \nu^2 (w_{j+1}^n - 2w_j^n + w_{j-1}^n)$$

(4) 1st order upwind scheme (weakest diffusion)

$$\begin{aligned} w_j^{n+1} &= w_j^n - \nu (w_j^n - w_{j-1}^n) \\ &= w_j^n - \frac{1}{2} \nu (w_{j+1}^n - w_{j-1}^n) + \frac{1}{2} \nu (w_{j+1}^n - 2w_j^n + w_{j-1}^n) \\ &= w_j^n - \frac{1}{2} \nu (w_{j+1}^n - w_{j-1}^n) + \frac{1}{2} |\nu| (w_{j+1}^n - 2w_j^n + w_{j-1}^n) \end{aligned}$$

(Can be used also for negative c.)

von Neumann stability analysis.

- A method to analyze the stability of numerical scheme for linear PDE (assuming equally spaced grid points and periodic boundary condition). $w_j^n = \sum_{\theta \in \Theta} \tilde{c}_\theta^n \exp[ij\theta]$,
- Consider that the finite difference equation has the following solution.
- Then the perturbation $\delta_j^n = \tilde{w}_j^n - w_j^n$ can be also written

$$i = \sqrt{-1}, \text{ and, in each dimension,}$$

$$\theta_a := \frac{2\pi k}{N_a}, \quad k = 0, \dots, N_a - 1.$$

$$\delta_j^n = \sum_{\theta \in \Theta} c_\theta^n \exp[ij\theta].$$

The FD equation is $(B_0 w^{n+1})_j = (B_1 w^n)_j + (B_2 S^n)_j + (B_3 S^{n+1})_j$,

hence, perturbation satisfies, $(B_0 \delta^{n+1})_j = (B_1 \delta^n)_j$,

which can be rewritten,

$$\sum_k a_k \delta_{j+k}^{n+1} = \sum_k b_k \delta_{j+k}^n.$$

Substituting the Fourier transform above, we have $c_\theta^{n+1} = g(\theta) c_\theta^n$.

Amplification factor $g(\theta)$ is define $g(\theta) = \frac{\sum_k b_k \exp[ik\theta]}{\sum_k a_k \exp[ik\theta]}$.

$$\partial_t f + \partial_x(cf) = 0 \rightarrow w_j^{n+1} = w_j^n - \frac{\tau}{h} \left(F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n \right).$$

$$(1) \text{ Explicit Euler : } F_{j+\frac{1}{2}}^n = \frac{1}{2}(\tilde{f}_{j+1}^n + \tilde{f}_j^n) = \begin{cases} \tilde{f}_j^n + \frac{1}{2}(\tilde{f}_{j+1}^n - \tilde{f}_j^n) & c > 0 \\ \tilde{f}_{j+1}^n - \frac{1}{2}(\tilde{f}_{j+1}^n - \tilde{f}_j^n) & c < 0 \end{cases}$$

$$(2) \text{ Lax : } \begin{aligned} F_{j+\frac{1}{2}}^n &= \frac{1}{2}(\tilde{f}_{j+1}^n + \tilde{f}_j^n) - \frac{1}{2\nu}(\tilde{f}_{j+1}^n - \tilde{f}_j^n) \\ &= \frac{1}{2}\left(1 - \frac{1}{\nu}\right)\tilde{f}_{j+1}^n + \frac{1}{2}\left(1 + \frac{1}{\nu}\right)\tilde{f}_j^n \\ &= \begin{cases} \tilde{f}_j^n + \frac{1}{2}\left(1 - \frac{1}{\nu}\right)(\tilde{f}_{j+1}^n - \tilde{f}_j^n) & c > 0 \\ \tilde{f}_{j+1}^n - \frac{1}{2}\left(1 + \frac{1}{\nu}\right)(\tilde{f}_{j+1}^n - \tilde{f}_j^n) & c < 0 \end{cases} \end{aligned}$$

$$(3) \text{ Lax-Wendroff : } \begin{aligned} F_{j+\frac{1}{2}}^n &= \frac{1}{2}(\tilde{f}_{j+1}^n + \tilde{f}_j^n) - \frac{1}{2}\nu(\tilde{f}_{j+1}^n - \tilde{f}_j^n) \\ &= \frac{1}{2}(1 - \nu)\tilde{f}_{j+1}^n + \frac{1}{2}(1 + \nu)\tilde{f}_j^n \\ &= \begin{cases} \tilde{f}_j^n + \frac{1}{2}(1 - \nu)(\tilde{f}_{j+1}^n - \tilde{f}_j^n) & c > 0 \\ \tilde{f}_{j+1}^n - \frac{1}{2}(1 + \nu)(\tilde{f}_{j+1}^n - \tilde{f}_j^n) & c < 0 \end{cases} \end{aligned}$$

$$(4) \text{ 1}^{\text{st}} \text{ order upwind : } \begin{aligned} F_{j+\frac{1}{2}}^n &= \frac{1}{2}[\tilde{f}_{j+1}^n + \tilde{f}_j^n - |c|(w_{j+1}^n - w_j^n)] \\ &= \begin{cases} \tilde{f}_j^n & c > 0 \\ \tilde{f}_{j+1}^n & c < 0 \end{cases} \end{aligned}$$

κ – scheme : A parametrization of representative linear schemes. (Van Leer)

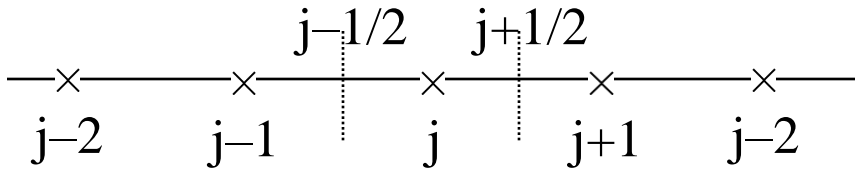
For a linear PDE $\partial_t f + c \partial_x f = 0$, write a Taylor expansion in time

$$\begin{aligned} f(t + \tau, x) &= f(t, x) + \tau \partial_t f + \frac{1}{2} \tau^2 \partial_t^2 f + O(\tau^3) \\ &= f(t, x) - c\tau \partial_x f + \frac{1}{2} (c\tau)^2 \partial_x^2 f + O(\tau^3) \end{aligned}$$

Approximate the second term in RHS as $-c\tau \partial_x f(t_n, x_j) \approx -\frac{\tau}{h} \left(F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n \right)$.

$$\begin{cases} F_{j+\frac{1}{2}}^n = c \left[\frac{1}{2}(w_j^n + w_{j+1}^n) + \frac{\kappa - 1}{4}(w_{j+1}^n - 2w_j^n + w_{j-1}^n) \right], & \text{for } c > 0. \\ F_{j+\frac{1}{2}}^n = c \left[\frac{1}{2}(w_j^n + w_{j+1}^n) + \frac{\kappa - 1}{4}(w_{j+2}^n - 2w_{j+1}^n + w_j^n) \right], & \text{for } c < 0. \end{cases}$$

And the third term as

$$\frac{1}{2} (c\tau)^2 \partial_x^2 f(t_n, x_j) \approx$$


$$\begin{cases} \frac{1}{2} \nu^2 \left[\mu(w_{j+1}^n - 2w_j^n + w_{j-1}^n) + (1 - \mu)(w_j^n - 2w_{j-1}^n + w_{j-2}^n) \right], & c > 0. \\ \frac{1}{2} \nu^2 \left[\mu(w_{j+1}^n - 2w_j^n + w_{j-1}^n) + (1 - \mu)(w_{j+2}^n - 2w_{j+1}^n + w_j^n) \right], & c < 0. \end{cases}$$

κ -scheme (continued)

Deriving a FD scheme explicitly for w_j^n , one finds that the coefficients of w_j^n are effectively proportional to $\kappa - 2\mu|v|$. Hence one parameter may be eliminated. A choice $\mu = \frac{1}{2} + \frac{3}{2} - \kappa|v|$ results in the form of κ -scheme derived by Van Leer.

κ -scheme becomes $\left\{ \begin{array}{ll} \kappa = 1/3 & \text{Quickest scheme} \\ \kappa = 1/2 & \text{Quick scheme} \\ \kappa = 0 & \text{Fromm scheme (optimal)} \end{array} \right\}$ Leonard (1979)

Different from the Van Leer's choice $\left\{ \begin{array}{ll} \mu = \kappa = 1 & \text{Lax-Wendroff} \\ \kappa - 2\mu v = -1 & \text{Warming \& Beam} \end{array} \right.$

Method of lines : (yet another idea for discretization.)

In the κ -scheme (and the linear scheme we have seen) a dependence on the time step τ is included in the Courant number v . To avoid this, one discretizes PDE along spatial direction first as $\partial_t f + c \partial_x f = 0 \rightarrow \frac{df}{dt} + L_h f = 0$

For the FD operator L_h , choose e.g. κ -scheme, then apply ODE integration scheme such as RK4.

Monotonicity preservation of a linear advection equation $\partial_t f + c\partial_x f = 0$

A function $f(s)$ is monotonically increasing (decreasing),
if $\forall x, y \in \mathbf{R}, \quad x \geq y \Rightarrow f(x) \leq f(y) \text{ , } (f(x) \geq f(y))$.

A linear advection equation $\partial_t f + c\partial_x f = 0, \quad t > 0, \quad x \in \mathbf{R}$,
preserves monotonicity i.e. if $f(0, x)$: monotonic) $f(t, x)$: monotonic,
since its general solution is $f(t, x) = f(x - ct)$.

Consider a finite difference scheme that generates numerical
approximation w_j^n to $f(t_n, x_j)$. $\{w_j^n\}_{x_j \in G_h}$: data at the time step n .

Definition: (Monotonicity preserving scheme)

A numerical scheme is called monotonicity preserving if for every non-
increasing (decreasing) initial data $\{w_j^0\}_{x_j \in G_h}$ the numerical solution
 $\{w_j^n\}_{x_j \in G_h}$ for $\forall n = 1, 2, \dots$ is non-increasing (decreasing).

Godunov's theorem

For the uniform grid $x_j = jh$ and the constant time step $t_n = n\tau$, the (explicit or implicit) one-step scheme, in which $\{w_j^{n+1}\}_{x_j \in G_h}$ at the $(n+1)$ th step is uniquely determined from $\{w_j^n\}_{x_j \in G_h}$ at the n th step, is written $w_j^{n+1} = \sum_m c_m w_{j+m}^n$

Theorem: (Godunov: Monotonicity preservation)

The above one-step scheme is monotonicity preserving if and only if

$$c_m \geq 0, \text{ for } \forall m.$$

Theorem: (Godunov's order barrier theorem)

Linear one-step second-order accurate numerical schemes for the convection equation $\partial_t f + c \partial_x f = 0$ cannot be monotonicity preserving, unless $\frac{|c|\tau}{h} \in \mathbb{N}$

Two 2nd order schemes: Lax-Wendroff and Warming & Beam schemes

From the local truncation error formula,

$$\tau_j^n = \frac{1}{\tau} \sum_{k=0}^N \left[\left(\frac{-c\tau}{h} \right)^k - \sum_m m^k c_m \right] \frac{\partial_x^k f(t_n, x_j)}{k!} h^k + O(\tau^N) + O\left(\frac{h^{N+1}}{\tau}\right)$$

the 2nd order scheme needs to satisfy $\left(\frac{-c\tau}{h} \right)^k - \sum_m m^k c_m, \quad k = 0, 1, 2$

Choice of 3 grid points $j-1, j, j+1$, ($m = -1, 0, +1$) results in Lax-Wendroff.

$$w_j^{n+1} = w_j^n - \frac{1}{2} \nu (w_{j+1}^n - w_{j-1}^n) + \frac{1}{2} \nu^2 (w_{j+1}^n - 2w_j^n + w_{j-1}^n)$$

(Explicit Euler + Diffusion term centered at j .)

Choice of 3 grid points $j-2, j-1, j$, ($m = -2, -1, 0$) results in Warming & Beam.

$$w_j^{n+1} = w_j^n - \nu (w_j^n - w_{j-1}^n) - \frac{1}{2} \nu(1 - \nu) (w_j^n - 2w_{j-1}^n + w_{j-2}^n)$$

(1st order upwind + Diffusion term centered at $j-1$.)

