Compressible Flow - TME085

Lecture 13

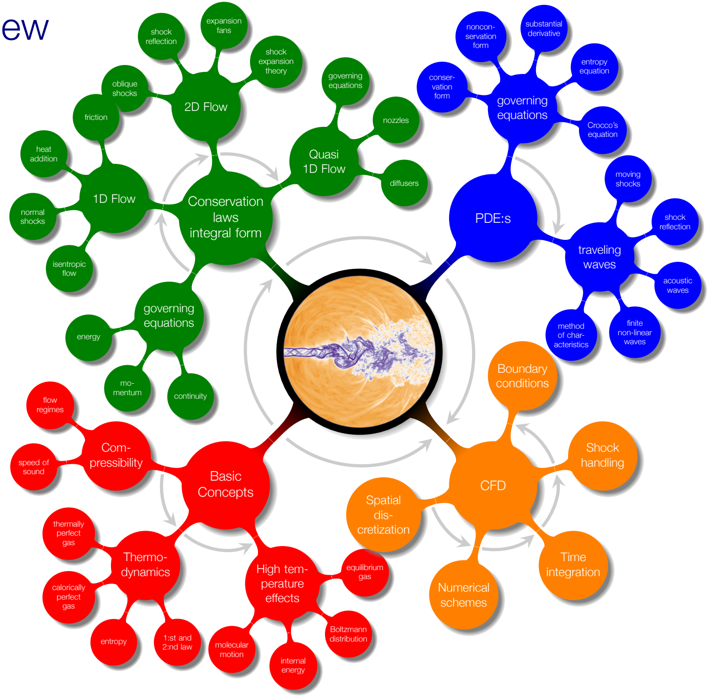
Niklas Andersson

Chalmers University of Technology
Department of Mechanics and Maritime Sciences
Division of Fluid Mechanics
Gothenburg, Sweden

niklas.andersson@chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY

Overview


Conservation laws integral form
2D Flow
shock reflection
expansion fans
shock expansion theory
oblique shocks
friction
1D Flow
heat addition
normal shocks
isentropic flow
Quasi 1D Flow
governing equations
nozzles
diffusers
governing equations
energy
mo-mentum
continuity
PDE:s
governing equations
noncon-servation form
substantial derivative
entropy equation
conser-vation form
Crocco's equation
traveling waves
moving shocks
shock reflection
acoustic waves
method of char-acteristics
finite non-linear waves
CFD
Boundary conditions
Shock handling
Spatial dis-cretization
Time integration
Numerical schemes
Basic Concepts
Com-pressibility
flow regimes
speed of sound
Thermo-dynamics
thermally perfect gas
calorically perfect gas
entropy
1:st and 2:nd law
High tem-perature effects
equilibrium gas
Boltzmann distribution
molecular motion
internal energy

Chapter 12
The Time-Marching Technique


method of char-acteristics
finite non-linear waves
Boundary conditions
Shock handling
CFD
Spatial dis-cretization
Time integration
Numerical schemes
equilibrium gas
Boltzmann distribution

Addressed Learning Outcomes

12 Explain the main principles behind a modern Finite Volume CFD code and such concepts as explicit/implicit time stepping, CFL number, conservation, handling of compression shocks, and boundary conditions

14 Analyze and verify the quality of the numerical solution

15 Explain the limitations in fluid flow simulation software

what about boundary conditions?

Spatial Discretization - Summary

1. Primary variables defined for all cells

$$\left\{\bar{\rho}, \overline{(\rho u)}, \overline{(\rho e_o)}\right\}_i, i \in \{1, 2, ..., N\}$$

2. Flux terms and lower-order terms may be computed

3. Temporal derivatives of the primary variables are defined for all cells

$$\left\{\frac{d}{dt}\bar{\rho}, \frac{d}{dt}\overline{(\rho u)}, \frac{d}{dt}\overline{(\rho e_o)}\right\}_i$$

