Computation Fluid and Solid Mechanics Summary

1. Approximations and their errors

Some phenomena can't be described algebraically. Then numerical techniques offer an outcome. However, these techniques are approximations. And where there are approximations, there are errors. We want to know in what way these errors behave when we refine our model. That's what we'll be looking at in this chapter.

1.1 Introducing the Landau symbols

1.1.1 Definitions of the Landau symbols

When examining approximation errors, we often use the two **Landau symbols** O (**big-o**) and o (**little-o**). We will examine their definitions now. Let's suppose we have two functions $u(x)$ and $v(x)$. We say that $u(x) = O(v(x))$ as $x \to a$ if

$$\lim_{x \to a} \frac{|u(x)|}{|v(x)|} = C, \tag{1.1.1}$$

with C a constant. In words this means that the function $u(x)$ starts behaving more or less like $v(x)$ as x gets close to a. Similarly, we say that $u(x) = o(v(x))$ as $x \to a$ if

$$\lim_{x \to a} \frac{|u(x)|}{|v(x)|} = 0. \tag{1.1.2}$$

Now how does it work? Suppose $u(x)$ is for example $2x^3 + 7x^4 + 5x^5$. Then the order of $u(x)$ is simply the term with the lowest power of x. (So in this case $u(x) = O(2x^3)$.) And we're even allowed to drop the constant! So in fact $u(x) = O(x^3)$. We also say that $u(x)$ is of order 3. By the way, the little-o of $u(x)$ is always one power smaller. So $u(x) = o(x^2)$.

There are a few handy rules which you can use with Landau symbols. As we just saw, we can ignore constants. So we can just say that $c_1 O(c_2 h^n) = O(h^n)$. The same, of course, works for additions. So $O(h^n) + O(h^n) = O(h^n)$, but also $O(h^n) - O(h^n) = O(h^n) \neq 0$. We can also multiply $O(h^n)$ with powers of h. We then get $h^m O(h^n) = O(h^{m+n})$. (This also works if $m < 0$.) Finally we have $O(h^n) + O(h^m) = O(h^n)$ if $n \leq m$.

There is one final addition which you should know. Suppose a function $u(x) = O(x^n)$ as $x \to a$. Now, as x gets 2 times as close to a, then $u(x)$ will be 2^n times as close to $u(a)$. In other words, the deviation of $u(x)$ from $u(a)$ has changed by a factor $(1/2)^n$.

1.1.2 The connection to Taylor polynomials

The **Taylor polynomial to the n^{th} degree** near $x = a$ of a function $u(x)$ is given by

$$p_n(x) = u(a) + \frac{Du(a)}{1!}(x-a) + \frac{D^2u(a)}{2!}(x-a)^2 + \ldots + \frac{D^nu(a)}{n!}(x-a)^n. \tag{1.1.3}$$

To find this polynomial, the function $u(x)$ must of course be differentiable, for as many times as necessary.

The above polynomial doesn't include all terms of the entire Taylor expansion $P_\infty(x)$. It is therefore only an approximation. We can now ask ourselves, how does the error $u(x) - p_n(x)$ behave as $x \to a$? To answer that question, we look at the entire Taylor expansion $p_\infty(x)$ of $u(x)$. By subtracting $p_n(x)$, we have removed all terms with a power x^i, where $i \leq n$. The term with the smallest power of x will then be x^{n+1}. The function $u(x) - p_n(x)$ is therefore $O(x^{n+1})$ as $x \to a$.

1.2 The finite-difference method

1.2.1 Basic finite-difference equations

It is possible to approximate an n^{th} derivative $D^n u(x)$ of a function $u(x)$, using only the function $u(x)$. This process is known as the **finite-difference method**. We can, for example, approximate $Du(x)$ as

$$\delta_h(x) = \frac{u(x+h) - u(x-h)}{2h}. \tag{1.2.1}$$

This approximation increases in accuracy as $h \to 0$. But there of course still is an error in this approximation. And once more we would like to know the order of this error. To find it, we have to use Taylor expansions of $u(x+h)$. The general equation for the Taylor expansion of $u(x+ih)$ is

$$u(x+ih) = u(x) + Du(x)(ih) + \frac{1}{2}D^2u(x)(ih)^2 + \frac{1}{6}D^3u(x)(ih)^3 + \frac{1}{24}D^4u(x)(ih)^4 + O(h^5). \tag{1.2.2}$$

In the above equation, we have cut off the expansion after the fourth term. You can of course also cut it off at any other term. Using this equation, we can find the order of $\delta_h(x)$. We do this according to

$$\delta_h(x) = \frac{u(x) + Du(x)h + \frac{1}{2}Du(x)h^2 + O(h^3) - u(x) + Du(x)h - \frac{1}{2}Du(x)h^2 + O(h^3)}{h} = Du(x) + O(h^2). \tag{1.2.3}$$

So the error $\delta_h(x) - Du(x)$ is $O(h^2)$. If we thus decrease h by a factor 2, then the error decreases approximately by a factor $2^2 = 4$.

