

LMECA2660 - Numerical Methods in Fluid Mechanics

ALEXANDRE ORÉKHOFF
SIMON DESMIDT

This summary may not be up-to-date, the newer version is available at this address:
<https://github.com/SimonDesmidt/Syntheses>

Academic year 2025-2026 - Q2



UCLouvain

Contents

1	Introduction - Finite differences with uniform grid	2
1.1	Classical finite differences	2
1.2	Richardson extrapolation	2
1.3	Operators	3
1.4	2D Laplacian	4
2	Space integration	5
2.1	Convection equation	5
2.2	Diffusion equation	7
2.3	One-sided finite differences	8
2.4	Convection-diffusion equation	10
2.5	Convection equation in 2D	12
2.6	Convection-diffusion equation in 2D	12
3	Numerical time-integration schemes for ODEs	13
3.1	Explicit Euler scheme	13
3.2	Implicit Euler scheme	14
3.3	Trapezoid rule	14
3.4	Explicit Runge-Kutta schemes	14
3.5	Multistep time schemes	18
4	Fourier document	20

Introduction - Finite differences with uniform grid

1.1 Classical finite differences

Let us define a function $u(\cdot)$ that depends on a variable x . Suppose that in the dimension x , we discretize the function uniformly with a step h and the values at the nodes are written u_i . Then, by a Taylor development series,

$$\begin{cases} u_{i+1} = u_i + h \left(\frac{\partial u}{\partial x} \right)_i + \frac{h^2}{2!} \left(\frac{\partial^2 u}{\partial x^2} \right)_i + \frac{h^3}{3!} \left(\frac{\partial^3 u}{\partial x^3} \right)_i + \frac{h^4}{4!} \left(\frac{\partial^4 u}{\partial x^4} \right)_i + \dots \\ u_{i-1} = u_i - h \left(\frac{\partial u}{\partial x} \right)_i + \frac{h^2}{2!} \left(\frac{\partial^2 u}{\partial x^2} \right)_i - \frac{h^3}{3!} \left(\frac{\partial^3 u}{\partial x^3} \right)_i + \frac{h^4}{4!} \left(\frac{\partial^4 u}{\partial x^4} \right)_i - \dots \end{cases} \quad (1.1)$$

This gives three possible finite-difference approximations:

$$\begin{aligned} \left(\frac{\partial u}{\partial x} \right)_i &= \frac{u_{i+1} - u_i}{h} + \mathcal{O}(h) && \text{(Forward differences)} \\ \left(\frac{\partial u}{\partial x} \right)_i &= \frac{u_i - u_{i-1}}{h} + \mathcal{O}(h) && \text{(Backward differences)} \\ \left(\frac{\partial u}{\partial x} \right)_i &= \frac{u_{i+1} - u_{i-1}}{2h} + \mathcal{O}(h^2) && \text{(Centered differences)} \end{aligned} \quad (1.2)$$

This also gives, for the second order,

$$\left(\frac{\partial^2 u}{\partial x^2} \right)_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} - \frac{h^2}{12} \left(\frac{\partial^4 u}{\partial x^4} \right)_i + \dots \quad (1.3)$$

→ Note: in general, discentered differences are only used for stability reasons.

1.2 Richardson extrapolation

Richardson extrapolation combines centered finite differences at different scales to get a better error:

$$\begin{aligned} \frac{4}{3} \left[\left(\frac{\partial u}{\partial x} \right)_i = \frac{u_{i+1} - u_{i-1}}{2h} - \frac{h^2}{6} \left(\frac{\partial^3 u}{\partial x^3} \right)_i - \frac{h^4}{120} \left(\frac{\partial^5 u}{\partial x^5} \right)_i - \dots \right] \\ - \frac{1}{3} \left[\left(\frac{\partial u}{\partial x} \right)_i = \frac{u_{i+2} - u_{i-2}}{2(2h)} - \frac{(2h)^2}{6} \left(\frac{\partial^3 u}{\partial x^3} \right)_i - \frac{(2h)^4}{120} \left(\frac{\partial^5 u}{\partial x^5} \right)_i - \dots \right] \\ \implies \left(\frac{\partial u}{\partial x} \right)_i = \frac{8(u_{i+1} - u_{i-1}) - (u_{i+2} - u_{i-2})}{12h} + \frac{h^4}{30} \left(\frac{\partial^5 u}{\partial x^5} \right)_i - \dots \end{aligned} \quad (1.4)$$

With this method, the truncation error is of order $\mathcal{O}(h^4)$. In the same way, for second order,

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_i = \frac{4}{3} \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} - \frac{1}{3} \frac{u_{i+2} - 2u_i + u_{i-2}}{(2h)^2} + \mathcal{O}(h^4) \quad (1.5)$$

1.3 Operators

Let us define the following operators:

- Forward difference: $\Delta u_i = u_{i+1} - u_i$;
- Backward difference: $\nabla u_i = u_i - u_{i-1}$;
- Centered difference: $\delta u_i = u_{i+1/2} - u_{i-1/2}$;
- Mean: $\mu u_i = \frac{1}{2}(u_{i+1/2} + u_{i-1/2})$;

→ Note: $u_{i+1/2}$ and $u_{i-1/2}$ are not computable because they are not grid values, but can be used for derivations of other formulae.

