

# Computational Fluid Dynamics

---

**Jacek Rokicki**

10 September 2014

Copyright © 2014, Jacek Rokicki



**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI

**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



## Table of content

1	Introduction.....	3
2	Conservation Laws for the continuous medium .....	4
2.1	Conservation of mass .....	5
2.2	Conservation (evolution) of momentum.....	5
2.3	Conservation (evolution) of energy.....	6
2.4	Reynolds transport theorem .....	7
2.5	Application to the conservation principles .....	7
2.6	The integral vs. differential form.....	7
2.7	Constitutive equations .....	9
2.7.1	Stress tensor .....	9
2.7.2	Fourier law.....	9
3	Boundary conditions .....	11
4	Initial conditions .....	12
5	The Navier-Stokes equations for compressible fluid .....	13
6	Navier-Stokes equations for incompressible fluid .....	15
7	Model problems .....	16
8	Discretisation methods.....	17
9	Finite Difference method .....	18
9.1	Consistency and the order of accuracy .....	19
9.2	Generation of finite difference formulas .....	20
9.2.1	Second-order derivative .....	20
9.2.2	Asymmetric first-order derivative .....	21
10	1D Boundary Value Problem .....	22
11	Diagonally dominant matrices.....	25
12	Properties of the 1D Boundary Value Problems .....	27
13	2D and 3D Boundary Value Problem.....	30
14	Consequences for the Navier-Stokes equations .....	34
15	Iterative methods to solve the large linear systems .....	35
15.1	Jacobi iterative algorithm.....	36
15.2	Gauss-Seidel iterative algorithm .....	37
15.3	Specialisation for sparse matrices.....	37
15.4	Direct iterations to solve nonlinear equations.....	37

- 16 Error of the approximate solution..... 39
- 17 Hyperbolic advection equation - initial value problem ..... 40
- 18 Parabolic equation - initial value problem ..... 41
- 19 Discretisation of the advection equation ..... 42
  - 19.1 Explicit Euler formula ..... 42
  - 19.2 Lax theorem..... 43
  - 19.3 One sided formulas ..... 43
  - 19.4 Implicit discretisation ..... 44
  - 19.5 Lax-Friedrichs discretisation ..... 45
  - 19.6 Higher-order discretisations..... 45
    - 19.6.1 The Lax-Wendroff discretisation ..... 45
    - 19.6.2 The Beam Warming discretisation ..... 45
- 20 Discretisation of the 1D parabolic equation..... 46
  - 20.1 Explicit Euler formula ..... 46
  - 20.2 Implicit Euler formula ..... 46
  - 20.3 Crank-Nicolson formula..... 47
- Annex A. Tridiagonal matrix algorithm ..... 48

## 1 Introduction

This book originates from the lectures of Computational Fluid Mechanics held since 1995, for the undergraduate and graduate students, at the Faculty of Power and Aeronautical Engineering of Warsaw University of Technology.

## 2 Conservation Laws for the continuous medium

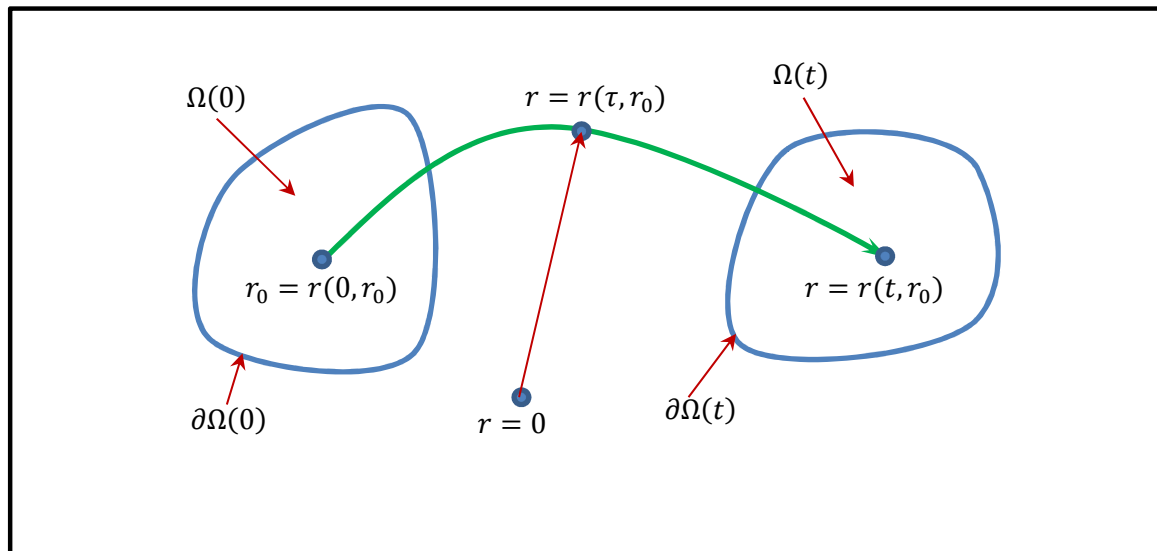


Figure 1, Evolution of the fluid region in time

In this chapter we will attempt to formulate general equation describing the general motion of the arbitrary continuous medium (which includes but is not limited to the classical fluids). This forms the subject in more advance course of Fluid Mechanics and therefore will be recalled here only shortly, without repeating the general information about continuum as opposed to molecular medium.

To describe the motion of the continuous medium, we consider that this medium fills some region  $X \in \mathbb{R}^3$ . We will be in particular interested what happens to a particular but otherwise arbitrary subdomain containing this medium consisting of selected fluid particles, which at time  $t = 0$  occupy the region  $\Omega_0$  (see Figure 1). The evolution of this fluid region in time is a subject of our study.

The position  $r(t, r_0) \in \mathbb{R}^3$  of each fluid particle in time is described, by its position at  $t = 0$  (e.g.,  $r_0$ ) and by the elapsed time  $t$ . Similarly the fluid region and its boundary at time  $t$  are described as  $\Omega(t)$  and  $\partial\Omega(t)$  respectively.

The most important observation now, is the fact that the fluid region consisting of always the same fluid particles, has to adhere to the conservation laws. In particular conservation of mass, momentum and energy (for the latter two it will be more evolution than strict conservation).

Therefore we should now calculate mass, momentum and energy of the medium present in this fluid region at the time  $t$ . We will assume that each fluid particle within  $\Omega(t)$  is characterised individually by the values of density, velocity and the total energy per unit mass  $\rho(t, r), V(t, r), e_t(t, r)$  respectively – see Figure 2 (the values of which are at the moment unknown). Other secondary local properties of the fluid can also be named, e.g., internal energy per unit mass  $e(t, r)$ , temperature  $T(t, r)$ , pressure  $p(t, r)$ , etc..

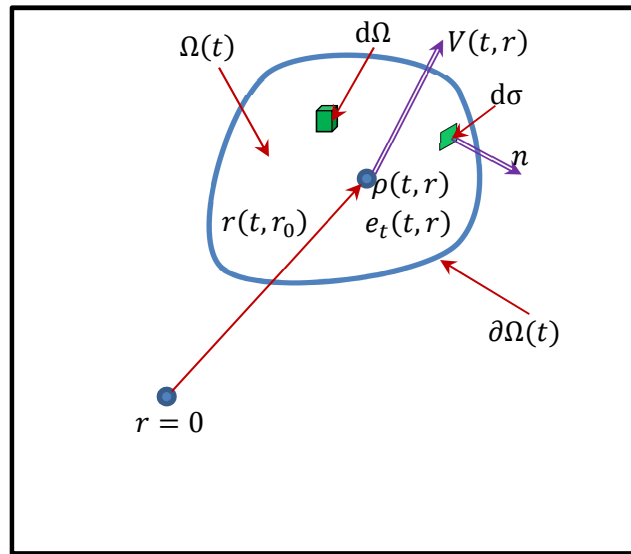


Figure 2, Local and global quantities

Now it is possible to calculate the Mass, Momentum and Energy of the fluid region  $\Omega(t)$ :

$$\begin{aligned} \text{MASS}(t) &= \int_{\Omega(t)} \rho(t, r) d\Omega \\ \text{MOMENTUM}(t) &= \int_{\Omega(t)} V(t, r) \rho(t, r) d\Omega \\ \text{ENERGY}(t) &= \int_{\Omega(t)} e_t(t, r) \rho(t, r) d\Omega = \int_{\Omega(t)} \rho e + \rho \frac{|V|^2}{2} d\Omega \end{aligned} \quad (1)$$

where  $e(t, r)$  denotes internal energy.

## 2.1 Conservation of mass

The mass of fluid region consisting of the same fluid particles does not change in time (its time derivative is zero):

$$\frac{d}{dt} \text{MASS}(t) \equiv \frac{d}{dt} \int_{\Omega(t)} \rho(t, r) d\Omega = 0 \quad (2)$$

## 2.2 Conservation (evolution) of momentum

The momentum of the fluid region is not strictly conserved, but evolves under the influence of forces (see Figure 3) acting on each fluid particle (volume forces) as well as acting on the surface of  $\Omega(t)$  (surface forces):

$$\frac{d}{dt} \text{MOMENTUM}(t) = \left\{ \begin{array}{l} \text{VOLUME} \\ \text{FORCES} \end{array} \right\} + \left\{ \begin{array}{l} \text{SURFACE} \\ \text{FORCES} \end{array} \right\} \quad (3)$$

This can be expressed in the differential form as:

$$\frac{d}{dt} \int_{\Omega(t)} V(t, r) \rho(t, r) d\Omega = \int_{\Omega(t)} f_v(t, r) \rho(t, r) d\Omega + \int_{\partial\Omega(t)} f_s(t, r, n) d\sigma \quad (4)$$

where  $f_v(t, r)$  and  $f_s(t, r, n)$  denotes volume and surface forces respectively.

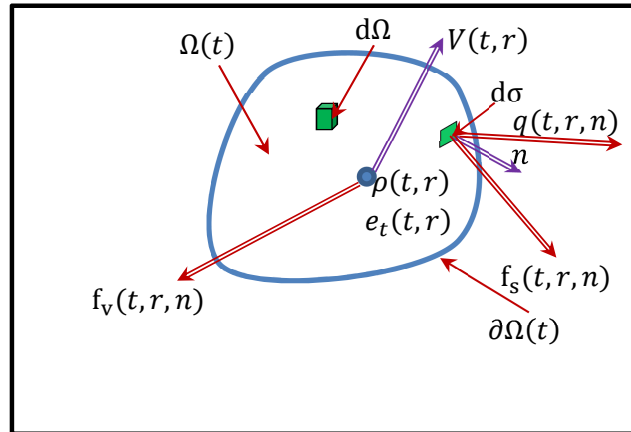


Figure 3, Forces and the heat flux

The volume forces are often well known (e.g., gravity), but may also be a part of the solution (e.g., electromagnetic field in the plasma flows). The surface forces, however, are as a rule a part of the solution and are unknown beforehand. One should note that the surface forces apart from position and time depend also on the orientation of the surface (i.e., direction of the normal vector  $n$ ).

