The Extension to the Navier-Stokes Equations

M. Ferch[2,] S. Roller[1], C.-D. Munz[2], M. Dumbser[2]

Universität Stuttgart
[1]Höchstleistungsrechenzentrum Stuttgart (HLRS)
www.hlrs.de
[2]Institut für Aerodynamik und Gasdynamik (IAG)
www.iag.uni-stuttgart.de

The Compressible Navier-Stokes Equations

$$
\begin{array}{cccccc}
\rho_t & + & \nabla\cdot(\rho U) & = & 0 \\
(\rho U)_t & + & \nabla\cdot((\rho U)\circ U)+\nabla p & = & \nabla\cdot\tau \\
e_t & + & \nabla\cdot(U(e+p)) & = & \nabla\cdot(\tau U)-\nabla\cdot q
\end{array}
$$

Equations include effects of

- Viscosity
- Heat conduction

Homogeneous System reduces to the Euler Equations

The nondimensional Navier-Stokes Equations

Since numerical problems can arise from variables with extremely differnt scales the equations are used in nondimensional form

$$
\begin{array}{cccccc}
\hat\rho_t & + & \nabla\cdot(\hat\rho\hat U) & = & 0 \\
(\hat\rho\hat U)_t & + & \nabla\cdot((\hat\rho\hat U)\circ\hat U)+\nabla\hat p & = & \dfrac{1}{\mathrm{Re}_{ref}}\nabla\cdot\hat\tau \\
\hat e_t & + & \nabla\cdot(\hat U(\hat e+\hat p)) & = & \dfrac{1}{\mathrm{Re}_{ref}}\nabla\cdot(\hat\tau\hat U)-\dfrac{\gamma}{(\gamma-1)\mathrm{Re}_{ref}\,\mathrm{Pr}_{ref}}\nabla\cdot\hat q
\end{array}
$$

The LHS only contains nondimensional variables. The additional Terms containing the Reynolds number and the Prandtl Number only appear on the RHS of the system.
→ The Euler Equations can be formulated in dimensional or nondimensional form arbitrarily

We will drop the ^ for nondimensional variables in the following slides

Flux Formulation

In order to construct a FV method we can formulate the 2D equations as

$$U_t + F_x^C + G_y^C = F_x^D + G_y^D$$

The LHS corresponds to the Euler Equations while the RHS can be written as:

$$
F^D = \begin{pmatrix} 0 \\ \frac{4}{3}\mu u_x - \frac{2}{3}\mu v_y \\ \mu(u_y+v_x) \\ u\left(\frac{4}{3}\mu u_x - \frac{2}{3}\mu v_y\right) + v\mu(u_y+v_x) - q_1 \end{pmatrix}
$$

$$
G^D = \begin{pmatrix} 0 \\ \mu(u_y+v_x) \\ \frac{4}{3}\mu v_y - \frac{2}{3}\mu u_x \\ u\mu(u_y+v_x) + v\left(\frac{4}{3}\mu v_y - \frac{2}{3}\mu u_x\right) - q_2 \end{pmatrix}
$$

Numerical Algorithm (1)

In order to simplify the process, the conservation law is presented in 1D:

$$U_t + F^C(U)_x - F^D(U)_x = 0$$

Source terms are usually treated using a cyclic operator splitting. First the homogenous system is solved separately:

$$\left.\begin{aligned} U_t + F^C(U)_x &= 0 \\ U(x,t^n) &= U^n \end{aligned}\right\} \Rightarrow U^{n+1,*}$$

Then in a second step the source term is treated using the intermediate solution of the first step as initial condition

$$\left.\begin{aligned} U_t - F^D(U)_x &= 0 \\ U(x,t^n) &= U^{n+1,*} \end{aligned}\right\} \Rightarrow U^{n+1}$$

Numerical Algorithm (2)

In the following timestep the procedure is repeated in in opposite order:

$$\left.\begin{aligned} U_t - F^D(U)_x &= 0 \\ U(x,t^n) &= U^{n+1} \end{aligned}\right\} \Rightarrow U^{n+2,*}$$

$$\left.\begin{aligned} U_t + F^C(U)_x &= 0 \\ U(x,t^n) &= U^{n+2,*} \end{aligned}\right\} \Rightarrow U^{n+2}$$

The cycling procedure ensures preservation of 2nd order accuracy.

