COMPUTATIONAL MODELLING SAMMARY II

BASED ON COMPUTATIONAL MODELLING BY S U HULSHOFF

SAM YAN ELSLOO

APRIL - JUNE 2017

ナマケモノの戦争馬の攻撃

Preface - 2018-2019 edition

Also, after all of my complaints, I felt obliged to admit that not *all* Casio-calculators are shit; their basic calculator is actually quite okay.

Their other calculators still suck tho

Also, since there seems to be some misunderstanding on how distributing the summaries and solution manuals and what not works:

- If something appears on the facebook group, it's fine to share it with everyone. If not, it's not fine. So don't put a summary or link somewhere if I did not publish it yet officially myself.
- As I mentioned before, after the course has ended, please remove the summaries and everything from your computer. You shouldn't be spreading around the summaries of last year in the first year chat when I did not publish it yet (see also first point above). It only makes it more annoying for me if I decide to make changes to the summary as it means I have to put extra effort into highlighting what changed, so just don't do it.
- Don't do any guessing of the next bit.ly link. Admittedly I make them very easy, but just be patient for the next one and be grateful when I release it but don't be greedy and try to get a hold of it before you should. It only makes me more reluctant to ever bother fixing or changing something about the summary/solution manual.

Even if you don't fully agree with the things above, it's like literally the least you can do in return for all of the summaries and solution manuals, so just abide with the points above honestly.

Contents

3 Discretisation with the finite-difference method

3.3 Confirming consistency and consistency

Of course, the question that has been burning on your mind since the first quiz is how to confirm whether a method is consistent or not, and whether it is stable or not. Today is your lucky day, cause in this section, we'll exactly see how.

Confirming consistency of a finite-difference method is again something you already learned in applied numerical analysis.

3.3.1 Confirming consistency: the modified equation

You actually already studied all of this when you did applied numerical analysis (the second quiz, at least). Remember the definition of consistency: a discretisation is **consistent** if when substituting an exact solution into the discrete equations the only terms which remain are those which tend to zero as the number of degrees of freedom is increased. However, like I said immediately afterwards, this amounts to plugging in a few Taylor expansion, check whether there's at least a Δx (or Δy or Δt) involved in the remaining error. To refresh your memory a bit: suppose we have the explicit Euler in time, central in space finite-difference approximation[1] for the linear advection equation:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \qquad \rightarrow \qquad \frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} = 0$$

Now, how did we check whether this was consistent? We have the following Taylor expansions:

$$\begin{aligned}
u_i^{n+1} = u(x, t+\Delta t) &= u_i^n + \Delta t\frac{\partial u}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 u}{\partial t^2} + \mathcal{O}\left(\Delta t^3\right) \\
u_i^n &= u_i^n \\
u_{i+1}^n = u(x+\Delta x, t) &= u_i^n + \Delta x\frac{\partial u}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 u}{\partial x^2} + \frac{1}{6}\Delta x^3\frac{\partial^3 u}{\partial x^3} + \mathcal{O}\left(\Delta x^4\right) \\
u_{i-1}^n = u(x+\Delta x, t) &= u_i^n - \Delta x\frac{\partial u}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 u}{\partial x^2} - \frac{1}{6}\Delta x^3\frac{\partial^3 u}{\partial x^3} + \mathcal{O}\left(\Delta x^4\right)
\end{aligned}$$

If you plug these in, and work it out a bit, you get the following:

$$\begin{aligned}
\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} + \frac{\Delta t^2}{2}\frac{\partial^2 u}{\partial t^2} + c\frac{\Delta x^2}{6}\frac{\partial^3 u}{\partial x^3} + \mathcal{O}\left(\Delta t^2, \Delta x^4\right) &= 0 \\
\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} &= -\frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2} - c\frac{\Delta x^2}{6}\frac{\partial^3 u}{\partial x^3} + \mathcal{O}\left(\Delta t^2, \Delta x^4\right)
\end{aligned}$$

This equation is called the **modified equation**. Looking at this equation, $\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x}$ is in fact not equal to zero when using this finite-difference scheme, but, *as the number of degrees of freedom is increased* (i.e. Δx and Δt are decreased), *the terms that remain reduce to zero*. This is good. Furthermore, if you think back to dissipation (solution too smooth) and dispersive (too many oscillations), if the leading term of the spatial derivative in the truncation error is even (i.e. $\frac{\partial^2 u}{\partial x^2}$, $\frac{\partial^4 u}{\partial x^4}$, $\frac{\partial^6 u}{\partial x^6}$) then basically, you included an additional artificial-dissipation term in it, thus the solution is likely to be dissipative. Alternatively, if the leading term of the spatial derivative in the truncation error is odd (i.e. $\frac{\partial^3 u}{\partial x^3}$, $\frac{\partial^5 u}{\partial x^5}$, etc.) then the solution is likely to be dispersive.

[1] In case you forgot what that meant: explicit means that we don't use any other unknown solution values for the computation; central in space means that the points used for the approximation of $\frac{\partial u}{\partial x}$ are symmetrically positioned around the point we're interested in.

Note that for example you'd been an idiot and you'd have derived something like

$$\frac{2u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} = 0$$

then you can verify yourself that you end up with

$$\begin{aligned}\frac{u_i^n}{\Delta t} - \frac{\partial u}{\partial t} + \Delta t\frac{\partial^2 u}{\partial t^2} + c\frac{\partial u}{\partial x} + c\frac{\Delta x^2}{6}\frac{\partial^3 u}{\partial x^3} + \mathcal{O}\left(\Delta t^2, \Delta x^4\right) &= 0 \\ \frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} &= -\frac{u_i^n}{\Delta t} + \frac{\partial u}{\partial t} - c\frac{\Delta x^2}{6}\frac{\partial^3 u}{\partial x^3} + \mathcal{O}\left(\Delta t^2, \Delta x^4\right)\end{aligned}$$

which is inconsistent because as Δt and Δx are decreased, we do not end up at 0 on the right-hand side of the equation.

Quiz 2: Question 1

Consider the following discretisation for the convection-reaction equation $\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = u$:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{u_i^n - u_{i-1}^n}{\Delta x} = u_i^n + TE$$

The leading terms of its truncation error, TE, are

The Taylor expansions are:

$$\begin{aligned}u_i^{n+1} &= u_i^n + \Delta t\frac{\partial u}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 u}{\partial t^2} + \ldots \\ u_i^n &= u_i^n \\ u_{i-1}^n &= u_i^n - \Delta x\frac{\partial u}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 u}{\partial x^2} + \ldots\end{aligned}$$

Plugging this in yields

$$\begin{aligned}\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{u_i^n - u_{i-1}^n}{\Delta x} &= u_i^n + TE \\ \frac{u_i^n + \Delta t\frac{\partial u}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 u}{\partial t^2} + \ldots - u_i^n}{\Delta t} + \frac{u_i^n - u_i^n + \Delta x\frac{\partial u}{\partial x} - \frac{\Delta x^2}{2}\frac{\partial^2 u}{\partial x^2} + \ldots}{\Delta x} &= u_i^n + TE \\ \frac{\partial u}{\partial t} + \frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2} + \ldots + \frac{\partial u}{\partial x} - \frac{\Delta x}{2}\frac{\partial^2 u}{\partial x^2} + \ldots &= u_i^n + TE\end{aligned}$$

Now, the governing equation is $\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = u_i^n$; thus, to obtain the truncation error, we can get rid of those terms, leaving us with

$$\frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2} - \frac{\Delta x}{2}\frac{\partial^2 u}{\partial x^2} = TE$$

This means that what you have to write down in Maple is the following: T/2*Diff(u,t,t)-X/2*Diff(u,x,x)

For your information, Diff(u,t,t) works as follows: it outputs u, first differentiated with respect to t, then again differentiated with respect to t. If you for example would want $\frac{\partial^3 u}{\partial x^2 t}$, you'd write Diff(u,x,x,t), etc.

Quiz 2: Question 4

The modified equation for a finite-difference approximation of the linear convection equation $\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0$

is given by:

$$\left(\frac{\partial u}{\partial t}\right)_{(i,n)} + \frac{\Delta t}{2!}\left(\frac{\partial^2 u}{\partial t^2}\right)_{(i,n)} + \mathcal{O}\left(\Delta t^2\right) + \left(\frac{\partial u}{\partial x}\right)_{(i,n)} - \frac{3\Delta x^2}{4}\left(\frac{\partial^3 u}{\partial x^3}\right)_{(i,n)} + \mathcal{O}\left(\Delta x^3\right) = 0$$

The approximation is:

- Inconsistent
- Consistent, first order accurate in space
- Consistent, second order accurate in space
- Consistent, third order accurate in space
- Stable, first order accurate in space
- Stable, second order accurate in space
- Stable, third-order accurate in space

The correct answer is **Consistent, second order accurate in space**. First of all, we can't determine stability from the modified equation, so that means the final three answers are incorrect. The method is consistent, because when $\Delta t \to 0$ and $\Delta x \to 0$, then you get the PDE you wanted to solve. Furthermore, it is second order accurate in space, as we have $\frac{-3\Delta x^2}{4}$ as leading term in the truncation error.

3.4 *Confirming stability: Fourier analysis*

Now that we've had consistency, let's discuss stability, for which we'll use your dearest friend Fourier. For this, we'll *assume* that we're on a periodic domain.

What was Fourier again? Basically, it was a guy who was frantically in love with sines and cosines. Remember that you learned during applied numerical analysis that every function could be written as the sum of polynomials, but Fourier was a radical guy and loved sines and cosines, and he wanted to write every function as a sum of them. He managed to in fact, as long as the original function is periodic. As we are on a periodic domain, it kinda makes sense that the solution is periodic.

Writing the numerical solution in terms of a spatial Fourier series that evolves in time looks like (with I being the imaginary number)

$$u_i^n = \sum_m b_m(t)\, e^{I k_m x}$$

What exactly am I writing here? Remember that $e^{I k_m x}$ simply becomes

$$e^{I k_m x} = \cos\left(k_m x\right) + I \sin\left(k_m x\right)$$

Thus, the trigonometric part of the function sort of only works in x-direction: the periodic waves propagate in x-direction only, they don't change over time t. However, by multiplying with $b_m(t)$, which is solely a function, we enable ourselves to "control" the solution in t direction anyway. Furthermore, note that this is a summation. u_i^n will be the *summation* of many trigonometric functions (each with its own frequency, governed by k_m) that only vary in x-direction, where each of these trigonometric functions have weights governed by its own, unique function $b_m(t)$, meaning some of the trigonometric functions will become more pronounced over time and others will be less influential in determining the value of u_i^n. If you think about it, you should see that this allows to exactly replicate the (numerical) solution, as long as you include enough terms in the solutions (i.e. m is sufficiently large).

Now, regarding the frequencies of each trigonometric component of the summation: $k_m = \frac{2\pi}{l_m}$, where l_m is the wavelength of its component. For increasing m, we take a smaller wavelength (just convention), so that k_m increases. Note that if we have a mesh with spacing Δx, the shortest wavelength which can be resolved is $2\Delta x$ (referred to as the "two-delta" wave); see figure 3.1 for clarification: we are unable to include a wavelength of Δx for example, as this would mean that you'd get another maximum at u_i, meaning you get maxima at u_{i-1}, u_i and u_{i+1}, meaning your wave form is essentially lost. This means that it is pointless to decrease the wavelength below $2\Delta x$ and thus that the maximum wave number k_m is $\pi/\Delta x$.

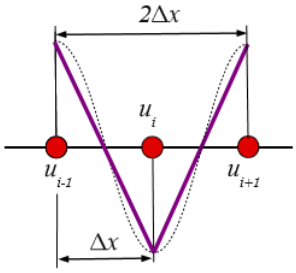


Figure 3.1: The shortest wavelength which can be represented on a mesh with spacing Δx is $2\Delta x$.

Let's put this into practice. Suppose we have the finite-difference equation

$$u_i^{n+1} = u_i^n - \frac{c\Delta t}{2\Delta x}\left(u_{i+1}^n - u_{i-1}^n\right)$$

Consider a solution made up of a single, arbitrary Fourier component, with an assumed exponential variation (i.e. $b_m(t) = e^{at}$. This would be

$$u_i^n = U e^{at} e^{Ik_m x}$$

Note that a may be complex. If we substitute this component into the finite-difference equation (I'll tell you later why we do this), we get (take into account that u_i^{n+1} occurs at $t + \Delta t$, etc.)

$$U e^{a(t+\Delta t)} e^{Ik_m x} = U e^{at} e^{Ik_m x} - \frac{c\Delta t}{2\Delta x}\left(U e^{at} e^{Ik_m (x+\Delta x)} - U e^{at} e^{Ik_m (x-\Delta x)}\right)$$

Now, we can divide everything by U, by e^{at} and $e^{Ik_m x}$ to get

$$e^{a\Delta t} = 1 - \frac{c\Delta t}{2\Delta x}\left(e^{Ik_m \Delta x} - e^{-Ik_m \Delta x}\right) = 1 - I\frac{c\Delta t}{\Delta x}\sin(\beta)$$

with $\beta = k_m \Delta x$, since (this is just a mathematical property which you should know, if not for this course then at least for the instrumentation and signals course)

$$\frac{e^x - e^{-x}}{2I} = \sin(x)$$

Now, we define[2]

AMPLIFICATION FACTOR

The **amplification factor** ρ is defined as

$$\rho \equiv e^{a\Delta t} \tag{3.1}$$

How is this related to stability? It makes sense that a numerical solution is stable if *none* of the Fourier components is to increase in magnitude over time, i.e. $|\rho| \leq 1$ for *all* β (since $\beta = k_m \Delta x$)[3] We can plot the amplification factor versus β for several values of $\frac{c\Delta t}{\Delta x}$ as done in figure 3.2.