Fortunately following Cauchy theorem we now more about this dependence. This theorem states that the surface force must be a linear function of the normal vector:

$$f_s = \mathbb{T}(t, r) \cdot n$$

where,  $\mathbb{T}$  denotes the stress tensor

$$\mathbb{T} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \quad (5)$$

### 2.3 Conservation (evolution) of energy

Similarly as for momentum the energy of the fluid region is not strictly conserved, but evolves under the influence of power of forces acting on each fluid particle (volume forces) as well as acting on the surface of  $\partial\Omega(t)$  (surface forces), the energy is also increased/decreased through the heat conduction and via the heat sources.

$$\frac{d}{dt} \text{ENERGY} = \left\{ \begin{array}{l} \text{POWER OF} \\ \text{VOLUME} \\ \text{FORCES} \end{array} \right\} + \left\{ \begin{array}{l} \text{POWER OF} \\ \text{SURFACE} \\ \text{FORCES} \end{array} \right\} - \left\{ \begin{array}{l} \text{HEAT FLUX} \\ \text{THROUGH} \\ \text{THE SURFACE} \end{array} \right\} + \left\{ \begin{array}{l} \text{POWER OF} \\ \text{HEAT} \\ \text{SOURCES} \end{array} \right\} \quad (6)$$

This can be expressed in the differential form (skipping the function arguments) as:

$$\frac{d}{dt} \int_{\Omega(t)} e_t \rho d\Omega = \int_{\Omega(t)} f_v \cdot V \rho d\Omega + \int_{\partial\Omega(t)} V \cdot f_s(t, r, n) - q \cdot n d\sigma + \int_{\Omega(t)} Q_s d\Omega \quad (7)$$

where  $q$  denotes a vector heat flux through the surface and  $Q_s$  is a local heat source.

In order to make the next step we need to learn how to differentiate integrals in which the fluid region depends on time.

## 2.4 Reynolds transport theorem

Suppose now, we have an arbitrary scalar function  $F(t, r)$  integrated over  $\Omega(t)$ , which we want to differentiate with respect to  $t$ ; the theorem below tells us how to do it:

$$\frac{d}{dt} \int_{\Omega(t)} F(t, r) d\Omega = \int_{\Omega(t)} \frac{\partial F}{\partial t} + \text{div}(FV) d\Omega = \int_{\Omega(t)} \frac{\partial F}{\partial t} d\Omega + \int_{\partial\Omega(t)} F V n d\Omega \quad (8)$$

The second equality is based on the Gauss theorem.

The proof of this theorem can be found in the literature.

## 2.5 Application to the conservation principles

The mass conservation principle can be now expressed as:

$$\frac{d}{dt} \int_{\Omega(t)} \rho d\Omega = 0 \Rightarrow \int_{\Omega(t)} \frac{\partial \rho}{\partial t} + \text{div}(\rho V) d\Omega = 0 \quad (9)$$

The momentum conservation/evolution principle is now:

$$\int_{\Omega(t)} \frac{\partial \rho V}{\partial t} + \text{Div}(\rho V \otimes V) d\Omega = \int_{\Omega(t)} f_v \rho + \text{Div}(\mathbb{T}) d\Omega \quad (10)$$

where the Reynolds transport theorem is used for the vector function rather than for the scalar one, and the tensor divergence is defined as follows:

$$\rho V \otimes V = \rho \begin{bmatrix} uu & uv & uw \\ vu & vv & vw \\ wu & wv & ww \end{bmatrix}, \quad \text{Div}(\rho V \otimes V) \stackrel{\text{def}}{=} \begin{bmatrix} \text{div}(\rho u V) \\ \text{div}(\rho v V) \\ \text{div}(\rho w V) \end{bmatrix}, \quad V \stackrel{\text{def}}{=} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (11)$$

The energy conservation/evolution principle can be expressed now as:

$$\int_{\Omega(t)} \frac{\partial e_t \rho}{\partial t} + \text{div}(e_t \rho V) d\Omega = \int_{\Omega(t)} f_v \cdot V \rho + \text{div}(\mathbb{T}V) - \text{div}(q) + Q_s d\Omega \quad (12)$$

## 2.6 The integral vs. differential form

We have obtained the previous equations (for conservation of mass, momentum and energy) in the form:



$$\int_{\Omega} F(r) d\Omega = 0 \quad (13)$$

where  $\Omega$  is an arbitrary subdomain within the full/total domain  $\Omega_{\text{total}}$  (filled by a fluid or by a general continuous medium). In such a case also the function itself has to vanish:

$$F(r) = 0 \quad (14)$$

This is a well-known theorem in a standard calculus.

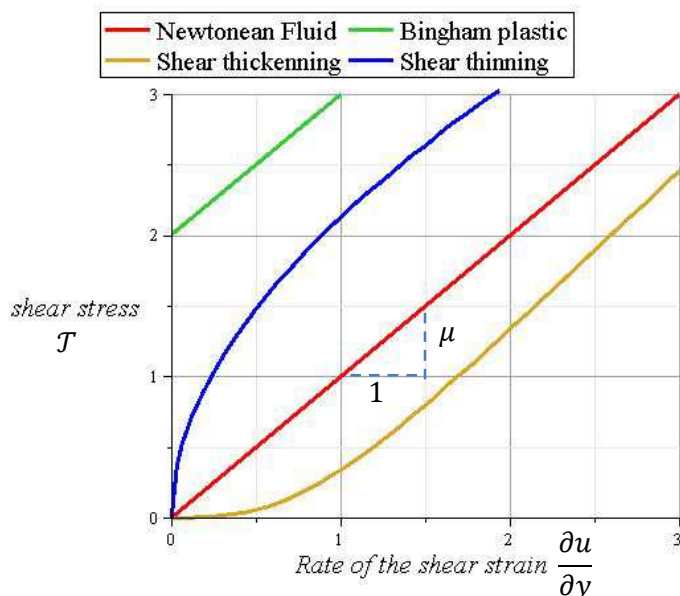
## 2.7 Constitutive equations

### 2.7.1 Stress tensor

The equations (9) (10) (12) form the integral system describing the motion of the general continuous medium (i.e., solid, fluid, plasma, viscoelastic medium, etc.). In order to specify it for the particular fluid type, the stress tensor  $\mathbb{T}$  has to be expressed through the value of some other quantity related, e.g., to the kinematics (through the linear or nonlinear formula).

The simplest linear Newtonian model valid for air, water and many other gases and liquids, assumes that the shear stress tensor  $\tau$  depends linearly on the deformation velocity tensor:

$$\begin{aligned}\mathbb{T} &= -p\mathbb{I} + \tau = -p\mathbb{I} + 2\mu \left( \mathbb{D} - \frac{1}{3} \operatorname{div} V \right) \\ \mathbb{D} &= \frac{1}{2} (\nabla V + \nabla V^T) = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)\end{aligned}\tag{15}$$



Many nonlinear fluids exist which deviate from the Newton model, nevertheless two most important air and water follow the linear Newton model.

### 2.7.2 Fourier law

There exist also constitutive equation (Fourier law) relating in linear manner the heat flux  $q$ , to the temperature gradient  $\nabla T$

$$q = -\lambda \nabla T\tag{16}$$

The material coefficients  $\mu$  (dynamic viscosity) and  $\lambda$  (thermal conductivity) are of the same nature, as they describe transversal transport of momentum/energy (this fact is well documented in the

statistical mechanics). Therefore both phenomena should be included at the same time into consideration.

Both coefficients may depend on temperature and to lesser extent also on pressure:

$$\mu(T, p), \lambda(T, p) \quad (17)$$

In many practical computations (in which temperature does not change significantly) both coefficients are regarded as constant.

### 3 Boundary conditions

The simplest boundary conditions can be prescribed at the solid wall  $\Gamma$ , for the velocity field  $V(r, t)$ :

$$V(r, t)|_{\Gamma} = V_B(r, t) \quad (18)$$

where  $V_B(r, t)$  denotes a prescribed velocity at the boundary. In case the boundary is not permeable  $V_B(r, t) \cdot n = 0$ , while if in addition it does not move in tangential direction  $V_B(r, t) \equiv 0$  ( $n$  denotes here the external normal vector). This is called a non-slip boundary condition and is appropriate for all viscous fluids on physical boundaries.

Other boundary conditions are appropriate for symmetry planes, for inlets and outlets, and for the so called *far-field*.

Similar boundary conditions are prescribed for the temperature field  $T(r, t)$ :

$$T(r, t)|_{\Gamma} = T_B(r, t) \quad (19)$$

where again  $T_B(r, t)$  is a prescribed temperature at the boundary. This boundary condition can be alternatively formulated for the heat flux:

$$\lambda \frac{\partial T(r, t)}{\partial n} \Big|_{\Gamma} = q_n(r, t) \quad (20)$$

where  $\lambda$  is a coefficient of thermal conductivity, while  $q_n(r, t)$  stands for the prescribed (normal) heat flux at the boundary  $\Gamma$ . In many physical situations, for simplicity, the adiabatic wall is assumed, for which  $q_n(r, t) = 0$  (this is a reasonable assumption for aeronautic fast external flows, for which it is expected, that the boundary's temperature will eventually become equal to that of the neighbouring fluid).

Very often only stationary solutions to the Navier-Stokes equations are sought, as is the case for the flow around the aircraft flying with a constant velocity. Such stationary solutions are found using specialised algorithms in which so-called marching in pseudo-time is used numerically. These algorithms are much cheaper than those, which employ true time resolved computations.

In such cases (when stationary solution is expected to exist) it is very important that, the boundary conditions fulfil certain integral constraints expressing the general conservation principles. For example, the mass conservation implies that the total mass flux through the boundary has to be equal to zero:

$$\int_{\partial\Omega_{total}} \rho V_B(r, t) \cdot n \, d\sigma \equiv 0 \quad (21)$$

## 4 Initial conditions

Simulation of time-dependent phenomena require providing also initial conditions to start the computations. These, for compressible flows, usually/often are the velocity, the pressure and the temperature:

$$\begin{aligned}V(r, t = 0) &= V_{init}(r) \\p(r, t = 0) &= p_{init}(r) \\T(r, t = 0) &= T_{init}(r) \\&\text{for all } r \in \Omega_{total}\end{aligned}\tag{22}$$

For incompressible flows it sufficient to provide the velocity field (additionally the Temperature field may be required if the viscosity coefficient depends on temperature). One needs to remember that the initial velocity field needs to remain divergence-free  $\text{div}[V_{init}(r)] = 0$ .

## 5 The Navier-Stokes equations for compressible fluid

The Navier-Stokes equation for compressible medium are best presented now in the unified manner which underlines their conservative structure.

$$\frac{\partial U}{\partial t} + \text{Div } F_c(U) = \text{Div } F_v(U, \nabla U) \quad (23)$$

where:

$$U = \begin{bmatrix} \rho \\ \rho V \\ \rho E \end{bmatrix} \in \mathbb{R}^5, \quad V = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \in \mathbb{R}^3, \quad F_c(U) = \begin{bmatrix} \rho V^T \\ \rho V V^T + p I_{3 \times 3} \\ (\rho E + p) V^T \end{bmatrix}_{5 \times 3} \quad (24)$$

In the above  $U_{5 \times 1}$  stands for the composite conservative variable,  $\rho$  is the density,  $V_{3 \times 1} = [u, v, w]^T$  denotes a velocity vector,  $E$  is a total energy per unit mass,  $p$  stands for pressure, while  $H = E + \frac{p}{\rho}$  is the total enthalpy per unit mass. The  $F_c(U)$  and  $F_v(U, \nabla U)$  stand for the convective and viscous fluxes respectively. The viscous flux can be expressed as:

$$F_v(U, \nabla U) = \begin{bmatrix} 0 \\ \tau_{3 \times 3} \\ V^T \tau_{3 \times 3} - q^T \end{bmatrix}_{5 \times 3} \quad (25)$$

where  $q$  stands for a heat flux, while  $\tau$  denotes the stress tensor. Both these quantities in Fluid Mechanics (and especially the stress tensor) can be defined via very different formulas, in particular when modelling of turbulence is attempted. Nevertheless for simple Newtonian, linear fluid they are defined/calculated as:

$$\begin{aligned} \tau_{3 \times 3} &= \mu \left[ -\frac{2}{3} (\nabla \cdot V) I_{3 \times 3} + (\nabla V)^T + \nabla V \right] \\ q &= -\lambda \nabla T \end{aligned} \quad (26)$$

where  $\mu$  and  $\lambda$  stand for coefficients of dynamic viscosity and thermal conductivity respectively. It is good to remember that for air and water  $\frac{\mu}{\rho}$  is very small and equals  $\sim 10^{-5} \frac{m^2}{s}$  and  $\sim 10^{-6} \frac{m^2}{s}$  respectively. This is the reason why in the Navier-Stokes equation the convective flux plays, in a sense, more important role than the viscous flux (at least for higher Reynolds numbers).