Writing these in the flux form $w_j^{n+1} = w_j^n - \frac{\tau}{h} \left(F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n \right).$

Lax-Wendroff:
$$F_{j+\frac{1}{2}}^n = \tilde{f}_j^n + \frac{1}{2}(1 - \nu) (\tilde{f}_{j+1}^n - \tilde{f}_j^n)$$

Warming & Beam:
$$F_{j+\frac{1}{2}}^n = \tilde{f}_j^n + \frac{1}{2}(1 - \nu) (\tilde{f}_j^n - \tilde{f}_{j-1}^n)$$

Flux Numerical Methods

Flux Basics

The finite-volume formulation of the conservation equations resulted in the equation

$$\frac{d \hat{Q}}{dt} = \hat{P} - \hat{F}$$

where \hat{F} was the flux of the flow across the control surface resulting from the approximation of the surface integral. For a finite-volume cell, the flux was expressed as

$$\hat{F} = \sum_{f=1}^{nf} \hat{F}_f$$

Where

$$\hat{F}_f = [(\vec{v} - \vec{g})Q - \mathbf{D}]_f \cdot (\hat{n} dS)_f$$

It was assumed that the flux was uniform over the cell face.

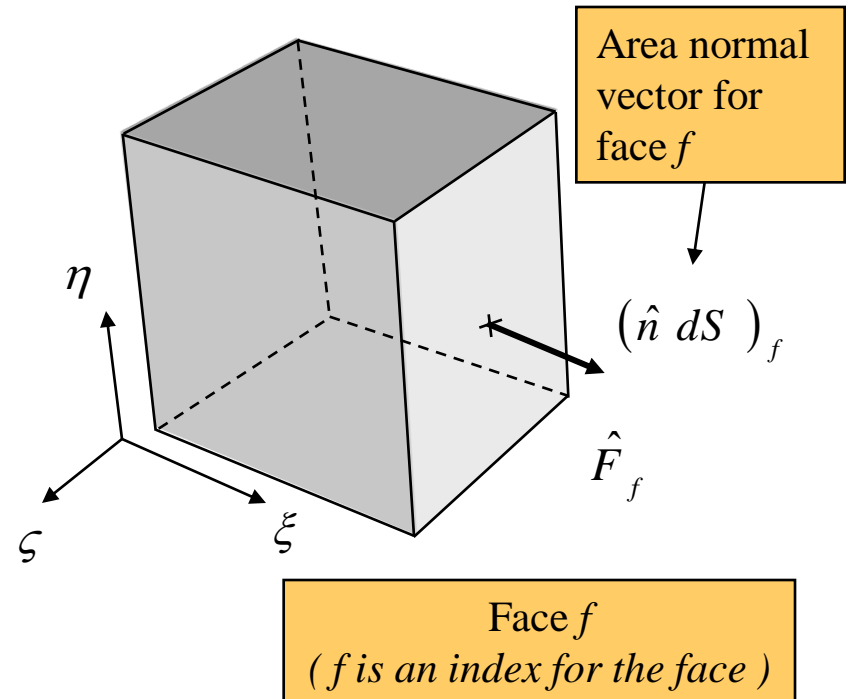
Fluxes on a Hexahedral Cell

A hexahedral cell contains 6 quadrilateral faces, thus

$$\hat{F} = \sum_{f=1}^{nf=6} \hat{F}_f$$

$$\hat{F}_f = [(\vec{v} - \vec{g})Q - \mathbf{D}]_f \cdot (\hat{n} dS)_f$$

where again,



Numerical Flux on a Cell Face

The numerical flux on a cell face is

$$\hat{F}_f = [(\vec{v} - \vec{g})Q - \mathbf{D}]_f \cdot (\hat{n} dS)_f$$

The normal area vector $(\hat{n} dS)_f$ is usually easily defined for a quadrilateral or triangular cell face. The focus of the rest of this discussion is on numerical methods for computing

$$[(\vec{v} - \vec{g})Q - \mathbf{D}]_f$$

at a cell face.

We first will assume that \vec{g} is a known velocity for the cell face.

Numerical Flux on a Cell Face

We first consider that we have the states of the flow on the “left” and “right” of the cell face, Q_L and Q_R . Our objective is to find the cell face flux.

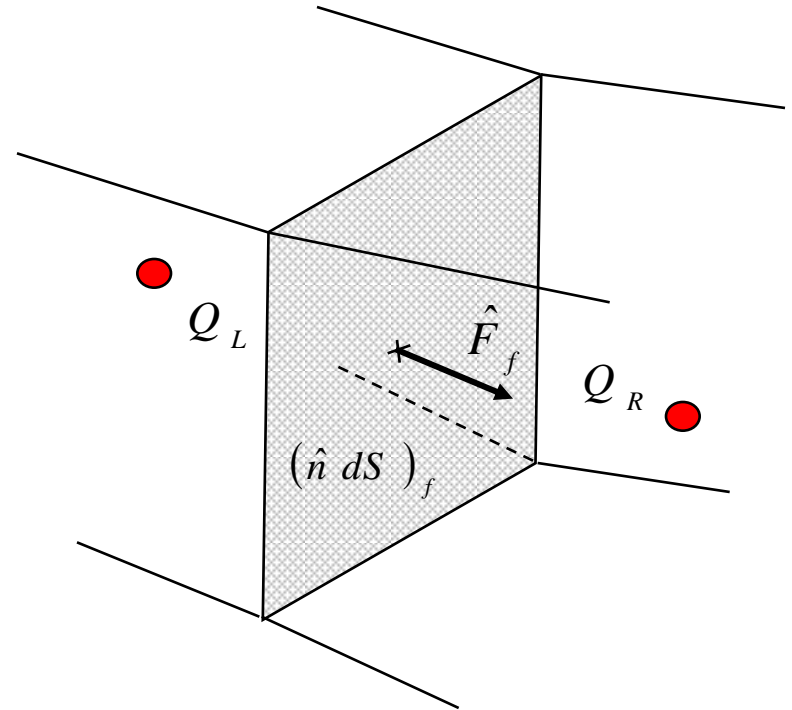
$$\hat{F}_f = \hat{F}_f(Q_L, Q_R)$$

One can define

$$\hat{F}_L = \hat{F}_f(Q_L) \quad \hat{F}_R = \hat{F}_f(Q_R)$$

A **consistency condition** for the numerical flux is that if $Q_L = Q_R$, then

$$\hat{F}_L = \hat{F}_R$$



Central Difference Method

A central difference method for computing the flux is simply

$$\hat{F}_f = \frac{1}{2}(\hat{F}_L + \hat{F}_R)$$

The central-difference method works okay for elliptic components of the flux because there is no preferred direction for the propagation of information.

A simple central difference is often unstable, especially in the presence of strong gradients. One solution is to add some second-order and fourth-order dissipation (artificial viscosity) to the flux.

$$\hat{F}_f = \frac{1}{2}(\hat{F}_L + \hat{F}_R) + D^{(2)} + D^{(4)}$$

Methods for computing $D^{(2)}$ and $D^{(4)}$ vary, but generally use second and fourth-order differences with switches to handle variations in Q .

Use of Central Difference Method

For the Navier-Stokes equations, the viscous shear stress and heat flux terms in the viscous component \mathbf{D}^V of the non-convective component \mathbf{D} are elliptic and those flux components can be computed with the central difference method.

Similarly, the fluxes of the turbulence and chemistry equations can be computed using the central difference method.

The convective portion of the flux and the pressure term in the inviscid component \mathbf{D}^I of the Navier-Stokes equation have a hyperbolic character. This wave nature can be put to use to compute the flux using *upwind methods*.

Upwind Methods

We expressed the non-convective portion of the flux of the Navier-Stokes equation as

$$\mathbf{D}_{NS} = \mathbf{D}_{NS}^I + \mathbf{D}_{NS}^V$$

This results in the cell-face flux being expressed as

$$\hat{F}_f = [(\vec{v} - \vec{g}) Q_{NS} - \mathbf{D}_{NS}^I - \mathbf{D}_{NS}^V]$$

or

$$\hat{F}_f = \hat{F}_f^I + \hat{F}_f^V$$

We will now focus on computing the inviscid flux \hat{F}_f^I using upwind methods. The focus will be on the use of **Roe's Upwind Flux-Difference Splitting Method**.

Roe Upwind Flux-Difference Method

The Roe upwind flux-difference method computes the inviscid flux as:

$$\hat{F}_f^I = \frac{1}{2}(\hat{F}_L^I + \hat{F}_R^I) - \frac{1}{2}\Delta\hat{F}_f$$

where $\Delta\hat{F}$ is the flux difference computed as,

$$\Delta\hat{F} = \Delta\hat{F}^+ + \Delta\hat{F}^-$$

$$\Delta\hat{F}^+ = \sum_{m=1}^5 \lambda_m^{(+)} \vec{r}_m \delta w_m$$

$$\Delta\hat{F}^- = \sum_{m=1}^5 \lambda_m^{(-)} \vec{r}_m \delta w_m$$

Roe's method is the default flux method in WIND.

Roe Upwind Flux-Difference Method

The λ_m are the eigen values that represent the speed of the waves. The (+) indicate positive eigen values and the (-) indicates negative eigen values.

The \vec{r}_m are the right eigenvectors that represent the direction of propagation of the waves.

$$\delta w_1 = \delta \rho - \frac{\delta p}{c^2}$$

The δw_m are the Riemann invariants and represent the strength of the wave,

$$\delta w_2 = n_1 \delta w - n_3 \delta u$$

$$\delta w_3 = n_2 \delta u - n_1 \delta v$$

$$\delta w_4 = \frac{\delta p}{\rho c} + n_1 \delta u + n_2 \delta v + n_3 \delta w$$

$$\delta w_5 = \frac{\delta p}{\rho c} - n_1 \delta u + n_2 \delta v + n_3 \delta w$$

Roe Upwind Flux-Difference Method

The differentials are computed as

$$\delta \rho = \rho_R - \rho_L$$

$$\delta p = p_R - p_L$$

$$\delta u = u_R - u_L$$

$$\delta v = v_R - v_L$$

$$\delta w = w_R - w_L$$

Flow properties at the face are computed using Roe-averaging

$$\rho^2 = \rho_R \rho_L$$

$$u = \frac{\rho_L^{1/2} u_L + \rho_R^{1/2} u_R}{\rho_L^{1/2} + \rho_R^{1/2}}$$

Similar for computing v , w , and h_t .

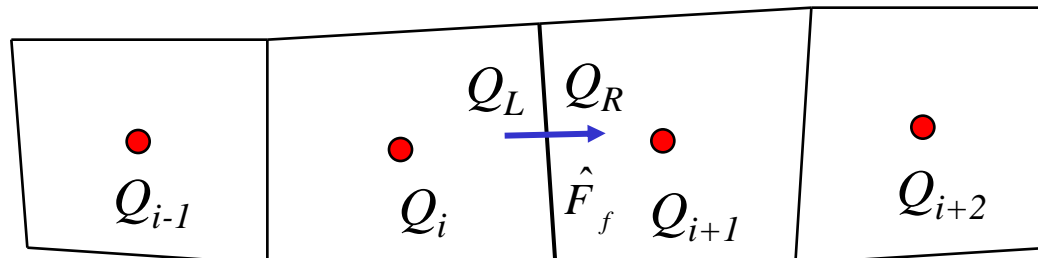
Higher-Order Projection

The choice of values of Q_L and Q_R have several option:

1) Use the values of the finite-volume cells to the “left” and “right” of the face. This is a zero-order evaluation and will result in a *spatially first-order flux*.

2) Use an extrapolation of neighboring finite-volume cells to form a first-order evaluation of Q at the face. This will result in a *spatially second-order flux*.

$$Q_L = Q_i + \frac{1}{2}(Q_i - Q_{i-1}) \quad Q_R = Q_{i+1} - \frac{1}{2}(Q_{i+2} - Q_{i+1})$$



Variation Limiting

The simple extrapolation formulas assume a smooth variation of Q ; however, discontinuities in Q are possible (i.e. shocks). Need some mechanism to sense such discontinuities and limit the variation of Q in these extrapolation formulas. Modify the extrapolations by introducing a limiter ϕ ,

$$Q_L = Q_i + \frac{1}{2}\phi (Q_i - Q_{i-1})$$

$$Q_R = Q_{i+1} - \frac{1}{2}\phi (Q_{i+2} - Q_{i+1})$$

This gets into the topic of TVD (Total Variational Diminishing) flux limiting methods, which we will not get into here. The essential role of the limiter is to make $\phi \rightarrow 0$ in the presence of large variations, which make the flux spatially first-order.