Time Discretisation (1)

Explicit methods are computationally less costly and therefore well-suited for non-stationary problems where small and therefore many timesteps are required in order to resolve the physical phenomena.

Implicit methods have the advantage that in theory there is no stability limit which theoretically allows us to choose the timestep size arbitrarily. While large timesteps are suitable to reach the steady-state solution quickly, for unsteady problems the timestep needs to be small enough in order to resolve the physical phenomena.

Implicit methods are, compared to explicit methods, computationally extremely costly since they require the solution of a linear equation system with a high number of unknowns. Should the equations be nonlinear, which is the case here, they also need to be linearized in order to obtain a linear equation system.

Time Discretisation (2)

The convective part of the equations reduces to the Euler Equations, which, in most cases, can be treated explicitly in an efficient manner. In case of a steady-state solution it would be advantageous to apply an implicit discretisation since there are no small-scale phenomena which need to be resolved.

The nonconvective part of the equations proves to have big impact on stability resulting in a very restrictive timestep size for explicit methods due to its parabolic nature. An explicit time discretisation would cause extremely small timesteps resulting in a high computation time which makes an implicit time discretisation the method of choice.

Time Discretisation (3)

In the present case both methods are combined. The explicit timestep limit for the convective part ensures resolution of small-scale convective physical phenomena which is well-suited for unsteady flows. The nonconvective part is discretized implicitly in order to use the explicit timestep size with no destabilizing effects.

Time Discretisation (4)

In general the timeupdate of a 2D finite volume scheme can be formulated as follows:

$$U^{n+1} = U^n - \Delta t\left(\frac{g_{i+\frac{1}{2}} - g_{i-\frac{1}{2}}}{\Delta x} + \frac{h_{i+\frac{1}{2}} - h_{i-\frac{1}{2}}}{\Delta y}\right)$$

This can be rewritten to

$$U^{n+1} = U^n - \Delta t \cdot R\left(U^n\right) \leftrightarrow \delta U^n = -\Delta t \cdot R\left(U^n\right)$$

resulting in the implicit scheme

$$U^{n+1} = U^n - \Delta t\left(\theta R\left(U^{n+1}\right) + (1-\theta) R\left(U^n\right)\right)$$

Time Discretisation (5)

In order to obtain a linear equation system, $Z(U^{n+1})$ needs to be linearized:

$$R\left(U^{n+1}\right) \approx R\left(U^n\right) + \left.\frac{\partial R}{\partial U}\right|^n \delta U^n$$

This yields

$$\delta U^n = -\Delta t\left(\theta\left(R\left(U^n\right) + \left.\frac{\partial R}{\partial U}\right|^n \delta U^n\right) + (1-\theta) R\left(U^n\right)\right)$$

$$\rightarrow \delta U^n = -\Delta t\left(\theta \left.\frac{\partial R}{\partial U}\right|^n \delta U^n + R\left(U^n\right)\right)$$

$$\rightarrow \left(\frac{I}{\Delta t} + \theta \left.\frac{\partial R}{\partial U}\right|^n\right)\delta U^n = -R\left(U^n\right)$$

Time Discretisation (5)

With $\theta = 1$ we obtain the so-called fully implicit scheme, which is first order accurate while with $\theta = 0.5$ we obtain the Crank-Nicholson scheme, which is second order accurate. $\theta = 0$ would yield an explicit scheme.

The resulting equation system is solved for δU^n. With the solution we obtain the timeupdate

$$U^{n+1} = U^n + \delta U^n$$

Spatial Discretisation (1)

For the viscous terms a central differences scheme is used as a spatial discretisation. While this may seem to be a violation of the finite volume idea, it can actually be shown that the central differences scheme is equivalent to a finite volume scheme.