- Identity operator: $Iu_i = u_i$;
- Forward operator: $E u_i = u_{i+1}$;
- Backward operator: $E^{-1} u_i = u_{i-1}$;

→ Note: $E^{-1}E = I$.

Those operators have the following properties:

- $\mu\delta = \frac{1}{2}(E - E^{-1})$;
- $\mu^2 = I + \delta^2/4$;

The forward operator can be re-expressed using a Taylor development series:

$$\begin{aligned} E u_i &= u_{i+1} = u_i + h \frac{\partial}{\partial x} u_i + \frac{h^2}{2!} \frac{\partial^2}{\partial x^2} u_i + \frac{h^3}{3!} \frac{\partial^3}{\partial x^3} u_i + \dots \\ &= \left(I + hD + \frac{(hD)^2}{2!} + \frac{(hD)^3}{3!} + \dots \right) u_i = \exp(hD) u_i \end{aligned} \quad (1.6)$$

From this, using a second Taylor development series,

$$hD = \log(I + \Delta) = \Delta - \frac{\Delta^2}{2} + \frac{\Delta^3}{3} - \frac{\Delta^4}{4} + \dots \quad (1.7)$$

And, in the same way,

$$hD = -\log(I - \nabla) = \nabla + \frac{\nabla^2}{2} + \frac{\nabla^3}{3} + \frac{\nabla^4}{4} + \dots \quad (1.8)$$

We can do the same for another operator:

$$\mu\delta = \frac{1}{2}(E - E^{-1}) = \frac{1}{2}(\exp(hD) - \exp(-hD)) = \sinh(hD) \quad (1.9)$$

and we can use the Taylor series for $\text{arc sinh}(x)$ but it is not very useful. By the property that $\mu^2 = I + \delta^2/4$, we get another form:

$$hD = \mu\delta \left(I - \frac{1}{6}\delta^2 + \frac{1}{30}\delta^4 - \frac{140}{\delta^6} + \dots \right) \quad (1.10)$$

If we keep only the first order term, we find the centered-difference scheme, and the terms up to second order give the Richardson extrapolation.

→ Note: in any scheme, using more information (more values, e.g. u_{i+2}, u_{i+3}, \dots) gives a more accurate solution and the order of the truncation error increases (e.g. to $\mathcal{O}(h^3)$).

1.4 2D Laplacian

For finite differences in 2D, we can define several types of stencils. For a second-order error, there is the cross operator, which is simply the sum of classical centered finite differences on both axes, and the box operator. This operator is a linear combination of the cross operator using the medians of the square, and the one that uses the diagonals of the square (see 1.1).

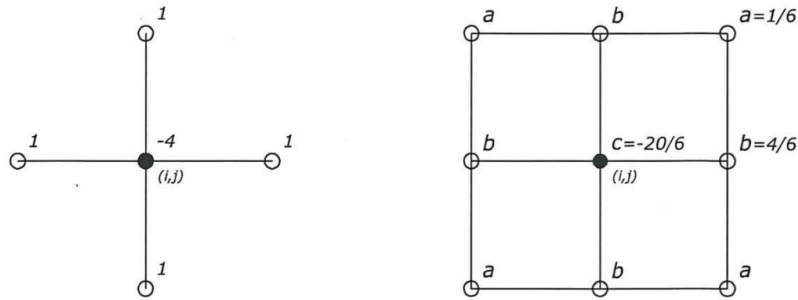


Figure 1.1: Cross operator and box operator.

The box operator expresses the following quantity:

$$h^2 \nabla^2 \left(u + \frac{h^2}{12} \nabla^2 u + \dots \right) = h^2 \left(\nabla^2 u + \frac{h^2}{12} \nabla^2 (\nabla^2 u) \right) \quad (1.11)$$

Those stencils can be generalized to higher orders using more points (bigger cross and bigger square).

→ Note: the coefficients are found using the constraint that the truncation error is independent of orientation of the stencil.