Examples of Limiters

The possible functions (and theory) for limiters is varied. A couple examples include:

Superbee: $\phi(r) = \max [0, \min (2r, 1), \min (r, 2)]$

Chakravarthy: $\phi(r) = \max [0, \min \{r, \beta\}]$

Where r is some ratio of the flow properties and indicates the amount of variation in the solution. An example is

$$r = \frac{\delta Q_{i+1}}{\delta Q_i}$$

The β is a compression parameter $1 \leq \beta \leq 2$, where a value toward 1 makes the limiter more dissipative.

Flux Vector Splitting

An alternative to flux-difference splitting is flux-vector splitting that considers that the inviscid flux can be linearly separated

$$\hat{F}_f^I = \hat{F}^+ + \hat{F}^-$$

Van Leer's flux-vector splitting has the general form of

$$\hat{F}^{\pm} = f_{\pm}^{mass} \begin{bmatrix} 1 \\ u + n_1 (-v_n \pm 2c)/\gamma \\ v + n_2 (-v_n \pm 2c)/\gamma \\ w + n_3 (-v_n \pm 2c)/\gamma \\ [(q^2 - v_n^2)/2 + [(\gamma - 1)v_n \pm 2c]^2 / [2(\gamma^2 - 1)]] \end{bmatrix}$$

$$f_{\pm}^{mass} = \pm \frac{\rho c (M_n \pm 1)^2}{4} \quad v_n = \vec{v} \cdot \hat{n} \quad M_n = \frac{v_n}{c}$$

Other RHS Methods

Other methods for the “right-hand-side (RHS)” that will not be discussed:

- Methods available for 3rd to 5th –order spatial accuracy.
- Roe’s method is modified to allow non-uniform grids.
- Roe’s method as used in the OVERFLOW code is available.
- Coakley method is available
- HLLE method is available (similar to Roe’s method)

Total variation diminishing (TVD) property.

- Total variation of a function $f(t, x)$ ($t > 0, x \in \mathbb{R}$), $\text{TV}(f)$ is defined by

$$\text{TV}(f) := \lim_{\epsilon \rightarrow 0} \sup \frac{1}{\epsilon} \int_{-\infty}^{\infty} |f(t, x) - f(t, x - \epsilon)| dx, \quad \left(\text{or } \int_{-\infty}^{\infty} |\partial_x f(t, x)| dx \right)$$

Definition: (TVD). If $\text{TV}(f)$ does not increase in time, $f(t, x)$ is called total variation diminishing or **TVD**.

$$\partial_t f + c \partial_x f = 0 \quad f(t, x) = f(x - ct)$$

- For $f(t, x)$ a solution to, we have

$$\text{TV}(f) := \lim_{\epsilon \rightarrow 0} \sup \frac{1}{\epsilon} \int_{-\infty}^{\infty} |f(x') - f(x' - \epsilon)| dx'$$

which is independent of t , hence $f(t, x)$ is TVD.

Definition: A numerical scheme with this property is called **TVD scheme**.

$$w_j^{n+1} = \sum_m c_m w_{j+m}^n$$

$$c_m \geq 0 \quad \text{for } \forall m \quad \text{and} \quad \sum_m c_m \leq 1.$$

Monotonicity preserving scheme with flux limiter function. (Flux limited schemes)

- Godunov's theorem does not allow the 2nd order linear one-step scheme.
- Conditions to be satisfied by the 1st order monotonicity preserving scheme are \longrightarrow $\left\{ \begin{array}{l} (1) \ c_m \geq 0, \text{ for } \forall m, \\ (2) \ \sum_m c_m = 1, \end{array} \right.$
- Considering that the number of grid points for the 1st order scheme are 2 points, resulting scheme is the 1st-order upwind. $\nu := \frac{c \tau}{h} \sum_m m c_m = -\frac{c \tau}{h}.$

$$w_j^{n+1} = \begin{cases} (1 - \nu)w_j^n + \nu w_{j-1}^n & \text{for } 0 \leq \nu \leq 1 \\ (1 + \nu)w_j^n - \nu w_{j+1}^n & \text{for } -1 \leq \nu \leq 0 \end{cases}$$

Lax-Wendroff scheme is understood as modifying the flux of 1st order upwind.

$$\partial_t f + \partial_x(cf) = 0 \qquad w_j^{n+1} = w_j^n - \frac{\tau}{h} \left(F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n \right).$$

$$1^{\text{st}} \text{ order upwind } (c > 0): \quad F_{j+\frac{1}{2}}^n = \tilde{f}_j^n \qquad \tilde{f}_j^n := c w_j^n$$

$$\text{Lax-Wendroff :} \qquad F_{j+\frac{1}{2}}^n = \tilde{f}_j^n + \frac{1}{2}(1 - \nu) (\tilde{f}_{j+1}^n - \tilde{f}_j^n)$$

Consider a non-linear scheme that modify the flux with a
limiter function

$$F_{j+\frac{1}{2}}^n = \tilde{f}_j^n + \frac{1}{2}(1 - \nu) \textcolor{red}{B}_{j+\frac{1}{2}} (\tilde{f}_{j+1}^n - \tilde{f}_j^n) \qquad \textcolor{red}{B}_{j+\frac{1}{2}}$$

(The value of ν differs at each cell boundary.) $0 \leq \nu \leq 1$

UNIT – V

TIME DEPENDENT METHODS

Explicit Methods

- The forward time / central space (FTCS) method.

$$u_i^{n+1} = u_i^n + \frac{\alpha(\Delta t)}{(\Delta x)^2} (u_{i+1}^n - 2u_i^n + u_{i-1}^n)$$

order $[(\Delta t), (\Delta x)^2]$

- Stability condition $\alpha\Delta t/(\Delta x)^2 \leq 1/2$

- The Richardson method (central differencing for both of time and space derivatives):

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = \alpha \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2}$$

order $[(\Delta t)^2, (\Delta x)^2]$

- Unconditionally unstable and, therefore, has no practical value

The DuFort-Frankel method

- Due to stability considerations, u_i^n in the diffusion term is replaced by the average value of u_i^{n+1} and u_i^{n-1} .

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = \alpha \frac{u_{i+1}^n - 2\frac{u_i^{n+1} + u_i^{n-1}}{2} + u_{i-1}^n}{(\Delta x)^2}$$

- the equation can be solved explicitly:

$$u_i^{n+1} = u_i^{n-1} + \frac{2\alpha(\Delta t)}{(\Delta x)^2} [u_{i+1}^n - u_i^{n+1} - u_i^{n-1} + u_{i-1}^n]$$

- unconditionally stable!

$$\left[1 + \frac{2\alpha(\Delta t)}{(\Delta x)^2}\right] u_i^{n+1} = \left[1 - 2\frac{\alpha(\Delta t)}{(\Delta x)^2}\right] u_i^{n-1} + \frac{2\alpha(\Delta t)}{(\Delta x)^2} [u_{i+1}^n + u_{i-1}^n]$$

order $[(\Delta t)^2, (\Delta x)^2, (\Delta t/\Delta x)^2]$

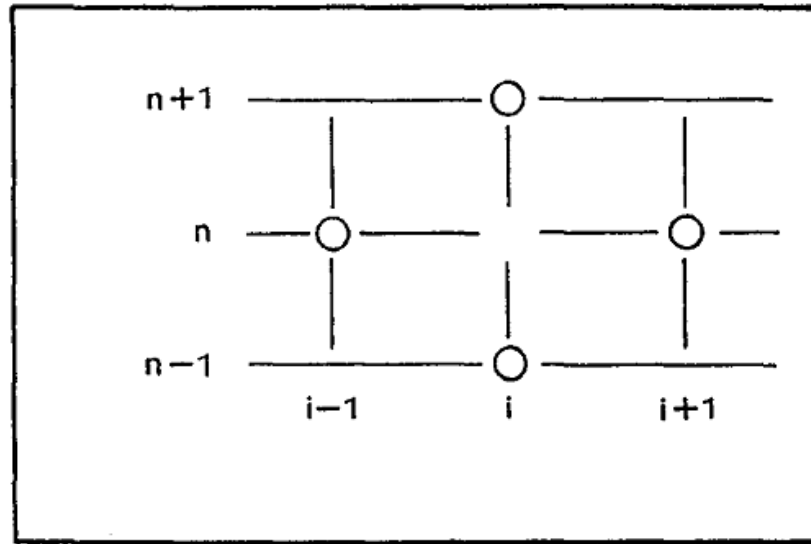


Figure 3-4. Grid points for the DuFort-Frankel method.

$$\left[1 + \frac{2\alpha(\Delta t)}{(\Delta x)^2}\right] u_i^{n+1} = \left[1 - 2\frac{\alpha(\Delta t)}{(\Delta x)^2}\right] u_i^{n-1} + \frac{2\alpha(\Delta t)}{(\Delta x)^2} [u_{i+1}^n + u_{i-1}^n]$$

- Either two sets of data must be specified, or, from a practical point of view, a one-step method can be used as a starter. Of course, for the one-step (in Δt) starter solution, only one set of initial data, say at $n - 1$, is required to generate the solution at n .
- With the values of u , at $n - 1$ and n specified, the DuFort-Frankel method can be used.
- The accuracy of the solution provided by the Dufort-Frankel method is affected by the accuracy of the starter solution.
- Since the solution at the unknown station requires data from two previous stations, computer storage requirements will increase.

Implicit Methods

- The Laasonen method:

$$\frac{u_i^{n+1} - u_i^n}{(\Delta t)} = \alpha \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{(\Delta x)^2}$$

- Implicit methods offer great advantage on the stability of the finite difference equations, since most are unconditionally stable.
- Therefore, a larger step size in time is permitted;
- however, the selection of a larger time step is limited due to accuracy consideration because an increase in time step will increase the truncation error of the finite difference equation.

The Crank-Nicolson method:

- If the diffusion term in Laasonen method is replaced by the average of the central differences at time levels n and $n + 1$:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \left(\frac{1}{2} \right) \left[\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{(\Delta x)^2} + \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} \right]$$

$$\frac{\partial u}{\partial t} = \frac{u_i^{n+1} - u_i^n}{2 \left(\frac{\Delta t}{2} \right)}$$

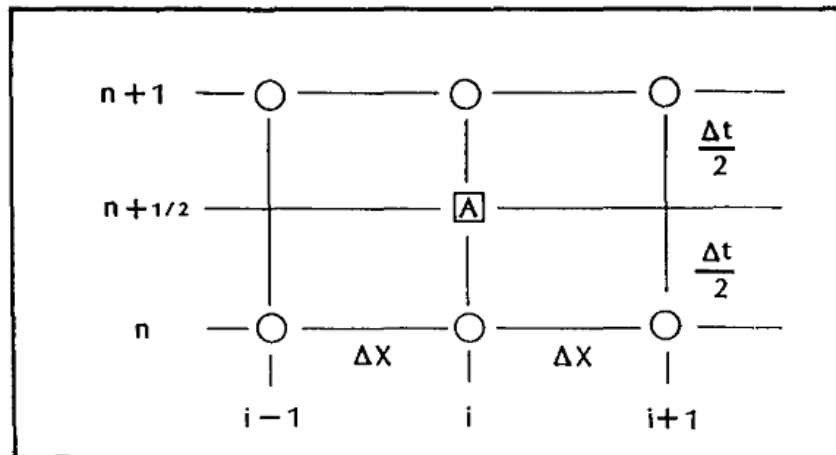


Figure 3-5. Grid points for the Crank-Nicolson implicit method.

$$\frac{u_i^{n+\frac{1}{2}} - u_i^n}{\frac{\Delta t}{2}} = \alpha \frac{u_{i-1}^n - 2u_i^n + u_{i+1}^n}{(\Delta x)^2}$$



$$\frac{u_i^{n+1} - u_i^{n+\frac{1}{2}}}{\frac{\Delta t}{2}} = \alpha \frac{u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}}{(\Delta x)^2}$$



