# Compressible Flow - TME085 

## Lecture 14

Niklas Andersson

Chalmers University of Technology<br>Department of Mechanics and Maritime Sciences<br>Division of Fluid Mechanics<br>Gothenburg, Sweden<br>niklas.andersson@chalmers.se

## Chapter 12

The Time-Marching Technique


## 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
time for CFD!

## Roadmap - The Time-Marching Technique



Time Stepping

## Quasi-One-Dimensional Flow - Spatial Discretization

## cell-averaged quantity

face-averaged quantity
source term

$$
\begin{aligned}
& \left.V O L_{i} \frac{d}{d t} \bar{\rho}_{i}-\overline{(\rho u)}_{i-\frac{1}{2}} A_{i-\frac{1}{2}}+\overline{(\rho u}\right)_{i+\frac{1}{2}} A_{i+\frac{1}{2}}=0 \\
& V O L_{i} \frac{d}{d t} \overline{(\rho u)_{i}}-\overline{\left(\rho u^{2}+p\right)_{i-\frac{1}{2}}} A_{i-\frac{1}{2}}+\overline{\left(\rho u^{2}+p\right)_{i+\frac{1}{2}}} A_{i+\frac{1}{2}}=\bar{p}_{i}\left(A_{i+\frac{1}{2}}-A_{i-\frac{1}{2}}\right) \\
& \operatorname{VOL} \frac{d}{} \frac{d}{d t} \overline{\left(\rho e_{0}\right)_{i}}-{\overline{\left(\rho u h_{0}\right)_{i-\frac{1}{2}}}}^{A_{i-\frac{1}{2}}}+{\overline{\left(\rho u h_{0}\right)_{i+\frac{1}{2}}}}^{A_{i+\frac{1}{2}}}=0
\end{aligned}
$$

Application of these equations to all cells $i \in\{1,2, \ldots ., N\}$ of the computational domain results in a system of ODEs

## Quasi-One-Dimensional Flow - Spatial Discretization

cell-averaged quantity

face-averaged quantity
source term

$$
\begin{aligned}
& V^{V} L_{i} \frac{d}{d t} \bar{\rho}_{i}=\overline{(\rho u)}_{i-\frac{1}{2}} A_{i-\frac{1}{2}}-\overline{(\rho u)}_{i+\frac{1}{2}} A_{i+\frac{1}{2}} \\
& V^{\prime} L_{i} \frac{d}{d t} \overline{(\rho u)}_{i}={\overline{\left(\rho u^{2}+p\right)_{i-\frac{1}{2}}} A_{i-\frac{1}{2}}-{\overline{\left(\rho u^{2}+p\right)_{i+\frac{1}{2}}}}_{i+\frac{1}{2}}+\bar{p}_{i}\left(A_{i+\frac{1}{2}}-A_{i-\frac{1}{2}}\right)}_{V_{O L} \frac{d}{d t}{\overline{\left(\rho e_{0}\right)_{i}}}_{i}={\overline{\left(\rho u h_{0}\right)_{i-\frac{1}{2}}}} A_{i-\frac{1}{2}}-{\overline{\left(\rho u h_{0}\right)_{i+\frac{1}{2}}}} A_{i+\frac{1}{2}}}
\end{aligned}
$$

## Quasi-One-Dimensional Flow - Spatial Discretization

## cell-averaged quantity

face-averaged quantity
source term

$$
\begin{aligned}
& \left.\left.\frac{d}{d t} \bar{\rho}_{i}=\frac{1}{V O L_{i}}[\overline{(\rho u})_{i-\frac{1}{2}} A_{i-\frac{1}{2}}-\overline{(\rho u}\right)_{i+\frac{1}{2}} A_{i+\frac{1}{2}}\right] \\
& \frac{d}{d t} \overline{(\rho u)_{i}}=\frac{1}{V O L_{i}}\left[{\left.\overline{\left(\rho u^{2}+\rho\right)_{i-\frac{1}{2}}} A_{i-\frac{1}{2}}-{\overline{\left(\rho u^{2}+p\right)_{i+\frac{1}{2}}}} A_{i+\frac{1}{2}}+\bar{\rho}_{i}\left(A_{i+\frac{1}{2}}-A_{i-\frac{1}{2}}\right)\right]}_{\frac{d}{d t}{\overline{\left(\rho e_{O}\right)_{i}}}=\frac{1}{V O L_{i}}\left[{\overline{\left(\rho u h_{O}\right)_{i-\frac{1}{2}}}} A_{i-\frac{1}{2}}-{\overline{\left(\rho u h_{O}\right)_{i+\frac{1}{2}}}} A_{i+\frac{1}{2}}\right]} . l\right.
\end{aligned}
$$

$$
\frac{d}{d t} \overline{\mathbf{Q}}_{i}=\mathbf{F}\left(\overline{\mathbf{Q}}_{i}\right) \text { where } \overline{\mathbf{Q}}_{i}=\left[\bar{\rho}, \overline{\rho u}, \overline{\rho e_{o}}\right]_{i}, i \in\{1: \text { NCells }\}
$$