1.2.2 Finite-difference equations in matrix form

In our previous example, we only had the terms $u(x+h)$ and $u(x-h)$. It turns out that if you want to approximate higher derivatives of $u(x)$, or if you want more accurate approximations of $Du(x)$, then you also need more terms. Although we could continue to write these terms in equations, it would be a lot better to put it in vector form. (At least in this way our equations will still fit on one sheet of paper.) So we say that $\mathbf{u} = M\mathbf{d} + O(h^{\alpha+1})$, where

$$\mathbf{u} = \begin{bmatrix} u(x-jh) \\ \vdots \\ u(x-h) \\ u(x) \\ u(x+h) \\ \vdots \\ u(x+kh) \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} u(x) \\ Du(x)h \\ D^2u(x)h^2 \\ D^3u(x)h^3 \\ \vdots \\ D^{\alpha-1}u(x)h^{\alpha-1} \\ D^\alpha u(x)h^\alpha \end{bmatrix} \quad \text{and} \quad M = \begin{bmatrix} 1 & -j/1! & (-j)^2/2! & \dots & (-j)^\alpha/\alpha! \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & -1 & 1/2 & \dots & (-1)^\alpha/\alpha! \\ 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1/2 & \dots & a/\alpha! \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & k/1! & k^2/2! & \dots & k^\alpha/\alpha! \end{bmatrix} \tag{1.2.4}$$

Note that M is a $(j+k+1) \times (\alpha+1)$ matrix. Let's suppose that M is a square matrix, and thus that $j+k=\alpha$. In this case it can be shown that M is invertible, and thus that $\mathbf{d} = M^{-1}\mathbf{u} + O(h^{\alpha+1}$. From

this equation we can derive expressions for $D^i u(x)$ (with $0 \leq i \leq \alpha$). This expression will then usually have an order of accuracy $O(h^{\alpha+1-i})$. So what can we note from this? If we increase α, we can either get very accurate approximations, or we can approximate higher-order derivatives.

1.3 Numerically solving differential equations

1.3.1 An outline of the solving method

Let's suppose we have some differential equation we want to solve. For example, we want to solve

$$Du(x) = f(x) \quad \text{on the interval } [0,1], \text{ with the initial value } u(0) = \phi. \tag{1.3.1}$$

To solve this, we divide the interval $[0,1]$ in a **grid** of $N+1$ equally spaced points. For our example, these points are $x_i = ih$, where $i = 0, 1, 2, \ldots, N$ and $h = 1/N$. We will now try to find the corresponding values of $u(x_i)$. We will denote our approximations by u_i.

But the question remains, how can we find these approximations u_i? For that, we use the finite-difference approximation. For our example, we therefore approximate that

$$\frac{u_i - u_{i-1}}{h} = f(x_i). \tag{1.3.2}$$

The value of u_0 follows from the initial condition. (In fact, $u_0 = u(0) = \phi$.) Then, using the above equation, we can find every other value of u_i. And even when we have a more difficult equation, with other initial conditions, it is usually possible to find a solution in this way.

The above method is based on the idea of **convergence**. This means that, as h decreases, our approximations converge to the actual solutions. In other words, the more data points we use, the more accurate we are.

1.3.2 Using residuals to determine the approximation accuracy

Of course we are interested in the error of our approximation. To examine this error, we define two residuals. First we define the **residual of the differential equation** $R_i(u)$. This is the error in our finite-difference approximation. For our example, we thus have

$$R_i(u) = \frac{u(x_i) - u(x_{i-1})}{h} - f(x) = \frac{u(x_i) - \big(u(x_i) - Du(x_i)h + O(h^2)\big)}{h} - f(x_i) = O(h). \tag{1.3.3}$$

(The latter part follows from the relation $Du(x_i) = f(x_i)$, which was our differential equation.) We of course want that $R_i(u) \to 0$ as $h \to 0$. If this is indeed the case, we say that our finite-difference approximation is **consistent** with the differential equation.

The second residual we will define, is the **residual of the initial conditions** $R_{IC}(u)$. This residual is the difference between the given initial conditions $u(x_i)$, and our approximations u_i to them. For our example we thus have

$$R_{IC}(u) = u_0 - u(0) = u_0 - \phi. \tag{1.3.4}$$

Although for our example we have $R_{IC}(u) = 0$, this is of course not always the case. It may very well occur that $R_{IC}(u) = O(h^2)$, for example.

Of course we want our actual solution to be accurate. But then both our initial conditions and our finite-difference approximation should be accurate. If only one of these is accurate, while the other is inaccurate, then our solution isn't accurate either. In fact, our solution is as accurate as the worst of these two residuals. If, for example, $R_i(u) = O(h^3)$ and $R_{IC}(u) = O(h^2)$, then our solution has an error which is of order 2 (meaning it is $O(h^2)$). This order is referred to as the **order of consistency**.

1.3.3 Using plots to find the order of consistency

Another way to find the order of consistency, is by making a plot. Let's suppose r is the error in our approximation. We will now plot ${}^2\log r$ with respect to ${}^2\log h$. As $h \to 0$, this graph converges to a straight line. Let's call the slope of this line n. It turns out that this n is actually the order of consistency of our approximation!