$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{1}{2}\alpha \left[\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{(\Delta x)^2} + \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} \right]$$

- unconditionally stable

order $[(\Delta t)^2, (\Delta x)^2]$

The Beta Formulation

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \left[\beta \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{(\Delta x)^2} + (1 - \beta) \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} \right]$$

- unconditionally stable: $1/2 \leq \beta \leq 1$
- Crank-Nicolson implicit method: $\beta = 1/2$
- conditionally stable: $0 \leq \beta < 1/2$
- FTCS explicit method: $\beta = 0$

- A fluid bounded by two parallel plates extended to infinity such that no end effects are encountered. The planar walls and the fluid are initially at rest. Now, the lower wall is suddenly accelerated in the x-direction

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2}$$

$$\begin{aligned} \text{(a) Initial condition} \quad & t = 0, \quad u = U_0 \quad \text{for} \quad y = 0 \\ & u = 0 \quad \text{for} \quad 0 < y \leq h \\ \text{(b) Boundary conditions} \quad & t \geq 0, \quad u = U_0 \quad \text{for} \quad y = 0 \\ & u = 0 \quad \text{for} \quad y = h \end{aligned}$$

Parabolic Equations in Two-Space Dimensions

$$\frac{\partial u}{\partial t} = \alpha \left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right]$$

- FTCS

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} = \alpha \left[\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{(\Delta x)^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{(\Delta y)^2} \right]$$

- stability

$$\left[\frac{\alpha \Delta t}{(\Delta x)^2} + \frac{\alpha \Delta t}{(\Delta y)^2} \right] \leq \frac{1}{2}$$

$$d_x = \frac{\alpha \Delta t}{(\Delta x)^2}$$

$$d_y = \frac{\alpha \Delta t}{(\Delta y)^2}$$

$$\Rightarrow \Delta x = \Delta y \Rightarrow d_x = d_y = d \Rightarrow d \leq 0.25$$

Implicit formulation

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{(\Delta t)} = \alpha \left[\frac{u_{i+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i-1,j}^{n+1}}{(\Delta x)^2} + \frac{u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}}{(\Delta y)^2} \right]$$

$$d_x u_{i+1,j}^{n+1} + d_x u_{i-1,j}^{n+1} - (2d_x + 2d_y + 1)u_{i,j}^{n+1} + d_y u_{i,j-1}^{n+1} + d_y u_{i,j+1}^{n+1} = -u_{i,j}^n$$

$$a_{i,j} u_{i+1,j}^{n+1} + b_{i,j} u_{i-1,j}^{n+1} + c_{i,j} u_{i,j}^{n+1} + d_{i,j} u_{i,j-1}^{n+1} + e_{i,j} u_{i,j+1}^{n+1} = f_{i,j}^n$$

$$\begin{bmatrix}
c_{2,2} & e_{2,2} & 0 & a_{2,2} & 0 & 0 & 0 & 0 & 0 \\
d_{2,3} & c_{2,3} & e_{2,3} & 0 & a_{2,3} & 0 & 0 & 0 & 0 \\
0 & d_{2,4} & c_{2,4} & 0 & 0 & a_{2,4} & 0 & 0 & 0 \\
b_{3,2} & 0 & 0 & c_{3,2} & e_{3,2} & 0 & a_{3,2} & 0 & 0 \\
0 & b_{3,3} & 0 & d_{3,3} & c_{3,3} & e_{3,3} & 0 & a_{3,3} & 0 \\
0 & 0 & b_{3,4} & 0 & d_{3,4} & c_{3,4} & 0 & 0 & a_{3,4} \\
0 & 0 & 0 & b_{4,2} & 0 & 0 & c_{4,2} & e_{4,2} & 0 \\
0 & 0 & 0 & 0 & b_{4,3} & 0 & d_{4,3} & c_{4,3} & e_{4,3} \\
0 & 0 & 0 & 0 & 0 & b_{4,4} & 0 & d_{4,4} & c_{4,4}
\end{bmatrix}
\begin{bmatrix}
u_{2,2} \\
u_{2,3} \\
u_{2,4} \\
u_{3,2} \\
u_{3,3} \\
u_{3,4} \\
u_{4,2} \\
u_{4,3} \\
u_{4,4}
\end{bmatrix}
=
\begin{bmatrix}
f_{2,2} - b_{2,2}u_{1,2} - d_{2,2}u_{2,1} \\
f_{2,3} - b_{2,3}u_{1,3} \\
f_{2,4} - b_{2,4}u_{1,4} - e_{2,4}u_{2,5} \\
f_{3,2} - d_{3,2}u_{3,1} \\
f_{3,3} \\
f_{3,4} - e_{3,4}u_{3,5} \\
f_{4,2} - a_{4,2}u_{5,2} - d_{4,2}u_{4,1} \\
f_{4,3} - a_{4,3}u_{5,3} \\
f_{4,4} - a_{4,4}u_{5,4} - e_{4,4}u_{4,5}
\end{bmatrix}$$

Alternating Direction Implicit method or ADI

- x sweep: implicit in the x-direction and explicit in the y-direction:

$$\frac{u_{i,j}^{n+\frac{1}{2}} - u_{i,j}^n}{\left(\frac{\Delta t}{2}\right)} = \alpha \left[\frac{u_{i+1,j}^{n+\frac{1}{2}} - 2u_{i,j}^{n+\frac{1}{2}} + u_{i-1,j}^{n+\frac{1}{2}}}{(\Delta x)^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{(\Delta y)^2} \right]$$

- y sweep: implicit in the y-direction and explicit in the x-direction:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n+\frac{1}{2}}}{\left(\frac{\Delta t}{2}\right)} = \alpha \left[\frac{u_{i+1,j}^{n+\frac{1}{2}} - 2u_{i,j}^{n+\frac{1}{2}} + u_{i-1,j}^{n+\frac{1}{2}}}{(\Delta x)^2} + \frac{u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}}{(\Delta y)^2} \right]$$

$$\begin{aligned} -d_1 u_{i-1,j}^{n+\frac{1}{2}} + (1 + 2d_1) u_{i,j}^{n+\frac{1}{2}} - d_1 u_{i+1,j}^{n+\frac{1}{2}} &= \\ &= d_2 u_{i,j+1}^n + (1 - 2d_2) u_{i,j}^n + d_2 u_{i,j-1}^n \end{aligned}$$

$$\begin{aligned} -d_2 u_{i,j-1}^{n+1} + (1 + 2d_2) u_{i,j}^{n+1} - d_2 u_{i,j+1}^{n+1} &= \\ &= d_1 u_{i+1,j}^{n+\frac{1}{2}} + (1 - 2d_1) u_{i,j}^{n+\frac{1}{2}} + d_1 u_{i-1,j}^{n+\frac{1}{2}} \end{aligned}$$

$$d_1 = \frac{1}{2} d_x = \frac{1}{2} \frac{\alpha \Delta t}{(\Delta x)^2}$$

$$d_2 = \frac{1}{2} d_y = \frac{1}{2} \frac{\alpha \Delta t}{(\Delta y)^2}$$

Extension to Three-Space Dimensions

$$\frac{\partial u}{\partial t} = \alpha \left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right]$$

$$\frac{u_{i,j,k}^{n+\frac{1}{3}} - u_{i,j,k}^n}{\frac{\Delta t}{3}} = \alpha \left[\frac{\delta_x^2 u_{i,j,k}^{n+\frac{1}{3}}}{(\Delta x)^2} + \frac{\delta_y^2 u_{i,j,k}^n}{(\Delta y)^2} + \frac{\delta_z^2 u_{i,j,k}^n}{(\Delta z)^2} \right]$$

$$\frac{u_{i,j,k}^{n+\frac{2}{3}} - u_{i,j,k}^{n+\frac{1}{3}}}{\frac{\Delta t}{3}} = \alpha \left[\frac{\delta_x^2 u_{i,j,k}^{n+\frac{1}{3}}}{(\Delta x)^2} + \frac{\delta_y^2 u_{i,j,k}^{n+\frac{2}{3}}}{(\Delta y)^2} + \frac{\delta_z^2 u_{i,j,k}^{n+\frac{1}{3}}}{(\Delta z)^2} \right]$$

$$\frac{u_{i,j,k}^{n+1} - u_{i,j,k}^{n+\frac{2}{3}}}{\frac{\Delta t}{3}} = \alpha \left[\frac{\delta_x^2 u_{i,j,k}^{n+\frac{2}{3}}}{(\Delta x)^2} + \frac{\delta_y^2 u_{i,j,k}^{n+\frac{2}{3}}}{(\Delta y)^2} + \frac{\delta_z^2 u_{i,j,k}^{n+1}}{(\Delta z)^2} \right]$$

$$[(\Delta t), (\Delta x)^2, (\Delta y)^2, (\Delta z)^2] \quad (d_x + d_y + d_z) \leq (3/2).$$

Consistency Analysis of the Finite Difference Equations

- By previous definition, an FDE approximation of a PDE is consistent if the FDE reduces to the original PDE as the step sizes approach zero.

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2}$$

- FTCS explicit method.

$$u_i^{n+1} = u_i^n + \frac{\partial u}{\partial t}(\Delta t) + \frac{\partial^2 u}{\partial t^2} \frac{(\Delta t)^2}{2!} + O(\Delta t)^3$$

$$u_{i+1}^n = u_i^n + \frac{\partial u}{\partial x}(\Delta x) + \frac{\partial^2 u}{\partial x^2} \frac{(\Delta x)^2}{2!} + \frac{\partial^3 u}{\partial x^3} \frac{(\Delta x)^3}{3!} + O(\Delta x)^4$$

$$u_{i-1}^n = u_i^n - \frac{\partial u}{\partial x}(\Delta x) + \frac{\partial^2 u}{\partial x^2} \frac{(\Delta x)^2}{2!} - \frac{\partial^3 u}{\partial x^3} \frac{(\Delta x)^3}{3!} + O(\Delta x)^4$$

$$\left[\frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial t^2} \frac{\Delta t}{2} + O(\Delta t)^2 \right] = \alpha \left[\frac{\partial^2 u}{\partial x^2} + O(\Delta x)^2 \right]$$

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} - \left[\frac{\Delta t}{2} \right] \frac{\partial^2 u}{\partial t^2} + O \left[(\Delta t)^2, (\Delta x)^2 \right]$$

- Consistency requires that as the step sizes Δx and Δt approach zero, the FDE must reduce to the original PDE.