In addition, for perfect gas, the equation of state is assumed in the usual form:  $\frac{p}{\rho} = RT$  (where  $T$  stands for temperature, while  $R$  is an ideal gas constant).

The tensor divergence operator  $\text{Div}$  present in (23) can be expressed for clarity in an extended form as:

$$\text{Div } F_c(U) = \begin{bmatrix} \text{div}(\rho V) \\ \text{div}(\rho u V) + \frac{\partial p}{\partial x} \\ \text{div}(\rho v V) + \frac{\partial p}{\partial y} \\ \text{div}(\rho w V) + \frac{\partial p}{\partial z} \\ \text{div}(\rho H V) \end{bmatrix}_{5 \times 1} \tag{27}$$

where div denotes a usual scalar divergence operator acting on vector functions.

## 6 Navier-Stokes equations for incompressible fluid

The Navier-Stokes equations for incompressible fluid ( $\rho = \text{const}$ ) differ significantly, in their mathematical structure, from their compressible counterpart. They no longer have the form of pure evolution PDE, but include the constraint in the form of the continuity equation:

$$\text{div } V = 0 \quad (28)$$

This constraint is supplemented by the usual momentum equation in which time derivative of velocity is present, and can be used to progress the computations to the next time level:

$$\frac{\partial V}{\partial t} + \text{Div} \left[ VV^T + \frac{p}{\rho} I_{3 \times 3} \right] = \text{Div} \left[ \frac{\mu}{\rho} \left[ -\frac{2}{3} (\nabla \cdot V) I_{3 \times 3} + (\nabla V)^T + \nabla V \right] \right] \quad (29)$$

The additional energy equation is needed only if the viscosity coefficient  $\mu$  is a function of the temperature. Otherwise both velocity and pressure are fully determined by (29) and the suitable initial and boundary conditions.



## 7 Model problems

In order to better understand the principles of discretisation, various model problems will be considered in the next Sections, including:

1. 1D elliptic problem	$\begin{cases} \frac{d^2u}{dx^2} = f(x) \\ u(a) = u_a \\ u(b) = u_b \end{cases}$ <p>or for more complex equations:</p> $\begin{cases} p(x)\frac{d^2u}{dx^2} + q(x)\frac{du}{dx} + r(x)u = f(x) \\ u(a) = u_a \\ u(b) = u_b \end{cases}$
2. 2D Poisson equation (2D elliptic problem)	$\Delta u \equiv \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$ $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \gamma \cdot \left( \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \right) = f(x, y)$
3. 1D Advection equation (1D hyperbolic problem)	$\begin{cases} \frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 \quad \text{or} \quad u_t + cu_x = 0 \\ u(x, t = 0) = f(x) \end{cases}$
4. 1D Heat conduction equation (1D parabolic problem)	$\begin{cases} \frac{\partial u}{\partial t} = v \frac{\partial^2 u}{\partial x^2} \quad \text{or} \quad u_t = \nu u_{xx} \\ u(x, t = 0) = e^{ikx} \end{cases}$

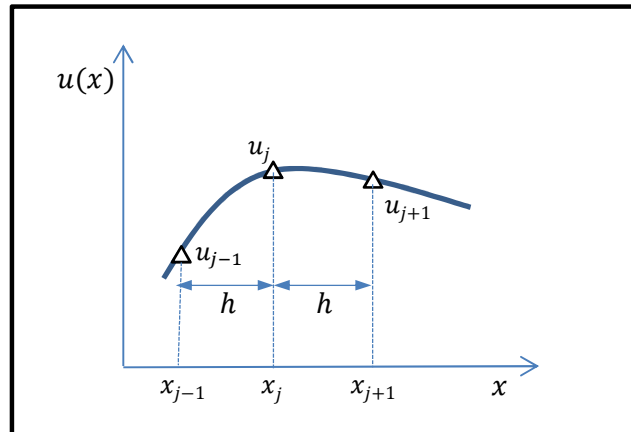
## 8 Discretisation methods

Various discretisation methods are available to replace Partial Differential Equation (PDE) by the system of algebraic (linear or nonlinear) equations. Among the most often used are:

- (1) Finite Difference methods
- (2) Finite Element methods
- (3) Finite Volume methods
- (4) Spectral methods
- (5) Particle methods
- (6) Lattice Boltzmann methods

We will present Finite Difference methods as their properties are easiest to investigate and asses. The conclusions, however, will remain valid also for the other approaches. In engineering practice the finite difference methods are of very limited applicability, as only simplest geometries can be treated by this kind of discretisation.

## 9 Finite Difference method



Suppose we would like to estimate the first derivative of function  $u(x)$  at some point  $x_j$  knowing the value of the function at the neighbouring points:  $u_j = u(x_j), u_{j+1}, u_{j-1}$ , where  $x_j = jh$ . For this purpose one may get inspired by the direct definition of the first derivative:

$$u'_j \equiv \left. \frac{du}{dx} \right|_{x_j} \stackrel{\text{def}}{=} \lim_{h \rightarrow 0} \frac{u(x_j + h) - u(x_j)}{h} \quad (30)$$

and to propose different algebraic formulas:

$$\begin{aligned} u'_j &\approx \frac{u_{j+1} - u_j}{h} && \text{Forward finite difference} \\ u'_j &\approx \frac{u_j - u_{j-1}}{h} && \text{Backward finite difference} \\ u'_j &\approx \frac{u_{j+1} - u_{j-1}}{2h} && \text{Central finite difference (the average of} \\ &&& \text{the first two)} \end{aligned} \quad (31)$$

This is, however, a purely heuristic procedure and it is difficult to see what are the differences between the formulas above (e.g., with respect to their accuracy) or how to generate analogous formulas for higher derivatives.

More systematic procedure is based on the Taylor expansion technique. This technique will be first illustrated for the forward and central difference formulas. Let's expand  $u(x_j \pm h)$  for small values of the step size  $h$ :

$$u(x_j \pm h) = u(x_j) \pm \left. \frac{du}{dx} \right|_{x_j} h + \frac{d^2u}{dx^2} \Big|_{x_j} \frac{h^2}{2!} \pm \left. \frac{d^3u}{dx^3} \right|_{x_j} \frac{h^3}{3!} + \dots \quad (32)$$

or using briefer notation:

$$u_{j\pm 1} = u_j \pm hu'_j + \frac{h^2}{2!}u''_j \pm \frac{h^3}{3!}u'''_j + \dots \quad (33)$$

Then the forward difference can be expressed as:

$$\frac{u_{j+1} - u_j}{h} = \frac{1}{h} \left[ u_j + hu'_j + \frac{h^2}{2!} u''_j + \frac{h^3}{3!} u'''_j + \dots - u_j \right] = u'_j + \frac{h}{2!} u''_j + \frac{h^2}{3!} u'''_j + \dots \quad (34)$$

We have then shown, that the forward difference is equal to the first derivative and the error is equal to  $\frac{h}{2!} u''_j + \frac{h^2}{3!} u'''_j + \dots$ . As a result the error converges to zero when the step size  $h$  tends to zero (the leading term of the error is proportional to  $h$ ).

The same analysis can be carried out for the central difference:

$$\begin{aligned} \frac{u_{j+1} - u_{j-1}}{2h} &= \\ &= \frac{1}{2h} \left\{ u_j + hu'_j + \frac{h^2}{2!} u''_j + \frac{h^3}{3!} u'''_j + \dots - \left( u_j - hu'_j + \frac{h^2}{2!} u''_j - \frac{h^3}{3!} u'''_j + \dots \right) \right\} = \\ &= u'_j + \frac{h^2}{6} u'''_j + \dots \end{aligned} \quad (35)$$

In this case again the central difference is equal to the first derivative, while the error is proportional to  $h^2$ . This means that the central difference is more accurate than the forward difference (the error drops down by a factor of 100 for the step size lowered by a factor of 10).

## 9.1 Consistency and the order of accuracy

The procedure above can be generalised also for other operators, describing also the notion of accuracy and consistency.

For this purpose suppose that  $\Phi(u)$  denotes the differential operator acting on the function  $u$ , while  $\Phi_h(u)$  denotes the finite difference formula approximating the differential operator. The error of finite difference formula can be defined now as:

$$\mathcal{E}_h(u) \stackrel{\text{def}}{=} \Phi(u) - \Phi_h(u) \quad (36)$$

We say now that,  $\Phi_h(u)$  is consistent with  $\Phi(u)$  if (in functional sense):

$$\Phi(u) = \lim_{h \rightarrow 0} \Phi_h(u)$$

or

$$\lim_{h \rightarrow 0} \mathcal{E}_h(u) = 0 \quad (37)$$

We say also that, the finite difference formulas has order of accuracy  $p$  if the error (its leading term), is proportional to  $h^p$  (for small values of  $h$ ). Thus the forward finite difference is of first order of accuracy, while the central difference of the second.

From this point of view, formulas with higher order of accuracy are better than these of lower order (more accurate).

It is indeed so if we are interested in simple calculation of the value of the derivative. However, higher order formulas are not necessarily better when it comes to solving the actual ordinary or partial differential equations. This problem will be tackled again in detail in the present course – as in

the past this problem formed a main obstacle in development of numerical methods for simulation of fluid and thermal flows.

## 9.2 Generation of finite difference formulas

We have shown how to analyse the accuracy of the given finite-difference formula. It remains to show a systematic method of generation of finite difference formulas (possible of different accuracy) for a given differential operator.

### 9.2.1 Second-order derivative

Suppose now, we have function values  $u_{j-1}, u_j, u_{j+1}$  and would like to generate a finite-difference formula (as accurate as possible) for the second order derivative  $u_j'' \equiv \frac{d^2u}{dx^2}\Big|_{x_j}$

$$u_j'' \approx \alpha u_{j-1} + \beta u_j + \gamma u_{j+1}$$

by selecting appropriate values of  $\alpha, \beta, \gamma$ . Using again the expansion in the Taylor series one obtains a series of algebraic equations (third column below):

$$\begin{array}{l} \alpha u_{j-1} + \beta u_j + \gamma u_{j+1} = \\ \quad u_j \cdot (\alpha + \beta + \gamma) \\ \quad + h u_j' \cdot (-\alpha + \gamma) \\ \quad + \frac{h^2}{2!} u_j'' \cdot (\alpha + \gamma) \\ \quad + \frac{h^3}{3!} u_j''' \cdot (-\alpha + \gamma) \\ \quad + \frac{h^4}{4!} u_j^{(4)} \cdot (\alpha + \gamma) \\ \quad \dots \end{array} \quad \left| \begin{array}{l} \Rightarrow \alpha + \beta + \gamma = 0 \\ \Rightarrow -\alpha + \gamma = 0 \\ \Rightarrow \frac{h^2}{2!} (\alpha + \gamma) = 1 \\ \Rightarrow -\alpha + \gamma = 0 \\ \Rightarrow \alpha + \gamma = 0 \\ \dots \end{array} \right. \quad (38)$$

These equations are generated by requirement that only second derivative should be left for the selected values of unknowns  $\alpha, \beta, \gamma$ . The system is in principle infinite, but one attempts to fulfil only as many first equations as possible. In the above, only four equations can be fulfilled as the second and the fourth are identical while the third and the fifth are already in contradiction. As a result the fifth term of the Taylor expansion will form the leading term of the error and give the information on the order of accuracy.