Spatial Discretization - Summary

cell-averaged quantity
face-averaged quantity
source term

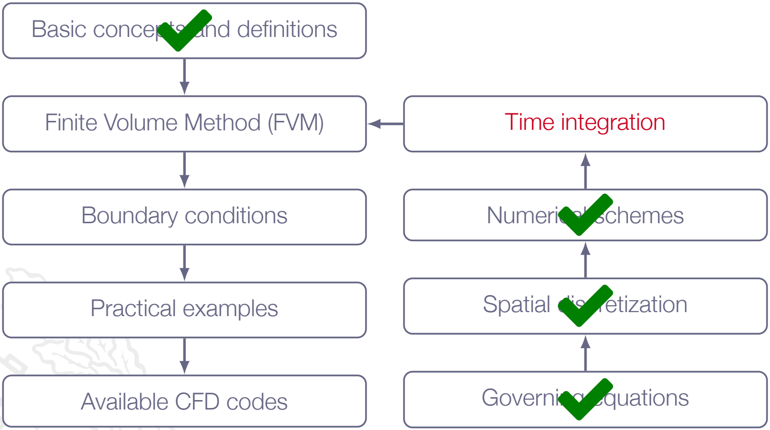
$$\frac{d}{dt}\bar{\rho}_i = \frac{1}{VOL_i}\left\{\overline{(\rho u)}_{i-\frac{1}{2}}A_{i-\frac{1}{2}} - \overline{(\rho u)}_{i+\frac{1}{2}}A_{i+\frac{1}{2}}\right\}$$

$$\frac{d}{dt}\overline{(\rho u)}_i = \frac{1}{VOL_i}\left\{\overline{(\rho u^2+p)}_{i-\frac{1}{2}}A_{i-\frac{1}{2}} - \overline{(\rho u^2+p)}_{i+\frac{1}{2}}A_{i+\frac{1}{2}} + \right.$$

$$\left.\bar{p}_i\left(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}\right)\right\}$$

$$\frac{d}{dt}\overline{(\rho e_o)}_i = \frac{1}{VOL_i}\left\{\overline{(\rho u h_o)}_{i-\frac{1}{2}}A_{i-\frac{1}{2}} - \overline{(\rho u h_o)}_{i+\frac{1}{2}}A_{i+\frac{1}{2}}\right\}$$

Roadmap - The Time-Marching Technique



Time Stepping

Time Stepping

The system of ODEs obtained from the spatial discretization in vector notation

$$\frac{d}{dt}\mathbf{Q} = \mathbf{F}(\mathbf{Q})$$

- $\mathbf{Q}$ is a vector containing all DOFs in all cells
- $\mathbf{F}(\mathbf{Q})$ is the time derivative of $\mathbf{Q}$ resulting from above mentioned flux approximations
  non-linear vector-valued function

Time Stepping

Three-stage Runge-Kutta - *one example of many*:

- Explicit time-marching scheme
- Second-order accurate

Time Stepping - Three-stage Runge-Kutta

$$\frac{d}{dt}\mathbf{Q} = \mathbf{F}(\mathbf{Q})$$

Let $\mathbf{Q}^n = \mathbf{Q}(t_n)$ and $\mathbf{Q}^{n+1} = \mathbf{Q}(t_{n+1})$

- t_n is the current time level and t_{n+1} is the next time level
- $\Delta t = t_{n+1} - t_n$ is the solver time step

Algorithm:

$$\begin{aligned}
1.\quad \mathbf{Q}^{*} &= \mathbf{Q}^n + \Delta t \mathbf{F}(\mathbf{Q}^n) \\
2.\quad \mathbf{Q}^{**} &= \mathbf{Q}^n + \frac{1}{2}\Delta t \mathbf{F}(\mathbf{Q}^n) + \frac{1}{2}\Delta t \mathbf{F}(\mathbf{Q}^{*}) \\
3.\quad \mathbf{Q}^{n+1} &= \mathbf{Q}^n + \frac{1}{2}\Delta t \mathbf{F}(\mathbf{Q}^n) + \frac{1}{2}\Delta t \mathbf{F}(\mathbf{Q}^{**})
\end{aligned}$$