## Time Stepping

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

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

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

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

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

1. $\mathbf{Q}^{*}=\mathbf{Q}^{n}+\Delta t \mathbf{F}\left(\mathbf{Q}^{n}\right)$
2. $\mathbf{Q}^{* *}=\mathbf{Q}^{n}+\frac{1}{2} \Delta t \mathbf{F}\left(\mathbf{Q}^{n}\right)+\frac{1}{2} \Delta t \mathbf{F}\left(\mathbf{Q}^{*}\right)$
3. $\mathbf{Q}^{n+1}=\mathbf{Q}^{n}+\frac{1}{2} \Delta t \mathbf{F}\left(\mathbf{Q}^{n}\right)+\frac{1}{2} \Delta t \mathbf{F}\left(\mathbf{Q}^{* *}\right)$

## Time Stepping - Three-stage Runge-Kutta

```
void RungeKutta:: fwd(Domain *dom){
    G3DCopy(dom->cons,cons0);
    /* Runge-Kutta step 1 */
    dom->update () ;
    if(!G3DMode : : constdt) {LocalTimeStep(dom) ;}
    dcons->evaluate(dom);
    G3DWAXPY(dom }->\mathrm{ cons,1.0, dcons, cons0) ;
    G3DAXPBY (cons0,0.5,0.5,dom->cons) ;
    /* Runge-Kutta step 2 */
    dom->update();
    dcons }->evaluate(dom)
    G3DWAXPY (dom }->\mathrm{ cons,0.5, dcons, cons0);
    /* Runge-Kutta step 3 */
    dom->update();
    dcons->evaluate(dom)
    G3DWAXPY(dom>>cons,0.5, dcons, cons0);
}
```


## 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:

$$
C F L_{i}=\frac{\Delta t\left(\left|u_{i}\right|+a_{i}\right)}{\Delta x_{i}} \leq 1
$$

Interpretation: The fastest characteristic $\left(C^{+}\right.$or $\left.C^{-}\right)$must not travel longer than $\Delta x$ during one time step

Time Stepping - Explicit Schemes


## Time Stepping - Explicit Schemes



Steady-state problems:
local time stepping
each cell has an individual time step

$$
\Delta t_{i} \text { maximum allowed value based on CFL criteria }
$$

Unsteady problems:

> time accurate
all cells have the same time step

$$
\Delta t_{i}=\min \left\{\Delta t_{1}, \ldots, \Delta t_{N}\right\}
$$

## 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 specifed at the boundaries

## Left Boundary - Subsonic Inflow

we have three PDEs, and are solving for three unknowns

Subsonic inflow: $0<u<a$

$$
\begin{aligned}
& u-a<0 \\
& u>0 \\
& u+a>0
\end{aligned}
$$


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: $-a<u<0$

$$
\begin{aligned}
& u-a<0 \\
& u<0 \\
& u+a>0
\end{aligned}
$$


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: $u>a$

$$
\begin{aligned}
& u-a>0 \\
& u>0 \\
& u+a>0
\end{aligned}
$$


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: $u<-a$

$$
\begin{aligned}
& u-a<0 \\
& u<0 \\
& u+a<0
\end{aligned}
$$


three outgoing characteristics no ingoing characteristics

No variables should be specified at the boundary
All variables must be left free

## Right Boundary - Subsonic Inflow

we have three PDEs, and are solving for three unknowns

Subsonic inflow: $-a<u<0$

$$
\begin{aligned}
& u-a<0 \\
& u<0 \\
& u+a>0
\end{aligned}
$$

two ingoing characteristics one outgoing characteristic

Two variables should be specified at the boundary
The third 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 - Supersonic Inflow

we have three PDEs, and are solving for three unknowns

Supersonic inflow: $u<-a$

$$
\begin{aligned}
& u-a<0 \\
& u<0 \\
& u+a<0
\end{aligned}
$$

three ingoing characteristics
no outgoing characteristics

All three variables should be specified at the boundary No variables must be left free

## Right Boundary - Supersonic Outflow

we have three PDEs, and are solving for three unknowns

Supersonic outflow: $u>a$

$$
\begin{aligned}
& u-a>0 \\
& u>0 \\
& u+a>0
\end{aligned}
$$

no ingoing characteristics
three outgoing characteristics

No variables should be specified at the boundary All three variables must be left free