With the numerical flux

$$g_{i+\frac{1}{2}}^{num} = \frac{1}{2}\left(f_1(U_{i+1}) + f_1(U_i)\right)$$

We obtain the timeupdate for a 1D scheme

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}\left(g_{i+\frac{1}{2}}^{num} - g_{i-\frac{1}{2}}^{num}\right)$$

Spatial Discretisation (2)

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{2\Delta x}\left(\left(f_1(U_{i+1}^n) + f_1(U_i^n)\right) - \left(f_1(U_i^n) + f_1(U_{i-1}^n)\right)\right)$$

$$\rightarrow U_i^{n+1} = U_i^n - \frac{\Delta t}{2\Delta x}\left(f_1(U_{i+1}^n) - f_1(U_{i-1}^n)\right)$$

which is identical to a central differences formualtion. The essential ingredient is the flux formulation of the scheme. This ensures the conservation properties. The difference to other flux calculations is that other schemes (e.g. the ROE scheme) have a more sophisticated way of computing the intercell flux.

While the central differences scheme is unconditionally unstable for an explicit time discretisation it is stable for an implicit time discretisation making it a suitable approach for both the convective and the nonconvective part of the equations.

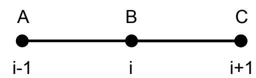
The Complete Scheme

The complete scheme reads:

$$\frac{I}{\Delta t} -$$

Setting up the Equation System (1)

On a structured 1D-Grid the stencil would look like



Giving the equation system for 8 grid points in x-direction the shape

$$\begin{bmatrix} B & C & & & \cdots & & & 0 \\ A & B & C & & & & & \\ & A & B & C & & & & \\ & & A & B & C & & & \vdots \\ \vdots & & & A & B & C & & \\ & & & & A & B & C & \\ & & & & & A & B & C \\ 0 & & & \cdots & & & A & B \end{bmatrix} \begin{bmatrix} \delta U_1^n \\ \delta U_2^n \\ \delta U_3^n \\ \delta U_4^n \\ \delta U_5^n \\ \delta U_6^n \\ \delta U_7^n \\ \delta U_8^n \end{bmatrix} = \begin{bmatrix} R_1^n \\ R_2^n \\ R_3^n \\ R_4^n \\ R_5^n \\ R_6^n \\ R_7^n \\ R_8^n \end{bmatrix}$$

Solving the Equation System (1)

We obtain a block-tridiagonal matrix that can be solved by i.e. using an iterative method like the LU-SSOR scheme.

The scheme is based on a splitting of the system matrix (S) into three separate matrices, one containing the diagonal elements (B), one containing the elements under the diagonal (A) and one containing the elements above the diagonal (C).

$$(B+A)B^{-1}(B+C)\delta U^n = -R^n$$

This yields the following rule for the iteration steps:

$$\delta U^{p+1} = -(B+A)^{-1} C\,\delta U^p + (B+A)^{-1} R^n$$

Solving the Equation System (2)

The component-wise formulation is:

$$\delta U_k^{p+1} = \frac{1}{s_{kk}}\left(R_k^n - \sum_{j=1}^{k-1} s_{kj}\,\delta U_j^{p+1} - \sum_{j=k+1}^{k_{\max}} s_{kj}\,\delta U_j^{p} \right)$$

Making use of the knowledge of the system matrix‘ shape we can optimize the above algorithm signigficantly by only treating the nonzero elements of the system matrix.

The algorithm is stopped when the residual drops below a pre-defined lower limit or the iteration count reaches an upper limit.

The ADI Scheme

Having derived the scheme for the one-dimensional case we now extend it to the second dimension in space. This can either be done directly by formulating the equation system in 2D or by simply using the ADI scheme

$$1.\ \left\{ I + \theta\delta t\left[\frac{\partial}{\partial x}\left(-P + R_x\right)^n - \frac{\partial^2}{\partial x^2}(R)^n \right] \right\}\delta U^*$$

$$= \delta t\left[\frac{\partial}{\partial x}(V_1+V_2)^n + \frac{\partial}{\partial y}(W_1+W_2)^n \right] + \Theta\delta t\left[\frac{\partial}{\partial x}(\delta V_2)^{n-1} + \frac{\partial}{\partial y}(\delta W_1)^{n-1} \right]$$

$$2.\ \left\{ I + \theta\delta t\left[\frac{\partial}{\partial y}\left(-Q + S_y\right)^n - \frac{\partial^2}{\partial y^2}(S)^n \right] \right\}\delta U^n = \delta U^*$$

Boundary Conditions

Boundary conditions need to be set at all boundaries. The most common boundary types are:

- Walls
- Inflow
- Outflow
- Periodic

Apart from walls the boundary conditions are handled the same way they are handled for the Euler Equations.