Don't believe me? Then let's show it. We have thus defined n as

$$n = \frac{d\left({}^2\log r\right)}{d\left({}^2\log h\right)}, \qquad \text{which, after working out, gives} \qquad \frac{1}{r}dr = \frac{n}{h}dh. \tag{1.3.5}$$

Let's suppose our error r is $O(h^m)$. As $h \to 0$, we can thus approximate r as ch^m. It follows that $dr = mch^{m-1}dh$. By inserting this in the above equation, we find

$$\frac{mch^{m-1}}{ch^m}dh = \frac{n}{h}dh, \qquad \text{which implies that} \qquad m = n. \tag{1.3.6}$$

In other words, the slope of this graph n is equal to the order of our approximation error m. And this is what we wanted to show.

2. Hyperbolic and elliptic equations

There are multiple types of partial differential equations (PDEs). Tackling one equation differs from solving another one. So first we need to look at what kind of equations there are. Then we will try to solve them using numerical techniques.

2.1 Types of partial differential equations

2.1.1 Linear, quasi-linear and nonlinear equations

Let's examine a partial differential operator $L(u)$. We say that it is **linear** if

$$L\left(\sum c_i u_i\right) = \sum c_i L(u_i), \tag{2.1.1}$$

where c_i are constants. If $L(u)$ is not linear, it can be **quasi-linear**. Just look for the highest derivatives. Replace all other terms and coefficients by constants. If the remaining operator satisfies equation (2.1.1) (and is thus linear), then the original PDE is quasi-linear. In every other case, it is **nonlinear**. Solving these kind of equations is usually hardest.

2.1.2 Hyperbolic, parabolic and elliptic equations

We can also classify PDEs in hyperbolic, parabolic and elliptic equations. **Hyperbolic** PDEs usually describe phenomena in which features propagate in preffered directions, while keeping its strength (like supersonic flow). **Elliptic** PDEs usually describe phenomena in which features propagate in all directions, while decaying in strength (like subsonic flow). **Parabolic** PDEs are just a limit case of hyperbolic PDEs. We will therefore not consider those.

There is a way to check whether a PDE is hyperbolic or elliptic. For that, we have first have to rewrite our PDE as a system of first-order PDEs. If we can then transform it to a system of ODEs, then the original PDE is hyperbolic. Otherwise it is elliptic.

The above method might sounds a bit vague. S so we'll demonstrate it with an example. Let's consider the equation

$$\frac{\partial^2 u}{\partial x^2} + c\frac{\partial^2 u}{\partial y^2} = 0, \tag{2.1.2}$$

where c is an unknown constant. We want to know for what c the above equation is hyperbolic, and for which it is elliptic.

First we need to split the equation up into a system of first-order PDEs. So we define $p = \partial u/\partial x$ and $q = \partial u/\partial y$. We can then find two equations. One follows from our differential equation, and the second one is almost per definition true.

$$\frac{\partial p}{\partial x} + c\frac{\partial q}{\partial y} = 0 \qquad \text{and} \qquad \frac{\partial q}{\partial x} - \frac{\partial p}{\partial y} = 0. \tag{2.1.3}$$

We now examine the linear combinations of the above two equations. In other words, we consider

$$\left(\frac{\partial p}{\partial x} + c\frac{\partial q}{\partial y}\right) + \alpha\left(\frac{\partial q}{\partial x} - \frac{\partial p}{\partial y}\right) = 0. \tag{2.1.4}$$

We can rewrite this to

$$\left(\frac{\partial}{\partial x} - \alpha\frac{\partial}{\partial y}\right)p + \left(\frac{\partial}{\partial x} + \frac{c}{\alpha}\frac{\partial}{\partial y}\right)(\alpha q) = 0. \tag{2.1.5}$$

So, we want to know whether we can transform the system of PDEs to a system of ODEs. If we can, then there should be some (real) direction s, in which we can differentiate both p and q (or equivalently, both p and αq). In other words, we should have

$$\frac{\partial}{\partial x} - \alpha\frac{\partial}{\partial y} = \frac{\partial}{\partial x} + \frac{c}{\alpha}\frac{\partial}{\partial y}, \qquad \text{which implies that} \qquad \alpha = \pm\sqrt{-c}. \tag{2.1.6}$$

What can we conclude from this? If $c > 0$, then there is no real direction in which we can differentiate p and q. So the equation is elliptic. If $c < 0$, then the equation is hyperbolic. We can differentiate p and q in two directions. To find these directions s_1 and s_2, we just insert $\alpha = \sqrt{-c}$ and $\alpha = -\sqrt{-c}$ in equation (2.1.5). We then find

$$\frac{d}{ds1}(p + \sqrt{c}q) = 0, \qquad \text{where} \qquad \frac{d}{ds1} = \frac{\partial}{\partial x} - \sqrt{-c}\frac{\partial}{\partial y}, \tag{2.1.7}$$

$$\frac{d}{ds2}(p - \sqrt{-c}q) = 0, \qquad \text{where} \qquad \frac{d}{ds2} = \frac{\partial}{\partial x} + \sqrt{-c}\frac{\partial}{\partial y}. \tag{2.1.8}$$

Now what does this mean? To find that out, we look at the so-called **Riemann invariants** $p + \sqrt{-c}q$ and $p - \sqrt{-c}q$. These values are constant (invariant) along the lines where $\sqrt{-c}x + y$ and $\sqrt{-c}x - y$ are constant, respectively. By the way, these lines are called the **characteristic lines**.