From the second (and fourth) equation one obtains  $\alpha = \gamma$ , while from the third one  $\alpha = \frac{1}{h^2}$ . Finally one obtains:

$$\beta = -\frac{2}{h^2}, \quad \alpha = \gamma = \frac{1}{h^2} \quad (39)$$

Therefore the final formula is:

$$u_j'' \approx \frac{u_{j-1} - 2u_j + u_{j+1}}{h^2} \quad (40)$$

while the leading term of the error is  $\frac{h^2}{12} |u_j''|$  (only the absolute value of the error is of interest). Thus the finite-difference formula above is second-order accurate.

### 9.2.2 Asymmetric first-order derivative

Sometimes it is impossible to use central-difference to calculate the first derivative (this happens usually at the boundary where we do not have data on one side of the boundary) and yet second-order of accuracy should be maintained.

In such cases the values of  $u_j, u_{j+1}, u_{j+2}$  are available and we seek for the unknown coefficients in the formula:

$$u_j' \approx \alpha u_j + \beta u_{j+1} + \gamma u_{j+2} \quad (41)$$

Keeping in mind that:

$$u_{j+2} = u_j + 2hu_j' + \frac{4h^2}{2!}u_j'' + \frac{8h^3}{3!}u_j''' + \dots$$

one gets the equation system:

$$\begin{aligned} \alpha u_j + \beta u_{j+1} + \gamma u_{j+2} &= u_j \cdot (\alpha + \beta + \gamma) && \Rightarrow \alpha + \beta + \gamma = 0 \\ &+ hu_j' \cdot (\beta + 2\gamma) && \Rightarrow h(\beta + 2\gamma) = 1 \\ &+ \frac{h^2}{2!} u_j'' \cdot (\beta + 4\gamma) && \Rightarrow \beta + 4\gamma = 0 \\ &+ \frac{h^3}{3!} u_j''' \cdot (\beta + 8\gamma) && \Rightarrow \beta + 8\gamma = 0 \\ &\dots && \dots \end{aligned} \quad (42)$$

from which only the first three equations can be simultaneously fulfilled, giving:

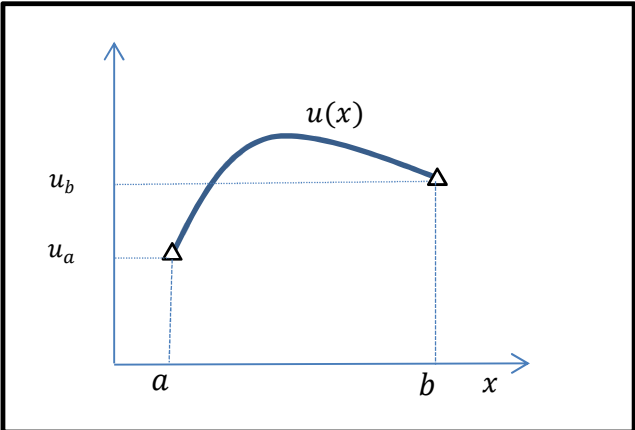
$$\begin{aligned} \beta &= -4\gamma, & \gamma &= -\frac{1}{2h}, & \alpha &= -\beta - \gamma = \frac{1}{2h} - \frac{4}{2h} = -\frac{3}{2h} \\ u_j' &\approx \frac{1}{2h} [-3u_j + 4u_{j+1} - u_{j+2}] \end{aligned} \quad (43)$$

and with the leading term error  $\frac{h^2}{2} |u_j'''|$  the finite-difference formula is second-order as required.

Many other formulas for different operators can be generated in this manner, nevertheless those shown already are sufficient for the present purposes.

## 10 1D Boundary Value Problem

The simplest possible 1D Boundary Value Problem (BVP) is formulated for the interval  $\langle a, b \rangle$ . We seek for the function  $u(x)$ , which fulfils the Ordinary Differential Equation (ODE) and assumes the prescribed values at the ends of this interval.

$$\begin{cases} \frac{d^2 u}{dx^2} = f(x) \\ u(a) = u_a \\ u(b) = u_b \end{cases} \quad (44)$$


This is a very simple linear problem, which can be easily solved analytically, however the analytical solution method cannot be extended to more complicated equations (even in 1D but especially in 2D and 3D) and therefore an alternative - the finite-difference approach should be investigated. It will become clear that the latter technique can be easily extended to higher dimensions and much more complicated equations (including nonlinear cases).

In order to introduce finite difference method the interval  $\langle a, b \rangle$  has to be divided into (equal) subintervals, each of the length  $h$ :

$$\begin{aligned} x_j &\stackrel{\text{def}}{=} a + (j - 1)h, \quad h \stackrel{\text{def}}{=} \frac{b - a}{n - 1}, \quad j = 1, \dots, n \\ x_1 &\equiv a, \quad x_n \equiv b \end{aligned} \quad (45)$$

We will also adopt simplified notation:

$$u_j \stackrel{\text{def}}{=} u(x_j), \quad f_j \stackrel{\text{def}}{=} f(x_j)$$

At the internal points  $j = 2, \dots, n - 1$  the discretisation via finite difference formula is used, while at the boundary the first and the last equation will result from the boundary condition. The system of equations consists then of  $n$  equations and  $n$  unknowns:

$$\begin{aligned}
u_1 &= u_a \\
u_1 - 2u_2 + u_3 &= h^2 f_2 \\
&\dots \\
u_{j-1} - 2u_j + u_{j+1} &= h^2 f_j \\
&\dots \\
u_{n-2} - 2u_{n-1} + u_n &= h^2 f_{n-1} \\
u_n &= u_b
\end{aligned} \tag{46}$$

This equation system has the tridiagonal matrix of the form:

$$Au = g \quad A = \begin{bmatrix} 1 & 0 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 0 & 1 \end{bmatrix}, u = \begin{bmatrix} u_1 \\ u_2 \\ \dots \\ u_{n-1} \\ u_n \end{bmatrix}, g = \begin{bmatrix} u_a \\ h^2 f_2 \\ \dots \\ h^2 f_{n-1} \\ u_b \end{bmatrix} \tag{47}$$

and can be easily solved by a suitable numerical algorithm (see Annex A). This is provided that the matrix  $A$  is not singular (which is not known in advance). This matter will be further investigated in the next Sections.

Similarly one can discretise more complex boundary value problems, in which the ordinary differential equation has additional terms as well as variable but known coefficients ( $p(x), q(x), r(x)$ ):

$$\begin{cases} p(x) \frac{d^2 u}{dx^2} + q(x) \frac{du}{dx} + r(x)u = f(x) \\ u(a) = u_a \\ u(b) = u_b \end{cases} \tag{48}$$

The algebraic equations generated by this procedure have the form:

$$\begin{aligned}
p_j \frac{u_{j-1} - 2u_j + u_{j+1}}{h^2} + q_j \frac{u_{j+1} - u_{j-1}}{2h} + r_j u_j &= f_j \\
\left( p_j - \frac{hq_j}{2} \right) u_{j-1} + (2p_j - h^2 r_j) u_j + \left( p_j + \frac{hq_j}{2} \right) u_{j+1} &= h^2 f_j
\end{aligned} \tag{49}$$

where

$$p_j \stackrel{\text{def}}{=} p(x_j), \quad q_j \stackrel{\text{def}}{=} q(x_j), \quad r_j \stackrel{\text{def}}{=} r(x_j)$$

The matrix of the system is again tridiagonal, but with a different matrix entries:

$$Au = g \quad A = \begin{bmatrix} 1 & 0 & & & \\ \bar{c}_2 & \bar{a}_2 & \bar{b}_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \bar{c}_j & \bar{a}_j & \bar{b}_j \\ & & & & \\ & & & \bar{c}_{n-1} & \bar{a}_{n-1} & \bar{b}_{n-1} \\ & & & & 0 & 1 \end{bmatrix} \tag{50}$$

$$\bar{c}_j = p_j - \frac{hq_j}{2}, \quad \bar{a}_j = 2p_j - h^2 r_j, \quad \bar{b}_j = p_j + \frac{hq_j}{2} \tag{51}$$



Also in this case, we cannot be sure that the matrix is not singular.

## 11 Diagonally dominant matrices

It is generally difficult (or numerically expensive) to decide whether the matrix is non-singular. In many cases however the very simple property of the matrix (called diagonal dominance) can become useful and provide a sufficient condition for the non-singularity.

The matrix  $A = (a_{ij})$  is said to be *strictly diagonally dominant* if:

$$|a_{ii}| > \sum_{\substack{j=1 \\ i \neq j}}^n |a_{ij}|, \quad i = 1, \dots, n \quad (52)$$

That is, if for every matrix row the magnitude of the diagonal element is larger than the sum of all magnitudes of extra-diagonal elements in the row.

### Property:

*Strictly diagonally dominant matrices are non-singular.*

It is quite easy to verify the strict diagonal dominance, nevertheless this is only a sufficient condition for the matrix to be non-singular. In addition the matrix  $A$  from Section 10 is not strictly diagonally dominant ( $2 \not> 1 + 1$ ), so this condition is too strong and cannot be used to obtain the required information. As result a weaker condition has to be used.

The matrix  $A = (a_{ij})$  is said to be *weakly diagonally dominant* if:

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ i \neq j}}^n |a_{ij}|, \quad i = 1, \dots, n \quad (\text{weak inequality})$$

and for at least one row  $i_0$  (53)

$$|a_{i_0 i_0}| > \sum_{\substack{j=1 \\ j \neq i_0}}^n |a_{i_0 j}| \quad (\text{strong inequality})$$

The matrix  $A = (a_{ij})$  is said to be *irreducible* if it cannot be expressed in the block diagonal form:

$$A = \begin{bmatrix} B & 0 \\ 0 & C \end{bmatrix} \quad (54)$$

where  $B$  and  $C$  are square matrices. The possibility to express the matrix in the form above would mean, that the corresponding linear equation system consists of two independent, separate equation systems (and each can be solved separately). Such reducible systems can always be broken into the sequence of irreducible systems, for which further analysis is possible.

### Property:

*Weakly diagonally dominant, irreducible matrices are non-singular.*

*Example:*

It is worth demonstrating that irreducibility is indeed important:

$$\det \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} = 0 \quad \det \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} = 1 \neq 0 \quad \det \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} = 1 \neq 0 \quad (55)$$

The first matrix above is weakly diagonally dominant but reducible and therefore it happens to be singular. The second matrix has the same properties but is non-singular. The third matrix is weakly diagonally dominant and irreducible and therefore is non-singular.

The matrix (47) obtained by discretisation of the simplest BVP, being irreducible and weakly diagonally dominant is therefore non-singular. Thus the numerical solution vector  $(u_1, \dots, u_n)$  exists and might be expected to approximate the exact analytical solution of the BVP. We have not provided here a strict proof of this approximation property, as this requires much more advanced theory.