Clearly, we see that for the explicit Euler in time, second-order central in space discretisation, no matter what $\frac{c\Delta t}{\Delta x}$ we choose, the amplification is larger than 1 for all $\beta \neq 0$ or $\beta \neq \pi$. This means that almost all components are amplified with each step, meaning the method is unstable. Therefore, this discretisation, although consistent, is not convergent.

Now, consider the explicit Euler in time, first-order upwind method:

$$u_i^{n+1} = u_i^n - \frac{c\Delta t}{\Delta x}\left(u_i^n - u_{i-1}^n\right)$$

[2]Note that the amplification factor is also equal to

$$\frac{u_i^{n+1}}{u_i^n} = \frac{U e^{a(t+\Delta t)} e^{Ik_m x}}{U e^{at} e^{Ik_m x}} = e^{at}$$

This is, in fact, the reason why it is called the amplification factor, and we now see it reappearing when we plugged in the Fourier components.

[3]If you don't see why: note that the amplification factor is equal to $\frac{u_i^{n+1}}{u_i^n}$.

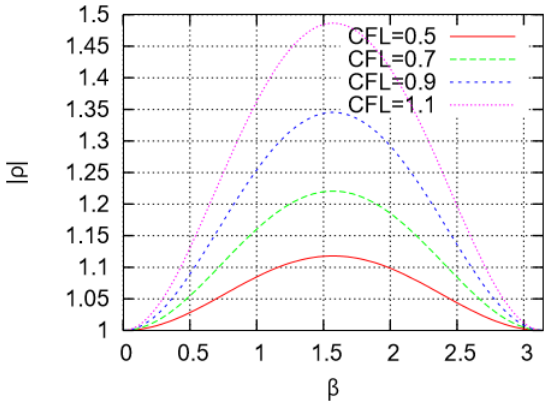


Figure 3.2: Amplification factor versus $\beta = k_m \Delta x$ for an explicit Euler in time, second-order central in space discretisation of the linear advection equation. Results for different values of $CFL = \frac{c\Delta t}{\Delta x}$ are shown.

This time, we find

$$\begin{aligned} U e^{a(t+\Delta t)} e^{I k_m x} &= U e^{at} e^{I k_m x} - \frac{c\Delta t}{\Delta x}\left(U e^{at} e^{I k_m x} - U e^{at} e^{I k_m (x-\Delta x)}\right) \\ e a^{\Delta t} = \rho &= 1 - \frac{c\Delta t}{\Delta x}\left(1 - e^{-I k_m x}\right) = 1 - \frac{c\Delta t}{\Delta x}\left(1 - e^{-I\beta}\right) \end{aligned}$$

Now, $e^{-I\beta} = \cos(-\beta) + I \sin(-\beta) = \cos(\beta) - I \sin(\beta)$ and thus we can write

$$\rho = 1 - \frac{c\Delta t}{\Delta x}\left(1 - \cos(\beta) + I \sin(\beta)\right)$$

Now, $\cos(\beta) = 1 - 2\sin^2\left(\frac{\beta}{2}\right)$ and $\sin(\beta) = 2\sin\left(\frac{\beta}{2}\right)\cos\left(\frac{\beta}{2}\right)$, meaning we can write this as

$$\rho = 1 - \frac{c\Delta t}{\Delta x}\left(1 - 1 + 2\sin^2\left(\frac{\beta}{2}\right) + 2I \sin\left(\frac{\beta}{2}\right)\cos\left(\frac{\beta}{2}\right)\right) = 1 - 2\frac{c\Delta t}{\Delta x}\sin^2\frac{\beta}{2} - 2I\frac{c\Delta t}{\Delta x}\sin\left(\frac{\beta}{2}\right)\cos\left(\frac{\beta}{2}\right)$$

Then,

$$\begin{aligned} |\rho|^2 &= \mathrm{Re}(\rho)^2 + \mathrm{Im}(\rho)^2 = \left(1 - 2\frac{c\Delta t}{\Delta x}\sin^2\frac{\beta}{2}\right)^2 + \left(-2\frac{c\Delta t}{\Delta x}\sin\left(\frac{\beta}{2}\right)\cos\left(\frac{\beta}{2}\right)\right)^2 \\ &= 1 - 4\frac{c\Delta t}{\Delta x}\sin^2\frac{\beta}{2} + 4\left(\frac{c\Delta t}{\Delta x}\right)^2\sin^4\frac{\beta}{2} + 4\left(\frac{c\Delta t}{\Delta x}\right)^2\sin^2\left(\frac{\beta}{2}\right)\cos^2\left(\frac{\beta}{2}\right) \\ &= 1 - 4\frac{c\Delta t}{\Delta x}\sin^2\frac{\beta}{2} + 4\left(\frac{c\Delta t}{\Delta x}\right)^2\sin^2\frac{\beta}{2}\cdot\left(\sin^2\frac{\beta}{2} + \cos^2\frac{\beta}{2}\right) \\ &= 1 - 4\frac{c\Delta t}{\Delta x}\left(1 - \frac{c\Delta t}{\Delta x}\right)\sin^2\frac{\beta}{2} \end{aligned}$$

so that

$$|\rho| = \sqrt{1 - 4\frac{c\Delta t}{\Delta x}\left(1 - \frac{c\Delta t}{\Delta x}\right)\sin^2\frac{\beta}{2}}$$

Again, we can plot this as function of β for several $\frac{c\Delta t}{\Delta x}$, as shown in figure 3.3. Note that this method is conditionally stable provided $\frac{c\Delta t}{\Delta x} \leq 1$. You've seen $\frac{c\Delta t}{\Delta x}$ a few times now; it even has a fancy name:

COURANT-FRIEDRICHS-LEWY NUMBER

The **Courant-Friedrichs-Lewy number** is used to refer to the ratio $\frac{c\Delta t}{\Delta x}$.

This also explains the behaviour of the Python program during the first work session: you could not simply alter j_{max} with no repercussions; at some point, your CFL would increase beyond 1, meaning that your method

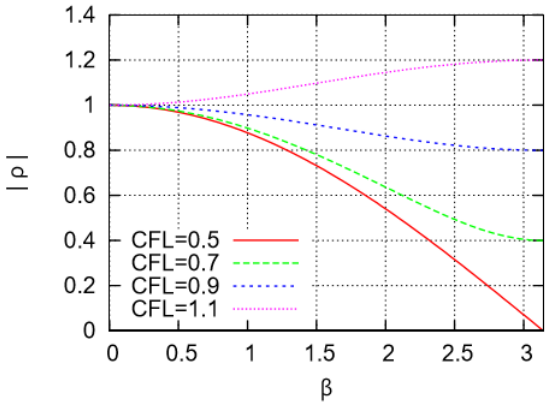


Figure 3.3: Amplification factor versus $\beta = k\Delta x$ for the an explicit Euler in time, first-order upwind in space discretisation of the linear advection equation.

becomes unstable. Furthermore, at stable CFL numbers, the highest frequencies of the solution are significantly decreased in amplitude each time step (higher values of β are associated with higher frequency components, as $\beta = k_m \Delta x$ and k_m is the component frequency). Depending on how much you've already studied for instrumentation and signals, you'll realize that this is the reason for the dissipative effects of an upwind scheme you saw in the previous part of this course, once again shown in figure 3.4 (if you haven't studied so far, in short, you need very high frequency components to accurately portray those very steep gradients; if they are damped out quickly then you can't portray them any more).

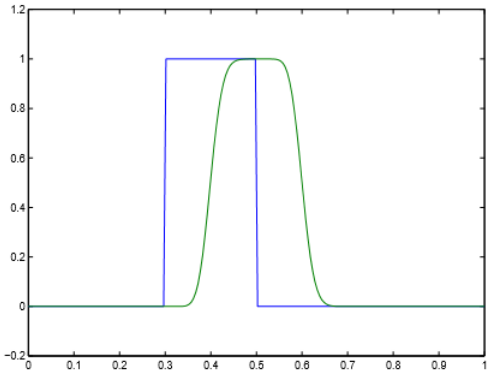


Figure 3.4: Solution from an explicit Euler in time, first-order upwind in space discretisation of the linear advection equation (CFL number = 0.1).

Finally, if you have to determine the amplification factor yourself (note that we make use of the fact that we divide by $U e^{at} e^{Ik_m x}$ somewhere along the way):

DETERMINING THE AMPLIFICATION FACTOR

1. Replace u_i^n with 1.
2. For the nodes in the time direction, use the following substitution:

$$u_i^{n+p} = e^{ap\Delta t} \tag{3.2}$$

3. For the nodes in spatial direction, use the following substitution:

$$u_{i+q}^n = e^{Ik_m q \Delta x} \tag{3.3}$$

4. Rewrite to find an explicit expression for $\rho = e^{a\Delta t}$.

5. If required use the following equations:

$$e^{Ix} = \cos(x) + I\sin(x) \tag{3.4}$$
$$\cos(2x) = \cos^2(x) - \sin^2(2x) = 1 - 2\sin^2(x) \tag{3.5}$$
$$\sin(2x) = 2\sin(x)\cos(x) \tag{3.6}$$

6. Work out

$$|\rho| = \sqrt{(\mathrm{Re}(\rho))^2 + (\mathrm{Im}(\rho))^2} \tag{3.7}$$

Quiz 2: Question 2

A A numerical method with the amplification factor $\rho = 1 - I \cdot \left(0.1\sin^2\beta\right)$ is:

- stable
- conditionally stable
- neutrally stable
- unstable

The correct answer is **unstable**: we have

$$|\rho| = \sqrt{(\mathrm{Re}(\rho))^2 + (\mathrm{Im}(\rho))^2} = \sqrt{1^2 + \left(0.1\sin^2\beta\right)^2} = \sqrt{1 + 0.01\sin^4\beta}$$

which is larger than 1 for values of $0 < \beta < \pi$ (which are the relevant values of β).

To be clear, as long as $|\rho| > 1$ for any value of β between 0 and π (including these end points), then the method is unstable, even if $|\rho| \leq$ for all other values of β.

Quiz 2: Question 5

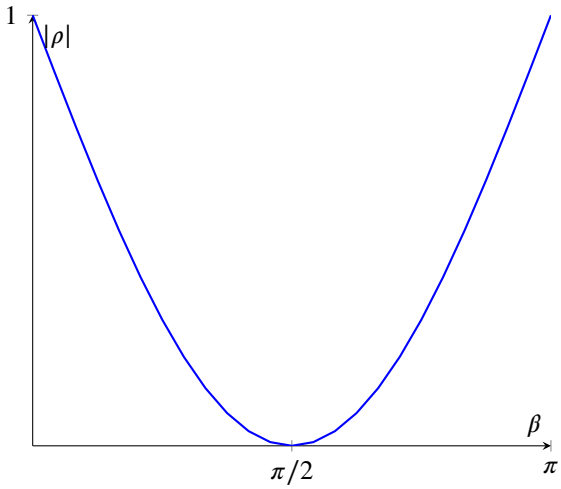
A numerical method with the amplification factor $\rho = 1 - \sin(\beta)$ has its strongest damping for:

- low-frequency components of the solution
- low and medium components of the solution
- medium-frequency components of the solution
- high-frequency components of the solution
- low and high frequency components of the solution
- high and medium components of the solution

The correct answer is **medium-frequency components of the solution**: note that β takes values $0 \leq \beta \leq \pi$. The absolute value of ρ is simply equal to

$$|\rho| = |1 - \sin(\beta)| = 1 - \sin(\beta)$$

as this is always ≥ 0; the graph looks like



Thus, we see that for medium-frequency components (around $\beta = \pi/2$), $|\rho| \approx 0$, whereas for low (near $\beta = 0$) and high (near $\beta = \pi$) frequency components, $\rho \approx 1$. This means that amplification is only really low for the medium-frequency components, meaning the damping is strongest for medium-frequency components.