Adiabatic Walls (1)

boundary cells

$\begin{pmatrix}\rho_{i,j}\\ u_{i,j}\\ v_{i,j}\\ p_{i,j}\end{pmatrix}$ i,j	$\begin{pmatrix}\rho_{i+1,j}\\ u_{i+1,j}\\ v_{i+1,j}\\ p_{i+1,j}\end{pmatrix}$ i+1,j
$\begin{pmatrix}\rho_{i,j}\\ 2u_{wall}-u_{i,j}\\ -v_{i,j}\\ p_{i,j}\end{pmatrix}$ i,j-1	$\begin{pmatrix}\rho_{i+1,j}\\ 2u_{wall}-u_{i+1,j}\\ -v_{i+1,j}\\ p_{i+1,j}\end{pmatrix}$ i+1,j-1

u_{wall}

wall ghost cells

The normal momentum at a wall needs to be zero so that a wall does not apply any force onto the flow. The most simple approach, which is to simply extrapolate the pressure, is shown here.

Adiabatic Walls (2)

boundary cells

$\begin{pmatrix}\rho_{i,j}\\ u_{i,j}\\ v_{i,j}\\ p_{i,j}\end{pmatrix}$ i,j	$\begin{pmatrix}\rho_{i+1,j}\\ u_{i+1,j}\\ v_{i+1,j}\\ p_{i+1,j}\end{pmatrix}$ i+1,j
$\begin{pmatrix}\rho_{i,j}\\ 2u_{wall}-u_{i,j}\\ -v_{i,j}\\ p^*\left(U_{i,j-\frac{1}{2}}\right)\end{pmatrix}$ i,j-1	$\begin{pmatrix}\rho_{i+1,j}\\ 2u_{wall}-u_{i+1,j}\\ -v_{i+1,j}\\ p^*\left(U_{i+1,j-\frac{1}{2}}\right)\end{pmatrix}$ i+1,j-1

u_{wall}

wall ghost cells

A more advanced approach would be to use the normal momentum in order to calculate the pressure:

$$p^* = p_i - \Delta y \times \left[\frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial y}\right) - \frac{\partial}{\partial y}\left(\frac{2}{3}\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{4}{3}\mu\frac{\partial v}{\partial y}\right)\right]$$

Inflow

All ghost cell values are prescribed:

inflow ghost cells	boundary cells
$\begin{pmatrix}\rho_{\infty}\\ u_{\infty}\\ v_{\infty}\\ p_{\infty}\end{pmatrix}$ i-1,j	$\begin{pmatrix}\rho_{i,j}\\ u_{i,j}\\ v_{i,j}\\ p_{i,j}\end{pmatrix}$ i,j
$\begin{pmatrix}\rho_{\infty}\\ u_{\infty}\\ v_{\infty}\\ p_{\infty}\end{pmatrix}$ i-1j-1	$\begin{pmatrix}\rho_{i,j-1}\\ u_{i,j-1}\\ v_{i,j-1}\\ p_{i,j-1}\end{pmatrix}$ i,j-1

Outflow

All ghost cell values are extrapolated:

boundary cells	outflow ghost cells
$\begin{pmatrix}\rho_{i,j}\\ u_{i,j}\\ v_{i,j}\\ p_{i,j}\end{pmatrix}$ i,j	$\begin{pmatrix}\rho_{i,j}\\ u_{i,j}\\ v_{i,j}\\ p_{i,j}\end{pmatrix}$ i+1,j
$\begin{pmatrix}\rho_{i,j-1}\\ u_{i,j-1}\\ v_{i,j-1}\\ p_{i,j-1}\end{pmatrix}$ i,j-1	$\begin{pmatrix}\rho_{i,j-1}\\ u_{i,j-1}\\ v_{i,j-1}\\ p_{i,j-1}\end{pmatrix}$ i+1,j-1

Problems can arise from the extrapolation of the pressure. More advanced methods would enhance the results at the boundary significantly.