- DuFort-Frankel method:

$$(1 + 2d)u_i^{n+1} = (1 - 2d)u_i^{n-1} + 2d(u_{i+1}^n + u_{i-1}^n) \quad d = \alpha \frac{\Delta t}{(\Delta x)^2}$$

$$\begin{aligned} (1 + 2d) \left[u_i^n + \frac{\partial u}{\partial t} \Delta t + \frac{\partial^2 u}{\partial t^2} \frac{(\Delta t)^2}{2} + \frac{\partial^3 u}{\partial t^3} \frac{(\Delta t)^3}{3!} + O(\Delta t)^4 \right] = \\ (1 - 2d) \left[u_i^n - \frac{\partial u}{\partial t} \Delta t + \frac{\partial^2 u}{\partial t^2} \frac{(\Delta t)^2}{2} - \frac{\partial^3 u}{\partial t^3} \frac{(\Delta t)^3}{3!} + O(\Delta t)^4 \right] \\ + 2d \left\{ \left[u_i^n + \frac{\partial u}{\partial x} \Delta x + \frac{\partial^2 u}{\partial x^2} \frac{(\Delta x)^2}{2} + \frac{\partial^3 u}{\partial x^3} \frac{(\Delta x)^3}{3!} + O(\Delta x)^4 \right] \right. \\ \left. + \left[u_i^n - \frac{\partial u}{\partial x} \Delta x + \frac{\partial^2 u}{\partial x^2} \frac{(\Delta x)^2}{2} - \frac{\partial^3 u}{\partial x^3} \frac{(\Delta x)^3}{3!} + O(\Delta x)^4 \right] \right\} \end{aligned}$$

Linearization

- Method 1. Lagging: $u(\partial u / \partial x)$

- Method II. Iterative: $u_{i,j} \frac{u_{i+1,j} - u_{i,j}}{\Delta x}$

- Method III. Newton's iterative linearization: $u_{i+1,j}^k \frac{u_{i+1,j}^{k+1} - u_{i,j}}{\Delta x}$

$$\delta A = A^{k+1} - A^k$$

$$\delta B = B^{k+1} - B^k$$

$$A^{k+1} = A^k + \delta A$$

$$B^{k+1} = B^k + \delta B$$

$$\begin{aligned} A^{k+1}B^{k+1} &= (A^k + \delta A)(B^k + \delta B) \\ &= A^k B^k + B^k \delta A + A^k \delta B + (\delta A)(\delta B) \end{aligned}$$

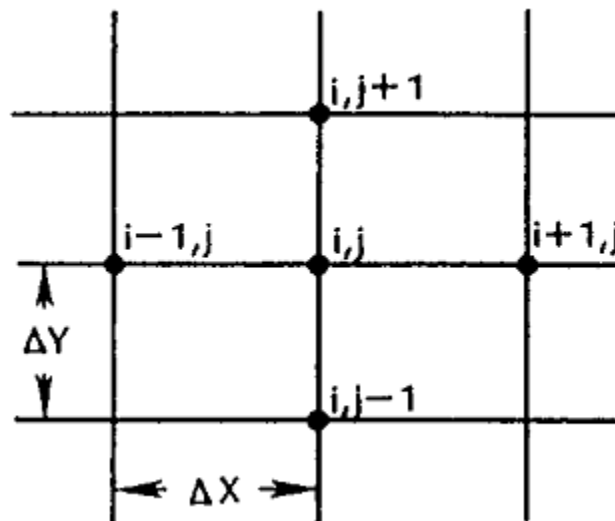
$$\begin{aligned} A^{k+1}B^{k+1} &= A^k B^k + B^k(A^{k+1} - A^k) + A^k(B^{k+1} - B^k) \\ &= A^k B^k + A^{k+1}B^k - A^k B^k + A^k B^{k+1} - A^k B^k \\ &= A^k B^{k+1} + A^{k+1}B^k - A^k B^k \end{aligned}$$

Elliptic Equations

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

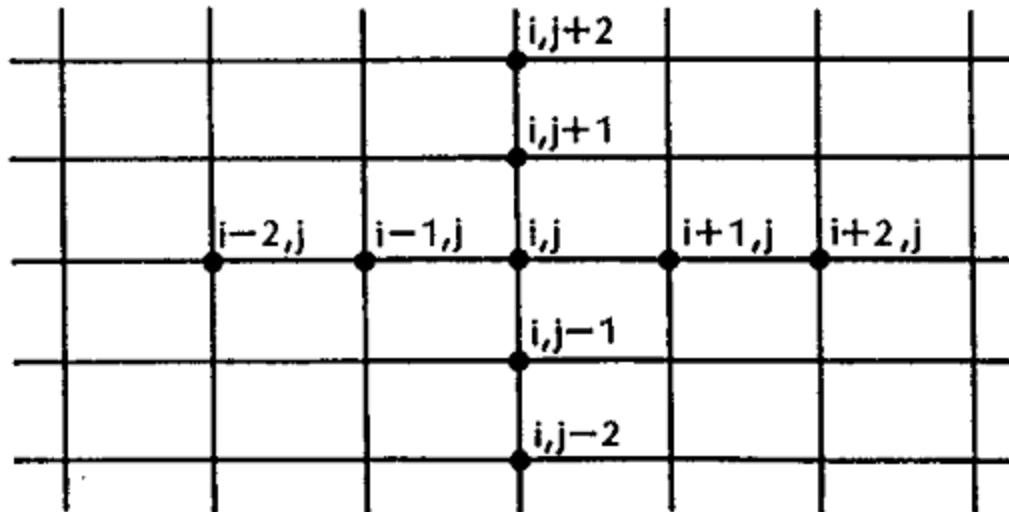
- Five-point formula" is the most commonly used. central differencing (second-order accurate):

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} = 0$$



- Nine-point formula (fourth-order accurate):

$$\frac{-u_{i-2,j} + 16u_{i-1,j} - 30u_{i,j} + 16u_{i+1,j} - u_{i+2,j}}{12(\Delta x)^2} + \frac{-u_{i,j-2} + 16u_{i,j-1} - 30u_{i,j} + 16u_{i,j+1} - u_{i,j+2}}{12(\Delta y)^2} = 0$$

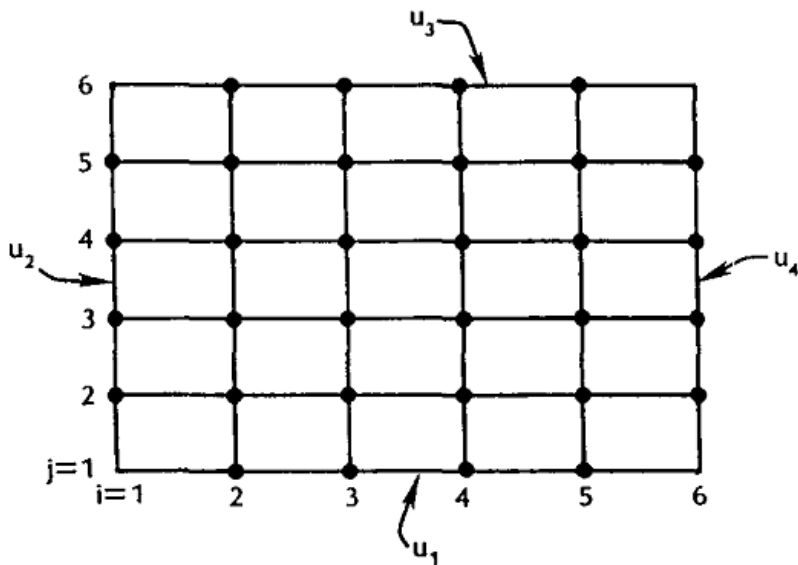


- Implementation of the boundary conditions?

Five point

$$u_{i+1,j} - 2u_{i,j} + u_{i-1,j} + \left(\frac{\Delta x}{\Delta y}\right)^2 (u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) = 0$$

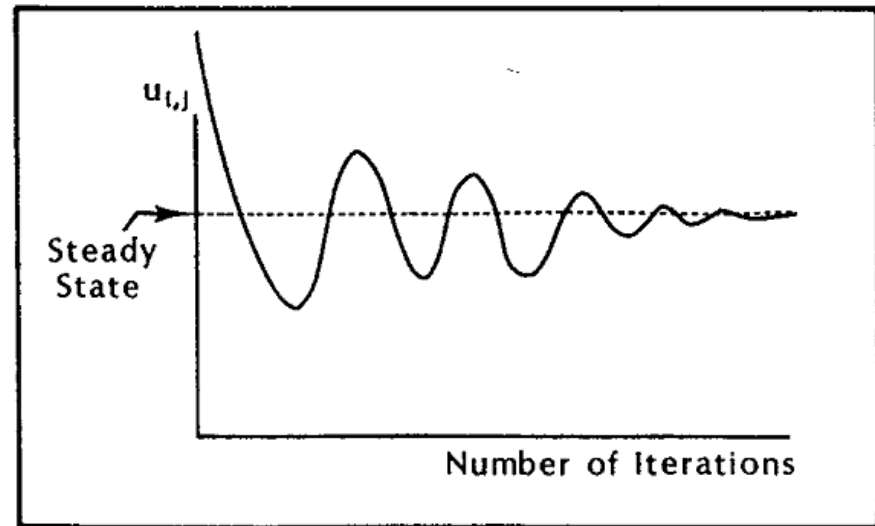
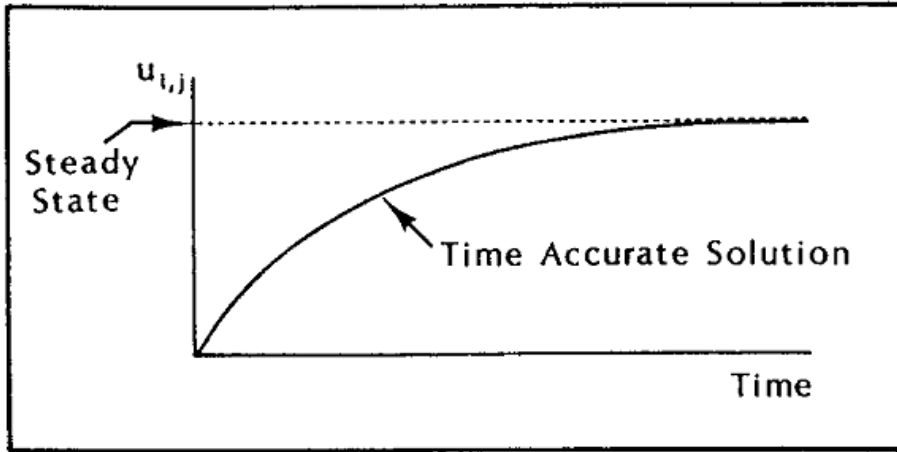
$$u_{i+1,j} + u_{i-1,j} + \beta^2 u_{i,j+1} + \beta^2 u_{i,j-1} - 2(1 + \beta^2)u_{i,j} = 0$$



$$\begin{array}{llll} x = 0 & u = u_2 & , & y = 0 & u = u_1 \\ x = L & u = u_4 & , & y = H & u = u_3 \end{array}$$

$$\begin{aligned}
u_{3,2} + u_{1,2} + \beta^2 u_{2,3} + \beta^2 u_{2,1} - 2(1 + \beta^2) u_{2,2} &= 0 \\
u_{4,2} + u_{2,2} + \beta^2 u_{3,3} + \beta^2 u_{3,1} - 2(1 + \beta^2) u_{3,2} &= 0 \\
u_{5,2} + u_{3,2} + \beta^2 u_{4,3} + \beta^2 u_{4,1} - 2(1 + \beta^2) u_{4,2} &= 0 \\
u_{6,2} + u_{4,2} + \beta^2 u_{5,3} + \beta^2 u_{5,1} - 2(1 + \beta^2) u_{5,2} &= 0 \\
u_{3,3} + u_{1,3} + \beta^2 u_{2,4} + \beta^2 u_{2,2} - 2(1 + \beta^2) u_{2,3} &= 0 \\
u_{4,3} + u_{2,3} + \beta^2 u_{3,4} + \beta^2 u_{3,2} - 2(1 + \beta^2) u_{3,3} &= 0 \\
u_{5,3} + u_{3,3} + \beta^2 u_{4,4} + \beta^2 u_{4,2} - 2(1 + \beta^2) u_{4,3} &= 0 \\
u_{6,3} + u_{4,3} + \beta^2 u_{5,4} + \beta^2 u_{5,2} - 2(1 + \beta^2) u_{5,3} &= 0 \\
u_{3,4} + u_{1,4} + \beta^2 u_{2,5} + \beta^2 u_{2,3} - 2(1 + \beta^2) u_{2,4} &= 0 \\
u_{4,4} + u_{2,4} + \beta^2 u_{3,5} + \beta^2 u_{3,3} - 2(1 + \beta^2) u_{3,4} &= 0 \\
u_{5,4} + u_{3,4} + \beta^2 u_{4,5} + \beta^2 u_{4,3} - 2(1 + \beta^2) u_{4,4} &= 0 \\
u_{6,4} + u_{4,4} + \beta^2 u_{5,5} + \beta^2 u_{5,3} - 2(1 + \beta^2) u_{5,4} &= 0 \\
u_{3,5} + u_{1,5} + \beta^2 u_{2,6} + \beta^2 u_{2,4} - 2(1 + \beta^2) u_{2,5} &= 0 \\
u_{4,5} + u_{2,5} + \beta^2 u_{3,6} + \beta^2 u_{3,4} - 2(1 + \beta^2) u_{3,5} &= 0 \\
u_{5,5} + u_{3,5} + \beta^2 u_{4,6} + \beta^2 u_{4,4} - 2(1 + \beta^2) u_{4,5} &= 0 \\
u_{6,5} + u_{4,5} + \beta^2 u_{5,6} + \beta^2 u_{5,4} - 2(1 + \beta^2) u_{5,5} &= 0
\end{aligned}$$

$$\begin{bmatrix}
\alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & \alpha & 0 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\beta^2 & 0 & 0 & 0 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 0 & 0 & 0 & \beta^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 0 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 0 & 0 & 0 & \beta^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 0 & \alpha & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha
\end{bmatrix}
\begin{bmatrix}
u_{2,2} \\
u_{3,2} \\
u_{4,2} \\
u_{5,2} \\
u_{2,3} \\
u_{3,3} \\
u_{4,3} \\
u_{5,3} \\
u_{2,4} \\
u_{3,4} \\
u_{4,4} \\
u_{5,4} \\
u_{2,5} \\
u_{3,5} \\
u_{4,5} \\
u_{5,5}
\end{bmatrix}
=
\begin{bmatrix}
-u_{1,2} - \beta^2 u_{2,1} \\
-\beta^2 u_{3,1} \\
-\beta^2 u_{4,1} \\
-u_{6,2} - \beta^2 u_{5,1} \\
-u_{1,3} \\
0 \\
0 \\
-u_{6,3} \\
-u_{1,4} \\
0 \\
0 \\
-u_{6,4} \\
-u_{1,5} - \beta^2 u_{3,6} \\
-\beta^2 u_{3,6} \\
-\beta^2 u_{4,6} \\
-u_{6,5} - \beta^2 u_{5,6}
\end{bmatrix}$$