Quiz 2: Question 7

The following one-sided approximation for the heat equation

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = v \frac{u_i^n - 2u_{i+1}^n + u_{i+2}^n}{\Delta x^2}$$

is to be applied on a uniform mesh for a periodic domain. Defining $g = \frac{v\Delta t}{\Delta x^2}$, $b = k_m \Delta x$ and $I = \sqrt{-1}$, the amplification factor ρ of this method equals:

Using the problem solving guide above, we get

$$\begin{aligned} \frac{e^{a\Delta t} - 1}{\Delta t} &= v \frac{1 - 2e^{Ik_m\Delta x} + e^{Ik_m 2\Delta x}}{\Delta x^2} \\ e^{a\Delta t} = \rho &= 1 + \frac{v\Delta t}{\Delta x^2}\left(1 - 2e^{Ib} + e^{I2b}\right) = 1 + g\left(1 - 2\left(\cos(b) + I\sin(b)\right) + \cos(2b) + I\sin(2b)\right) \end{aligned}$$

and this is the answer. Fortunately, we don't need to calculate the absolute value of the amplification error.

4 Verification

Verification is the process of evaluating the errors occurring between the model and the numerical representation of it (i.e. discretisation errors, iteration errors and rounding errors). Verification can be split in two parts:

VERIFICATION ACTIVITIES

During **code verification**, the consistency of the program used for the simulations with the model PDEs is established.

During **solution verification**, the errors associated with a particular case of interest are estimated.

4.1 Code verification

We may always make errors in our programming code (even though the underlying algorithm is mathematically correct). Code verification consists of a number procedures meant for testing different aspects of an algorithm. We will focus on the most important procedure, the **order-of-accuracy test**, which verifies that the solutions generated by the code converge to the exact solutions of the model equations with the correct order of accuracy. It makes sense to do this: if our discretisation method has order of accuracy $\mathcal{O}\left(\Delta x^2\right)$, but it only converges with order of accuracy $\mathcal{O}\left(\Delta x\right)$, then we probably made a mistake somewhere in our code.

Now, for the order-of-accuracy test, we need the exact solution, otherwise we don't know how fast the error decreases if we decrease Δx. Now, typically speaking, we don't have an exact solution at hand (why would we use numerical solutions if we had them), so what do we do instead?

4.1.1 The method of manufactured solutions (MMS)

We can manufacture our own exact solution. For example, consider the non-linear Burgers equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - v\frac{\partial^2 u}{\partial x^2} = 0; \qquad \text{on} \quad 0 \leq x \leq \pi$$

Now, *assume* the exact solution $u_e(x,t) = \sin(x)\cos(t)$. If we'd substitute this into the PDE, we'd get:

$$\begin{aligned} \frac{\partial u}{\partial t} &= -\sin(x)\sin(t) \\ \frac{\partial u}{\partial x} &= \cos(x)\cos(t) \\ \frac{\partial^2 u}{\partial x^2} &= -\sin(x)\cos(t) \end{aligned}$$

and thus

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - v\frac{\partial^2 u}{\partial x^2} = -\sin(x)\sin(t) + \sin(x)\cos(t)\cos(x)\cos(t) + v\sin(x)\cos(t) = S(x,t)$$

where $S(x,t)$ is the *source term*. So, we'll modify our code slightly to make it numerically solve this equation. The exact solution to this PDE is known (it's simply $u_e(x,t) = \sin(x)\cos(t)$), so we can easily verify how close the numerical solution is to the exact solution[1].

[1] Yes this means that you are verifying a different PDE (although the differential operators on the left are the same, there is a source term now on the right which was not there before). However, note that the inclusion of a source term is not bad at all. If your code is correct for the PDE with this source term, then it is natural to expect that it'll also work with other source terms; this means that it also works for the source term $S(x,t) = 0$.

To make it complete, note that the boundary conditions would now become

$$\begin{aligned} u(0,t) = u(\pi,t) &= 0 \\ u(x,0) &= \sin(x) \end{aligned}$$

to make it comply with the assumed exact solution.

Final thing, what exact solution should you assume? Suppose you'd assumed $u_e(x,t) = x\cos(t)$, then you'd have $\frac{\partial^2 u}{\partial x^2} = 0$. This is bad, as this means the $\upsilon\frac{\partial^2 u}{\partial x^2}$ is not really used in the manufactured solution, meaning it's impossible to verify the implementation of the viscous term. Aside from this, it is generally advantageous to use smooth assumed solutions so that the correct order of accuracy can be obtained without resorting to excessively fine meshes.

Quiz 2: Question 6

Implementing a manufactured solution requires which of the following (multiple answers possible):

- a source term
- compatible boundary conditions
- trigonometric functions
- a stability analysis

The correct answer is **a source term** and **compatible boundary conditions**. It has nothing to do with a stability analysis, and it indeed requires a source term and compatible boundary conditions. Although it requires smooth functions as exact solution, it does not require trigonometric functions for the exact solution per se.

Quiz 2: Question 8

A manufactured solution for the heat equation

$$\frac{\partial u}{\partial t} - a\frac{\partial^2 u}{\partial x^2} = 0$$

of the form $u(x,t) = \sin^2(t)\cos(x)$ requires the addition of a term of the form $S(x,t) = ...$ on the right hand side of the equation.

We have

$$\begin{aligned} u(x,t) &= \sin^2(t)\cos(x) \\ \frac{\partial u}{\partial t} &= 2\sin(t)\cos(t)\cos(x) \\ \frac{\partial u}{\partial x} &= -\sin^2(t)\cos(x) \end{aligned}$$

so that

$$\frac{\partial u}{\partial t} - a\frac{\partial^2 u}{\partial x^2} = 2\sin(t)\cos(t)\cos(x) - a\cdot -\sin^2(t)\cos(x) = 2\sin(t)\cos(t)\cos(x) + a\sin^2(t)\cos(x)$$

Note that in Maple, you need to put an asterisk between a and $sin^2(t)$, additionally, you need to write $\sin(t)^2$, and not $\sin^2(t)$. Always preview your answer before submitting.

Quiz 2: Question 9

An inappropriate form for a manufactured solution for a discretisation of Laplace's equation $\nabla^2 u = 0$ on a rectangular 2D domain in (x, y) is:

- y^2x^2
- y^3x^3

- $(1-x)(1-y)$
- $\left(1-x^2\right)\left(1-y^2\right)$
- $\left(1-x^3\right)\left(1-y^3\right)$
- $\sin(x)\sin(y)$
- $\sin(x)\cos(y)$
- $\cos(x)\cos(y)$

The correct answer is $(1-x)(1-y)$: as I explained, the exact solution needs to be differentiable sufficiently many times. Due to the ∇^2, we need to have the exact solution to have $\frac{\partial^2 u}{\partial x^2} \neq 0$ and $\frac{\partial^2 u}{\partial y^2} \neq 0$. Only for the solution $(1-x)(1-y)$ do these second order derivatives reduce to zero (just work it out yourself, you'll quickly see I'm right). Thus, only this solution is inappropriate.

Quiz 2: Question 10

Choosing a smooth manufactured solution:

- makes it easier to define an appropriate initial condition
- minimises the computational effort required for the order-of-accuracy test
- can violate the assumptions used in the order-of-accuracy test
- makes it easier to define appropriate boundary values

The correct answer is **minimises the computational effort required for the order-of-accuracy test**. Please note: this question does not refer to the fact that solutions need to be sufficiently many times differentiable without reducing to zero (which is the previous question about), but beyond that, what is the advantage of using smooth functions (i.e. once that is many, many times continuously differentiable) compared to less smooth functions (even if they still meet the requirement of being sufficiently many times differentiable). The advantage of using smooth functions is that you don't have to use excessively fine meshes to find the correct order of accuracy. For your information, "can violate the assumptions used in the order-of-accuracy test" refers to the use of discontinuous solutions.

4.1.2 *The order-of-accuracy test*

Now that we have an exact solution, remember what the order of accuracy meant: suppose we have the 1D finite-difference discretisation that is known to produce errors of the form

$$\epsilon = u_e(x,t) - u(x,t) = \mathcal{O}(\Delta x^p) + O\left(\Delta x^{p+1}\right) + ...$$

Such a discretisation is said to have order of accuracy p. If Δx is sufficiently small, then $\mathcal{O}\left(\Delta x^{p+1}\right)$ and higher terms should be much smaller than the $\mathcal{O}(\Delta x^p)$ term. When this is true the discretisation is said to be in the **asymptotic region**. Inside the asymptotic region, if we then for example multiply Δx with a factor $\frac{1}{2}$, then the error should be multiplied by a factor $\left(\frac{1}{2}\right)^p$. Thus, we could plot the error versus Δx as done in figure 4.1, and see whether this is indeed the case[2] if it is, then the code is correct apparently; if it isn't, then there must be a mistake somewhere.

Now, in figure 4.1, we see a few interesting things:

a) It can be that the error becomes very small, but starts to show random behaviour; these would be round-off errors.
b) It can be that you have too large tolerances for iterative solution procedures, meaning you get a minimum error there.
c) It can be that you have incompatible refinement: suppose we have a finite-difference discretisation for a 1D space-time problem which produces errors of the form

$$\epsilon = u_e(x,t) - u(x,t) = \mathcal{O}(\Delta t) + \mathcal{O}\left(\Delta x^2\right) + ...$$

[2] The slope of the straight line should be p.

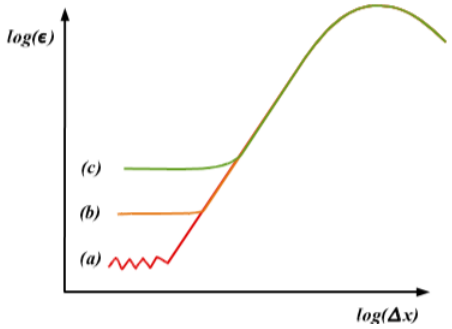


Figure 4.1: Levelling off of convergence due to (a) round-off error, (b) insuffcient tolerances for iterative solution procedures, (c) incompatible refinement.

If we'd keep on decreasing Δx then without changing Δt, then the error would level off at a fixed level of (time) discretisation error. Note that in this case, we'd need a factor of 4 refinement in time to be compatible with a factor 2 refinement in space.

Personally, I think this section is pretty logical. As a final note, the order-of-accuracy test is not able to catch all coding errors. Errors that affect the iterative solution are not detected, for example. However, experience has shown that the order-of-accuracy test is pretty good at even detecting subtle errors.

4.2 Solution verification

Now, let's estimate the numerical error of our method. Of course, we have already something related to that, namely the order of accuracy of the method. However, we'd like to have an actual estimation of the discretisation error. How would we do that? Suppose we have a method that is p-th order accurate. Then we can write that the numerical solution f_d is related to the exact solution f_e as

$$f_d = f_e + gh^p + \mathcal{O}\left(h^{p+1}\right)$$

where g is a function of the gradients (Taylor series and all that joy) of the continuous solution, and h is the step size (e.g. Δx, Δy or Δt).

The trick to compute the estimation for the error is by comparing the solutions for two different meshes: one fine mesh h_1 and one coarser mesh h_2, with mesh spacings related by the ratio:

$$r = \frac{h_2}{h_1}$$

so that we get (if we set $h = h_1$)

$$\begin{aligned} f_1 &= f_e + gh^p + \mathcal{O}\left(h^{p+1}\right) \\ f_2 &= f_e + g\,(rh)^p + \mathcal{O}\left((rh)^{p+1}\right) \end{aligned}$$

If we neglect higher order terms, these equations can be solved for the exact solution:

$$f_e = f_1 + \frac{f_1 - f_2}{r^p}$$

so that the error of the discretisation error is

ESTIMATION OF DISCRETISATION ERROR

$$\epsilon = f_e - f_1 = \frac{f_1 - f_2}{r^p - 1} \tag{4.1}$$

In other words: to estimate the discretisation error (at a given point), then we calculate the numerical estimations for that point f_1 and f_2 using two different mesh spacings, h_1 and h_2, of which the ratio is h_2/h_1; using the expected order of accuracy p, we can estimate the discretisation error.

Now, in above derivation, we assumed that we were in the asymptotic region (as we assumed that higher order terms were negligible). However, this may not be the case: in that case, the **observed order of accuracy** will be different. Let's find a way to calculate the observed order of accuracy, so that we can check whether this is close the expected order of accuracy. We introduce a third mesh with spacing h_3 where $h_3 = rh_2$, so that we get

$$\begin{aligned} f_1 &= f_e + gh^{p_o} + \mathcal{O}\left(h^{p_o+1}\right) \\ f_2 &= f_e + g\left(rh\right)^{p_o} + \mathcal{O}\left((rh)^{p_o+1}\right) \\ f_3 &= f_e + g\left(r^2h\right)^{p_o} + \mathcal{O}\left(\left(r^2h\right)^{p_o+1}\right) \end{aligned}$$

which, neglecting higher-order terms, can be solved for the observed order of accuracy:

OBSERVED ORDER OF ACCURACY

The observed order of accuracy is equal to

$$p_0 = \frac{\ln\left(\frac{f_3-f_2}{f_2-f_1}\right)}{\ln\left(r\right)} \tag{4.2}$$

If p_0 differs substantially from the expected order of accuracy, the use of Richardson extrapolation is invalid. Occasionally, it is necessary to compute the observed order of accuracy from solutions on meshes which are not related by the same grid spacing ratio, i.e. $r_{12} = \frac{h_2}{h_1}$ and $r_{23} = \frac{h_3}{h_2}$. In that case, we must solve

$$\frac{f_3 - f_2}{r_{23}^{p_0} - 1} = r_{12}^{p_0}\left(\frac{f_2 - f_1}{r_{12}^{p_o} - 1}\right)$$

by iterative procedures.