DOFs in all cells updated from time level t_n to time level t_{n+1}, repeat procedure for $t_{n+2}, t_{n+3}, \ldots$

Time Stepping - Explicit Schemes

Properties of explicit time-stepping schemes:

+ Easy to implement in computer codes
+ Efficient execution on most computers
+ Easy to adapt for parallel execution on distributed memory systems (*e.g.* Linux clusters)

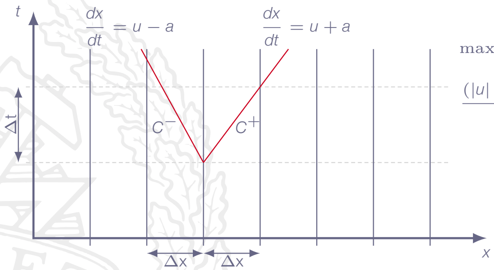
- Time step limitation (CFL number)
- Convergence to steady-state often slow (there are, however, some remedies for this)

Time Stepping - Explicit Schemes

Courant-Friedrich-Levy (CFL) number - *one-dimensional case*:

$$CFL_i = \frac{\Delta t(|u_i| + a_i)}{\Delta x_i} \leq 1$$

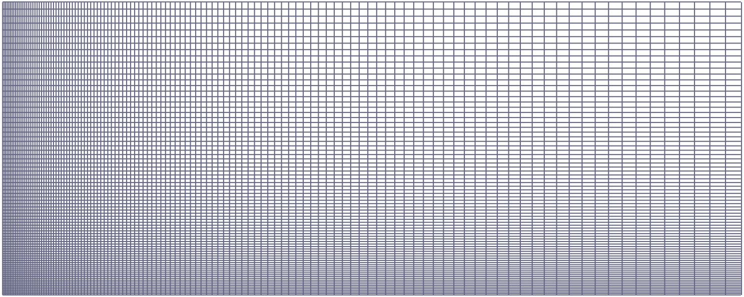
Interpretation: The fastest characteristic (C^+ or C^-) must not travel longer than Δx during one time step



$$\max(|u-a|, |u+a|)\Delta t = (|u| + a)\Delta t \leq \Delta x \Rightarrow$$

$$\frac{(|u| + a)\Delta t}{\Delta x} = CFL \leq 1$$

Time Stepping - Explicit Schemes



Steady-state problems:

- local time stepping
- each cell has an individual time step
- Δt_i maximum allowed value based on CFL criteria

Unsteady problems:

- time accurate
- all cells have the same time step
- $\Delta t_i = \min\{\Delta t_1, ..., \Delta t_N\}$

Explicit Finite-Volume Method - Summary

The described numerical scheme is an example of a density-based, fully coupled scheme

Explicit Finite-Volume Method - Summary

- density-based schemes
  - solve for density in the continuity equation
  - in general preferred for high-Mach-number flows and for unsteady compressible flows
- pressure-based schemes
  - the continuity and momentum equations are combined to form a pressure correction equation
  - were first used for incompressible flows but have been adapted for compressible flows also
  - quite popular for steady-state subsonic/transonic flows

Explicit Finite-Volume Method - Summary

- fully-copuled schemes
  - all equations (continuity, momentum, energy) are solved for simultaneously
- segregated schemes
  - alternate between the solution of the velocity field and the pressure field (pressure-based solver)

Explicit Finite-Volume Method - Summary

Spatial discretization:

- Control volume formulations of conservation equations are applied to the cells of the discretized domain
- Cell-averaged flow quantities ($\overline{\rho}, \overline{\rho u}, \overline{\rho e_o}$) are chosen as degrees of freedom (DOFs)
- Flux terms are approximated in terms of the chosen DOFs
  - high-order, upwind type of flux approximation is used for optimum results
- A fully conservative scheme is obtained
  - the flux leaving one cell is identical to the flux entering the neighboring cell
- The result of the spatial discretization is a system of ODEs