Space integration

2.1 Convection equation

The convection equation is

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 \quad (2.1)$$

The analytic solution of this equation on an infinite domain, for a speed c constant, is

$$u(x, t) = A(t)e^{ikx} = A(0)e^{ik(x-ct)} \quad (2.2)$$

On a periodic domain of length L , the solution is

$$\begin{aligned} u(x, t) &= \sum_{k=-\infty}^{\infty} A_k(t)e^{ikx} \quad k = \frac{2\pi}{L}p \quad p \in \mathbb{Z} \\ \implies u(x, t) &= \sum_{p=-\infty}^{\infty} A_p(t)e^{i\frac{2\pi x}{L}p} \end{aligned} \quad (2.3)$$

where the coefficients verify the condition $A_k = A_{-k}^*$ for all k , since the solution must be real. In the exact solution, all the modes have the same speed c .

2.1.1 Explicit schemes

However, all modes do not have the same velocity when we use explicit finite differences. Let us show it in the case of an infinite domain (same thing happens for a periodic domain, adding the sum on p):

Let $u_i(t) = A(t)e^{jkx_i}$. Then,

$$\begin{aligned} \left. \frac{\partial u}{\partial x} \right|_i &= A j k^* \exp(j k x_i) \implies \frac{dA}{dt} + j k^* c A = 0 \\ \implies u_i(t) &= A(0)e^{j(kx_i - k^* ct)} = A(0)e^{ik\left(x_i - \frac{k^* h}{kh} ct\right)} \end{aligned} \quad (2.4)$$

we call k^* the modified wave number. It is different from k because all the modes do not move at the same speed. For example, for the E2 stencil, its expression is derived in the following way:

$$\begin{aligned} \left. \frac{\partial u}{\partial x} \right|_i &= \frac{u_{i+1} - u_i}{2h} = \frac{A}{h} \frac{1}{2} (e^{jkh} - e^{-jkh}) e^{jkx_i} = \frac{A}{h} j \sin(kh) e^{jkx_i} = A j k^* e^{jkx_i} \\ \implies k^* h &= \sin(kh) \end{aligned} \quad (2.5)$$

This is a **phase** error, as the modes do not **move** with the right **velocity**.
By a Taylor development,

$$\frac{k^*h}{kh} = 1 - \frac{(kh)^2}{6} + \mathcal{O}((kh)^4) \quad (2.6)$$

and although k is not constant for all modes, the error of the stencil is still of the same order: $\mathcal{O}((kh)^2)$. Moreover, the speed of the modes is $c^* = \frac{k^*h}{kh}c$.

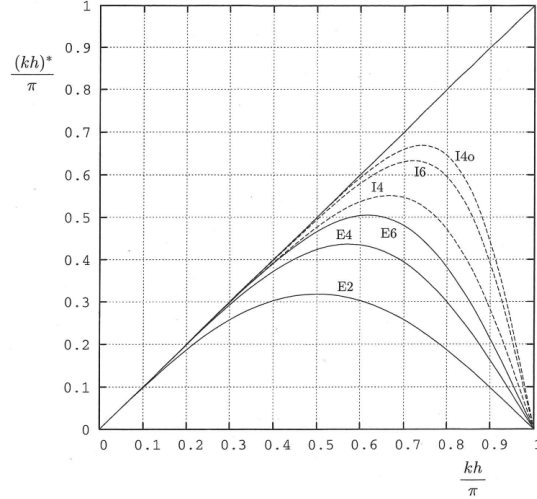


Figure 2.1: Evolution of the exact wave number and the modified wave number.

As the error increases when k increases, it is important to use a very refined grid so that all points whose amplitude is non negligible have $k^*h \approx kh$.

2.1.2 Implicit schemes

The general scheme of implicit finite differences is the following:

$$\begin{aligned} \frac{\partial u}{\partial x}_i + \alpha \frac{1}{2} \left(\frac{\partial u}{\partial x} \Big|_{i+1} + \frac{\partial u}{\partial x} \Big|_{i-1} \right) + \beta \frac{1}{2} \left(\frac{\partial u}{\partial x} \Big|_{i+2} + \frac{\partial u}{\partial x} \Big|_{i-2} \right) \\ = a \frac{u_{i+1} - u_{i-1}}{2h} + b \frac{u_{i+2} - u_{i-2}}{4h} + c \frac{u_{i+3} - u_{i-3}}{6h} \end{aligned} \quad (2.7)$$

Usually, we use $\beta = c = 0$ to keep only the nearest grid points to i . Those schemes are called compact.