2.1.3 Boundary conditions for elliptic and hyperbolic PDEs

Let's suppose we have a rectangle, spanning from $x = 0$ to $x = w$ and from $y = 0$ to $y = h$. Also suppose that this rectangle is subject to the PDE of equation (2.1.2). When solving for u, we will need boundary conditions. We want enough boundaries to have a unique solution. If this is indeed the case, then we say that the problem is **well-posed**.

First let's ask ourselves, how many boundary conditions do we need? There is a second-order derivative with respect to x, so we need two boundary conditions for given x. This can be something like $u(0, y) = f_1(y)$, $u(w, y) = f_2(y)$ or $\frac{\partial u}{\partial x}(0, y) = f_3(y)$. Similarly, we need two boundary conditions for given y.

Now let's ask ourselves, where should we apply these boundary conditions? This differs for hyperbolic and elliptic PDEs. For hyperbolic equations we should have one side with two boundary conditions. So there usually are two boundary conditions at the line $x = 0$. For elliptic PDEs things are different. If we have an elliptic PDE, then we should have exactly one boundary condition at every side of our rectangle.

2.2 Numerical methods for hyperbolic equations

2.2.1 Finite-difference schemes

Let's suppose we have, as an example equation, the PDE

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \tag{2.2.1}$$

with the constant a being bigger than zero. We want to solve this equation over a certain interval using a finite-difference approximation. To do that, we first have to turn our interval into a **grid**: We divide it in small rectangles. These rectangles span Δt in t-direction and Δx in x-direction. We will now approximate $u(x, t)$ by u_m^l (with l and m integers), where $l = \frac{x}{\Delta x}$ and $m = \frac{t}{\Delta t}$. So we simply find u_m^l, and then we say that $u(x, t) \approx u_m^l$.

But how do we find u_m^l? Well, to do that, we have to use a **finite-difference scheme**. Such a scheme expresses u_m^l in its neighbouring cells. There are many finite-difference schemes for equation (2.2.1). One

example is

$$\frac{u_m^{l+1} - u_m^l}{\Delta t} + a\frac{u_{m+1}^l - u_{m-1}^l}{\Delta x} = 0. \tag{2.2.2}$$

It is often handy to visualize which points are used by a finite-difference scheme. A good way to do that, is by making a stencil of the scheme. A stencil of the scheme above can be seen in figure 2.1.

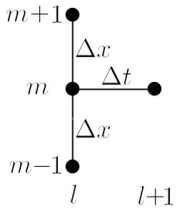

Figure 2.1: Stencil of a finite difference scheme.

There are multiple types of schemes. If you can express the next time level u_m^{l+1} in previous time levels, we say that the scheme if **explicit (forward) in time**. Otherwise it is **implicit (backward) in time**. (You then need to have an equation with matrices to solve the problem.) If the expressions in x-direction are symmetric with respect to point m (or some other point), then we say the scheme is **central in space**. If this is not the case, then we use data from only one direction. We then call the scheme **upwind in space**. We can see that our example scheme is forward in time. (We can express u_m^{l+1} as a function of $u_{\dots}^l$.) It is, however, central in space. (The scheme is symmetric about m.)

Once we have selected a scheme, we will use our boundary conditions. We usually know the value of u at the boundaries $x = 0$ and $t = 0$. (So we usually know u_0^l and u_m^0 for all l, m.) Using our finite-difference scheme, we can then find the values of u for $l = 1$, $l = 2$, and so on. At least, this works for a hyperbolic PDE.

You may be wondering, why doesn't it work for an elliptic PDE? Well, we have previously seen that hyperbolic PDEs often don't have a boundary condition at the boundary $x = w$. So, by applying our finite-difference scheme we can just work from left (where $x = 0$) to right (up to $x = w$) without a problem. When we deal with an elliptic PDE, there is also a boundary condition at $x = w$. And we actually need to combine the boundary condition at $x = 0$ and $x = w$! We will take a look on that difficult problem at the end of this chapter. For now we will just stick to our hyperbolic example PDE.

2.2.2 Testing for consistency

We want our approximation to be **consistent**. This means that, as $\Delta t \to 0$ and $\Delta x \to 0$, our finite-difference scheme converges to the actual PDE. To test for consistency, we use Taylor series expansions. The general equations for Taylor expansions in this case are

$$u_m^{l+a} = u_m^l + a\Delta t\left(\frac{\partial u}{\partial t}\right)_m^l + \frac{a^2}{2}\Delta t^2\left(\frac{\partial^2 u}{\partial t^2}\right)_m^l + \frac{a^3}{6}\Delta t^3\left(\frac{\partial^3 u}{\partial t^3}\right)_m^l + \frac{a^4}{24}\Delta t^4\left(\frac{\partial^4 u}{\partial t^4}\right)_m^l + O(\Delta t^5), \tag{2.2.3}$$

$$u_{m+a}^l = u_m^l + a\Delta x\left(\frac{\partial u}{\partial x}\right)_m^l + \frac{a^2}{2}\Delta x^2\left(\frac{\partial^2 u}{\partial x^2}\right)_m^l + \frac{a^3}{6}\Delta x^3\left(\frac{\partial^3 u}{\partial x^3}\right)_m^l + \frac{a^4}{24}\Delta x^4\left(\frac{\partial^4 u}{\partial x^4}\right)_m^l + O(\Delta x^5). \tag{2.2.4}$$

By the way, usually the indices l and m are omitted. We can apply these relations to our finite-difference scheme. We then find

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + \frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2} + a\frac{\Delta x^2}{6}\frac{\partial^3 u}{\partial x^3} + O(\Delta t^2, \Delta x^4) = 0. \tag{2.2.5}$$

So, as $\Delta t \to 0$ and $\Delta x \to 0$, our finite-difference scheme does converge to our PDE. So it is consistent! By the way, the above equation is called the **modified equation** belonging to the finite-difference scheme.