## 12 Properties of the 1D Boundary Value Problems

It was shown in the previous Sections that the linear equation system corresponding to the simplest 1D BVP is non-singular for all values of step size  $h$ . This analysis will be repeated for more complex equations as well as, subsequently, for higher dimensions. The analysis will start with the equation supplemented by the term containing the first derivative, which models to some degree the presence of convective term in the Navier-Stokes equations (the coefficient  $\gamma$  corresponds to  $\nu^{-1}$  or to the Reynolds number  $Re$ ):

$$\begin{cases} \frac{d^2u}{dx^2} + \gamma \frac{du}{dx} = f(x) \\ u(a) = u_a \\ u(b) = u_b \end{cases} \quad (56)$$

After discretisation (using central difference for the first derivative) one obtains the generic equation - see (49):

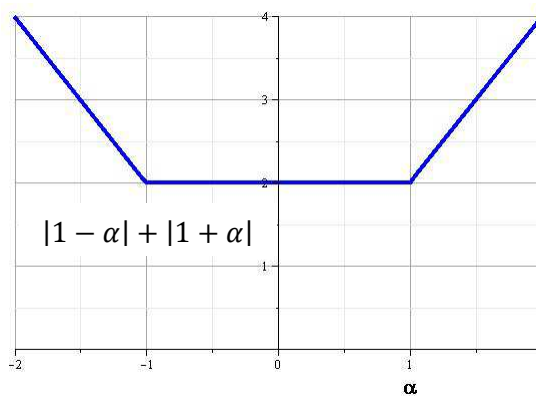
$$\left(1 - \frac{h\gamma}{2}\right)u_{j-1} - 2u_j + \left(1 + \frac{h\gamma}{2}\right)u_{j+1} = h^2f_j \quad (57)$$

The matrix of the whole equation system will be weakly diagonally dominant, if:

$$\left|1 - \frac{h\gamma}{2}\right| + \left|1 + \frac{h\gamma}{2}\right| \leq 2 \quad (58)$$

To solve this inequality (to find  $\alpha \equiv h\gamma/2$ ) we present below the graph of the left hand side, i.e., the function  $|1 - \alpha| + |1 + \alpha|$ . The solution therefore is:

$$|\alpha| \leq 1 \Rightarrow h \leq \frac{2}{|\gamma|} \quad (59)$$



The condition (59) states, that for large  $\gamma$  the stepsize  $h$  has to be sufficiently small to guarantee the non-singularity of the matrix. This means that for large  $\gamma$  the equation system must be larger and therefore more expensive to solve (this is very important in multidimensional cases in which the computational cost can be quite prohibitive).

Another special case forms the ODE with the additional functional term:

$$\begin{cases} \frac{d^2u}{dx^2} + Ku = f(x) \\ u(a) = u_a \\ u(b) = u_b \end{cases} \quad (60)$$

where  $K$  stands for the known coefficient.

After discretisation one obtains the generic equation:

$$u_{j-1} - (2 - Kh^2)u_j + u_{j+1} = h^2 f_j \quad (61)$$

therefore, the condition for diagonal dominance is:

$$|2 - Kh^2| \geq 2 \quad (62)$$

This condition is fulfilled if (i)  $K \leq 0$  or if (ii)  $K > 0$  and  $Kh^2 \geq 4$ . The latter condition cannot be considered, because it limits the stepsize  $h$  from below, while for convergence it is necessary to be able to take arbitrary small value of  $h$ .

We have obtained the conclusion that for eq. (60) the condition for non-singularity of the discretised equation system concerns the type of equation rather than the discretisation itself. It seems therefore that for  $K > 0$  the equation itself may be not *well posed* (the equation is *well posed* if its solution (i) exists, (ii) is unique, and (iii) changes continuously with the initial conditions).

To demonstrate this fact we will consider the homogeneous problem for  $K = 1$ :

$$\begin{cases} \frac{d^2u}{dx^2} + u = 0 \\ u(0) = 0 \\ u(\pi) = 0 \end{cases} \quad (63)$$

If the problem is well posed, then the only solution should be  $u(x) \equiv 0$ . However it is easy to check that  $u(x) \equiv C \sin(x)$ , with the arbitrary value of  $C$  is also a solution of (63). We may conclude then, that for  $K > 0$  the equation (60) may have infinite number of solutions (the problem is not well posed). In fact for nonzero values of  $f(x), u_a, u_b$  the BVP may have no solution. This fact should be mimicked by the discretised problem and therefore the equation system is singular for all  $K > 0$ .

Similar problem appears if Neumann instead of Dirichlet boundary conditions are applied.

$$\begin{cases} \frac{d^2u}{dx^2} = f(x) \\ \frac{du}{dx}\Big|_{x=a} = u_{a1} \\ \frac{du}{dx}\Big|_{x=b} = u_{b1} \end{cases} \quad (64)$$

We consider now the homogeneous problem ( $f(x) \equiv 0, u_{a1} \equiv 0, u_{b1} \equiv 0$ ). After finite difference discretisation, the matrix  $A$  of the equation system has the form:

$$A = \begin{bmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & -1 & 1 \end{bmatrix} \quad (65)$$

since the derivatives in the boundary condition were discretised by backward and forward finite differences:

$$\frac{du}{dx}\Big|_{x=a} \approx \frac{u_2 - u_1}{h}, \quad \frac{du}{dx}\Big|_{x=b} \approx \frac{u_n - u_{n-1}}{h} \quad (66)$$

The matrix is not diagonally dominant, as in all rows

$$|a_{ii}| = \sum_{\substack{j=1 \\ i \neq j}}^n |a_{ij}|, \quad i = 1, \dots, n \quad (67)$$

In this case, the matrix is also singular, as sum of each row elements is zero ( $\sum_{j=1}^n a_{ij}, \quad i = 1, \dots, n$ ).

As a consequence the equation system has an infinite number of solutions. This is also the property of the original BVP (64); every constant  $u(x) \equiv C$  forms its valid solution.

We have shown therefore that lack of diagonal dominance (not being equivalent to singularity of the matrix), may still add important information about the discretised system and the original BVP.

### 13 2D and 3D Boundary Value Problem

We will consider the simplest Laplace ( $f(x, y) \equiv 0$ ) and the Poisson equations defined over the square domain  $\Omega = \langle a, b \rangle \times \langle a, b \rangle$  supplemented with the Dirichlet condition at the boundary  $\partial\Omega$ .

Both functions,  $f(x, y)$  on  $\Omega$  and  $z(x, y)$  on  $\partial\Omega$ , are known (prescribed):

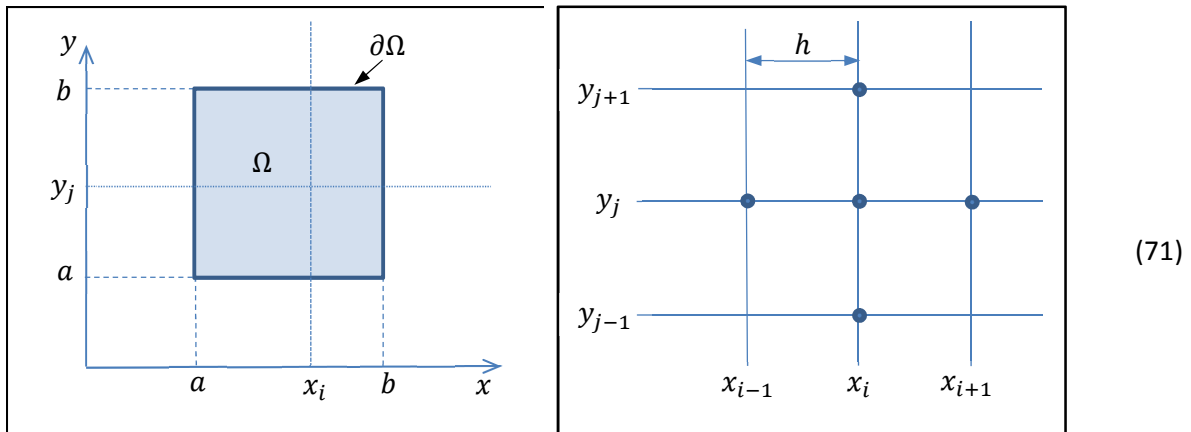
$$\begin{cases} \Delta u \equiv \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y), & (x, y) \in \Omega \\ u(x, y)|_{\partial\Omega} = z(x, y) \end{cases} \quad (68)$$

For discretisation, the domain  $\Omega$  is divided into square cells (cell side length is  $h = \frac{1}{n-1}$ ). The resulting mesh nodes are therefore equidistant:

$$(x_i, y_j), \quad x_i = a + (i - 1)h, \quad y_j = a + (j - 1)h, \quad h = \frac{b - a}{n - 1} \quad (69)$$

For simplification we will assume also the following notation:

$$u_{ij} \stackrel{\text{def}}{=} u(x_i, y_j), \quad f_{ij} \stackrel{\text{def}}{=} f(x_i, y_j), \quad z_{ij} \stackrel{\text{def}}{=} z(x_i, y_j) \quad (70)$$



The domain  $\Omega = \langle a, b \rangle \times \langle a, b \rangle$

The computational stencil

The Poisson equation (68) is discretised using 5-point computational stencil (71), using the same finite difference formulas as in 1D case:

$$\begin{cases} \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2} = f_{ij}, & i, j = 2, \dots, n - 1, \quad N = n^2 \\ u_{i,j}|_{(x_i, y_j) \in \partial\Omega} = z_{ij} \\ h = \frac{1}{n - 1}, \quad N = n^2 \end{cases} \quad (72)$$

The PDE is discretised in the internal points of the grid and additionally the boundary condition is prescribed at the boundary. The total number of equations is therefore equal to the number of grid points  $N = n^2$ , which is in turn equal to the number of unknowns (we formally can regard the boundary values of the solution as unknown).

The discretised equations can be rearranged by grouping the coefficients of the central term and multiplication of both sides by  $h^2$ :

$$u_{i,j-1} + u_{i-1,j} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1} = h^2 f_{ij}, \quad (73)$$

the remaining boundary equations remain unchanged.

It is clear that this is a system of linear equations (all coefficients are constant), nevertheless its form is different than (46) as the unknowns  $u_{ij}$  have two indices instead of one. In order to cast this system in matrix-vector form we introduce therefore a new index  $s = i + n(j - 1)$  which corresponds to row-wise numbering (any other numbering can also be used). The notation of unknowns and the other elements change now, to:

$$u_{s=i+n(j-1)} \equiv u_{ij}, \quad f_{s=i+n(j-1)} \equiv f_{ij}, \quad z_{s=i+n(j-1)} \equiv z_{ij}, \quad (74)$$

while the equation (73) can be rewritten for all internal nodes of the grid as:

$$u_{s-n} + u_{s-1} - 4u_s + u_{s+1} + u_{s+n} = h^2 f_s \quad (75)$$

For the remaining nodes:

$$u_s = z_s, \quad \text{for all } s = i + n(j - 1) \text{ for which } (x_i, y_j) \in \partial\Omega \quad (76)$$

The linear system can be presented now in the matrix-vector form:

$$Au = z \quad (77)$$

in which the matrix, as well as both vectors, have the block form, each block corresponding to a separate grid row. It should be noted that the matrix  $A$  and both vectors  $x$  and  $z$  have  $n$  block rows and  $N = n^2$  scalar rows:

$$A_{N \times N} = \begin{bmatrix} I & 0 & & & & & & & \\ D & T & D & & & & & & \\ & \ddots & \ddots & \ddots & & & & & \\ & & D & T & D & & & & \\ & & & \ddots & \ddots & \ddots & & & \\ & & & & D & T & D & & \\ & & & & & 0 & I & & \end{bmatrix}, \quad N = n^2 \quad (78)$$

$$D_{n \times n} = \begin{bmatrix} 0 & 0 & & & \\ 0 & 1 & 0 & & \\ & \ddots & \ddots & \ddots & \\ & & 0 & 1 & 0 \\ & & & 0 & 1 \end{bmatrix} \quad T_{n \times n} = \begin{bmatrix} 1 & 0 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ & & & 0 & 1 \end{bmatrix} \quad (79)$$

The empty entries (as well as 0) correspond to  $n \times n$  zero matrices, while  $I$  stands for  $n \times n$  identity matrix. The right-hand-side vector  $z$  can be now presented in the block-vector form, each block corresponding to a consecutive grid row:



$$Z_{N \times 1} = \begin{bmatrix} Z_{(1)} \\ Z_{(2)} \\ \vdots \\ Z_{(i)} \\ \vdots \\ Z_{(n-1)} \\ Z_{(n)} \end{bmatrix}, \text{ where } z_{(1)} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_j \\ \vdots \\ g_{n-1} \\ g_n \end{bmatrix}, z_{(i)} = \begin{bmatrix} g^{(i-1)n+1} \\ h^2 f_{(i-1)n+2} \\ \vdots \\ h^2 f_{(i-1)n+j} \\ \vdots \\ h^2 f_{(i-1)n+n-1} \\ g_{i-n} \end{bmatrix}, z_{(n)} = \begin{bmatrix} g_{N-n+1} \\ g_{N-n+2} \\ \vdots \\ g_{N-n+j} \\ \vdots \\ g_{N-1} \\ g_N \end{bmatrix} \quad (80)$$

$$i = 2, \dots, n-1$$

Inspecting row entries of the matrix  $A$  it is now straightforward to verify, that this matrix is weakly diagonally dominant, as each row originating from (73) satisfies the weak inequality, while each row originating from the boundary condition satisfies the strong inequality in (53). Thus the matrix is non-singular and the solution  $u$  always exists.