- The Point Gauss-Seidel Iteration Method

$$u_{i,j} = \frac{1}{2(1 + \beta^2)} [u_{i+1,j} + u_{i-1,j} + \beta^2(u_{i,j+1} + u_{i,j-1})]$$

- The Line Gauss-Seidel Iteration Method :

$$u_{i-1,j}^{k+1} - 2(1 + \beta^2)u_{i,j}^{k+1} + u_{i+1,j}^{k+1} = -\beta^2 u_{i,j+1}^k - \beta^2 u_{i,j-1}^{k+1}$$

Point Successive Over-Relaxation Method (PSOR)

$$u_{i,j}^{k+1} = \frac{1}{2(1 + \beta^2)} \left[u_{i+1,j}^k + u_{i-1,j}^{k+1} + \beta^2(u_{i,j+1}^k + u_{i,j-1}^{k+1}) \right]$$

Adding $u_{i,j}^k - u_{i,j}^k$

$$u_{i,j}^{k+1} = u_{i,j}^k + \frac{1}{2(1 + \beta^2)} \left[u_{i+1,j}^k + u_{i-1,j}^{k+1} + \beta^2(u_{i,j+1}^k + u_{i,j-1}^{k+1}) - 2(1 + \beta^2)u_{i,j}^k \right]$$

$$u_{i,j}^{k+1} = u_{i,j}^k + \frac{\omega}{2(1 + \beta^2)} \left[u_{i+1,j}^k + u_{i-1,j}^{k+1} + \beta^2(u_{i,j+1}^k + u_{i,j-1}^{k+1}) - 2(1 + \beta^2)u_{i,j}^k \right]$$

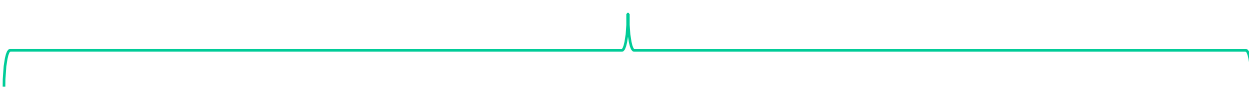
$$u_{i,j}^{k+1} = (1 - \omega)u_{i,j}^k + \frac{\omega}{2(1 + \beta^2)} \left[u_{i+1,j}^k + u_{i-1,j}^{k+1} + \beta^2(u_{i,j+1}^k + u_{i,j-1}^{k+1}) \right]$$

Line Successive Over-Relaxation Method (LSOR)

$$u_{i-1,j}^{k+1} - 2(1 + \beta^2)u_{i,j}^{k+1} + u_{i+1,j}^{k+1} = -\beta^2 u_{i,j+1}^k - \beta^2 u_{i,j-1}^{k+1}$$

$$\begin{aligned} \omega u_{i-1,j}^{k+1} - 2(1 + \beta^2)\omega u_{i,j}^{k+1} + \omega u_{i+1,j}^{k+1} = \\ -(1 - \omega) \left[2(1 + \beta^2) \right] u_{i,j}^k - \omega \beta^2 (u_{i,j+1}^k + u_{i,j-1}^{k+1}) \end{aligned}$$

The Alternating Direction Implicit (ADI) Method

$$u_{i,j}^{k+1} = \frac{1}{2(1+\beta^2)} \left[u_{i+1,j}^k + u_{i-1,j}^k + \beta^2(u_{i,j+1}^k + u_{i,j-1}^k) \right]$$


$$u_{i-1,j}^{k+\frac{1}{2}} - 2(1+\beta^2)u_{i,j}^{k+\frac{1}{2}} + u_{i+1,j}^{k+\frac{1}{2}} = -\beta^2(u_{i,j+1}^k + u_{i,j-1}^k)$$

$$\beta^2 u_{i,j-1}^{k+1} - 2(1+\beta^2)u_{i,j}^{k+1} + \beta^2 u_{i,j+1}^{k+1} = -u_{i+1,j}^{k+\frac{1}{2}} - u_{i-1,j}^{k+\frac{1}{2}}$$

The Point Gauss-Seidel Iteration

Method

- (sufficient condition) for the convergence of the method:

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|$$

- and, at least for one row $|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|$

Leap Frog (space and time)

- We use centered differencing for both space and time.

$$\frac{u_m^{n+1} - u_m^{n-1}}{2 \, dt} = c \left(\frac{u_{m+1}^n - u_{m-1}^n}{2 \, dx} \right)$$

- We know that the leap frog time stepping method is only stable for operators with eigenvalues in the range: $dt \lambda \in i[-1, 1]$
- However, we also know that the centered difference derivative matrix is a skew symmetric matrix with eigenvalues: $\frac{ic}{dx} \sin \left(\frac{2\pi m}{M} \right)$
- So we are left with a condition: $\left| \frac{c \, dt}{dx} \right| \leq 1$ i.e. $dt \leq \frac{dx}{c}$

cont

- We can perform a full truncation analysis (in space and time):

$$\begin{aligned} T_m^n &= \frac{u(x_m, t_{n+1}) - u(x_m, t_{n-1}))}{2} - c dt \left(\frac{u(x_{m+1}, t_n) - u(x_{m-1}, t_n)}{2 dx} \right) \\ &= O(dt^3) - \frac{c dt}{dx} O(dx^3) \end{aligned}$$

- We know that $dt \leq dx$ so

$$T_m^n = O(dt^3) - c O(dx^3)$$

Lax-Friedrichs Method

- We immediately conclude that the following method is not stable:

$$\frac{u_m^{n+1} - u_m^n}{dt} = c \delta_0 u_m^n$$

- Because the Euler-Forward time integrator only admits one stable point (the origin) on the imaginary axis, but the central differencing matrix has all eigenvalues on the imaginary axis.
- However, we can stabilize this formula by replacing the second term in the time-stepping formula:

$$\frac{u_m^{n+1} - 0.5 \left(u_{m+1}^n + u_{m-1}^n \right)}{dt} = c \delta_0 u_m^n$$

- This formula does not quite fit into our constructive process (method of lines approach).

$$\frac{u_m^{n+1} - 0.5(u_{m+1}^n + u_{m-1}^n)}{dt} = c \delta_0 u_m^n$$

- We have admitted spatial averaging into the discretization of the time derivative.
- We can rearrange this:

$$\begin{aligned} u_m^{n+1} &= \frac{1}{2} (u_{m+1}^n + u_{m-1}^n) + c \frac{dt}{2 dx} (u_{m+1}^n - u_{m-1}^n) \\ &= \frac{1}{2} \left(1 + c \frac{dt}{dx} \right) u_{m+1}^n + \frac{1}{2} \left(1 - c \frac{dt}{dx} \right) u_{m-1}^n \end{aligned}$$

$$u_m^{n+1} = \frac{1}{2} \left(1 + c \frac{dt}{dx} \right) u_{m+1}^n + \frac{1}{2} \left(1 - c \frac{dt}{dx} \right) u_{m-1}^n$$

- We can immediately determine that this is a stable method as long as $c \cdot dt/dx \leq 1$
- Given, this condition we observe that the time updated solution is always bounded between the values of the left and right neighbor at the previous time because this is an interpolation formula.

- A formal stability analysis would involve:

$$\begin{aligned}
 u_m^{n+1} &= \frac{1}{2} \left(1 + c \frac{dt}{dx} \right) u_{m+1}^n + \frac{1}{2} \left(1 - c \frac{dt}{dx} \right) u_{m-1}^n \\
 \Rightarrow \vec{u}^{n+1} &= \frac{1}{2} \left(1 + c \frac{dt}{dx} \right) \mathbf{R} \vec{u}^n + \frac{1}{2} \left(1 - c \frac{dt}{dx} \right) \mathbf{L} \vec{u}^n \\
 \Rightarrow \tilde{u}_m^{n+1} &= \frac{1}{2} \left(1 + c \frac{dt}{dx} \right) e^{i\theta_m} \tilde{u}_m^n + \frac{1}{2} \left(1 - c \frac{dt}{dx} \right) e^{-i\theta_m} \tilde{u}_m^n \\
 &= \left\{ \cos(\theta_m) + c \frac{dt}{dx} i \sin(\theta_m) \right\} \tilde{u}_m^n
 \end{aligned}$$

- Which gives stability for each mode if:

$$\left| \cos(\theta_m) + c \frac{dt}{dx} i \sin(\theta_m) \right| \leq 1$$

- Thus considering all the possible modes
- We note that the middle mode requires:

$$\theta_m = \frac{2\pi m}{M}$$

$$\left| \cos\left(\frac{\pi}{2}\right) + c \frac{dt}{dx} i \sin\left(\frac{\pi}{2}\right) \right| \leq 1$$

$$\Rightarrow \left| c \frac{dt}{dx} \right| \leq 1$$

- This condition gives a sufficient condition for all modes to be bounded.
- By the invertibility and boundedness of the Fourier transform we conclude that the original equation is stable.

- We can recast the Lax-Friedrichs method again

$$\begin{aligned}
 u_m^{n+1} &= \frac{1}{2} \left(1 + c \frac{dt}{dx} \right) u_{m+1}^n + \frac{1}{2} \left(1 - c \frac{dt}{dx} \right) u_{m-1}^n \\
 &\approx \\
 u_m^{n+1} &= (1 + c dt \delta_0) u_m^n + \frac{1}{2} dx^2 \delta^2 u_m^n
 \end{aligned}$$

- This method consists of Euler Forward, central differencing for the space derivative and an extra dissipative term (i.e. a grid dependent advection diffusion equation)

$$\frac{\partial u}{\partial t} = c \frac{\partial u}{\partial x} + \frac{dx^2}{2} \frac{\partial^2 u}{\partial x^2}$$

CFL Number

- The ratio $\lambda = \frac{dt}{dx}$ appears repeatedly, in particular in the estimates for the maximum possible time step.
- We refer to this quantity as the CFL (Courant-Friedrichs-Lewy) number.
- Bounds of the form: $\lambda = \frac{dt}{dx} \leq C$ which result from a stability analysis are frequently referred to as CFL conditions.

Lax-Wendroff Method

- We are fairly free to choose the parameter in the stabilizing term:

$$u_m^{n+1} = (1 + c d t \delta_0) u_m^n + \frac{1}{2} c d t^2 \delta^2 u_m^n$$

- The *artificial viscosity* term acts to shift the eigenvalues of the spatial operator into the left half-plane.
- Recall – the Euler-Forward stability region is the unit circle centered at -1 in the complex plane. So pushing the eigenvalues off the imaginary axis allows us to choose a dt small enough to push the eigen values of the discrete spatial operator into the stability region...