2.2.3 Testing for stability and convergence

We also want our scheme to be **stable**. This means that our solution remains bounded. One method to see whether our solution remains bounded is the **Von Neumann stability analysis**. For this we assume that we can write the exact solution to our PDE as

$$u(x,t) = Ue^{\sigma t}e^{i\omega x} \qquad \Rightarrow \qquad u_m^l = Ue^{\sigma l\Delta t}e^{i\omega m\Delta x} = U\rho^l e^{ikm}. \tag{2.2.6}$$

Here U and ω are given real constants, and σ is still to be determined. We have also defined the **amplification factor** $\rho = e^{\sigma\Delta t}$ and the **grid frequency** $k = \omega\Delta x$.

To apply the stability analysis, we should find ρ, or, more specifically, $|\rho|$. If $|\rho| > 1$, our solution is unstable. If $|\rho| < 1$ it is stable. If $|\rho| = 1$, our solution has neutral stability.

Let's try to find ρ for our example scheme. We insert our new relation for u_m^l in the finite-difference scheme (2.2.2). After we divide by common terms, we find

$$\rho = 1 - \frac{a\Delta t}{2\Delta x}\left(e^{ik} - e^{-ik}\right) = 1 - i\frac{a\Delta t}{\Delta x}\sin k. \tag{2.2.7}$$

So ρ is a complex number! (Note that we have used the relations $e^{ik} = \cos k + i\sin k$ and $e^{-ik} = \cos k - i\sin k$.) To find the length $|\rho|$ of ρ, we have to sum up the squares of the real and the complex part, and take the square root of that. We then get

$$|\rho| = \sqrt{(1)^2 + \left(\frac{a\Delta t}{\Delta x}\sin k\right)^2}. \tag{2.2.8}$$

It turns out that $|\rho|$ is always bigger than 1! So our example scheme isn't stable at all! How annoying. Luckily there are plenty other schemes that are stable. We'll look at those in the next paragraph.

Finally, we also want our finite-difference scheme to be **convergent**. This means that our solution converges to the continuous solution as $\Delta x \to 0$ and $\Delta t \to 0$. Luckily, the **Lax equivalence theorem** states that any scheme that is both consistent and stable is also convergent. This means we don't have to worry about convergence. Great!

2.2.4 Numerical schemes for our example equation

There are a lot more numerical schemes belonging to our example PDE. And they all got names too! Let's examine a few.

First, there is the **upwind scheme**, reading

$$\frac{u_m^{l+1} - u_m^l}{\Delta t} + a\frac{u_m^l - u_{m-1}^l}{\Delta x} = 0. \tag{2.2.9}$$

This scheme is stable for $a\frac{\Delta t}{\Delta x} \leq 1$. It is even exact for $a\frac{\Delta t}{\Delta x} = 1$. (This usually means that the finite-difference solution coincides with the actual solution.)

Second, there is the **Lax-Friedrichs scheme**. This one is defined as

$$\frac{2u_m^{l+1} - (u_{m+1}^l - u_{m-1}^l)}{2\Delta t} + a\frac{u_{m+1}^l - u_{m-1}^l}{2\Delta x} = 0. \tag{2.2.10}$$

This scheme is stable if $-1 \leq a\frac{\Delta t}{\Delta x} \leq 1$.

Third, we have the **Leapfrog scheme**, defined as

$$\frac{u_m^{l+1} - u_m^{l-1}}{2\Delta t} + a\frac{u_{m+1}^l - u_{m-1}^l}{2\Delta x} = 0. \tag{2.2.11}$$

This scheme has $|\rho| = 1$ if $-1 \leq a\frac{\Delta t}{\Delta x} \leq 1$. So it is neutrally stable. Theoretically this is great. But in practice this often poses difficulties.

There is also another disadvantage to the Leapfrog scheme. To calculate u_m^{l+1} we need data from both the l layer and the $l-1$ layer. Usually this data isn't present in just our boundary conditions. Instead, we first need to user another scheme to acquire enough data. The Leapfrog scheme is thus often combined with the Lax-Friedrichs scheme. Together, they form the **Lax-Wendroff two-step scheme**. First, this scheme uses the Lax-Friedrichs scheme to find the points u_{m+1}^{l+1} and u_{m-1}^{l+1}. It then continous with the Leapfrog scheme to find u_m^{l+2}. It can be derived that we then have

$$\frac{u^{l+2} - u_m^l}{2\Delta t} + \frac{a}{4\Delta x}\left(u_{m+2}^l - u_{m-2}^l\right) - \frac{a^2\Delta t}{4\Delta x^2}\left(u_{m+2}^l - 2u_m^l + u_{m-2}^l\right) = 0. \tag{2.2.12}$$