It should be noted, that the weak diagonal dominance (and therefore also the non-singularity of the equation system) could have been confirmed already at the basis of the original form of the equation system (73), prior to introduction of a particular renumbering to a single index. This renumbering served solely the purpose of better understanding the matrix properties. In practical computations it does not need to be used at all, especially for iterative solution methods.

We will consider now a Poisson equation supplemented by the first derivative as 2D analogy of (60).

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \gamma \cdot \left( \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \right) = f(x, y), & (x, y) \in \Omega \\ u(x, y)|_{\partial\Omega} = z(x, y) \end{cases} \quad (81)$$

Discretising it on the same square mesh, we obtain similarly to (57):

$$\left(1 - \frac{h\gamma}{2}\right) u_{i,j-1} + \left(1 - \frac{h\gamma}{2}\right) u_{i-1,j} - 4u_{i,j} + \left(1 + \frac{h\gamma}{2}\right) u_{i+1,j} + \left(1 + \frac{h\gamma}{2}\right) u_{i,j+1} = h^2 f_{ij} \quad (82)$$

To preserve diagonal dominance of the linear system we have to use sufficiently small stepsize

$$h \leq \frac{2}{|\gamma|} \quad (83)$$

As a result one obtains the condition on the minimum number of points/cells that need to be used in the computational mesh (see ):

$$N = n^2 = \left(\frac{1}{h} + 1\right)^2 \sim h^{-2} \geq \frac{\gamma^2}{4} \quad \text{for } \gamma \sim 10^3, N \sim 10^6 \quad (84)$$

If  $\gamma$  is large the number of cells  $N$  becomes very large (and sometimes so large that it cannot be tackled numerically because of the limitation of computational resources). For 3D cases the similar estimation gives:

$$N = n^3 = \left(\frac{1}{h} + 1\right)^3 \sim h^{-3} \geq \frac{\gamma^3}{8} \quad \text{for } \gamma \sim 10^3, N \sim 10^9 \quad (85)$$

This proves, that the same computational problem becomes even more demanding in 3D.

It will be indicated further on, that for the Navier-Stokes equations a similar problem arises even for moderate Reynolds numbers.

In a similar way to 1D case, it is possible to discretise still more complex 2D and 3D elliptic PDE's obtaining large and sparse linear equation systems, possibly with some conditions on the stepsize  $h$  to preserve diagonal dominance of the system. However, it must be pointed out, that the numerical method of solution of the discretised system typical for 1D problems is not applicable to higher dimensional problems, so other algorithms need to be presented.

## 14 Consequences for the Navier-Stokes equations

If we look at the simplest differential form of the Navier-Stokes equation for the 3D incompressible fluid (omitting the continuity equation):

$$\frac{\partial V}{\partial t} + \underbrace{V \cdot \nabla V}_{\text{CT}} = -\frac{1}{\rho} \nabla p + \underbrace{\nu \Delta V}_{\text{VT}}, \quad V = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (86)$$

we can observe that the convective term (CT) and the viscous term (VT) together are similar to the equation (81), with the exception that the constant coefficient  $\gamma$  in (81) corresponds to a variable  $V$  in the Navier-Stokes equation. However, what is really important for the estimation (59) of the stepsize, is how big the coefficient is (and not that it is constant). We can now neglect the other terms and consider the simplified equation:

$$\nu \Delta V - V_{max} \nabla V = 0 \quad \text{where} \quad V_{max} = \max_{\Omega}(|u|, |v|, |w|) \quad (87)$$

which due to its vector forms has the same limitation for the stepsize:

$$h \leq \frac{2\nu}{V_{max}} \quad (88)$$

We should now consider the characteristic size  $L$  of the fluid phenomenon taking place (chord of the aerofoil/blade, span of the wing, diameter of the pipe). The stepsize in each direction should be now approximately equal:

$$h \sim \frac{L}{n} \quad (89)$$

Therefore the lower limit for the number of grid points has to be:

$$n \geq \frac{LV_{max}}{2\nu} = \frac{\text{Re}}{2}, \quad N = n^3 \geq \frac{\text{Re}^3}{8} \quad (90)$$

where the Reynolds number is based on the maximum velocity. Since in practical engineering/aeronautic applications (aerofoil, wing, engine), the Reynolds number is at least of order  $10^5$  and easily exceeds  $10^7$ , we obtain the required number of mesh points on the level of  $10^{15} - 10^{21}$ . Today, with the largest supercomputers, we cannot take larger grids than consisting of  $10^8 - 10^9$  meshpoints (due to the memory and CPU limitations).

We see then, that (also due to numerical reasons) we must limit Direct Numerical Simulation of viscous flows to much lower Reynolds numbers, than occurring in engineering practice. For higher Reynolds number some form of turbulence modelling is therefore a necessity (RANS, LES, DES, ...).

It should be noted that the presented estimation of the stepsize is perhaps too stringent, and other estimations based, e.g., on Kolmogorov hypothesis can extend somehow the upper limit of allowable Re. Nevertheless the conclusion considering the necessity of turbulence modelling still remains correct.

## 15 Iterative methods to solve the large linear systems

The well-known Gauss algorithm to solve the linear system of equations, as well as presented in the Annex A the method to solve the tridiagonal system, belong to the class of finite algorithms. This means that the solution is obtained by performing a known in advanced, finite number of arithmetic operations (e.g.,  $n^3/3$  arithmetic operations for the Gauss and  $8n$  for the tridiagonal matrix)<sup>1</sup>.

The Gauss method is most suitable for dense matrices, for which the stored non-zero elements fill the matrix. In case of sparse matrices, the number of non-zero elements is proportional to the number of rows/columns (e.g.,  $\sim 3n$  for 1D BVP,  $\sim 5n$  for 2D BVP,  $\sim 7n$  for 3D BVP). For 2D/3D problems the Gauss method, in the process of elimination, fills large number of original zeros significantly increasing the memory requirement (to  $\sim n^{3/2}$  for 2D BVP,  $\sim n^{5/3}$  for 3D BVP). As number of equations  $n$  is typically very large (e.g.,  $10^6 \div 10^8$ ) the required memory may well exceed the available one.

On the other hand the nonlinear systems of equations are always solved by the iterative algorithms and for these algorithms the number of iterations is never well known in advance. Since nonlinear PDE's are our ultimate goal, we will concentrate on the algorithms that might be applicable for both linear and nonlinear systems.

### Example

Suppose we would like to solve the very simple linear equation, with the solution  $x_* = 5$ :

$$3x = 15 \quad (91)$$

For this purpose we can rewrite the equation in two equivalent forms:

$$2x = 15 - x \quad x = 15 - 2x \quad (92)$$

Each form can generate an iterative algorithm (the upper subscript denotes the iteration number):

$$\begin{aligned} 2x^{(m+1)} &= 15 - x^{(m)} \\ x^{(m+1)} &= (15 - x^{(m)})/2 \end{aligned} \quad x^{(m+1)} = 15 - 2x^{(m)} \quad (93)$$

Both iterative processes can start from  $x^{(0)} = 0$ , giving the following sequence of consecutive approximations of the solution:

$$\begin{array}{ll} x_1 = 7.5 & x_1 = 15 \\ x_2 = 3.75 & x_2 = -15 \\ x_3 = 5.625 & x_3 = 15 \\ x_4 = 4.6875 & \dots \\ x_5 = 5.15625 & \\ \dots & \end{array} \quad (94)$$

The first sequence seems to converge to the correct exact solution, the second one does not. This is a very trivial example, but it seems to indicate that the convergence is somehow related to the fact

<sup>1</sup> In this Section  $n$  denotes the size of the matrix, rather than the number of mesh points in one direction or the size of the block.

that the bigger part of the unknown  $x$  was used to form a new iteration (if one compares both processes).

## 15.1 Jacobi iterative algorithm

This entirely heuristic approach can now be used to propose the iterative algorithm to solve the system of linear equations:

$$\begin{aligned} \sum_{j=1}^n a_{ij}x_j &= b_i, \quad i = 1, \dots, n \\ &\text{or} \\ \sum_{j=1}^{i-1} a_{ij}x_j + a_{ii}x_i + \sum_{j=i+1}^n a_{ij}x_j &= b_i, \quad i = 1, \dots, n \\ &\text{or} \\ x_i &= \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^n a_{ij}x_j \right), \quad i = 1, \dots, n \\ &\text{or} \\ x_i &= \frac{1}{a_{ii}} \left( b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}x_j \right), \quad i = 1, \dots, n \end{aligned} \tag{95}$$

As the system is diagonally dominant we have:  $|a_{ii}| \geq \sum_{j=1, j \neq i}^n |a_{ij}|$ . By analogy with the previous approach we can propose an iterative solution process describing how to get the next iteration vector:  $x^{(m+1)}$  out of the already existing:  $x^{(m)}$ :

$$x_i^{(m+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}x_j^{(m)} \right), \quad i = 1, \dots, n \tag{96}$$

or

$$x_i^{(m+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(m)} - \sum_{j=i+1}^n a_{ij}x_j^{(m)} \right), \quad i = 1, \dots, n \tag{97}$$

It is easy to verify that the cost of the single step of this iterative procedure is  $\sim n^2$ . The number of iterations necessary to reach the requested accuracy of the solution depends on the properties and the size of the matrix (and this depends on the BVP which was discretised).

The algorithm (96) is the *Jacobi iterative algorithm*, and it is known to converge for the diagonally dominant systems. Nevertheless, the convergence is very slow and many iterations are necessary to obtain the good approximation of the exact solution.

## 15.2 Gauss-Seidel iterative algorithm

A modification of the Jacobi algorithm is based on the simple observation, that in the process (97) the variables with lower indices are already known from the current  $(m + 1)$  iteration, so we may hope to accelerate convergence by taking the newest values of available unknowns. Such variant is called the *Gauss-Seidel iterative method* and has the following form:

$$x_i^{(m+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(m+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(m)} \right), \quad i = 1, \dots, n \quad (98)$$

In addition to acceleration of convergence, the Gauss-Seidel method requires to store only one solution vector as the new iteration can overwrite the old one (for Jacobi method both have to be kept until the end of each iteration).

## 15.3 Specialisation for sparse matrices

Both Jacobi and Gauss-Seidel methods can be efficiently specialised for the sparse matrices, which originate from the discretisation of the BVP. For example for the multidimensional Poisson problems (73) we have:

$$\begin{aligned} u_{i,j-1} + u_{i-1,j} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1} &= h^2 \\ \text{or} & \\ u_{i,j} &= (u_{i,j-1} + u_{i-1,j} + u_{i+1,j} + u_{i,j+1} - h^2 f_{ij})/4 \end{aligned} \quad (99)$$

The Gauss-Seidel iteration method has now the very simple form:

$$u_{i,j}^{(m+1)} = (u_{i,j-1}^{(m+1)} + u_{i-1,j}^{(m+1)} + u_{i+1,j}^{(m)} + u_{i,j+1}^{(m)} - h^2 f_{ij})/4 \quad (100)$$

provided that  $u_{i,j-1}^{(m+1)}$  and  $u_{i-1,j}^{(m+1)}$  are evaluated prior to  $u_{i,j}^{(m+1)}$  (this follows from the natural, row-wise numbering of the mesh points).