Explicit Finite-Volume Method - Summary

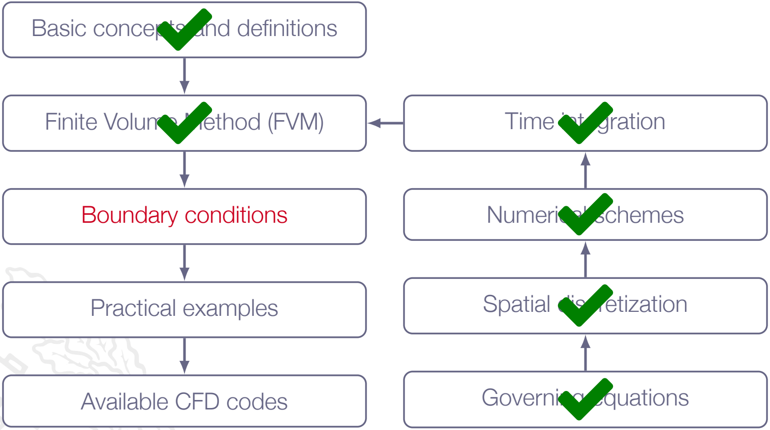
Time marching:

- Three-stage, second-order accurate Runge-Kutta scheme
  - Explicit time-stepping
  - Time step length limited by the *CFL* condition ($CFL \leq 1$)

Classification of numerical scheme:

- density-based
  - includes the continuity equation
- fully coupled
  - all equations are solved simultaneously

Roadmap - The Time-Marching Technique



Boundary Conditions

Boundary Conditions

Boundary conditions are very important for numerical simulation of compressible flows

Main reason: both flow and acoustics involved!

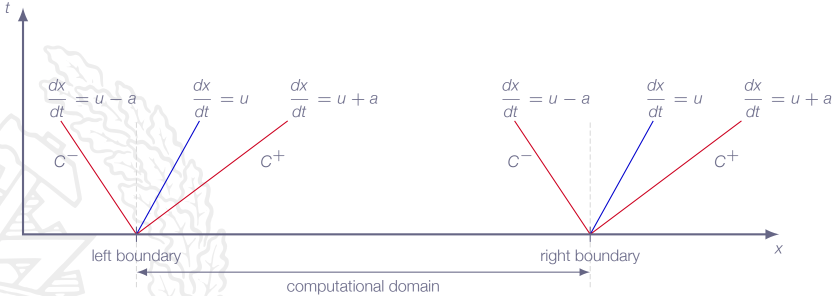
Example 1:
Finite-volume CFD code for Quasi-1D compressible flow
(Time-marching procedure)

What boundary conditions should be applied at the left and right ends?

Boundary Conditions

three characteristics:

1. C^+
2. C^-
3. advection



Boundary Conditions

- C^+ and C^- characteristics describe the transport of isentropic pressure waves (often referred to as acoustics)
- The advection characteristic simply describes the transport of certain quantities with the fluid itself (for example entropy)
- In one space dimension and time, these three characteristics, together with the quantities that are known to be constant along them, give a complete description of the time evolution of the flow
- We can use the characteristics as a guide to tell us what information that should be specify at the boundaries

Left Boundary - Subsonic Inflow

we have three PDEs, and are solving for three unknowns

- Subsonic inflow: $0 < u < a$

  $$u - a < 0$$
  $$u > 0$$
  $$u + a > 0$$

  - one outgoing characteristic
  - two ingoing characteristics
- Two variables should be specified at the boundary
- The third variable must be left free

Left Boundary - Subsonic Outflow

we have three PDEs, and are solving for three unknowns

- Subsonic outflow: $-a < u < 0$

  $$u - a < 0$$
  $$u < 0$$
  $$u + a > 0$$

  - two outgoing characteristics
  - one ingoing characteristic
- One variable should be specified at the boundary
- The second and third variables must be left free