2.3 Numerical methods for elliptic equations

2.3.1 Implicitly solving the system

Elliptic differential equations are much harder to solve then hyperbolic differential equations. This is because elliptic equations have boundary conditions at all boundaries of the interval. So any finite-difference scheme has to combine multiple boundary conditions at different points. To see how we deal with this, we examine the **Laplace equation** $\nabla^2 u = 0$. (Note that we now don't have a time-derivate, but a derivative w.r.t. y.) A possible finite-difference scheme for this equation is

$$\frac{u_{m+1,n} - 2u_{m,n} + u_{m-1,n}}{2\Delta x} + \frac{u_{m,n+1} - 2u_{m,n} + u_{m,n-1}}{2\Delta y}, \tag{2.3.1}$$

where $m\Delta x = x$ and $n\Delta y = y$. If we choose $\Delta x = \Delta y = h$, we get

$$u_{m-1,n} + u_{m,n-1} - 4u_{m,n} + u_{m+1,n} + u_{m,n+1} = 0. \tag{2.3.2}$$

We can't just use four points to find the fifth now. So how do we solve it? One way is by solving for all points simultaneously (we call this **implicit solving**). We do that using a matrix. We first put all (unknown) data points in a vector

$$\mathbf{x} = \begin{bmatrix} \cdots & u_{m,n-1} & \cdots & u_{m-1,n} & u_{m,n} & u_{m+1,n} & \cdots & u_{m,n+1} & \cdots \end{bmatrix}^T. \tag{2.3.3}$$

Then we can write our system of equations, being

$$A\mathbf{x} = \begin{bmatrix} & & 1 & & & 1 & -4 & 1 & & & 1 & & & \\ & 1 & & & 1 & & -4 & & 1 & & & 1 & & \\ & & & 1 & & & 1 & -4 & & 1 & & & 1 & \\ & & & & & & & & & & & & & \end{bmatrix} \mathbf{x} = \mathbf{0}. \tag{2.3.4}$$

We then need to solve this equation. Although it is possible, it is quite a lot of work. So usually other methods are prefered.

2.3.2 Iteration methods

Luckily, there is a slightly easier way to solve this problem. This is by using **iteration methods**. This method uses iterative steps. To apply it, we first choose some values for $u_{m,n}^0$ (for every m, n). We then

refine our initial choice using an iterative equation. One example of such an equation is the **point Jacobi iteration**, given by

$$u^l_{m-1,n} + u^l_{m,n-1} - 4u^{l+1}_{m,n} + u^l_{m+1,n} + u^l_{m,n+1} = 0. \tag{2.3.5}$$

So during every iteration step the value of $u_{m,n}$ is updated, using the values of its neighbours. This is done for every number $u^{l+1}_{m,n}$, after which the next iteration step commences. And, after a while, the values of $u^l_{m,n}$ will (hopefully) converge.

The point Jacobi iteration methods only 'upgrades' one point every iteration step. There are several other iteration methods, which 'upgrade' more points in one step. For example, there is the **point Gauss-Seidel iteration**. This method has two variants. In the first method a point $u_{m,n}$ and all points to the left-bottom of it are upgraded. (So all $u_{i,j}$ with $i + j \leq m + n$.) In the second method a point $u_{m,n}$ and all points to the right-top are upgraded. (So all $u_{i,j}$ with $i + j \geq m + n$.) This gives us the equations

$$\begin{aligned} u^{l+1}_{m-1,n} + u^{l+1}_{m,n-1} - 4u^{l+1}_{m,n} + u^l_{m+1,n} + u^l_{m,n+1} &= 0, & (2.3.6)\\ u^l_{m-1,n} + u^l_{m,n-1} - 4u^{l+1}_{m,n} + u^{l+1}_{m+1,n} + u^{l+1}_{m,n+1} &= 0. & (2.3.7) \end{aligned}$$

There is also the **line Jacobi iteration**. Here we iterate over one line every time. This can be either a horizontal or a vertical line. So,

$$\begin{aligned} u^{l+1}_{m-1,n} + u^l_{m,n-1} - 4u^{l+1}_{m,n} + u^{l+1}_{m+1,n} + u^l_{m,n+1} &= 0, & (2.3.8)\\ u^l_{m-1,n} + u^{l+1}_{m,n-1} - 4u^{l+1}_{m,n} + u^l_{m+1,n} + u^{l+1}_{m,n+1} &= 0. & (2.3.9) \end{aligned}$$

Finally there is the **line Gauss-Seidel iteration**. It is similar to the line Jacobi iteration. But if we iterate over a horizontal line now, we don't only update all points in that line, but also all points below it. Similarly, if we iterate over a vertical line, we also update all points left to it. So we then get the equations

$$\begin{aligned} u^{l+1}_{m-1,n} + u^{l+1}_{m,n-1} - 4u^{l+1}_{m,n} + u^{l+1}_{m+1,n} + u^l_{m,n+1} &= 0, & (2.3.10)\\ u^{l+1}_{m-1,n} + u^{l+1}_{m,n-1} - 4u^{l+1}_{m,n} + u^l_{m+1,n} + u^{l+1}_{m,n+1} &= 0. & (2.3.11) \end{aligned}$$