Periodic Boundary Conditions

left ghost cells	boundary cells		boundary cells	right ghost cells
$\begin{pmatrix}\rho_{I_{\max},j}\\ u_{I_{\max},j}\\ v_{I_{\max},j}\\ p_{I_{\max},j}\end{pmatrix}$ 0,j	$\begin{pmatrix}\rho_{1,j}\\ u_{1,j}\\ v_{1,j}\\ p_{1,j}\end{pmatrix}$ 1,j	....	$\begin{pmatrix}\rho_{I_{\max},j}\\ u_{I_{\max},j}\\ v_{I_{\max},j}\\ p_{I_{\max},j}\end{pmatrix}$ I_{max},j	$\begin{pmatrix}\rho_{1,j}\\ u_{1,j}\\ v_{1,j}\\ p_{1,j}\end{pmatrix}$ I_{max}+1,j
$\begin{pmatrix}\rho_{I_{\max},j-1}\\ u_{I_{\max},j-1}\\ v_{I_{\max},j-1}\\ p_{I_{\max},j-1}\end{pmatrix}$ 0,j-1	$\begin{pmatrix}\rho_{1,j-1}\\ u_{1,j-1}\\ v_{1,j-1}\\ p_{1,j-1}\end{pmatrix}$ 1,j-1	....	$\begin{pmatrix}\rho_{I_{\max},j-1}\\ u_{I_{\max},j-1}\\ v_{I_{\max},j-1}\\ p_{I_{\max},j-1}\end{pmatrix}$ I_{max},j-1	$\begin{pmatrix}\rho_{1,j-1}\\ u_{1,j-1}\\ v_{1,j-1}\\ p_{1,j-1}\end{pmatrix}$ I_{max}+1,j-1

Some Examples for the Equation System (1)

Periodic Boundary Conditions

$$U_0 = U_{I_{\max}};\quad U_{I_{\max}+1} = U_1$$

$$\begin{bmatrix} M_1 & N_2 & & & L_0 \\ L_1 & M_2 & N_3 & & \\ & \ddots & \ddots & \ddots & \\ & & L_{I_{\max}-2} & M_{I_{\max}-1} & N_{I_{\max}} \\ N_{I_{\max}+1} & & & L_{I_{\max}-1} & M_{I_{\max}} \end{bmatrix} \begin{bmatrix} \delta U_1^{n+1,*} \\ \delta U_2^{n+1,*} \\ \vdots \\ \delta U_{I_{\max}-1}^{n+1,*} \\ \delta U_{I_{\max}}^{n+1,*} \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{I_{\max}-1} \\ H_{I_{\max}} \end{bmatrix}$$

Some Examples for the Equation System (2)

Inflow (left) and Outflow (right) Boundary Conditions

$$U_0 = U_\infty;\quad U_{I_{\max}+1} = U_{I_{\max}}$$

$$\begin{bmatrix} M_1 & N_2 & & & \\ L_1 & M_2 & N_3 & & \\ & \ddots & \ddots & \ddots & \\ & & L_{I_{\max}-2} & M_{I_{\max}-1} & N_{I_{\max}} \\ & & & L_{I_{\max}-1} & M_{I_{\max}} + N_{I_{\max}+1} \end{bmatrix} \begin{bmatrix} \delta U_1^{n+1,*} \\ \delta U_2^{n+1,*} \\ \vdots \\ \delta U_{I_{\max}-1}^{n+1,*} \\ \delta U_{I_{\max}}^{n+1,*} \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{I_{\max}-1} \\ H_{I_{\max}} \end{bmatrix}$$

Some Examples for the Equation System (3)

Wall Boundary Conditions (left and right)

$$\begin{bmatrix} (M+C)_1 & N_2 & & & \\ L_1 & M_2 & N_3 & & \\ & \ddots & \ddots & \ddots & \\ & & L_{I_{\max}-2} & M_{I_{\max}-1} & N_{I_{\max}} \\ & & & L_{I_{\max}-1} & (M+C)_{I_{\max}} \end{bmatrix} \begin{bmatrix} \delta U_1^{n+1,*} \\ \delta U_2^{n+1,*} \\ \vdots \\ \delta U_{I_{\max}-1}^{n+1,*} \\ \delta U_{I_{\max}}^{n+1,*} \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{I_{\max}-1} \\ H_{I_{\max}} \end{bmatrix}$$

with C being a matrix that maps the boundary cell onto the wall ghost cell according to the rule for adiabatic wall ghost cells.