Left Boundary - Supersonic Inflow

we have three PDEs, and are solving for three unknowns

- Supersonic inflow: $u > a$

  $$u - a > 0$$
  $$u > 0$$
  $$u + a > 0$$

  - no outgoing characteristics
  - three ingoing characteristics
- All three variables should be specified at the boundary
- No variables must be left free

Left Boundary - Supersonic Outflow

we have three PDEs, and are solving for three unknowns

- Supersonic outflow: $u < -a$

  $u - a < 0$
  $u < 0$
  $u + a < 0$
  - three outgoing characteristics
  - no ingoing characteristics
- No variables should be specified at the boundary
- All variables must be left free

Right Boundary - Subsonic Outflow

we have three PDEs, and are solving for three unknowns

- Subsonic outflow: $0 < u < a$

  $u - a < 0$
  $u > 0$
  $u + a > 0$

  - one ingoing characteristic
  - two outgoing characteristics
- One variable should be specified at the boundary
- The second and third variables must be left free

Right Boundary - Subsonic Inflow

we have three PDEs, and are solving for three unknowns

- Subsonic inflow: $-a < u < 0$

  $$u - a < 0$$
  $$u < 0$$
  $$u + a > 0$$

  - two ingoing characteristics
  - one outgoing characteristic
- Two variables should be specified at the boundary
- The third variables must be left free

Right Boundary - Supersonic Outflow

we have three PDEs, and are solving for three unknowns

- Supersonic outflow: $u > a$

  $$u - a > 0$$
  $$u > 0$$
  $$u + a > 0$$

  - no ingoing characteristics
  - three outgoing characteristics
- No variables should be specified at the boundary
- All three variables must be left free

Right Boundary - Supersonic Inflow

we have three PDEs, and are solving for three unknowns

- Supersonic inflow: $u < -a$

  $u - a < 0$
  $u < 0$
  $u + a < 0$

  - three ingoing characteristics
  - no outgoing characteristics
- All three variables should be specified at the boundary
- No variables must be left free

Subsonic Inflow (Left Boundary) - Example

Subsonic inflow: we should specify two variables

Alt	specified variable 1	specified variable 2	well-posed	non-reflective
1	p_o	T_o	X	
2	ρu	T_o	X	
3	s	J^+	X	X

well posed:

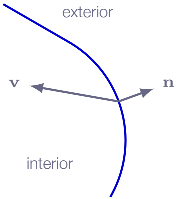
- the problem has a solution
- the solution is unique
- the solution's behaviour changes continuously with initial conditions

Subsonic Outflow (Left Boundary) - Example

Subsonic outflow: we should specify one variable

Alt	specified variable	well-posed	non-reflective
1	p	X	
2	ρu	X	
3	J^+	X	X

Subsonic Inflow 2D/3D



$\mathbf{n}$	unit normal vector
$\mathbf{v}$	fluid velocity at boundary

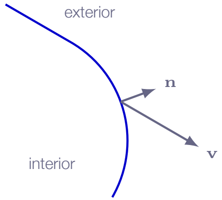
Subsonic inflow

- Assumption:

  $-a < \mathbf{v} \cdot \mathbf{n} < 0$

- Four ingoing characteristics
- One outgoing characteristic
- Specify four variables at the boundary:
  - example: p_o, T_o, flow direction (two angles)

Subsonic Outflow 2D/3D



$\mathbf{n}$	unit normal vector
$\mathbf{v}$	fluid velocity at boundary

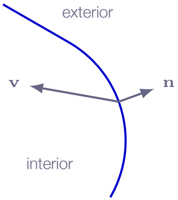
Subsonic outflow

- Assumption:

  $0 < \mathbf{v} \cdot \mathbf{n} < a$

- One ingoing characteristics
- Four outgoing characteristic
- Specify one variables at the boundary:
  - example: p