## 1D Boundary Conditions (Summary)

| Characteristic |  | 1D subsonic inflow (left) | 1D subsonic inflow (right) |
| :---: | :---: | :---: | :---: |
| advection | $\mathrm{v} \cdot \mathrm{n}$ | $(u, 0,0) \cdot(-1,0,0)=-u<0$ | $(-u, 0,0) \cdot(1,0,0)=-u<0$ |
| $C^{-}$ | $\mathbf{v} \cdot \mathbf{n}-\mathrm{a}$ | $-u-a<0$ | $-u-a<0$ |
| $C^{+}$ | $\mathbf{v} \cdot \mathbf{n}+\mathrm{a}$ | $-u+a>0$ | $-u+a>0$ |
| Characteristic |  | 1D subsonic outflow (left) | 1D subsonic outflow (right) |
| advection | $\mathrm{v} \cdot \mathrm{n}$ | $(-u, 0,0) \cdot(-1,0,0)=u>0$ | $(u, 0,0) \cdot(1,0,0)=u>0$ |
| $C^{-}$ | $\mathbf{v} \cdot \mathbf{n}-\mathrm{a}$ | $u-a<0$ | $u-a<0$ |
| $C^{+}$ | $\mathbf{v} \cdot \mathbf{n}+a$ | $u+a>0$ | $u+a>0$ |
| Characteristic |  | 1D supersonic inflow (left) | 1D supersonic inflow (right) |
| advection | $\mathrm{v} \cdot \mathrm{n}$ | $(u, 0,0) \cdot(-1,0,0)=-u<0$ | $(-u, 0,0) \cdot(1,0,0)=-u<0$ |
| $C^{-}$ | $\mathbf{v} \cdot \mathbf{n}-\mathrm{a}$ | $-u-a<0$ | $-u-a<0$ |
| $C^{+}$ | $\mathbf{v} \cdot \mathbf{n}+\mathrm{a}$ | $-u+a<0$ | $-u+a<0$ |
| Characteristic |  | 1D supersonic outflow (left) | 1D supersonic outflow (right) |
| advection <br> $C^{-}$ $c^{+}$ | $\begin{aligned} & \mathbf{v} \cdot \mathbf{n} \\ & \mathbf{v} \cdot \mathbf{n}-a \end{aligned}$ | $\begin{gathered} (-u, 0,0) \cdot(-1,0,0)=u>0 \\ u-a>0 \end{gathered}$ | $\begin{gathered} (u, 0,0) \cdot(1,0,0)=u>0 \\ u-a>0 \end{gathered}$ |
|  | $\mathbf{v} \cdot \mathbf{n}+\mathrm{a}$ | $u+a>0$ | $u+a>0$ |

## Subsonic Inflow (Left Boundary) - Example

Subsonic inflow: we should specify two variables

| Alt | specified <br> variable 1 | specified <br> variable 2 | well-posed | non-reflective |
| :---: | :---: | :---: | :---: | :---: |
| 1 | $p_{0}$ | $T_{0}$ | $X$ |  |
| 2 | $\rho u$ | $T_{0}$ | $X$ |  |
| 3 | $s$ | $J^{+}$ | $X$ | $X$ |

well posed:

1. the problem has a solution
2. the solution is unique
3. the solution's behaviour changes continuously with initial conditions

## Subsonic Outflow (Left Boundary) - Example

Subsonic outflow: we should specify one variable

| Alt | specified <br> variable | well-posed | non-reflective |
| :---: | :---: | :---: | :---: |
| 1 | $\rho$ | $X$ |  |
| 2 | $\rho u$ | $X$ |  |
| 3 | $J^{+}$ | $X$ | $X$ |

## Subsonic Inflow 2D/3D



Subsonic inflow

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

Four ingoing characteristics
One outgoing characteristic

Specify four variables at the boundary: $p_{0}, T_{0}$, and flow direction (two angles)

## Subsonic Outflow 2D/3D



Subsonic outflow

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

One ingoing characteristics
Four outgoing characteristic

Specify one variables at the boundary: static pressure

## Supersonic Inflow 2D/3D



Supersonic inflow

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

Five ingoing characteristics
No outgoing characteristics

Specify five variables at the boundary: solver variables

## Supersonic Outflow 2D/3D



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



Explicit Finite-Volume Method - Summary

The described numerical approach can be categorized as


## Fully coupled

## Structured

## Explicit

with the following features

High-order convective scheme

## 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 ( $\bar{\rho}, \overline{\rho u}, \overline{\rho e_{o}}$ ) are chosen as degrees of freedom

Flux terms are approximated in terms of the chosen degrees of freedom 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 $\leq 1$ )

## 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!

## CFD Codes - General Guidlines

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

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

1. coupled solver
2. equation of state
3. energy equation included

## 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 ${ }^{\circledR} /$ STAR-CCM $+{ }^{\circledR}$ - Typical Experiences

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

## Roadmap - The Time-Marching Technique