Quiz 2: Question 11

Computing the observed order of accuracy from solutions on three meshes with spacing Δx, $2\Delta x$, and $3\Delta x$

- can be done with an explicit formula
- requires the solution of a system of two linear equations
- requires the solution of a system of three linear equations
- requires the solution of a non-linear algebraic equation

The correct answer is **requires the solution of a non-linear algebraic equation**. Literally what I just explained. In this case, r is not constant (it's 2 for one, and 1.5 for the other). You then need to solve the non-linear algebraic equation I listed just before this example.

Quiz 2: Question 12

A sequence of computations using a third-order method on different meshes produces the following data for average shear stress τ_{avg}:

Mesh	h	τ_{avg}
1	2	1.048
2	4	1.237
3	8	2.479
4	16	14.01

where h is the mesh node spacing.

1. An approximation for the observed order of accuracy obtained using the three coarsest meshes is (three decimal places):
2. Now assuming that the observed order of accuracy is 3, the best estimate for the exact value of τ_{avg} is (three decimal places):

Hints:

$$p_0 = \frac{\ln\left(\frac{f_3 - f_2}{f_2 - f_1}\right)}{\ln(r)}, \qquad \epsilon = \frac{f_1 - f_2}{r^p - 1}$$

For question 1., we take meshes 2, 3 and 4 since they are the coarest meshes. Then, $r = 2$ and we get

$$p_0 = \frac{\ln\left(\frac{14.01 - 2.749}{2.749 - 1.237}\right)}{\ln(2)} = 2.897$$

For question 2., we simply get (using the meshes 1 and 2 as they are the finest meshes)

$$\epsilon = \frac{1.237 - 1.048}{2^3 - 1} = 0.027$$

so that the best estimate for $\tau_{avg} = f_1 - \epsilon = 1.048 - 0.027 = 1.021$. If you're not sure whether you should subtract or add the error: note that τ_{avg} decreases as we increase the mesh refinement. Thus, we'd expect the exact value to be even smaller than 1.048.

4.2.1 *Artificial boundary studies*

Truncating physical domains results in model errors. People study this.

5 Discretisation with spectral and finite-element methods

In chapter 3, we used finite-difference methods to find solutions. In this chapter, we'll discuss new methods that are very similar in principle: spectral methods and finite-element methods. Finite-difference methods have the disadvantage that they are complex to implement, as the stencil may vary throughout the mesh (and if the domain is shaped weirdly it also becomes difficult to implement).

5.1 Approximating the solution with functions

It makes sense that the solution to a PDE could be written as a combination of pre-specified basis functions, i.e.

$$\hat{u}(x) = \sum_{i=1}^{N} a_i \phi_i(x)$$

For example, one could use

$$\begin{aligned} \phi_1(x) &= 1 \\ \phi_2(x) &= x \\ \phi_3(x) &= x^2 \\ \phi_4(x) &= x^3 \end{aligned}$$

etc. to approximate the solution (it should make sense that this could be a way find solutions; the coefficients a_i would be determined at a later stage). These functions could either span the entire domain, or only locally, as depicted in figure 5.1.

SPECTRAL AND FINITE-ELEMENT METHODS

Spectral methods use global functions that span the entire domain.

Finite-element methods (FEM) use local functions that only span part of the domain.

In both methods, the a_i are referred to as **degrees of freedom**.

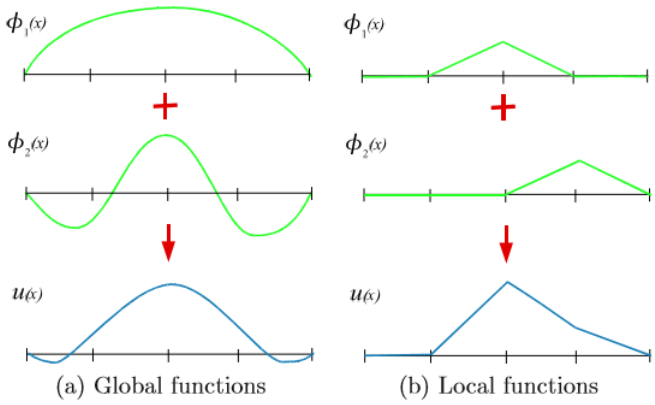


Figure 5.1: Approximation using global and local functions.

So, in finite-element and spectral methods, we need to determine a_i. We'll discuss several methods to do so in subsequent sections; one elementary method you've already seen in use is the **Rayleigh-Ritz approach**. In structural analysis, we used a combination of basis functions to estimate the deflection under buckling (e.g. $u(z) = a_1 + a_2 z + a_3 z^2 + a_4 z^3 + a_5 z^4$); we then set-up the equation for the potential energy of the beam under buckling. This potential energy is a **functional**, as it itself is a function of a function (the beam deflection $u(z)$) and let it be denoted by $I[u]$ (note the square brackets)[1]. If you remember correctly, the potential energy had to be minimized; this means that the derivatives of $I[u]$ with respect to any quantity should be zero:

$$\frac{\partial I[u]}{\partial a_i} = 0, \quad i = 1, ..., 5$$

yielding 5 equations with 5 unknowns, meaning we could, in principle, solve this system.

In many applications, it is difficult to come up with a functional, which is why we'll spend the rest of this chapter to discuss other methods to find a_i.

5.2 *Method of weighted residuals*

A common approach is the **method of weighted residuals** (MWR). To make explaining it a bit easier, suppose we have the following sample problem on the 1D Ω spanning the region from $x = 0$ to $x = 1$:

$$\begin{aligned} u_{xx} &= f \quad \text{on} \quad \Omega \\ u(0) &= g \\ u_x(1) &= q \end{aligned}$$

The first equation is referred to as the **strong form** of the governing equation. The boundary conditions are Dirichlet respectively Neumann boundary conditions. Suppose we'd want to approximate the solution using 4 basis functions, so that we have 4 degrees of freedom (i.e. a_1 to a_4). Our aim is to find these 4 coefficients.

Now, the trick is that we define another form of the governing equation, namely the **weak form**, which uses the method of weighted residuals. To be precise, the residual for this governing equation is $R(x) = u_{xx} - f$; we then require

$$\int_0^1 w(x) \left(u_{xx} - f \right) dx = 0$$

Now, what the fuck just happened? We have our numerical solution, u, which consists of the 4 basis functions with coefficients a_1, a_2, a_3 and a_4. These will produce errors/residuals equal to $u_{xx} - f$ since that's basically our PDE. The method of weighted residuals then states that if you integrate these residuals over the domain, using a weighting function $w(x)$, the result should be zero (i.e. a_1, a_2, a_3 and a_4 need to be determined such that that is indeed the result). Why exactly do we call $w(x)$ a weighting function? Because it essentially weights the residuals; suppose we'd have the weighting function $w(x) = x$, then residuals close to $x = 1$ would count more than the residuals close to $x = 0$ when determining the value of the integral. Of course, this will not lead to your numerical solution being the exact solution. But, we are using a numerical solution anyway: it is unreasonable to expect that it'll be able to perfectly describe the solution anyway. Using this method of weighted residuals will at least result in a solution that comes relatively close: when you 'average' the error in certain ways (as described by $w(x)$), the outcome should be 0. This is something reasonable to expect from your numerical solution, you'll agree with me.

Now, $w(x)$ is chosen by yourself, so suppose you choose a certain function for it and you calculate the integral: this will give you one equation with four unknowns: a_1, a_2, a_3 and a_4. So, in fact, you have to choose four functions $w(x)$. You'd then get four different integrals and four different equations, meaning you can solve for a_1, a_2, a_3 and a_4. This method of using N weighting functions to determine N degrees of freedom is called the **Galerkin method**.

[1] We use $I[u]$ often to denote functionals.

5.3 Equivalence of strong and weak forms

Just to assure you: solutions that are a solution of the strong form will satisfy the weak form of the governing equation. Furthermore, you can prove that there won't be any solutions that satisfy the weak form of the equation but don't satisfy the strong form (so don't worry about that).

5.4 The Bubnov-Galerkin method

Now, although you probably have no trouble understanding that $\phi_1(x)$, $\phi_2(x)$ etc. are just functions you choose yourself, how do we determine $w_1(x)$, $w_2(x)$ etc.? Well, again, it is something you pick yourself in principle; however, the popular **Bubnov-Galerkin method** uses the basis function as weighting functions, i.e. $w_i = \phi_i(x)$. The reason why this method is popular is that they maximise the reuse of information and produce symmetric matrices for certain classes of problems. In case you want to see it in practice, suppose we have once more those four basis functions: how would our system of equations look like, if we use $w_i = \phi_i(x)$? Disregarding boundaries, you'd get the following equations: we have

$$\int_0^1 w_i(x)\left(u_{xx} - f\right) dx = 0$$

where $u_{xx} = a_1\phi_{1_{xx}} + a_2\phi_{2_{xx}} + a_3\phi_{3_{xx}} + a_4\phi_{4_{xx}}$. Thus, we'd get (since $w_i(x) = \phi_i(x)$)

$$\int_0^1 w_i(x)\left(a_1\phi_{1_{xx}} + a_2\phi_{2_{xx}} + a_3\phi_{3_{xx}} + a_4\phi_{4_{xx}} - f\right) dx = 0$$

which can be written as (for clarification, $\phi_{1_{xx}}$ etc. and f are all functions of x as well):

$$\begin{aligned}
a_1\int_0^1 \phi_1(x)\phi_{1_{xx}}dx + a_2\int_0^1 \phi_1(x)\phi_{2_{xx}}dx + a_3\int_0^1 \phi_1(x)\phi_{3_{xx}}dx + a_4\int_0^1 \phi_1(x)\phi_{4_{xx}}dx &= \int_0^1 \phi_1(x) f dx \\
a_1\int_0^1 \phi_2(x)\phi_{1_{xx}}dx + a_2\int_0^1 \phi_2(x)\phi_{2_{xx}}dx + a_3\int_0^1 \phi_2(x)\phi_{3_{xx}}dx + a_4\int_0^1 \phi_2(x)\phi_{4_{xx}}dx &= \int_0^1 \phi_2(x) f dx \\
a_1\int_0^1 \phi_3(x)\phi_{1_{xx}}dx + a_2\int_0^1 \phi_3(x)\phi_{2_{xx}}dx + a_3\int_0^1 \phi_3(x)\phi_{3_{xx}}dx + a_4\int_0^1 \phi_3(x)\phi_{4_{xx}}dx &= \int_0^1 \phi_3(x) f dx \\
a_1\int_0^1 \phi_4(x)\phi_{1_{xx}}dx + a_2\int_0^1 \phi_4(x)\phi_{2_{xx}}dx + a_3\int_0^1 \phi_4(x)\phi_{3_{xx}}dx + a_4\int_0^1 \phi_4(x)\phi_{4_{xx}}dx &= \int_0^1 \phi_4(x) f dx
\end{aligned}$$

Note that the integrals can all be calculated! Thus, this is a simple linear 4×4 system which can be straightforwardly calculated. *This is the basic idea behind what you've seen so far.*

5.5 Suitable choices for weighting functions

Now, there is one limitation so far: u_{xx} meant that the solution (i.e. the basis functions) had to be differentiated twice. However, if we use piece-wise defined linear basis functions, then this is problematic: we'd get $u_{xx} = 0$ everywhere, except for the kinks in the curve where it'd be ∞. However, there's a way to circumvent it, by using

integration by parts:

$$\begin{aligned}\int_0^1 w\left(u_{xx}-f\right)dx = \int_0^1 wu_{xx}dx - \int_0^1 wf\,dx &= 0\\ wu_x|_0^1 - \int_0^1 w_x u_x dx - \int_0^1 wf\,dx &= 0\end{aligned}$$

INTEGRATED BY PARTS WEAK FORM OF GOVERNING EQUATION

$$wu_x|_0^1 - \int_0^1 w_x u_x dx - \int_0^1 wf\,dx = 0 \tag{5.1}$$

This equation only requires us to evaluate first derivatives, which are less problematic than second derivatives. If you forgot how to do integration by parts:

INTEGRATION BY PARTS FOR THE WEAK FORM

Starting with an integral in the form of

$$\int_a^b uv' \tag{5.2}$$

Choose the highest order derivative to be v', and let the other term be u. Then, integrate v' twice, and differentiate u only the second time, to arrive at

$$\int_a^b uv' = [uv]_a^b - \int_a^b u'v \tag{5.3}$$

Quiz 2: Question 3

Consider the equation:

$$u_x - vu_{xx} = f \quad \text{on} \quad 0 < x < 1; \qquad u(0) = u(1) = 0$$

It would be useful to use integration by parts in the derivation of the weak form for a Galerkin discretisation of this equation because:

- the continuity requirements for the test and solution basis functions would be similar
- it would simplify the application of Dirichlet boundary conditions
- the resulting matrices would be unsymmetric
- it would ensure that the test and solution basis functions are equal