The Jacobi iteration method is formed in the same manner:

$$u_{i,j}^{(m+1)} = (u_{i,j-1}^{(m)} + u_{i-1,j}^{(m)} + u_{i+1,j}^{(m)} + u_{i,j+1}^{(m)} - h^2 f_{ij})/4 \quad (101)$$

## 15.4 Direct iterations to solve nonlinear equations

The similar approach can also be used to solve nonlinear scalar equation as well as the systems of the nonlinear equations. Here we will present only an example how to solve the particular nonlinear equation:

$$x = \frac{1}{2} \cos x \quad (102)$$

The natural iteration procedure is:

$$x^{(m+1)} = \frac{1}{2} \cos x^{(m)} \quad (103)$$

Starting with  $x^{(0)} = 0$  one obtains:

$$\begin{aligned} x^{(1)} &= 0.5000 & x^{(4)} &= 0.4496 \\ x^{(2)} &= 0.4388 & x^{(5)} &= 0.4503 \\ x^{(3)} &= 0.4526 & x^{(6)} &= 0.4502 \end{aligned} \quad (104)$$

The convergence to the exact solution  $x_* \approx 0,4501836$  is in this case quite clear.

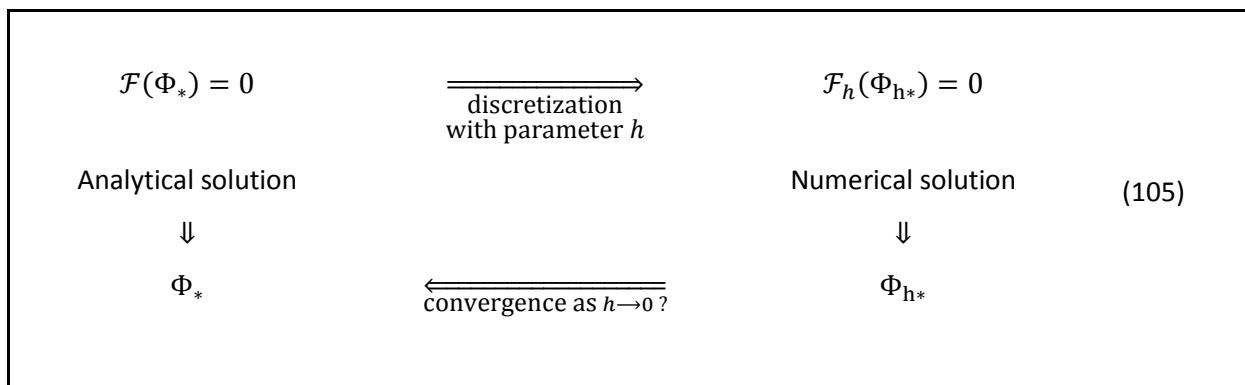
In the general case of nonlinear equation (or equation system) the convergence of direct iterations is quite difficult to achieve (the sufficient conditions for convergence are here difficult to verify).

## 16 Error of the approximate solution

In the previous Sections it was shown how the Partial Differential Equation, which may be impossible or difficult to solve analytically, can be replaced by the system of algebraic equations. It was also shown that such system can be easily solved numerically.

To analyse the process of discretization, let's assume that the original (linear or nonlinear) equation has the form  $\mathcal{F}(\Phi_*) = 0$ , while  $\Phi_*$  is the unknown exact solution. After the discretisation, the original equation is replaced by the system of (linear or nonlinear) equations denoted by  $\mathcal{F}_h(\Phi_{h*}) = 0$ , while  $\Phi_{h*}$  stands for the approximate solution.

The adopted discretisation scheme can be illustrated by the following diagram:



We have shown already, using the Taylor expansion technique, how to achieve the property, that for every sufficiently regular function  $\Phi$  the discretized equation approximates the original differential equation, i.e.:

$$\forall \Phi \quad \lim_{h \rightarrow 0} \mathcal{F}_h(\Phi) = \mathcal{F}(\Phi) \tag{106}$$

This property of discretisation is called *consistency*.

We have indicated also, that this **does not guarantee**, that the approximate solution  $\Phi_{h*}$  tends to the exact solution  $\Phi_*$  as the stepsize  $h$  tends to zero. This property is called convergence and is the one that we really need (but which is much more difficult to obtain and demonstrate)

In the next Sections we will investigate this problem for Partial Differential Equations of hyperbolic and parabolic types, for which it is major importance. In fact the lack of understanding of this issue significantly delayed the development of Computational Fluid Dynamics.



## 17 Hyperbolic advection equation - initial value problem

The simplest PDE of hyperbolic type is an advection equation. This equation models pure transport of some local property  $u(x, t)$  in space and time:

$$\begin{cases} \frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 & \text{or } u_t + cu_x = 0 \\ u(x, t = 0) = f(x) \end{cases} \quad (107)$$

The known function  $f(x)$  represents the value of  $u(x, t)$  at the initial time zero. This problem (107) has a very simple exact solution:

$$u(x, t) = f(x - c \cdot t) \quad (108)$$

The solution changes neither in shape nor in amplitude, but shifts in one direction with the constant speed – to the right or left, depending on the sign of the coefficient  $c$ .

It is also useful to consider a special case of the initial value, in which the function  $f(x)$  consists only of a single Fourier mode with a wavenumber  $k$ :

$$f(x) = e^{ikx} \quad (109)$$

We assume now, that the exact solution has a form:

$$\begin{aligned} u(x, t) &= e^{i(kx - \omega t)} \\ u_x &= iku, \quad u_t = -i\omega u \end{aligned} \quad (110)$$

In the above  $\omega$  is an unknown parameter depending on the wave number  $k$ . Substituting (110) into the advection equation one obtains dispersive relation:

$$-i\omega u + ciku = 0 \quad \Rightarrow \quad \omega = ck \quad (111)$$

The solution can be thus expressed as:

$$u(x, t) = e^{-ickt} e^{ikx} \quad (112)$$

## 18 Parabolic equation - initial value problem

As a model equation we will consider the Heat Conduction Equation, with a complex Fourier mode as an initial condition:

$$\begin{cases} \frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} & \text{or } u_t = \nu u_{xx} \\ u(x, t = 0) = e^{ikx} \end{cases} \quad (113)$$

The heat conduction coefficient, here denoted by  $\nu$ , is always positive. Again we seek the exact solution in the form:

$$\begin{aligned} u(x, t) &= e^{i(kx - \omega t)} \\ u_{xx} &= -k^2 u, \quad u_t = -i\omega u \end{aligned} \quad (114)$$

Substituting the above into the original equation (113), one obtains the dispersive relation:

$$-i\omega = -\nu k^2 \quad (115)$$

and the exact solution:

$$u(x, t) = e^{-\nu k^2 t} e^{ikx} \quad (116)$$

In this case the amplitude of the Fourier mode is damped in time (faster, for higher wave numbers  $k$ ).

## 19 Discretisation of the advection equation

We will attempt now to discretise and solve numerically the advection equation (107). We will assume that the time is divided into equal intervals  $\Delta$ , while  $h$  stands for the step size in  $x$ -direction.

The grid in the space-time consists of discrete points:

$$(x_j, t_p), \quad x_j = j \cdot h, \quad t_p = p \cdot \Delta, \quad j = 0, \pm 1, \pm 2, \dots, \quad p = 0, 1, 2, \dots \quad (117)$$

$$u_j^p \stackrel{\text{def}}{=} u(x_j, t_p)$$

### 19.1 Explicit Euler formula

The simplest finite difference discretisation, which corresponds to the explicit Euler formula for the ODE, is given by the formula:

$$\frac{u_j^{p+1} - u_j^p}{\Delta} + c \frac{u_{j+1}^p - u_{j-1}^p}{2h} = 0 \quad (118)$$

in which we assume that the values at the time-level  $p$  are known from the previous step (or directly from the initial condition). Therefore we obtain an explicit formula to calculate the solution values at the next time level:

$$u_j^{p+1} = u_j^p - \frac{\Delta c}{2h} (u_{j+1}^p - u_{j-1}^p) \quad (119)$$

This finite-difference formula is second-order accurate in space and first-order accurate in time (and therefore the whole formula is consistent with the PDE (107)).

We will show however, that the numerical solution generated by (119) does not converge to the analytic solution of (107). To prove this we will consider the Fourier mode as an initial condition, at the time level  $p$ :

$$u_j^p := A_p e^{ikx_j} \equiv A_p e^{ikjh} \quad (120)$$

In the above  $k$  stands for the wave number, while  $A_p$  denotes the amplitude (which should not increase in time, if the numerical solution is to be consistent with the exact solution). Substituting (120) into (119) one obtains:

$$A_{p+1} e^{ikjh} = A_p e^{ikjh} [1 - \beta (e^{ikh} - e^{-ikh})], \quad \beta = \frac{\Delta c}{2h}$$

$$A_{p+1} = A_p [1 - 2i\beta \sin kh] \quad (121)$$

$$|A_{p+1}| = G \cdot |A_p|, \quad G = \sqrt{1 + 4\beta^2 \sin^2 kh} > 1$$

It is clearly visible, that the amplification factor  $G$  is always greater than 1 and therefore the amplitude of the Fourier mode grows constantly.

## 19.2 Lax theorem

We have shown thus, that the explicit Euler formula cannot be used for numerical simulations.

The property, that the amplitude of the numerical Fourier mode does not grow in time, is called the stability of the numerical scheme.

The Lax theorem states, that if the numerical scheme is consistent and stable, then it is also convergent (and therefore can be used for numerical simulations).

## 19.3 One sided formulas

We will investigate now other numerical schemes to check this property. It will be started with one-sided scheme in which central difference in space is replaced by the one-sided first order finite difference:

$$\begin{aligned}\frac{u_j^{p+1} - u_j^p}{\Delta} + c \frac{u_j^p - u_{j-1}^p}{h} &= 0 \\ u_j^{p+1} &= u_j^p - \frac{\Delta c}{h} (u_j^p - u_{j-1}^p)\end{aligned}\tag{122}$$

Substituting the Fourier mode (120) one obtains:

$$\begin{aligned}A_{p+1} e^{ikjh} &= A_p e^{ikjh} [1 - \beta(1 - e^{-ikh})], \quad \beta = \frac{\Delta c}{h} \\ A_{p+1} &= A_p \cdot G = A_p [1 - \beta + \beta e^{-ikh}] \\ |G| \leq 1 &\Leftrightarrow \beta \geq 0 \wedge \beta \leq 1\end{aligned}\tag{123}$$

This formula is therefore stable (and convergent) for the positive values of  $c$  and for the time step smaller than:

$$\Delta \leq \frac{h}{c}\tag{124}$$

This is called a CFL (Courant-Friedrichs-Lewy) condition. This is a typical condition guaranteeing the stability of the numerical scheme. It says that the time step cannot be too large, otherwise the numerical scheme will produce a solution with unphysical amplitude growth.

The CFL condition can be interpreted as requirement, that during the time step the numerical information is travelling at most through a single computational cell (of size  $h$ ).