Supersonic Inflow 2D/3D



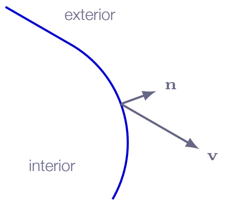
$\mathbf{n}$	unit normal vector
$\mathbf{v}$	fluid velocity at boundary

- Supersonic inflow
- Assumption:

  $\mathbf{v} \cdot \mathbf{n} < -a$

- Five ingoing characteristics
- No outgoing characteristics
- Specify five variables at the boundary:
  - all solver variables specified

Supersonic Outflow 2D/3D



$\mathbf{n}$	unit normal vector
$\mathbf{v}$	fluid velocity at boundary

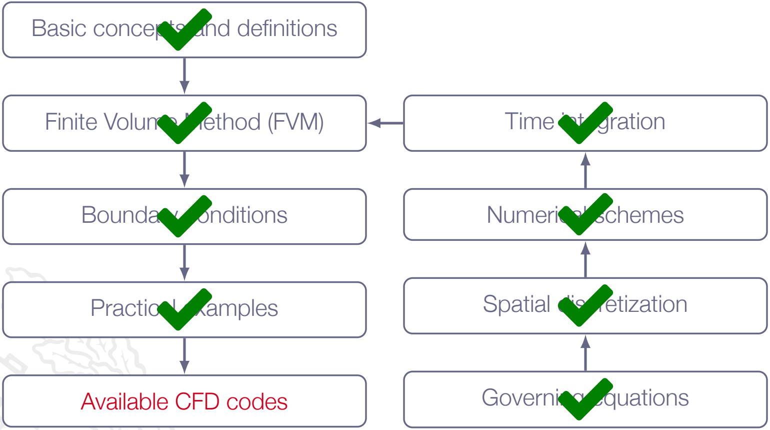
Supersonic outflow

- Assumption:

  $\mathbf{v} \cdot \mathbf{n} > a$

- No ingoing characteristics
- Five outgoing characteristics
- No variables specified at the boundary:

Roadmap - The Time-Marching Technique



Available CFD Codes

CFD Codes

List of free and commercial CFD codes:

http://www.cfd-online.com/Wiki/Codes

- Free codes are in general unsupported and poorly documented

- Commercial codes are often claimed to be suitable for all types of flows
  The reality is that the user must make sure of this!

- Industry/institute/university in-house codes not listed
  - non-commercial but proprietary
  - part of design/analysis system

CFD Codes - General Guidlines

Simulation of high-speed and/or unsteady compressible flows:

- Use correct solver options
  otherwise you may obtain completely wrong solution!

- Use a high-quality grid
  a poor grid will either not give you any solution at all (no convergence) or at best a very inaccurate solution!

ANSYS-FLUENT® - Typical Experiences

- Very robust solver - will almost always give you a solution
- Accuracy of solution depends a lot on grid quality
- Shocks are generally smeared more than in specialized codes
- Solver is generally very efficient for steady-state problems
- Solver is less efficient for truly unsteady problems, where both flow and acoustics must be resolved accurately

ANSYS-FLUENT® - Solver Options

- Coupled or Density-based *depends on version*
  - the continuity, momentum, energy equations are solved for simultaneously
    just like in the Quasi-1D code discussed previously
- Density = Ideal gas law
  - the calorically perfect gas assumption is activated
  - the energy equation is activated
- Explicit or Implicit time stepping
  - Explicit recommended for unsteady compressible flows
    CFL is set to 1 as default, but may be changed
  - Implicit more efficient for steady-state compressible flows
    CFL is set to 5 as default, but may be changed

ANSYS-FLUENT® - Solver Features

Spatial discretization:

- Finite-Volume Method (FVM)
- Unstructured grids
- Fully conservative, density-based scheme
- Flux approximations:
  first-order, second-order, upwind, ...
- Fully coupled solver approach

Explicit time stepping:

- Runge-Kutta time stepping

Implicit time stepping:

- Iterative solver based on Algebraic Multi-Grid (AGM)

Roadmap - The Time-Marching Technique