---

Correct is **the continuity requirements for the test and solution basis functions would be similar**. This is in general true, irrespective of the governing equation. If you want to see how we'd integrate this equation by parts, look at the following: the weak form would be

$$\int_0^1 w\left(u_x - vu_{xx} - f\right)dx = \int_0^1 wu_x dx - v\int_0^1 wu_{xx}dx - \int_0^1 wf\,dx$$

Now, you'll agree with me that integrating the first and third term by parts is pretty useless: you'd end up with the derivative of w instead of the derivative of u (and integrating f is even worse of an idea), which doesn't help much. Therefore, we only have to integrate the term in the middle by parts:

$$v\int_0^1 wu_{xx}dx = v\left(\left[wu_x\right]_0^1 - \int_0^1 w_x u_x dx\right)$$

and thus we get

$$\begin{aligned}\int_0^1 w\left(u_x - vu_{xx} - f\right)dx &= \int_0^1 wu_x dx - v\left(\left[wu_x\right]_0^1 - \int_0^1 w_x u_x dx\right) - \int_0^1 wf\,dx \\ &= -v\left[wu_x\right]_0^1 + \int_0^1 w\left(u_x - f\right)dx + \int_0^1 w_x u_x dx\end{aligned}$$

Quiz 2: Question 13

Consider the following problem:

$$u_x - u = f \qquad \text{on} \quad 0 < x < 1; \qquad u(0) = u(1) = 0$$

Defining w as an arbitrary weighting function, and assuming essential Dirichlet boundary conditions are used, an integrated-by-parts weak form of the problem is given by

$$\int_0^1 (M)\,dx = 0$$

where the quantity M equals:

We have that the weak form of this PDE is

$$\int_0^1 w\left(u_x - u - f\right)dx = \int_0^1 wu_x dx - \int_0^1 wu dx - \int_0^1 wf\,dx = 0$$

Now, applying integration by parts to the second and third integral is useless since there are no derivatives in those integrals (and thus it is not expected that you integrate those integrals by parts. However, for the first integral, we get

$$\int_0^1 wu_x dx = [wu]_0^1 - \int_0^1 w_x u dx$$

However, due to the Dirichlet boundary at $x = 0$ and $x = 1$, the left term completely reduces to zero. Thus, we get

$$\begin{aligned}\int_0^1 wu_x dx - \int_0^1 wu dx - \int_0^1 wf\,dx &= 0 \\ -\int_0^1 w_x u dx - \int_0^1 wu dx - \int_0^1 wf\,dx &= 0 \\ \int_0^1 \left(w_x u + wu + wf\right) &= 0\end{aligned}$$

Write this in Maple as Diff(w,x)*u+w*u+w*f.

Quiz 2: Question 14

Integration by parts is used to
- lower function continuity requirements
- make a Galerkin method
- make the final matrices less symmetric
- make a Bubnov-Galerkin method

The correct answer is **lower function continuity requirements**: you integrate u_{xx} beforehand so that you only have to plug in u_x. This means that the solution only needs to be differentiable once before it is allowed to reduce to zero. Thus, the function continuity requirements are lowered.

Now, what are suitable weighting and basis functions? For them, it is required that the integral from 0 to 1 remains integrable. For example, suppose we'd have $u_x = \frac{1}{x^2}$ and $w = x$, then

$$\int_0^1 w_x u_x dx$$

would go towards infinity (you'd integrate $1/x$, which integrates to infinity). For a Bubnov-Galerkin discretisation, where $w = \phi$ and thus $w_x = \phi_x$, the set of suitable functions are those which have

$$\int_0^1 \left(\phi_x\right)^2 dx < \infty$$

Note that the square of the derivative can be a stronger restriction than integrability of the function itself. For example, $\phi_x = 1/\sqrt{x}$ is integrable from 0 to 1 (its integral equals 2), but $\left(\phi_x\right)^2 = 1/x$ is not integrable from 0 to 1 (its integral goes towards infinity).

5.6 *An example spectral method*

Now, let's apply the spectral method to the sample problem listed earlier:

$$\begin{aligned} u_{xx} &= f \quad \text{on} \quad \Omega \\ u(0) &= g \\ u_x(1) &= q \end{aligned}$$

where we try to include the boundary conditions later on as well. Let us use the basis functions

$$\begin{aligned} \phi_1(x) &= 1 - x \\ \phi_2(x) &= x \\ \phi_3(x) &= x - x^2 \\ \phi_4(x) &= -2x + 6x^2 - 4x^3 \end{aligned}$$

which are shown in figure 5.2, resulting in

$$\hat{u}(x) = \sum_{i=1}^{4} a_i \phi_i(x)$$

We will use a Bubno-Galerkin method for which $w_i = \phi_i$.

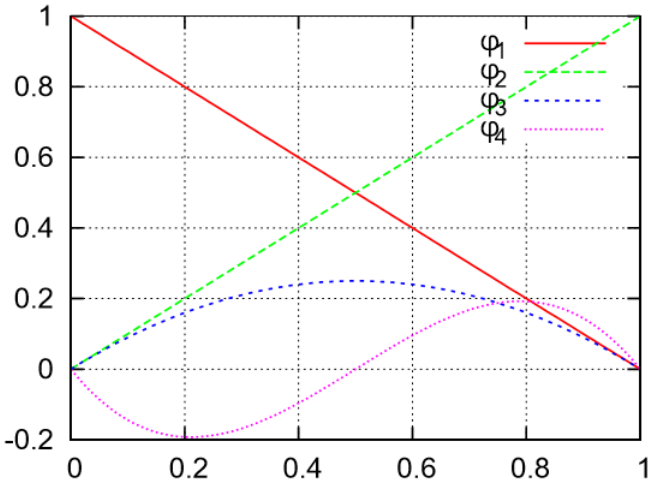


Figure 5.2: Modal basis functions.

5.6.1 *The basic system*

Without imposing boundary conditions yet, our basic system of equations would follow from

$$w_i u_x|_0^1 - \int_0^1 w_{i_x} u_x dx - \int_0^1 w_i f dx = 0$$

Now, regarding the first term: note that $w_3 = \phi_3$ and $w_4 = \phi_4$ are both zero at both $x = 0$ and $x = 1$ (look at figure 5.2); thus, for the equation where you use w_3 or w_4, this term completely disappears. Furthermore, $w_1 = \phi_1$ is only non-zero at $x = 0$ (but zero at $x = 1$), whereas $w_2 = \phi_2$ is non-zero at $x = 1$ (but zero at $x = 0$). This means we end up at the system (look at it carefully, you'll agree with me that this is correct; I haven't substituted $w_1 = \phi_1$ etc. yet):

$$\begin{aligned}
-w_1(0)\,u_x(0) - \int_0^1 w_{1_x} \sum_{i=1}^4 a_i \phi_{i_x} dx &= \int_0^1 w_1 f dx \\
w_2(1)\,u_x(1) - \int_0^1 w_{3_x} \sum_{i=1}^4 a_i \phi_{i_x} dx &= \int_0^1 w_2 f dx \\
- \int_0^1 w_{3_x} \sum_{i=1}^4 a_i \phi_{i_x} dx &= \int_0^1 w_3 f dx \\
- \int_0^1 w_{4_x} \sum_{i=1}^4 a_i \phi_{i_x} dx &= \int_0^1 w_4 f dx
\end{aligned}$$

Now, let's apply the boundary conditions.

5.6.2 *Dirichlet boundary condition*

Dirichlet boundary conditions can easily be implemented. In the example, we have $u(0) = g$. With the basis functions given, we see that only ϕ_1 is nonzero at $x = 0$, thus $u(0) = a_1 = g$. We then replace the weighting function associated with ϕ_1 (i.e. the one where you use $w_1 = \phi_1$) from the previous system with this equation,

i.e. we get

$$\begin{aligned} a_1 &= g \\ w_2(1)\,u_x(1) - \int_0^1 w_{3_x} \sum_{i=1}^{4} a_i \phi_{i_x} dx &= \int_0^1 w_2 f\, dx \\ - \int_0^1 w_{3_x} \sum_{i=1}^{4} a_i \phi_{i_x} dx &= \int_0^1 w_3 f\, dx \\ - \int_0^1 w_{4_x} \sum_{i=1}^{4} a_i \phi_{i_x} dx &= \int_0^1 w_4 f\, dx \end{aligned}$$

Note that the first equation is very neat this time; this does not always have to be the case, it can also be an equation involving a_2, a_3 and a_4 as well, depending on what basis functions you used[2].

5.6.3 *Neumann boundary condition*

Believe it or not, Neumann Boundary conditions are implemented even more easily. We have $u_x = q$ at $x = 1$ (i.e. $u_x(1) = q$), which we can substitute easily into our equations:

$$\begin{aligned} a_1 &= g \\ w_2(1)\,q - \int_0^1 w_{3_x} \sum_{i=1}^{4} a_i \phi_{i_x} dx &= \int_0^1 w_2 f\, dx \\ - \int_0^1 w_{3_x} \sum_{i=1}^{4} a_i \phi_{i_x} dx &= \int_0^1 w_3 f\, dx \\ - \int_0^1 w_{4_x} \sum_{i=1}^{4} a_i \phi_{i_x} dx &= \int_0^1 w_4 f\, dx \end{aligned}$$

Now that you've seen the two types of boundary conditions applied, I'd like to point two things:

- Neumann boundary conditions are ridiculously easy to implement; they are therefore called **natural** boundary conditions.
- On the other hand, Dirichlet boundary conditions require other equations to fuck off; these additional equations are called **essential** boundary conditions.

[2] For example, had we had $\phi_1(0) = 1$, $\phi_2(0) = 2$, $\phi_3(0) = -1$ and $\phi_4(0) = 0$, we'd have had as first equation $a_1 + 2a_2 - a_3 = g$. You'd then have replaced either the first, second or third equation in the basic system with this condition.

5.6.4 *The final system*

Replacing w_i with ϕ_i and writing out the sums yields

$$\begin{aligned} a_1 &= g \\ \phi_2(1)\,q - \int_0^1 \phi_{2_x}\left(a_1\phi_{1_x} + a_2\phi_{2_x} + a_3\phi_{3_x} + a_4\phi_{4_x}\right)dx &= \int_0^1 \phi_2 f\,dx \\ -\int_0^1 \phi_{3_x}\left(a_1\phi_{1_x} + a_2\phi_{2_x} + a_3\phi_{3_x} + a_4\phi_{4_x}\right)dx &= \int_0^1 \phi_3 f\,dx \\ -\int_0^1 \phi_{4_x}\left(a_1\phi_{1_x} + a_2\phi_{2_x} + a_3\phi_{3_x} + a_4\phi_{4_x}\right)dx &= \int_0^1 \phi_4 f\,dx \end{aligned}$$

which can be expressed as the matrix equation

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ K_{21} & K_{22} & K_{23} & K_{24} \\ K_{31} & K_{32} & K_{33} & K_{34} \\ K_{41} & K_{42} & K_{43} & K_{44} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} g \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}$$

where all terms like

$$\begin{aligned} K_{23} &= -\int_0^1 \left[\phi_{2_x}\phi_{3_x}(x)\right]dx \\ F_2 &= \int_0^1 \left[\phi_2(x)\,f(x)\right]dx - \phi_2(1)\,q \end{aligned}$$

being straightforward to calculate. With the exception of the first row, which contains an essential condition, each row in the K matrix corresponds to a test of the weak form with a single weighting function: the numerical solution u, consisting of the basis functions, is tested to see whether the weighted residuals indeed integrate to zero; it is for this reason that the weighting functions are often referred to as **testing functions**. Furthermore, K is often referred to as **stiffness matrix**. The matrix on the right is referred to as the F matrix.

Regarding the properties of the matrix: $\phi_i(x)$ must be linearly independent, otherwise no solution exists. This means that you should not be able to write any $\phi_i(x)$ as linear combination of the others. Furthermore, if we have an orthogonal basis, i.e. one for which

$$\int_\Omega \left[\phi_i(x)\,\phi_j(x)\right]d\Omega = 0; \quad i \neq j$$

the resulting K matrix will be diagonal and really easy to invert, yaay. An example of an orthogonal basis would for example be

$$\phi_n = \cos(2n\pi x), \quad 0 \leq n \leq N$$

over the domain $0 \leq x \leq 1$. Indeed, in reality, for spectral methods, often trigonometric series are used (at least according to Wikipedia).

5.6.5 *Observation and results*

For the sample problem, let $g = 0$, $q = \frac{1}{2}$ and $f = x$. We get the results shown in figure 5.3: if we only use the first two basis functions we get the solution of the dashed green line; if we include the third basis function as

well we get the dashed red line. If we include the fourth basis function as well, we get the thick light blue line; we see that this line exactly goes through the exact solution given by the blue diamonds. The reason for this is that the exact solution is given by $u(x) = qx + \frac{1}{6}\left(x^3 - 3x\right)$; if we include the first four basis functions, we are able to exactly replicate this polynomial using the numerical methods shown above (much like interpolation becomes exact when you are simply interpolating an nth order polynomial using $n+1$ nodes (and thus $n+1$ basis functions)). Ending up at the exact solution without error is called **superconvergence**.