An analogous discretisation with the forward finite difference:

$$\begin{aligned}\frac{u_j^{p+1} - u_j^p}{\Delta} + c \frac{u_{j+1}^p - u_j^p}{h} &= 0 \\ u_j^{p+1} &= u_j^p - \frac{\Delta c}{h} (u_{j+1}^p - u_j^p)\end{aligned}\tag{125}$$

is stable for the negative values of  $c$  and for the time step smaller than:

$$\Delta \leq \frac{h}{|c|} \quad (126)$$

These formulas can be put into one formula, which will be stable for both negative and positive values of  $c$

$$\begin{aligned} \frac{u_j^{p+1} - u_j^p}{\Delta} + c_+ \frac{u_j^p - u_{j-1}^p}{h} + c_- \frac{u_{j+1}^p - u_j^p}{h} &= 0 \\ u_j^{p+1} &= u_j^p - \frac{\Delta}{h} \left[ c_+ (u_j^p - u_{j-1}^p) + c_- (u_{j+1}^p - u_j^p) \right] \end{aligned} \quad (127)$$

where:

$$c_+ = \max(0, c), \quad c_- = \min(0, c) \quad (128)$$

and the CFL condition has the same form:

$$\Delta \leq \frac{h}{|c|} \quad (129)$$

It should be noted that this particular formula is no longer a linear discretisation.

## 19.4 Implicit discretisation

Still another discretisation can be obtained by the implicit approach in which space derivative is discretised on the new time level. It will be shown for the Euler discretisation, but the idea can be applied to any explicit formula. The following formula results:

$$\begin{aligned} \frac{u_j^{p+1} - u_j^p}{\Delta} + c \frac{u_{j+1}^{p+1} - u_{j-1}^{p+1}}{2h} &= 0 \\ u_j^{p+1} + \frac{\Delta c}{2h} (u_{j+1}^{p+1} - u_{j-1}^{p+1}) &= u_j^p \end{aligned} \quad (130)$$

Instead of explicit formula we obtain the equation system for the values of  $u_{j-1}^{p+1}, u_j^{p+1}, u_{j+1}^{p+1}$ . This significantly increases the computational effort.

On the other hand if we substitute the Fourier mode (120) into the formula above, we obtain:

$$\begin{aligned} A_{p+1} e^{ikjh} [1 + \beta (e^{ikh} - e^{-ikh})] &= A_p e^{ikjh}, \quad \beta = \frac{\Delta c}{2h} \\ A_{p+1} &= \frac{A_p}{1 + 2i\beta \sin kh} \\ |A_{p+1}| &= G \cdot |A_p|, \quad G = \frac{1}{\sqrt{1 + 4\beta^2 \sin^2 kh}} \leq 1 \end{aligned} \quad (131)$$

The amplification factor is always smaller than 1, therefore the formula is always stable.

This is the general property of implicit discretisation. For the price of solving the (possibly very large) linear system we obtain the unconditionally stable numerical discretisation.

## 19.5 Lax-Friedrichs discretisation

The unstable, explicit Euler discretisation can be improved through averaging of one of the terms:

$$\frac{u_j^{p+1} - \frac{u_{j+1}^p + u_{j-1}^p}{2}}{\Delta} + c \frac{u_{j+1}^p - u_{j-1}^p}{2h} = 0 \quad (132)$$

$$u_j^{p+1} = \frac{u_{j+1}^p + u_{j-1}^p}{2} - \frac{\Delta c}{2h} (u_{j+1}^p - u_{j-1}^p)$$

This formula is stable for the time step  $\Delta$  smaller than:

$$\Delta \leq \frac{h}{c} \quad (133)$$

## 19.6 Higher-order discretisations

It is still possible to generate higher-order discretisations, second order accurate both in space and time. The examples are presented below.

### 19.6.1 The Lax-Wendroff discretisation

$$u_j^{p+1} = u_j^p - \frac{\Delta c}{2h} (u_{j+1}^p - u_{j-1}^p) + \frac{\Delta^2 c^2}{2h^2} (u_{j+1}^p - 2u_j^p + u_{j-1}^p) \quad (134)$$

stable for  $\Delta \leq \frac{h}{c}$

### 19.6.2 The Beam Warming discretisation

$$u_j^{p+1} = u_j^p - \frac{\Delta c}{2h} (3u_j^p - 4u_{j-1}^p + u_{j-2}^p) + \frac{\Delta^2 c^2}{2h^2} (u_j^p - 2u_{j-1}^p + u_{j-2}^p) \quad (135)$$

stable for  $\Delta \leq \frac{h}{c}$

## 20 Discretisation of the 1D parabolic equation

Similarly as for the advection equation, we analyse various discretisation formulas for 1D parabolic equation (113).

### 20.1 Explicit Euler formula

The simplest explicit discretisation is obtained by applying forward finite difference to the time derivative and central scheme for the second spatial derivative:

$$\frac{u_j^{p+1} - u_j^p}{\Delta} = \nu \frac{u_{j+1}^p - 2u_j^p + u_{j-1}^p}{h^2} \quad (136)$$

$$u_j^{p+1} = u_j^p + \frac{\nu\Delta}{h^2} [u_{j+1}^p - 2u_j^p + u_{j-1}^p]$$

This formula is first order accurate in time and second order accurate in space. In order to assess convergence, following the Lax theorem, we have to analyse the stability, i.e., the evolution in time of the discrete Fourier mode (120). Substituting it into (136), we obtain:

$$A_{p+1} e^{ikjh} = A_p e^{ikjh} [1 + \beta (e^{ikh} - 2 + e^{-ikh})], \quad \beta = \frac{\Delta\nu}{h^2} \quad (137)$$

$$A_{p+1} = A_p \left[ 1 + \beta \left( e^{\frac{ikh}{2}} - e^{-\frac{ikh}{2}} \right)^2 \right] = A_p \left[ 1 - 4\beta \sin^2 \frac{kh}{2} \right]$$

The amplification factor can now be estimated as:

$$A_{p+1} = A_p \cdot G$$

$$|G| \leq 1 \Leftrightarrow 4\beta \sin^2 \frac{kh}{2} \leq 2 \Rightarrow \beta \leq \frac{1}{2} \Rightarrow \quad (138)$$

$$\Delta \leq \frac{h^2}{2\nu}$$

This last condition limiting the size of the time step is known as Courant condition and plays the role of CFL condition for parabolic problem. This condition may be very computationally demanding if the step size  $h$  is very small (however this might be mollified if the coefficient  $\nu$  is itself very small).

We conclude, that the explicit Euler formula (136) is stable and therefore convergent if the Courant condition for the time step is fulfilled.

### 20.2 Implicit Euler formula

Similarly implicit Euler discretisation (again first order accurate in time and second order accurate in space) can be presented:

$$\frac{u_j^{p+1} - u_j^p}{\Delta} = v \frac{u_{j+1}^{p+1} - 2u_j^{p+1} + u_{j-1}^{p+1}}{h^2} \quad (139)$$

$$u_j^{p+1} - \frac{v\Delta}{h^2} [u_{j+1}^{p+1} - 2u_j^{p+1} + u_{j-1}^{p+1}] = u_j^p$$

In order to obtain the solution at the next  $t_{p+1}$  time level, the tridiagonal system of linear equations has to be solved. This means a significant increase of the computational cost.

The stability analysis gives us, in this case:

$$A_{p+1} e^{ikjh} [1 - \beta(e^{ikh} - 2 + e^{-ikh})] = A_p e^{ikjh}, \quad \beta = \frac{\Delta v}{h^2}$$

$$A_{p+1} \left[ 1 + 4\beta \sin^2 \frac{kh}{2} \right] = A_p \Rightarrow A_{p+1} = \frac{A_p}{\left[ 1 + 4\beta \sin^2 \frac{kh}{2} \right]} \quad (140)$$

$$G = \frac{1}{1 + 4\beta \sin^2 \frac{kh}{2}} \leq 1$$

The amplification factor  $G$  is in this case always smaller than 1 and therefore the implicit Euler formula is unconditionally stable and convergent. This in turn means, that the time step can be larger than for the explicit case, which may balance the increased computational cost of a single time step.

### 20.3 Crank-Nicolson formula

In order to obtain higher order formula (second order in time at  $t_{p+1/2}$ ) one may take average of the explicit and implicit Euler formulas. This will correspond to the trapezoidal rule if we integrate the parabolic equation over single time step. As a result one obtains:

$$\frac{u_j^{p+1} - u_j^p}{\Delta} = \frac{v}{2} \left[ \frac{u_{j+1}^{p+1} - 2u_j^{p+1} + u_{j-1}^{p+1}}{h^2} + \frac{u_{j+1}^p - 2u_j^p + u_{j-1}^p}{h^2} \right] \quad (141)$$

$$u_j^{p+1} - \frac{v\Delta}{2h^2} [u_{j+1}^{p+1} - 2u_j^{p+1} + u_{j-1}^{p+1}] = u_j^p + \frac{v\Delta}{2h^2} [u_{j+1}^p - 2u_j^p + u_{j-1}^p]$$

This discretisation formula again leads to necessity to solve a linear tridiagonal system at each time step to obtain the solution  $u_j^{p+1}$  at the next time level. The stability analysis leads to the evaluation of the amplification factor in the following form:

$$A_{p+1} \left[ 1 + 4\beta \sin^2 \frac{kh}{2} \right] = A_p \left[ 1 - 4\beta \sin^2 \frac{kh}{2} \right] \Rightarrow A_{p+1} = \frac{\left[ 1 - 4\beta \sin^2 \frac{kh}{2} \right]}{\left[ 1 + 4\beta \sin^2 \frac{kh}{2} \right]} A_p \quad (142)$$

$$|G| = \frac{\left| 1 - 4\beta \sin^2 \frac{kh}{2} \right|}{1 + 4\beta \sin^2 \frac{kh}{2}} \leq 1$$

Again in this case the modulus of the amplification factor  $G$  is always smaller than 1 and therefore the Crank- Nicolson formula is unconditionally stable and convergent.



## Annex A. Tridiagonal matrix algorithm

The linear system with a tridiagonal matrix:

$$Tu = g \quad T = \begin{bmatrix} a_1 & b_1 & & & & \\ c_2 & a_2 & b_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & c_j & a_j & b_j & \\ & & & \ddots & \ddots & \ddots \\ & & & & c_n & a_n \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ \dots \\ u_j \\ \dots \\ u_n \end{bmatrix}, \quad g = \begin{bmatrix} g_1 \\ g_2 \\ \dots \\ g_j \\ \dots \\ g_n \end{bmatrix} \quad (143)$$

can be solved by a simplified Gauss elimination procedure, i.e., by reduction of consecutive elements of lower diagonal by the diagonal element of the previous row. In particular the first row multiplied by some  $\lambda$  is added to the second one. As a result the second row and the right hand side are modified:

$$c'_2 = c_2 + \lambda a_1, \quad a'_2 = a_2 + \lambda b_1, \quad g'_2 = g_2 + \lambda g_1 \quad (144)$$

The coefficient  $\lambda = -c_2/a_1$  is selected such that  $c'_2 \equiv 0$ . The first part of the algorithm consists then of the analogous steps repeated for the rows  $j = 2, \dots, n$ :

$$\lambda = -\frac{c_j}{a_{j-1}}, \quad a_j = a_j + \lambda b_{j-1}, \quad g_j = g_j + \lambda g_{j-1} \quad (145)$$

(the upper subscript ' is dropped for simplification, while  $c_j$  being eliminated does not require evaluation). The matrix has now the following bidiagonal form:

$$T = \begin{bmatrix} a_1 & b_1 & & & & \\ 0 & a_2 & b_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 0 & a_j & b_j & \\ & & & \ddots & \ddots & \ddots \\ & & & & 0 & a_n \end{bmatrix} \quad (146)$$

This system can be now easily solved by the sequence of substitutions (starting from the last equation):

$$u_n = g_n/a_n$$

$$u_j = \frac{g_j - b_j u_{j+1}}{a_j}, \quad j = n-1, \dots, 1 \quad (147)$$

The total numerical cost (number of arithmetic operations) is  $\sim 8n$ , so the algorithm is significantly cheaper than the full Gauss elimination, which requires  $\sim n^3/3$  operations..