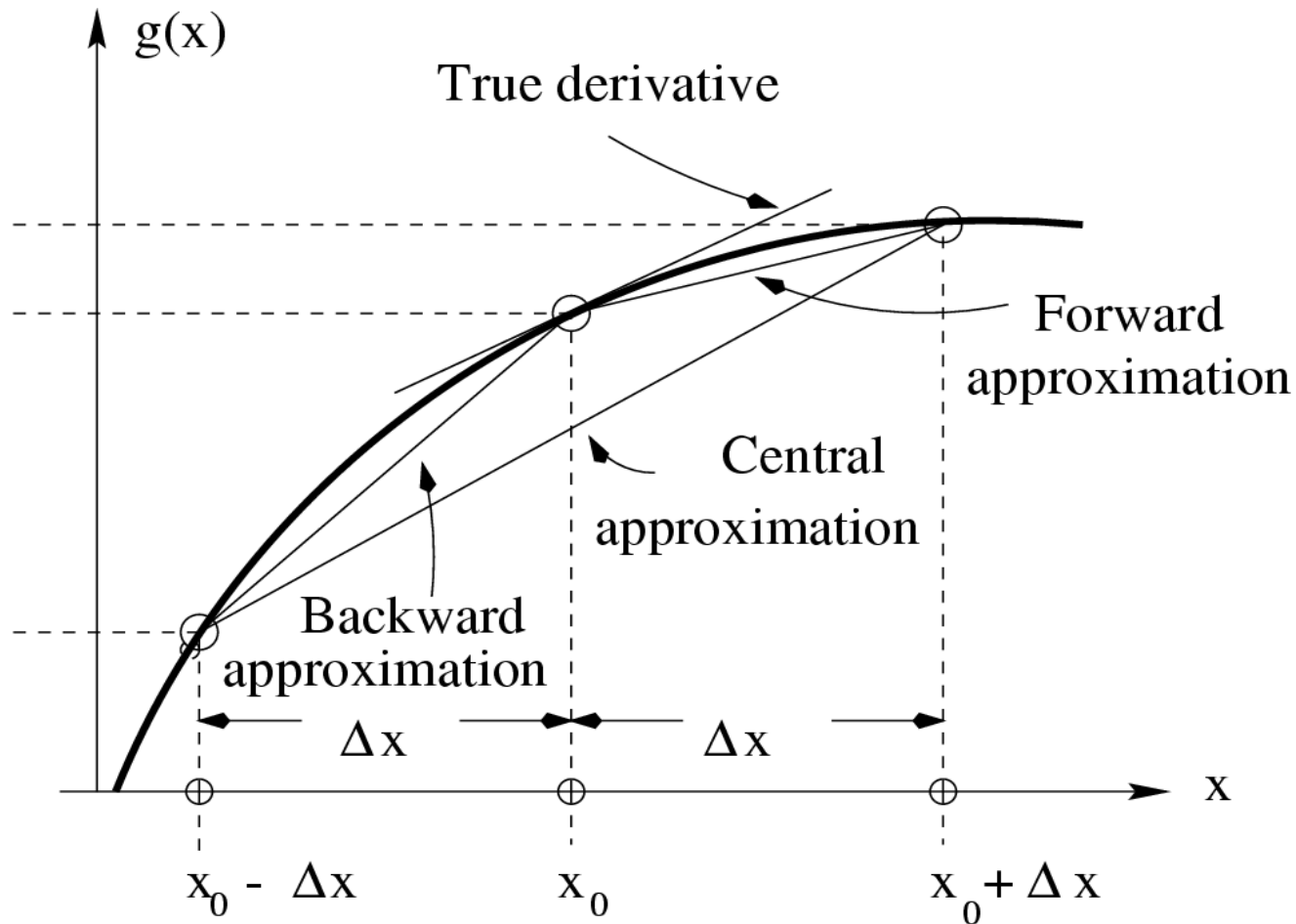
2-Step Predictor-Corrector

$$\text{Predictor} \quad : y_{i+1}^0 = y_i + h \left(\frac{3}{2} f(x_i, y_i) - \frac{1}{2} f(x_{i-1}, y_{i-1}) \right)$$

$$\text{Corrector} \quad : y_{i+1}^k = y_i + h \left(\frac{1}{2} f(x_{i+1}, y_{i+1}^{k-1}) + \frac{1}{2} f(x_i, y_i) \right)$$

- At each iteration one prediction step is done and as many correction steps as needed.
- y_{i+1}^k is the estimate of the solution at x_{i+1} after k correction steps.

Finite difference approximation to derivatives



Consider a smooth function $g(x)$. Taylor's theorem reads:

$$g(x_0 + \Delta x) = g(x_0) + \sum_k \frac{\Delta x^k}{k!} g^{(k)}(x_0)$$

In particular:

$$g(x_0 + \Delta x) = g(x_0) + \Delta x g^{(1)}(x_0) + O(\Delta x^2) \quad (1)$$

$$g(x_0 - \Delta x) = g(x_0) - \Delta x g^{(1)}(x_0) + O(\Delta x^2) \quad (2)$$

$$Eq.(1) \rightarrow g^{(1)}(x_0) = \frac{g(x_0 + \Delta x) - g(x_0)}{\Delta x} + O(\Delta x) \quad (3)$$

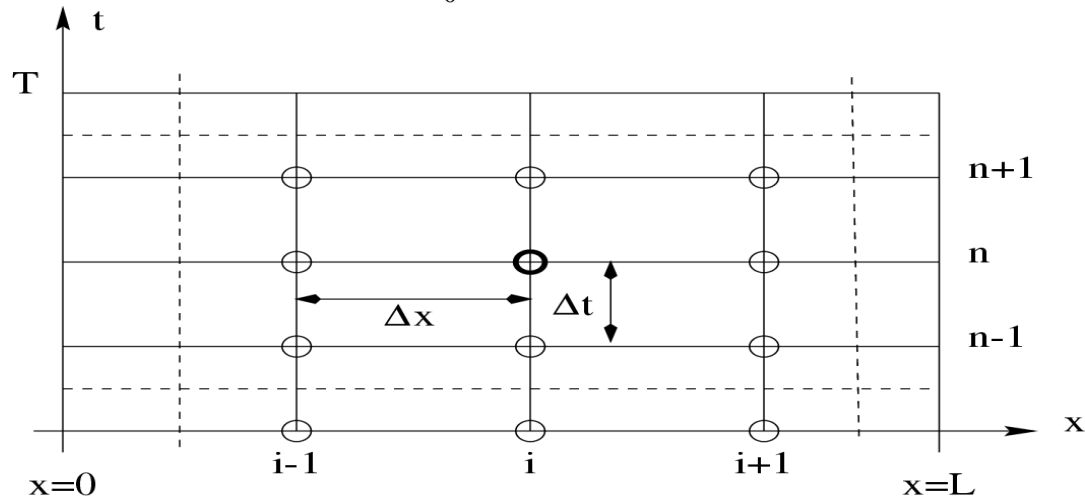
$$Eq.(2) \rightarrow g^{(1)}(x_0) = \frac{g(x_0) - g(x_0 - \Delta x)}{\Delta x} + O(\Delta x) \quad (4)$$

$$Eqs.(3) - (4) \rightarrow g^{(1)}(x_0) = \frac{g(x_0 + \Delta x) - g(x_0 - \Delta x)}{2\Delta x} + O(\Delta x^2)$$

Finite difference approximation of PDEs: FTCS

$$\text{PDE} : L_a(q) \equiv q_t + \lambda q_x = 0 \quad x \in [0, L] \quad t \in (0, T]$$

$$\text{IC} : q(x, 0) = q_0(x)$$



Discretise the x-t domain into a finite number of $M+1$ points, $I=0, \dots, M$

$$(x_i, t^n) \quad \text{with} \quad x_i = i\Delta x; \quad t^n = n\Delta t; \quad i, n \geq 0;$$

$$\text{Mesh} : \quad \Delta x = L / M; \quad \Delta t : \text{time step}$$

$$q_i^n \approx q(x_i, t^n)$$

Now approximate partial derivatives.

Use finite differences:

$$q_t \approx \frac{q_i^{n+1} - q_i^n}{\Delta t} + O(\Delta t) : \text{forward in time (FT)}$$

$$q_x \approx \frac{q_{i+1}^n - q_{i-1}^n}{2 \Delta x} + O(\Delta x^2) : \text{centred in space (CS)}$$

Then the PDE is replaced by a finite difference approximate operator:

$$L_a(q_i^n) = \frac{q_i^{n+1} - q_i^n}{\Delta t} + \lambda \frac{q_{i+1}^n - q_{i-1}^n}{2 \Delta x} = 0$$

$$\text{--- numerical scheme ---} \rightarrow q_i^{n+1} = q_i^n - \left(\frac{\lambda \Delta t}{\Delta x} \right) \left(\frac{q_{i+1}^n - q_{i-1}^n}{2 \Delta x} \right)$$

Introduce the dimensionless number:

$$c = \left(\frac{\lambda \Delta t}{\Delta x} \right) \quad \text{The Courant-Friedrichs-Lewy number or CFL number, or Courant number}$$

Note that c is a dimensionless quantity, it is the ratio of two velocities:

$$c = \frac{\lambda \Delta t}{\Delta x} = \frac{\lambda}{\Delta x / \Delta t} = \frac{\text{speed of PDE}}{\text{speed of mesh}}$$

Finally, the FTCS scheme reads:

$$q_i^{n+1} = q_i^n - \frac{1}{2} c (q_{i+1}^n - q_{i-1}^n) \quad \text{Stencil}$$

This formula allows us to calculate explicitly the evolution in time of discrete approximate values of the solution at every point, except for $i=0$ and $i=M$.

Convergence

Our ultimate goal is to construct schemes that converge, that is schemes that approach the true solution of the PDE when the mesh size tends to zero.

Lax's Equivalence theorem says that:

the only schemes that are CONVERGENT are those that
are CONSISTENT and STABLE

We must therefore work on CONSISTENCY AND STABILITY.

Local truncation error

Consider a particular scheme: FTCS.

The numerical analogue of the PDE is the approximate operator:

$$L_a(q_i^n) = \frac{q_i^{n+1} - q_i^n}{\Delta t} + \lambda \frac{q_{i+1}^n - q_{i-1}^n}{2\Delta x} = 0$$

We define the local truncation error:

$$E_T = L_a(q(i\Delta x, n\Delta t))$$

$$E_T = \frac{q(i\Delta x, (n+1)\Delta t) - q(i\Delta x, n\Delta t)}{\Delta t} + \lambda \frac{q((i+1)\Delta x, n\Delta t) - q((i-1)\Delta x, n\Delta t)}{2\Delta x}$$

Assuming the solution is sufficiently smooth we Taylor expand and obtain:

$$L_{TE} = (q_t + \lambda q_x)_i^n + \left[\frac{1}{2} \Delta t q_{tt} + \frac{1}{6} \lambda \Delta x^2 q_{xxx} + O(\Delta x^3) + O(\Delta t^2) \right]_i^n$$

Noting that: $(q_t + \lambda q_x)_i^n = 0 \rightarrow$

$$L_{TE} = \left[\frac{1}{2} \Delta t q_{tt} + \frac{1}{6} \lambda \Delta x^2 q_{xxx} + O(\Delta x^3) + O(\Delta t^2) \right]_i^n$$

$$L_{TE} = O(\Delta t) + O(\Delta x^2)$$

The FTCS is a first-order scheme

In general, if the local truncation error of a scheme is of the form:

$$L_{TE} = O(\Delta t^k) + O(\Delta x^l)$$

Then the scheme is said to be k-th order accurate in time and l-th order accurate in space.

Consistency

A numerical scheme is said to be consistent with the PDE being discretized if the local truncation error tends to zero as the mesh size tends to zero.

$$L_{TE} \rightarrow 0 \quad \text{as} \quad \Delta t \rightarrow 0 \quad \text{and} \quad \Delta x \rightarrow 0$$

For example, for the FTCS scheme we have

$$L_{TE} = O(\Delta t) + O(\Delta x^2)$$

Therefore FTCS is consistent with the PDE

Stability of a numerical method.

- If a method is consistent with the PDE, then all we need to bother is **stability**.
- One view of stability is that of **unbounded growth** of errors as the numerical scheme evolves the solution in time.
- Another view of stability is that of controlling **spurious oscillations**
- Stability in the sense on **unbounded growth** can be analysed by a variety of methods
- A popular method is the **von Neumann method**
- One performs a Fourier decomposition of the error. It is sufficient to consider a single component.