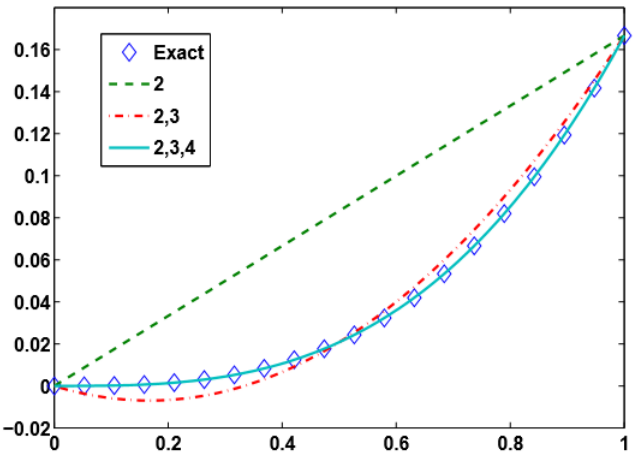


Figure 5.3: Spectral solution using an increasing number of basis functions from the modal p-type expansion.

Quiz 2: Question 17

Superconvergence occurs in spectral methods when

- the exact solution can be expressed as a combination of the basis functions
- basis functions which are infinitely differentiable are used
- the rate of convergence is one higher than the highest polynomial order of the basis functions
- the weak form is tested with an infinite number of basis functions

The correct answer is **the exact solution can be expressed as a combination of the basis functions**. See above discussion.

5.7 *An example finite-element method*

The example shown above uses the spectral method: it uses basis functions that span the entire domain. However, finite-element methods uses basis functions that span only part of the domain. At the end of this section, I'll give you the reasons why you'd want to do this, but let me first show you how it works. It consists of five steps:

1. Divide the domain into elements
2. Define basis functions which span a small number of elements
3. Compute per-element contributions to the weak form
4. Assemble the element contributions into a global matrix
5. Apply the boundary conditions and solve the resulting system

We will consider each of these steps in turn by solving our sample problem from before.

Step 1 We split the domain into three elements, each with equal width $h = 1/3$ (unequal element sizes are also allowed; for complex domains, we let a computer determine the grid).

Step 2 We define four basis functions, as shown in figure 5.4.

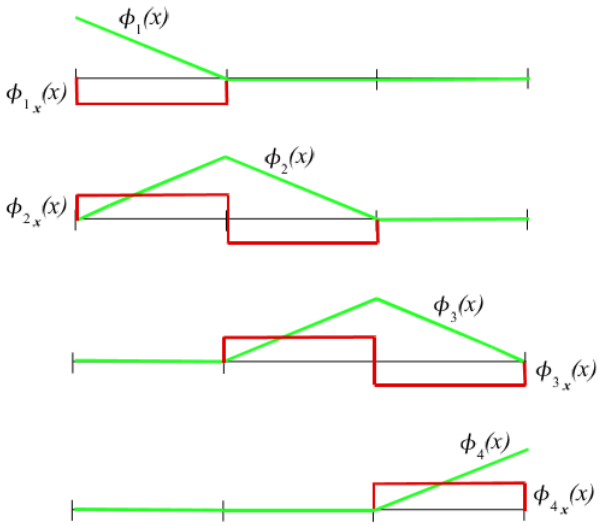


Figure 5.4: Basis functions (green) for a three-element discretisation of the sample problem. The derivatives of the basis functions are shown in red (we'll use them later).

Steps 3 and 4 This is the difficult step. Note that we now essentially have three domains on which we can apply the spectrum method. For example, if we focus on the first subdomain, $0 \leq x \leq \frac{1}{3}$, on which only $\phi_1(x)$ and $\phi_2(x)$ act. We'll use

$$w_i u_x|_0^{1/3} - \int_0^{1/3} w_{i_x} u_x dx - \int_0^{1/3} w_i f dx = 0$$

Then for the second subdomain, we'd use the same equation but then with limits from $\frac{1}{3} \leq x \leq \frac{2}{3}$ and finally the third subdomain we'd be using $\frac{2}{3} \leq x \leq 1$. Now, before I start working it out, I'd like you to realize that the first term can basically be excluded from our calculations. The reason for this is that if you include $w_i u_x|_0^{1/3}$ in the first subdomain, then $w_i u_x|_{1/3}^{2/3}$ for the second one and $w_i u_x|_{2/3}^{1}$ for third, and then you 'add them up', you essentially got $w_i u_x|_0^1$. What's the point I'm trying to make? $w_i u_x|_0^1$ only matters for the outer boundaries of the entire domain. We won't worry about boundaries right now, so I'll ignore them for now, meaning that I'll rather use

$$\int_0^{1/3} w_{i_x} u_x dx = -\int_0^{1/3} w_i f dx$$

to come up with the equations for the first element. Remember what I told you: only $\phi_1(x)$ and $\phi_2(x)$ acted on this element, meaning we get (with $w_1 = \phi_1$ and $w_2 = \phi_2$; due to space limitations, I'll leave out the arguments of the functions):

$$\begin{aligned} w_1 = \phi_1: \quad & \int_0^{1/3} \phi_{1_x} \left(a_1 \phi_{1_x} + a_2 \phi_{2_x} \right) dx = -\int_0^{1/3} \phi_1 f dx \\ w_2 = \phi_2: \quad & \int_0^{1/3} \phi_{2_x} \left(a_1 \phi_{1_x} + a_2 \phi_{2_x} \right) dx = -\int_0^{1/3} \phi_2 f dx \end{aligned}$$

or more elegantly put, the system

$$w_1 = \phi_1 : \quad a_1 \int_0^{1/3} \phi_{1_x}\phi_{1_x} dx + a_2 \int_0^{1/3} \phi_{1_x}\phi_{2_x} dx = -\int_0^{1/3} \phi_1 f\, dx \tag{5.4}$$

$$w_2 = \phi_2 : \quad a_1 \int_0^{1/3} \phi_{2_x}\phi_{1_x} dx + a_2 \int_0^{1/3} \phi_{2_x}\phi_{2_x} dx = -\int_0^{1/3} \phi_2 f\, dx \tag{5.5}$$

$$K_e a = F_e \tag{5.6}$$

$$\begin{bmatrix} \int_0^{1/3} \phi_{1_x}\phi_{1_x} dx & \int_0^{1/3} \phi_{1_x}\phi_{2_x} dx \\ \int_0^{1/3} \phi_{2_x}\phi_{1_x} dx & \int_0^{1/3} \phi_{2_x}\phi_{2_x} dx \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} -\int_0^{1/3} \phi_1 f\, dx \\ -\int_0^{1/3} \phi_2 f\, dx \end{bmatrix} \tag{5.7}$$

Now, in a very similar way, we get for the element in the middle

$$w_2 = \phi_2 : \quad a_2 \int_{1/3}^{2/3} \phi_{2_x}\phi_{2_x} dx + a_3 \int_{1/3}^{2/3} \phi_{2_x}\phi_{3_x} dx = -\int_{1/3}^{2/3} \phi_2 f\, dx \tag{5.8}$$

$$w_3 = \phi_3 : \quad a_2 \int_{1/3}^{2/3} \phi_{3_x}\phi_{2_x} dx + a_3 \int_{1/3}^{2/3} \phi_{3_x}\phi_{3_x} dx = -\int_{1/3}^{2/3} \phi_3 f\, dx \tag{5.9}$$

$$K_e a = F_e \tag{5.10}$$

$$\begin{bmatrix} \int_{1/3}^{2/3} \phi_{2_x}\phi_{2_x} dx & \int_{1/3}^{2/3} \phi_{2_x}\phi_{3_x} dx \\ \int_{1/3}^{2/3} \phi_{3_x}\phi_{2_x} dx & \int_{1/3}^{2/3} \phi_{3_x}\phi_{3_x} dx \end{bmatrix} \begin{bmatrix} a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} -\int_{1/3}^{2/3} \phi_2 f\, dx \\ -\int_{1/3}^{2/3} \phi_3 f\, dx \end{bmatrix} \tag{5.11}$$

Finally, the third element would become

$$w_3 = \phi_3 : \quad a_3 \int_{1/3}^{2/3} \phi_{3_x}\phi_{3_x} dx + a_4 \int_{1/3}^{2/3} \phi_{3_x}\phi_{4_x} dx = -\int_{2/3}^{1} \phi_3 f\, dx \tag{5.12}$$

$$w_4 = \phi_4 : \quad a_3 \int_{1/3}^{2/3} \phi_{4_x}\phi_{3_x} dx + a_4 \int_{1/3}^{2/3} \phi_{4_x}\phi_{4_x} dx = -\int_{2/3}^{1} \phi_4 f\, dx \tag{5.13}$$

$$K_e a = F_e \tag{5.14}$$

$$\begin{bmatrix} \int_{2/3}^{1} \phi_{3_x}\phi_{3_x} dx & \int_{2/3}^{1} \phi_{3_x}\phi_{4_x} dx \\ \int_{2/3}^{1} \phi_{4_x}\phi_{3_x} dx & \int_{2/3}^{1} \phi_{4_x}\phi_{4_x} dx \end{bmatrix} \begin{bmatrix} a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} -\int_{2/3}^{1} \phi_3 f\, dx \\ -\int_{2/3}^{1} \phi_4 f\, dx \end{bmatrix} \tag{5.15}$$

Now, one way to combine these matrices would of course be to write the following (I had to use two rows due to space, but you get what I mean, it should just be one row (also there are some dx missing but I don't have space

for them))

$$\begin{bmatrix} \int_0^{1/3} \phi_{1_x}\phi_{1_x}dx & \int_0^{1/3} \phi_{1_x}\phi_{2_x}dx & 0 & 0 & 0 & 0 \\ \int_0^{1/3} \phi_{2_x}\phi_{1_x}dx & \int_0^{1/3} \phi_{2_x}\phi_{2_x}dx & 0 & 0 & 0 & 0 \\ 0 & 0 & \int_{1/3}^{2/3} \phi_{2_x}\phi_{2_x}dx & \int_{1/3}^{2/3} \phi_{2_x}\phi_{3_x}dx & 0 & 0 \\ 0 & 0 & \int_{1/3}^{2/3} \phi_{3_x}\phi_{2_x}dx & \int_{1/3}^{2/3} \phi_{3_x}\phi_{3_x}dx & 0 & 0 \\ 0 & 0 & 0 & 0 & \int_{2/3}^{1} \phi_3(x)\phi_3(x)dx & \int_{2/3}^{1} \phi_3(x)\phi_4(x)dx \\ 0 & 0 & 0 & 0 & \int_{2/3}^{1} \phi_4(x)\phi_3(x)dx & \int_{2/3}^{1} \phi_4(x)\phi_4(x)dx \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_2 \\ a_3 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} -\int_0^{1/3} \phi_1 f dx \\ -\int_0^{1/3} \phi_2 f dx - \int_{1/3}^{2/3} \phi_2 f dx \\ -\int_{1/3}^{2/3} \phi_3 f dx - \int_{2/3}^{1} \phi_3 f dx \\ -\int_{2/3}^{1} \phi_4 f dx \end{bmatrix}$$

but you'll agree that this is a bit weird. Indeed, we realize that in the vector $\mathbf{a}$, the a_2 and a_3 both appear double: this means that we can actually 'merge' the second and third row of this matrix equation together, as well as the fourth and fifth row. In the K matrix, this means we need to merge the second and third column, and fourth and fifth column at the same time as well. This means we actually end up at (again, I couldn't write out all of the dx at the end of the integrals due to space limitations)

$$\begin{bmatrix} \int_0^{1/3} \phi_{1_x}\phi_{1_x}dx & \int_0^{1/3} \phi_{1_x}\phi_{2_x}dx & 0 & 0 \\ \int_0^{1/3} \phi_{2_x}\phi_{1_x}dx & \int_0^{1/3} \phi_{2_x}\phi_{2_x}dx + \int_{1/3}^{2/3} \phi_{2_x}\phi_{2_x}dx & \int_{1/3}^{2/3} \phi_{2_x}\phi_{3_x}dx & 0 \\ 0 & \int_{1/3}^{2/3} \phi_{3_x}\phi_{2_x}dx & \int_{1/3}^{2/3} \phi_{3_x}\phi_{3_x}dx + \int_{2/3}^{1} \phi_3(x)\phi_3(x)dx & \int_{2/3}^{1} \phi_3(x)\phi_4(x) \\ 0 & 0 & \int_{2/3}^{1} \phi_4(x)\phi_3(x)dx & \int_{2/3}^{1} \phi_4(x)\phi_4(x) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} -\int_0^{1/3} \phi_1(x) f dx \\ -\int_0^{1/3} \phi_2(x) f dx - \int_{1/3}^{2/3} \phi_2(x) f dx \\ -\int_{1/3}^{2/3} \phi_3(x) f dx - \int_{2/3}^{1} \phi_3(x) f dx \\ -\int_{2/3}^{1} \phi_4(x) f dx \end{bmatrix}$$

This can then straightforwardly be solved for a_1, a_2, a_3 and a_4 (okay we still have to implement those boundary conditions, but we'll do that in step 5). Now, compare this big matrix with the three smaller 2×2 matrices (called **element matrices**). I hope you see that these matrices are rather easily combined to a **global matrix**;

we can visualize the process as shown in figure 5.5. Basically, the element matrices sort of overlap each time and are then easily combined. Constructing the global matrix from element matrices is called **assembly**.