2.3.3 Testing for convergence

The above iteration techniques are nice, if they actually cause $u_{m,n}$ to converge to their actual value. But that isn't always the case. To investigate when this happens, we introduce the **iteration error** $\epsilon^l_{m,n} = u^l_{m,n} - u^*_{m,n}$, where $u^*_{m,n}$ is the exact (converged) solution in point m, n. We assume that

$$\epsilon^l_{m,n} = E\rho^l e^{i(k_1 m + k_2 n)}, \tag{2.3.12}$$

where k_1 and k_2 are the **grid frequencies** in x- and y-direction, respectively. Also E is a constant and ρ is the amplication factor. We are interested in ρ. Because, if we have found it, we can find the **rate of convergence** R_c using

$$R_c = \ln \frac{1}{|\rho|}. \tag{2.3.13}$$

So how do we find ρ? We just insert ϵ in our iteration scheme. We can do this for the point Jacobi iteration. We then get

$$E\rho^l e^{i(k_1(m-1)+k_2 n)} + E\rho^l e^{i(k_1 m + k_2(n-1))} - 4E\rho^l e^{i(k_1 m + k_2 n)} + E\rho^l e^{i(k_1(m+1)+k_2 n)} + E\rho^l e^{i(k_1 m + k_2(n+1))} = 0. \tag{2.3.14}$$

Solving this equation for $|\rho|$ will give

$$|\rho| = \left| \frac{e^{ik_1} + e^{-ik_1} + e^{ik_2} + e^{-ik_2}}{4} \right| = \frac{|\cos k_1 + \cos k_2|}{2}. \tag{2.3.15}$$

So we see that we always have $|\rho| \leq 1$ for this iteration method, which is nice. It means that eventually the values of $u^l_{m,n}$ will converge to $u^*_{m,n}$.

3. The panel method

A nice method to simulate the flow around an airfoil, is the so-called panel method. We'll examine this method in this chapter. But before we do that, we first have to examine some basic flow types.

3.1 Flow types

3.1.1 Incompressible flow

Let's examine a steady incompressible flow. Based on these assumptions, the continuity equation reduces to $\nabla \cdot \mathbf{u} = 0$, with $\mathbf{u}$ the velocity field. This is nice to know, but by itself it isn't very handy. We therefore also examine the **vorticity** $\omega = \nabla \times \mathbf{u}$. Let's assume that this vorticity is $\omega = 0$. If this is the case, then it can be shown that there is a function Φ such that

$$\mathbf{u} = \nabla \Phi. \tag{3.1.1}$$

Φ is called the **(total) potential function**. Using our earlier relation for $\mathbf{u}$, we find that Φ must satisfy $\nabla^2 \Phi = 0$. (This is called the **Laplace equation**.) From it, one can attempt to solve for Φ, using given boundary conditions. Once Φ is known, the pressure p in the flow can also be found. For that, we can use the momentum equation, being

$$\frac{1}{2}\rho |\nabla \Phi|^2 + p = \text{constant}. \tag{3.1.2}$$

What might be more useful is to find the pressure coefficient C_p. For that, we can use

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty V_\infty^2} = 1 - \frac{|\nabla \Phi|^2}{V_\infty^2}. \tag{3.1.3}$$

3.1.2 Compressible flow

For incompressible flows, we can not use the above equations. So we have to use different ones. Before we examine them, we first define the **disturbance potential** ϕ, such that $\Phi = V_\infty x + \phi$. It can be shown that this disturbance potential satisfies

$$\rho_\infty (1 - M_\infty^2) \frac{\partial^2 \phi}{\partial^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2}. \tag{3.1.4}$$

The factor $(1 - M_\infty^2)$ is slightly annoying. To get rid of it, we can change our coordinate system. Let's define $x' = x/\sqrt{1 - M_\infty^2}$, $y' = y$ and $z' = z$. We then get

$$\frac{\partial^2 \phi}{\partial x'^2} + \frac{\partial^2 \phi}{\partial y'^2} + \frac{\partial^2 \phi}{\partial z'^2}. \tag{3.1.5}$$

And this is simply the Laplace equation! So once more, we need to solve a Laplace equation to find ϕ. Once we have found ϕ, we can again find the pressure coefficient C_p. However, this time we have to use the approximation

$$C_p \approx -2 \frac{\partial \phi / \partial x}{V_\infty} \tag{3.1.6}$$

3.1.3 Boundary conditions

We see that we need to solve the Laplace equation. However, to solve it, we need boundary conditions.

We can, for example, set the value of ϕ along the boundary. This is a so-called **Dirichlet boundary conditions**. If we do this, then there is a unique solution for ϕ. (We say that the problem is **well-posed**.)

However, we can also set the value of $\partial\phi/\partial\mathbf{n}$ at the boundary. We now have **Neumann boundary conditions**. This time, there is no unique solution for ϕ. We can only determine ϕ up to a certain unknown constant.

Of course we can also use combinations of Dirichlet and Neumann boundary conditions. However, as long as ϕ is set somewhere, then there is always a unique solution for ϕ.

3.2 Solving the Laplace equation using singularity distributions

3.2.1 Singularity distributions

Sometimes it is very hard to solve the Laplace equation for given boundary conditions. Luckily the Laplace equation is a linear differential equation. So let's suppose that we have some solutions of the Laplace equation. Any linear combination of these solutions will then also be a solution. It thus also satisfies the Laplace equation!