Stability analysis of the FTCS

$q_i^n = A^n e^{li\theta}$: trial solution

$I = \sqrt{-1}$; i : spacial index $q_i^{n+1} = q_i^n - \frac{1}{2}c(q_{i+1}^n - q_{i-1}^n) \rightarrow$

θ : frequency angle

$$A^{n+1} e^{li\theta} = A^n e^{li\theta} - \frac{1}{2}c [A^n e^{I(i+1)\theta} - A^n e^{I(i-1)\theta}] \rightarrow$$

$$A = 1 - \frac{1}{2}c [e^{I\theta} - e^{-I\theta}] = 1 - \frac{1}{2}c [2I \sin \theta] \rightarrow$$

$$A = 1 - Ic \sin \theta \rightarrow$$

$$|A|^2 = 1 + c^2 \sin^2 \theta \geq 1$$

Thus FTCS is unstable under all circumstances:
UNCONDITIONALLY UNSTABLE (useless).

Godunov's first-order upwind scheme

- Approximate derivatives in $q_t + \lambda q_x = 0$, $\lambda > 0$

by

$$q_t = \frac{q_i^{n+1} - q_i^n}{\Delta t}, \quad q_x = \frac{q_i^n - q_{i-1}^n}{\Delta x}$$

then

$$L_a(q_i^n) = \frac{q_i^{n+1} - q_i^n}{\Delta t} + \lambda \left(\frac{q_i^n - q_{i-1}^n}{\Delta x} \right) = 0$$

- The scheme reads

$$q_i^{n+1} = q_i^n - c(q_i^n - q_{i-1}^n)$$

Illustrate the
stencil

Local truncation error:

The finite difference operator is:

$$L_a(q_i^n) \equiv \frac{q_i^{n+1} - q_i^n}{\Delta t} + \lambda \left(\frac{q_i^n - q_{i-1}^n}{\Delta x} \right) = 0$$

Substitution of the exact solution into the approximate operator gives:

$$L_{TE} = \frac{q(i\Delta x, (n+1)\Delta t) - q(i\Delta x, n\Delta t)}{\Delta t} + \lambda \frac{q(i\Delta x, n\Delta t) - q((i-1)\Delta x, n\Delta t)}{\Delta x}$$

Assuming sufficient smoothness and Taylor expanding:

$$L_{TE} = (q_t + \lambda q_x)_i^n + \left[\frac{1}{2} \Delta t q_{tt} + \frac{1}{6} \lambda \Delta x^2 q_{xxx} + O(\Delta x^3) + O(\Delta t^2) \right]_i^n$$
$$L_{TE} = \left[\frac{1}{2} \Delta t q_{tt} - \frac{1}{2} \lambda \Delta x q_{xx} + O(\Delta x^2) + O(\Delta t^2) \right]_i^n$$

The scheme is first-order accurate in space and time

Stability analysis

$$\|A\|^2 = (1 - c)^2 + 2c(1 - c)\cos\theta + c^2$$

and the stability condition $\|A\|^2 \leq 1$ becomes

$$\boxed{0 \leq c \leq 1} \quad \text{The CFL condition of Courant condition}$$

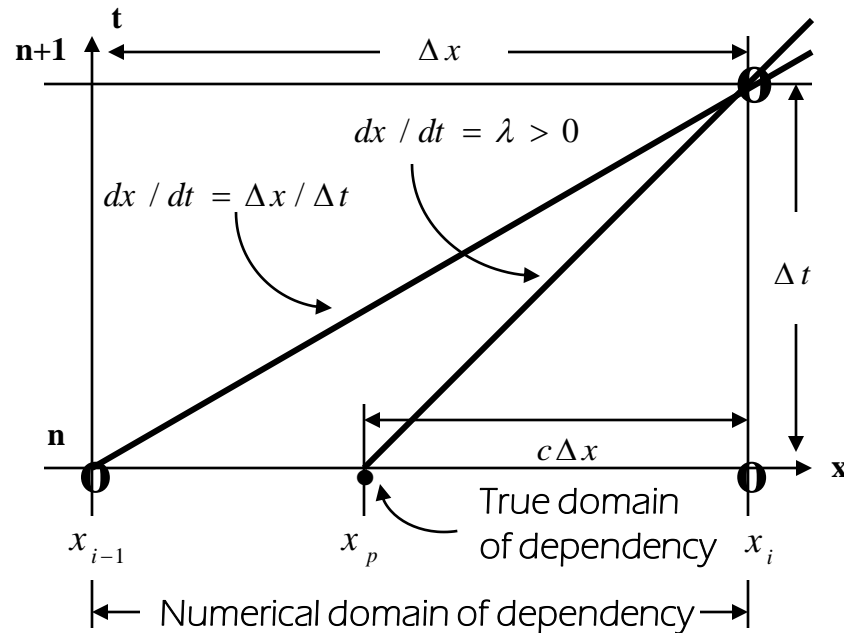
Given wave speed λ and mesh spacing Δx the time step Δt is determined from the stability condition

$$0 \leq \frac{\Delta t \lambda}{\Delta x} \leq 1 \Leftrightarrow 0 \leq \frac{\lambda}{\Delta x / \Delta t} \leq 1 \Rightarrow \Delta t \leq \frac{\Delta x}{\lambda}$$

$$\text{Set : } \Delta t = C_{cfl} \frac{\Delta x}{\lambda} \quad \text{with} \quad 0 < C_{cfl} \leq 1 \rightarrow$$

$$C_{cfl} = \frac{\lambda \Delta t}{\Delta x} : \text{the CFL number of the computation.}$$

The stencil (upwind)



The exact solution $q(x_i, t^n)$ is the value on the characteristic

$$\frac{dx}{dt} = \lambda \quad \text{at} \quad (x_i, t^{n+1}) \quad \text{that is :} \quad q(x_i, t^{n+1}) = q(x_p, t^n)$$

where x_p is the foot of the characteristic at time $t = t^n$

- For appropriate choices of Δx , Δt the point x_p lies between x_{i-1} and x_i
- Assume a linear interpolation between x_{i-1} and x_i

$$\tilde{q}(x) = q_{i-1}^n + (q_i^n - q_{i-1}^n) \left(\frac{x - x_{i-1}}{\Delta x} \right), \quad x \in [x_{i-1}, x_i]$$

- Evaluation of $\tilde{q}(x)$ at $x = x_p = (i - c) \Delta x$ gives

$$\tilde{q}(x_p) = q_i^n - c(q_i^n - q_{i-1}^n)$$

which is the Godunov scheme

The “downwind” scheme

- Approximate derivatives in $q_t + \lambda q_x = 0$, $\lambda > 0$

by

$$q_t = \frac{q_i^{n+1} - q_i^n}{\Delta t}, \quad q_x = \frac{q_{i+1}^n - q_i^n}{\Delta x}$$

then

$$L_a(q_i^n) = \frac{q_i^{n+1} - q_i^n}{\Delta t} + \lambda \left(\frac{q_{i+1}^n - q_i^n}{\Delta x} \right) = 0$$

- The scheme reads $q_i^{n+1} = q_i^n - c(q_{i+1}^n - q_i^n)$

Exercise: derive the local truncation error. Is the scheme consistent ?

Exercise: show that this scheme is unconditionally unstable.

General Form of the First-Order Upwind Scheme

- For $\lambda < 0$ the upwind scheme is $q_i^{n+1} = q_i^n - c(q_{i+1}^n - q_i^n)$
- For both $\lambda > 0$ and $\lambda < 0$ define:

$$\lambda^+ = \max(\lambda, 0) = \frac{1}{2}(\lambda + |\lambda|) \geq 0$$

$$\lambda^- = \min(\lambda, 0) = \frac{1}{2}(\lambda - |\lambda|) \leq 0$$

$$c^+ = \frac{\Delta t \lambda^+}{\Delta x}, \quad c^- = \frac{\Delta t \lambda^-}{\Delta x}$$

- The scheme reads: $q_i^{n+1} = q_i^n - c^+(q_i^n - q_{i-1}^n) - c^-(q_{i+1}^n - q_i^n)$

Exercise: show that the upwind scheme for negative speed is conditionally stable

Fully discrete and semi-discrete schemes

$$q_t + \lambda q_x = 0, \quad \lambda > 0$$

$$L_a(q_i^n) = \frac{q_i^{n+1} - q_i^n}{\Delta t} + \lambda \left(\frac{q_{i+1}^n - q_i^n}{\Delta x} \right) = 0 \quad \text{Fully discrete}$$

If the time derivative is left in its continuous form

$$\frac{d}{dt} q = -\lambda \left(\frac{q_{i+1}^n - q_i^n}{\Delta x} \right) \quad \text{Semi-discrete}$$

The method of lines

Explicit scheme and implicit schemes

$$q_t + \lambda q_x = 0, \quad \lambda > 0$$

$$\frac{q_i^{n+1} - q_i^n}{\Delta t} + \lambda \left(\frac{q_{i+1}^n - q_i^n}{\Delta x} \right) = 0 \rightarrow$$

$$q_i^{n+1} = q_i^n - c (q_{i+1}^n - q_i^n) \quad \text{Explicit}$$

$$\frac{q_i^{n+1} - q_i^n}{\Delta t} + \lambda \left(\frac{q_{i+1}^{n+1} - q_i^{n+1}}{\Delta x} \right) = 0 \rightarrow$$

$$q_i^{n+1} = q_i^n - c (q_{i+1}^{n+1} - q_i^{n+1}) \quad \text{Implicit}$$

Exercise: construct the fully discrete implicit version of FTCS

Monotone Schemes

$$q_i^{n+1} = H(q_{i-s}^n, \dots, q_i^n, \dots, q_{i+r}^n)$$

A monotone scheme satisfies:

$$\frac{\partial}{\partial q_k^n} H \geq 0 \quad \text{for all } k$$

Monotone schemes for the linear advection equation with constant speed of propagation are those whose coefficients are non-negative.

Example: The Godunov upwind scheme.

$$q_i^{n+1} = q_i^n - c(q_i^n - q_{i-1}^n)$$

$$H = cq_{i-1}^n + (1 - c)q_i^n$$

The Lax-Friedrichs scheme (LF)

$$q_i^{n+1} = \frac{1}{2}(1 + c)q_{i-1}^n + \frac{1}{2}(1 - c)q_{i+1}^n$$

- Conditionally stable $|c| \leq 1$
- Monotone
- Modified equation has $\alpha_{LF} = \frac{\Delta x}{2c}(1 - c^2)\lambda$

LF may also be seen as the FTCS scheme (unstable) with q_i^n replaced by $\frac{1}{2}(q_{i-1}^n + q_{i+1}^n)$

The Godunov Centered Scheme

$$q_i^{n+1} = \frac{1}{2}c(1+2c)q_{i-1}^n + (1-2c^2)q_i^n - \frac{1}{2}c(1-2c)q_{i+1}^n \quad \text{Stencil}$$

- Conditionally stable $0 \leq |c| \leq \frac{1}{2}\sqrt{2}$
- Monotone for $\frac{1}{2} \leq |c| \leq \frac{1}{2}\sqrt{2}$
- Oscillatory for $0 \leq |c| \leq \frac{1}{2} !!$

This is an interesting example of a first-order scheme that is NOT MONOTONE

The Lax-Wendroff Scheme (LW)

$$q_i^{n+1} = \frac{1}{2}c(1+c)q_{i-1}^n + (1-c^2)q_i^n - \frac{1}{2}c(1-c)q_{i+1}^n \quad \text{Stencil}$$

- Conditionally stable $0 \leq |c| \leq 1$
- Non-monotone (verify by inspection)

The FORCE scheme

$$q_i^{n+1} = \frac{1}{4}(1+c)^2 q_{i-1}^n + \frac{1}{2}(1-c^2) q_i^n + \frac{1}{4}(1-c)^2 q_{i+1}^n \quad \text{stencil}$$

- Conditionally stable: $|c| \leq 1$

- Monotone

- Modified equation: $\alpha_{force} = \frac{1}{4} \lambda \Delta x \left(\frac{1-c^2}{c} \right) = \frac{1}{2} \alpha_{LF}$