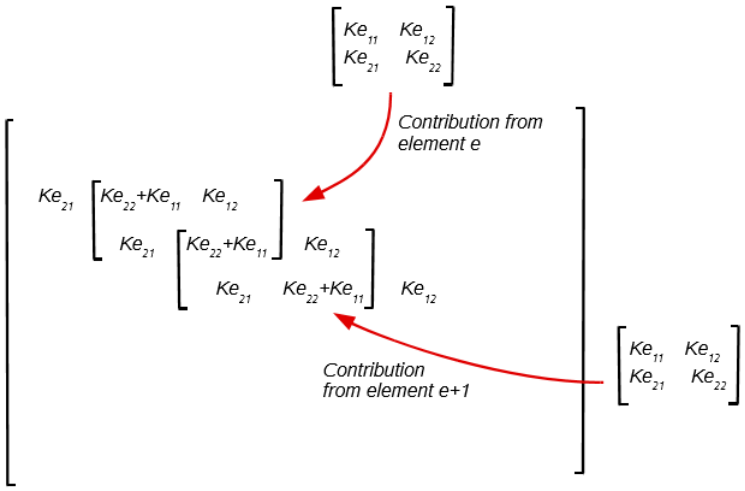


Figure 5.5: Overlapping element arrays in global matrix assembly.

5.7.1 *Transformation*

Now, although everything so far has been flawless, there is an important remark to make. In this case, our elements are very easy: they're just straight lines with the same length for all of them. However, most large finite-element codes use a limited number of element types (e.g. triangle, hexahedron, etc.) to which all elements in a mesh can be topologically related. Therefore, we use a master element: one fixed element for which the geometry is fixed, and we will transform all other elements, whether they are triangles, hexahedrons, whatever, to this master element. Now, don't worry about how this works for 2D and higher dimensional elements: in this course, we'll only consider 1D cases: physically, this means that we may divide the domain into line segments with unequal widths, which we'll then each transform to be of the master element form: a line segment with width 1.

Let's take the example we're discussing before: let an element in physical space have length h and starts at x_1,[3] then we define $\xi = \frac{x-x_1}{h}$. x derivatives can then be obtained from the chain rule, i.e. $\phi_x = \phi_\xi \xi_x$, with $\xi_x = \frac{1}{h}$. To change the integration coordinates, note that $dx = hd\xi$.

Consider the three element matrices again; note that we can write them in general as

$$K_e = \begin{bmatrix} \int_{x_i}^{x_f} \phi_{L_x}(x)\phi_{L_x}(x)\,dx & \int_{x_i}^{x_f} \phi_{L_x}(x)\phi_{R_x}(x)\,dx \\ \int_{x_i}^{x_f} \phi_{R_x}(x)\phi_{L_x}(x)\,dx & \int_{x_i}^{x_f} \phi_{R_x}(x)\phi_{R_x}(x)\,dx \end{bmatrix}$$

where x_i was the initial x-value of the line segment and x_f the final x-value. Now, if we substitute

$$\phi_{L_x} = \phi_{L_\xi}\xi_x = \frac{\phi_{L_\xi}}{h}, \quad \phi_{R_x} = \phi_{R_\xi}\xi_x = \frac{\phi_{R_\xi}}{h}, \quad dx = hd\xi$$

we get

$$K_e = \begin{bmatrix} \frac{1}{h}\int_0^1 \phi_{L_\xi}(\xi)\phi_{L_\xi}(\xi)\,d\xi & \frac{1}{h}\int_0^1 \phi_{L_\xi}(\xi)\phi_{R_\xi}(\xi)\,d\xi \\ \frac{1}{h}\int_0^1 \phi_{R_\xi}(\xi)\phi_{L_\xi}(\xi)\,d\xi & \frac{1}{h}\int_0^1 \phi_{R_\xi}(\xi)\phi_{R_\xi}(\xi)\,d\xi \end{bmatrix}$$

[3] For all three elements we're considering, $h = \frac{1}{3}$ and $x_1 = 0, 1/3$ and $2/3$ respectively.

and similarly, the element F vector becomes

$$F_e = \begin{bmatrix} -h \int_0^1 \phi_L(\xi) f(\xi) d\xi \\ -h \int_0^1 \phi_R(\xi) f(\xi) d\xi \end{bmatrix}$$

These element matrices can then once again be assembled to get the correct result.

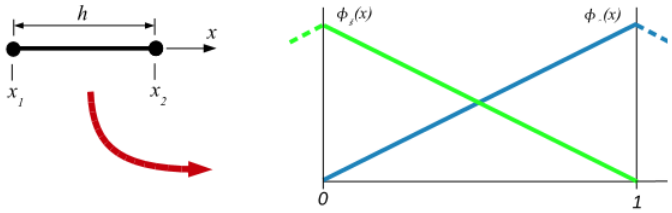


Figure 5.6: Physical element (left) and master element (right) with two basis functions, one centered on the left and one on the right.

Note the matrix (if it'd be larger especially) is very sparse: there are a lot of zeros in there. These matrices can be solved very efficiently using iterative or specialised direct techniques, which we'll discuss in the last chapter of the next quiz. Furthermore, note that the stiffness matrix for this specific PDE is symmetric. This is a consequence of the integrated-by-parts weak form combined with a Bubnov-Galerkin method. Symmetric matrices are easier to solve than unsymmetric ones.

Step 5 I said in the beginning that we wouldn't worry about boundaries until the very end, but fortunately, we've reached the very end now. Implementing the boundary conditions is very similar to how we did it for the spectral approach: the first row of the matrix equation is replaced with $a_1 = g$, and for the Neumann boundary condition at $x = 1$, we simply add $\phi_4(1)\, q$ to F_4.

The result of this is shown in figure 5.7. Remarkably, the solution is exact at some nodes. This type of superconvergence occurs due to the combination of piecewise linear basis functions with this particular weak form. Convergence rates for finite-elements of more general problems are discussed in the next section.

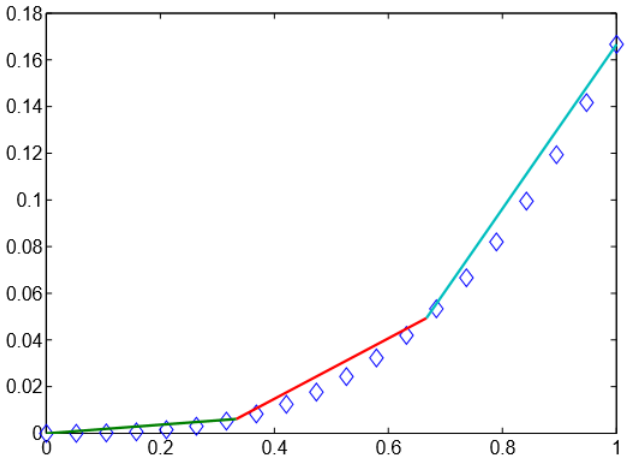


Figure 5.7: Finite-element solution of the sample problem using three elements.

Furthermore, compare figure 5.7 with figure 5.4, which showed the basis functions used. From comparison, we see that $\phi_1(x)$ was not actually used in the end: the numerical solution starts at $u = 0$ for $x = 0$, whereas $\phi_1(x)$ has a nonzero value for $x = 0$, meaning somehow we didn't end up using it. This is due to the Dirichlet

boundary: if there is a Dirichlet boundary applied and u is set equal to 0 at this boundary, then *all* basis functions will be zero at this boundary (if the basis function wasn't zero originally, it simply won't be used in the final solution, making it essentially 0). This in turn means, since we are using the Bubnov-Galerkin method, that all weighting functions are 0 at those boundaries as well[4].

Quiz 2: Question 15

To apply an essential Dirichlet boundary condition, the weighting functions on the boundary should

- match the imposed value
- be zero
- have continuous derivatives
- be square-integrable

The correct answer is **be zero**. Literally what I just said.

Quiz 2: Question 19

Consider a spectral Bubnov-Galerkin discretisation of the weak form:

$$\left(wu_x\right)\Big|_0^1 - \int_0^1 w_x u_x dx - \int_0^1 wf\,dx = 0$$

on a domain from $x = 0$ to $x = 1$ with an essential Dirichlet condition at $x = 0$ and a Neumann condition at $x = 1$. Which of the following is not a suitable test function?

- $1 - x$
- $x - 2x^2 + x^3$
- $x + x^2$
- x
- $-x - x^2 + 2x^3$
- $x - x^2$

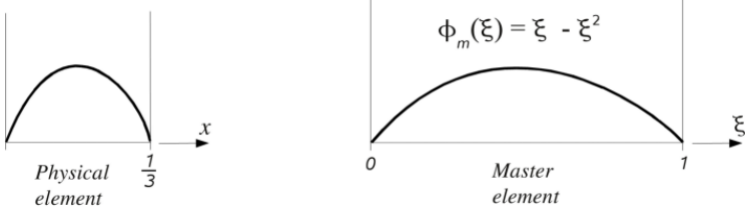
The correct answer is $1 - x$. As I said before: all weighting functions need to be zero at a Dirichlet condition. $1 - x$ is the only function of the listed functions that is not zero at $x = 0$ (where the Dirichlet condition is applied).

Quiz 2: Question 20

A coefficient of an element matrix to be computed for the physical element shown in the figure below (left) is defined by

$$K_{mm} = \int \phi_{m_x}\phi_{m_x} dx$$

where ϕ_m, defined in master element coordinates is $\phi_m = \xi - \xi^2$ (below right). The transformation between physical and master elements is given by $\xi = 3x$, so that $dx = \frac{d\xi}{3}$.



The numerical value of K_{mm} is ...

[4]There are some caveats to note here but for the present discussion, this is more or less how it works.

There are two ways to do this question. One way is to transform ϕ_m to the physical domain, i.e. we have

$$\phi_m = 3x - (3x)^2 = 3x - 9x^2$$

as $\xi = 3x$. We then have to integrate the derivative of this, $\phi_{m_x} = 3 - 18x$, squared, from 0 to $1/3$ (see the left figure):

$$K_{mm} = \int_0^{1/3} (3 - 18x)^2 \, dx = \int_0^{1/3} \left(9 - 108x + 324x^2\right) dx = \left[9x - 54x^2 + 108x^3\right]_0^{1/3} = 1$$

Alternatively, you can leave $\phi_m(\xi) = \xi - \xi^2$, from which

$$\phi_{m_x}(\xi) = \frac{d\phi_m}{d\xi}\frac{d\xi}{dx} = (1 - 2\xi) \cdot 3$$

where $d\xi/dx =$ comes from $dx = d\xi/3$. Furthermore, in the integral we need to replace dx with $d\xi/3$, meaning we get

$$\begin{aligned} K_{mm} &= \int_0^1 \phi_{m_x}(\xi)\,\phi_{m_x}(\xi)\,\frac{d\xi}{3} = \int_0^1 (3(1-2x)\cdot 3(1-2x))\,\frac{d\xi}{3} \\ &= 3\int_0^1 \left(1 - 4x + 4x^2\right) dx = 3\left[x - 2x^2 + \frac{4x^3}{3}\right]_0^1 = 1 \end{aligned}$$

Quiz 2: Question 21

Consider the weak form:

$$-w(0)\,b - \int_0^1 w_x u_x dx = \int_0^1 w f dx$$

associated with the PDE:

$$u_{xx} = f \qquad \text{on} \qquad 0 < x < 1$$

Assume only Dirichlet boundary conditions are applied as essential. If b is a specified value, and on both boundaries $u_x \neq 0$, the boundary conditions which have been applied at $x = 0$ and $x = 1$ are ... ($x = 0$) and ... ($x = 1$).

Note that the weak form ought to be

$$\begin{aligned} \left[w u_x\right]_0^1 - \int_0^1 w_x u_x dx &= \int_0^1 w f dx \\ w(1)\,u_x(1) - w(0)\,u_x(0) - \int_0^1 w_x u_x dx &= \int_0^1 w f dx \end{aligned}$$

Apparently, $w(1) = 0$, which corresponds to a Dirichlet boundary at $x = 1$. On the other hand, at $x = 0$, $w(0) \neq 0$, and $u_x(0)$ has a specified value, i.e. this is a Neumann boundary. Thus, at $x = 0$, there is a Neumann boundary; at $x = 1$, there is a Dirichlet boundary.