So the first thing we need to do is find some **elementary solutions** for the Laplace equation. Examples of such solutions are sources, sinks, dipoles, vortices and such. We will discuss some of them now.

3.2.2 Sources and sinks

Now it is time to examine some elementary solutions. One such solution is the so-called **source**. Let's suppose we have a source with strength $\sigma(Q)$, positioned at some point Q. The potential ϕ caused by this source at some point P then is

$$\phi(P) = \frac{\sigma(Q)}{2\pi} \ln\left(r(P,Q)\right). \tag{3.2.1}$$

By the way, if the source strength σ is negative, the source is often called a **sink**.

What can we do with these sources? Well, we can put a lot of them on a curve S. We then get a **source distribution**. To find the velocity potential at some point P, caused by this distribution, we use an integral. This velocity potential thus will be

$$\phi(P) = \int_S \frac{\sigma(Q)}{2\pi} \ln\left(r(P,Q)\right) ds. \tag{3.2.2}$$

A problem might occur if the point P lies on the source distribution itself. Such a situation should always be considered separately.

3.2.3 Doublets

Another elementary solution is the **doublet**. The potential at some point P, caused by a doublet at Q, is given by

$$\phi(P) = \mu(Q) \frac{\partial}{\partial \mathbf{n_Q}} \left(\frac{1}{2\pi} \ln\left(r(P,Q)\right) \right). \tag{3.2.3}$$

Here $\mu(Q)$ is the strength of the doublet and $\mathbf{n_Q}$ is the direction of the doublet.

Once more, we can put a lot of doublets in a row. We then get a **doublet distribution**. To find the velocity potential at P, we now have to use

$$\phi(P) = \int_S \mu(Q) \frac{\partial}{\partial \mathbf{n_Q}} \left(\frac{1}{2\pi} \ln\left(r(P,Q)\right) \right) ds. \tag{3.2.4}$$

Once more, the case $Q \in S$ should be considered separately.

3.3 The panel method

3.3.1 Using source distributions

We know that any combination of sources, doublets, vortices and such satisfies the Laplace equation. However, which combination describes our flow? To find that out, we have to use the boundary conditions. But how should we use them? We just use the panel method! Simply apply the following steps.

- First we take the edge of our airfoil. We split it up in N panels.
- We then place a source distribution (or similarly, a doublet distribution or vortex distribution) of constant strength σ_i on every panel. ($1 \leq i \leq N$.)
- At every panel, we can find the velocity potential ϕ (as a function of the source strengths σ). We can also find the flow velocity $\partial\phi/\partial\mathbf{n}$ normal to the airfoil. This flow velocity should of course be 0. This gives us N equations. Using these conditions, we can solve for the source strengths σ_i.

Solving for the source strengths σ might seem like a difficult task. However, these equations can simply be put in a matrix. This goes according to

$$\begin{bmatrix} A_{1,1} & \cdots & A_{1,N} \\ \vdots & \ddots & \vdots \\ A_{N,1} & \cdots & A_{N,N} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_N \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix}. \tag{3.3.1}$$

The left part of the above equation calculates the velocities caused by the source distributions. The right part of the equation then takes into account the parameters V_∞ and the angle of attack α.

3.3.2 Adding doublets/vortices

There is, however, one slight problem. Sources and sinks don't create any drag/lift. In other words, if we only place sources and sinks on our airfoil, then $c_l = c_d = 0$. To solve this problem, we also need to add doublets/vortices to the wing.

It is, for example, possible to add a constant vortex distribution with strength μ along the entire airfoil. This would give our airfoil lift. However, it would also give us an extra unknown. We thus also need an extra boundary condition. A condition that is often used, is the **Kutta condition**. It demands that the flow leaves the trailing edge smoothly. But what does that mean? To find that out, let's examine the velocity at the lower part of the trailing edge V_t^l. We also examine the velocity at the upper part of the trailing edge V_t^u. The Kutta condition claims that these velocities are equal: $V_t^l = V_t^u$.

Now that we have an extra unknown, and an extra equation, our system of equations also expands. We

thus get

$$\begin{bmatrix} A_{1,1} & \cdots & A_{1,N} & A_{1,\mu} \\ \vdots & \ddots & \vdots & \cdots \\ A_{N,1} & \cdots & A_{N,N} & A_{N,\mu} \\ A_{\mu,1} & \cdots & A_{\mu,N} & A_{\mu,\mu} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_N \\ \mu \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_N \\ f_\mu \end{bmatrix}. \tag{3.3.2}$$

From this the values of σ and also μ can be solved.

3.3.3 Finding the lift

But the question still remains, how can we find the lift? For that, we can use the **Kutta-Joukowski theorem**. This theorem gives the relation between the circulation Γ and the lift L. It is

$$L = \rho_\infty V_\infty \Gamma. \tag{3.3.3}$$

If we want to find the lift coefficient, we can use

$$c_l = \frac{2\Gamma}{Vc}. \tag{3.3.4}$$

By the way, by adding vortices/doublets, you do not effect the drag coefficient. This coefficient is still zero. And that is one of the major downsides of the panel method — you can't really use it to calculate drag.