Quiz 2: Question 23

With the exception of rows describing boundary conditions, each row in the assembled global matrix of a finite-element discretisation corresponds to

- an equation describing the solution at a point, u_i
- an equation for a single element, e_i
- a test with one of the weighting functions, w_i
- an equation for a single solution mode amplitude, a_i

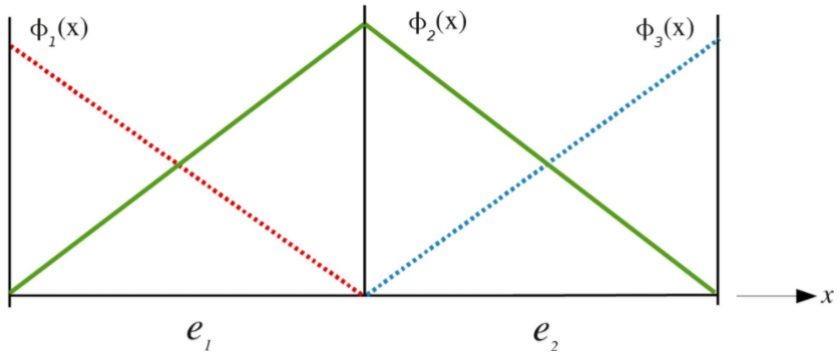
The correct answer is **a test with one of the weighting functions,** w_i. Remember what I wrote in section 5.6.4 for the spectral approach: each row in the K matrix corresponds to a test of the weak form with a single weighting function: the numerical solution u, consisting of the basis functions, is tested to see whether the weighted residuals indeed integrate to zero.

Quiz 2: Question 24

A 1D finite-element discretisation produces the element matrices

$$e_1 : \begin{bmatrix} 7 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$
$$e_2 : \begin{bmatrix} 6 & 2 \\ 9 & 5 \end{bmatrix} \begin{bmatrix} a_2 \\ a_3 \end{bmatrix}$$

for the elements shown in the figure,



where a_1, a_2, a_3 are the amplitudes of the modes Φ_1, Φ_2, Φ_3. What is the sequence of numbers that the row in the global matrix corresponding to a test with $w_2 = \Phi_2$ will have?

The global matrix will simply becomes

$$\begin{bmatrix} 7 & 3 & 0 \\ 4 & 5+6 & 2 \\ 0 & 9 & 2 \end{bmatrix} = \begin{bmatrix} 7 & 3 & 0 \\ 4 & 11 & 2 \\ 0 & 9 & 2 \end{bmatrix}$$

The row corresponding to $w_2 = \Phi_2$ is the second row, thus the correct answer is (4,11,2).

Quiz 2: Question 25

Consider the problem

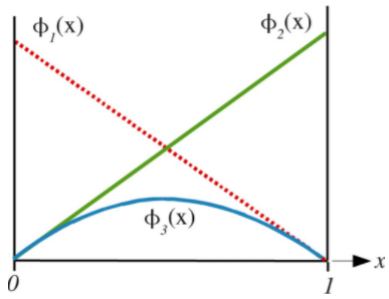
$$\frac{\partial^2 u}{\partial x^2} = x \qquad \text{on} \qquad 0 < x < 1 \qquad \text{with boundary conditions} \qquad u(0) = u(1) = 0$$

and the associated weak form $-\int_0^1 w_x u_x dx - \int_0^1 wx dx = 0$.

Using a Bubnov-Galerkin spectral method with the following basis functions:

$$\begin{aligned} \phi_1(x) &= 1-x \\ \phi_2(x) &= x \\ \phi_3(x) &= x - x^2 \end{aligned}$$

and considering Dirichlet boundary conditions as essential, the discrete solution for u is given by $u(x) = ...$



You can either do this the stupidly hard way and set up the global matrix, calculate each of the integrals, etc., or you can do it the easy way. Looking at the graph of the basis functions, and realizing that the Dirichlet boundary conditions dictate that each of the weighting functions should be zero at $x = 0$ and $x = 1$, we see that $w_1 = \phi_1$ and $w_2 = \phi_2$ are invalid weighting functions. Thus, ϕ_1 and ϕ_2 will not be used in the numerical solution of $u(x)$, since they are not zero at both boundaries. Instead, our solution will be only of the form

$$u = a_3\phi_3(x) = a_3\left(x - x^2\right)$$

With

$$\begin{aligned} u_x &= a_3(1-2x) \\ w_3 &= \phi_3 = x - x^2 \\ w_{3_x} &= 1 - 2x \end{aligned}$$

this means that we have to solve

$$\begin{aligned} -\int_0^1 w_x u_x dx - \int_0^1 wx dx &= 0 \\ \int_0^1 (1-2x)\, a_3 (1-2x)\, dx + \int_0^1 \left(x - x^2\right) x dx &= 0 \\ a_3 \int_0^1 \left(1 - 4x + 4x^2\right) dx + \int_0^1 \left(x^2 - x^3\right) dx &= 0 \\ a_3 \left[x - 2x^2 + \frac{4}{3}x^3\right]_0^1 + \left[\frac{x^3}{3} - \frac{x^4}{4}\right]_0^1 &= 0 \\ \frac{a_3}{3} + \frac{1}{12} &= 0 \\ a_3 &= -\frac{1}{4} \end{aligned}$$

So, the solution is

$$u(x) = -\frac{x - x^2}{4} = \frac{x^2}{4} - \frac{x}{4}$$

5.7.2 *Contrast with spectral approach*

Note that if you'd had used the spectral approach with the same basis functions, you'd have ended up at *exactly* the same matrix equation. However, there are several reasons not to use spectral approach:

- If the spectral approach is used, then all integrals are evaluated over the entire domain. Looking at figure 5.4, we realize that a lot of basis functions are 0 at certain parts of the domain, meaning it doesn't make sense to integrate over those parts as it'd just be a waste of computational effort.
- In a parallel computing environment[5], all functions and unknowns need to be present on all processors when using the spectral approach. In finite-element methods, this does not need to be the case, speeding up computations.
- In spectral methods, boundary conditions need to be considered for *all* functions. For finite-element methods, they only matter for the boundary elements.

Quiz 2: Question 22

An assembled global matrix from a finite-element method

- has a higher bandwidth than one from a spectral method with a similar number of unknowns
- is more symmetric than one from a spectral method with a similar number of unknowns
- is more sparse than one from a spectral method with a similar number of unknowns

The correct answer is **is more sparse than one from a spectral method with a similar number of unknowns**. In the spectral method, *all* entries of the matrix are filled; in finite element methods, this is definitely not the case.

5.8 *Convergence rates*

For problems where superconvergence does not occur, the convergence rate of spectral and finite-element discretisations is normally given by

$$|e| = \left(\int_\Omega (u - \hat{u})^2 \, d\Omega \right)^{1/2} \leq C h^p$$

where C is a constant determined by the discretisation and problem data, h is a measure of the domain (spectral) or element (FEM) size, and p depends on the basis functions chosen for the discretisation. For the piecewise-linear functions used in the previous section, normally $p = 2$. For the spectral method described before, the value of p is typically the order of the highest polynomial in the basis plus one.

It is possible to combine the spectral and finite-element methods, by using the more complicated basis functions we used in the spectral approach now on each single element of the finite-element method. Such *spectral-element methods*[6] also converge with a rate given by above equation. They can be refined by either introducing more elements (h refinement) or adding more basis functions (p refinement):

[5] Basically, you let one computer calculate one part of the problem, and a different computer another part of the problem. The result can then be combined to give the final solution.

[6] To be clear: instead of the simple linear functions we used for each of the elements, we use the same (more complicated) basis we used for the spectral approach. So you still divide the domain in elements, but on those elements, you apply more complicated basis functions. That is basically the main difference between finite element and spectral element methods (you typically speaking also use different quadrature rules (numerical schemes to compute the integrals)). This method has the benefit of still dividing the domain in elements (which was the advantage of FEM), but also still has the benefit of more complicated basis functions (leading to a more accurate solution).

- Since above equation is exponential in p, p refinement can be more effective than h refinement in spectral-element methods, especially when the solution is "smooth" (changes gradually between elements).
- When the solution is "rough" (changes rapidly between elements), h refinement tends to be more effective.

h refinement may also be less expensive for a given number of degrees of freedom, as p refinement can lead to uglier matrices that are more expensive to solve.

Quiz 2: Question 18

Finite-element methods are preferred over spectral methods
- when the weak form requires high-continuity functions
- when it is difficult to construct good global functions
- when higher-order accuracy is desired

The correct answer is **when it is difficult to construct good global functions**. Actually, spectral methods (which use high-continuity functions) would be preferable when the weak form requires high-continuity functions, since we use low-continuity functions for finite element methods. Furthermore, FEM is not necessarily more accurate than spectral methods. For your information, spectral element methods *would* be more accurate than finite element methods; after all, you still divide the domain into elements, but you use more complex basis functions, thereby increasing accuracy.

5.9 *FEM in multiple dimensions*

Pretty sure we don't have to most of the next few sections but anyway (I'll keep it short).

Classical discretisation methods such as the finite-difference method consider the boundary of the domain to be defined by a sequence of points; in FEM, this leads to elements with edges defined by straight-line segments, as shown in figure 5.8.

Figure 5.8: Approximate domain implied by straight element edges.

This can be problematic, especially in problems where smoothness matters; consider the flow around a body. Therefore, one solution to this problem is not to use straight lines as boundaries. This improves the accuracy in the solution.

Quiz 2: Question 16

Which of the following statements is not true?
- Spectral methods with highest polynomial order p may converge with order $p + 1$.
- h refinement tends to be more effective than p refinement when the solution is rough.
- h refinement is generally more computationally expensive than p refinement for a given number of degrees of freedom.
- Approximation of boundary surfaces using elements defined by straight-line segments can be a significant source of error.

The correct answer is h **refinement is generally more computationally expensive than** p **refinement**

for a given number of degrees of freedom. See above two sections. I literally wrote all these things down.

5.10 Unsteady problems

Now, something slightly important to remark is that everything you've so far was just spectral/FEM applied to the simply PDE

$$u_{xx} = f$$

which can't even be really classified as PDE. How could we for example apply these methods for

$$\begin{aligned} u_t - u_{xx} &= 0 \qquad \text{on} \quad \Omega \\ u(0,t) &= 0 \\ u(1,t) &= 0 \end{aligned}$$

you may ask? Let's find out.

5.10.1 Semi-discrete approach

In the semi-discrete approach, we do it in a rather logical way: the solution is now assumed to be of the form

$$\hat{u}(x,t) = \sum_{i=1}^{N} a_i(t)\,\phi_i(x)$$

Then, the weak form of the solution is

$$\int_\Omega w\left(u_t - u_{xx}\right) d\Omega = \int_\Omega w u_t d\Omega - \int_\Omega w u_{xx} dt = 0$$

Again, using integration by parts, this can be rewritten to

$$\int_\Omega w u_t d\Omega - w u_x\big|_0^1 + \int_\Omega w_x u_x d\Omega = 0$$

However, as we have those Dirichlet boundaries being equal to zero, all w are zero at both $x = 0$ and $x = 1$; thus, the term in the middle disappears, and we only have to deal with

$$\int_\Omega w u_t d\Omega + \int_\Omega w_x u_x d\Omega = 0$$

Now, this leads to a system of the form

$$M_{ij}\frac{d\mathbf{a}_j}{dt} + K_{ij}\mathbf{a}_j = \mathbf{0}$$

where the stiffness matrix K_{ij} has elements similar to as how they were derived previously, and the mass matrix M_{ij} is defined by

$$M_{ij} = \int_\Omega = \phi_i(\xi)\,\phi_j(\xi)\,d\Omega$$

Note that the derivative $\frac{da_j}{dt}$ comes from the fact that we differentiate u with respect to t for $\int_\Omega w u_t d\Omega$, and thus we get the derivative of a_j with respect to t rather than simply a_j.

Note that the resulting system is a system of ODEs which can be integrated using standard time-marching techniques, e.g.

$$a_j^{n+1} = a_j^n + (\Delta t)\,M_{ij}^{-1} K_{ij} a_j^n$$

where Δt is the time step and n indicates the n^{th} time level.

5.10.2 *Fully-discrete approach*

In the previous subsection we assumed a_i to be a function of time, but we can also assume ϕ_i to be a function of both x and t, i.e.

$$\hat{u}(x,t) = \sum_{i=1}^{N} a_i \phi_i (x,t)$$

We then get

$$\int_Q w \left(u_t - u_{xx}\right) dQ = 0$$

and integrating by parts gives

$$\int_\Omega \left(wu|_0^{\Delta t} - \int_0^{\Delta t} w_t u dt \right) dx - \int_0^{\Delta t} \left(wu_x|_0^1 - \int_\Omega w_x u_x dx \right) dt = 0$$

Again, since the Dirichlet boundary conditions at $x = 0$ and $x = 1$ are 0, we are left with

$$\int_{\Omega^{n+1}} wu dx - \int_{\Omega^n} wi dx - \int_Q w_t u dQ + \int_Q w_x u_x dQ = 0$$

If you don't understand what's happening, don't worry this is not part of the exam (also not for the next quiz).

5.11 *Further developments*

Read the latest scientific literature to be the shining star at any party who impresses all of his friends and family with his knowledge about the newest methods used in FEM.

Index