We can use $\frac{\partial u}{\partial x} \Big|_i = A j k^* e^{j k x_i}$ and $\frac{u_{i+1} - u_{i-1}}{2h} = \sin(kh)$ to show that

$$k^*h = \frac{a \sin(kh) + \frac{b}{2} \sin(2kh) + \frac{c}{3} \sin(3kh)}{1 + \alpha \cos(kh) + \beta \cos(2kh)} \quad (2.8)$$

We can do a Taylor development of this quantity to get different schemes:

$$\begin{aligned} \text{Taylor of order 1:} \quad & 1 + \alpha + \beta = a + b + c \implies \text{Error of order 2} \\ \text{Taylor of order 2:} \quad & 3(\alpha + 2^2\beta) = a + 2^2b + 3^2c \implies \text{Error of order 4} \\ \text{Taylor of order 3:} \quad & 5(\alpha + 2^4\beta) = a + 2^4b + 3^4c \implies \text{Error of order 6} \\ \text{Taylor of order 4:} \quad & 7(\alpha + 2^6\beta) = a + 2^6b + 3^6c \implies \text{Error of order 8} \\ \text{Taylor of order 5:} \quad & 9(\alpha + 2^8\beta) = a + 2^8b + 3^8c \implies \text{Error of order 10} \end{aligned} \quad (2.9)$$

For example, in the case where $\beta = 0$ and $\alpha \neq 0$, the system is tridiagonal and the solver has a time complexity of $\mathcal{O}(N)$. For $\beta = 0$ and $\alpha = 0$, we find the explicit scheme.

→ Note: The I4o scheme is the scheme with the smallest constant before the $(kh)^4$ using all parameters. For this one, the sign of that constant is positive, meaning that it goes above the correct velocity. This behaviour is not present on the other scheme.

2.2 Diffusion equation

The diffusion equation is

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} \quad (2.10)$$

This equation is not reversible because it represents a loss of information: the high-wave-number terms decay fast and have no impact after a small time.

The exact solution is $u(x, t) = A_k(t)e^{jkx}$ with $A_k(t) = A_k(0)e^{\alpha k^2 t}$.

2.2.1 Explicit schemes

The explicit finite differences scheme is

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_i = a \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + b \frac{u_{i+2} - 2u_i + u_{i-2}}{4h^2} + c \frac{u_{i+3} - 2u_i + u_{i-3}}{9h^2} \quad (2.11)$$

Let $u_i(t) = A(t)e^{jkx_i}$. Then,

$$\frac{dA}{dt} = -\alpha(k^2)^* A \implies A(t) = A(0)e^{-\alpha(k^2)^* t} e^{jkx_i} \implies u_i(t) = A(0)e^{-\alpha k^2 \left(\frac{(k^2)^* h^2}{k^2 h^2} \right)} e^{jkx_i} \quad (2.12)$$

This is a **amplitude** error, as the mode do not decay with the proper **rate**.

For example, for the explicit scheme of order 2 ($a = 1$, $b = c = 0$), the modified k^2 is given by

$$(k^1)^* h^2 = 4 \sin^2 \left(\frac{kh}{2} \right) \iff \frac{(k^2)^* h^2}{k^2 h^2} = \frac{\sin^2 \left(\frac{kh}{2} \right)}{\left(\frac{kh}{2} \right)^2} \quad (2.13)$$

2.2.2 Implicit schemes

The general form of the implicit scheme follows the same reasoning as for (2.7).

$$\begin{aligned} & \left. \frac{\partial^2 u}{\partial x^2} \right|_i + \alpha \frac{1}{2} \left(\left. \frac{\partial^2 u}{\partial x^2} \right|_{i+1} + \left. \frac{\partial^2 u}{\partial x^2} \right|_{i-1} \right) + \beta \frac{1}{2} \left(\left. \frac{\partial^2 u}{\partial x^2} \right|_{i+2} + \left. \frac{\partial^2 u}{\partial x^2} \right|_{i-2} \right) \\ &= a \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + b \frac{u_{i+2} - 2u_i + u_{i-2}}{4h^2} + c \frac{u_{i+3} - 2u_i + u_{i-3}}{9h^2} \end{aligned} \quad (2.14)$$

By using $\left. \frac{\partial^2 u}{\partial x^2} \right|_i = -A(t)(k^2)^* e^{jkx_i}$, we can show that

$$(k^2)^* h^2 = \frac{4 \left[a \sin^2 \left(\frac{kh}{2} \right) + \frac{b}{4} \sin^2 \left(\frac{2kh}{2} \right) + \frac{c}{9} \sin^2 \left(\frac{3kh}{2} \right) \right]}{1 + \alpha \cos(kh) + \beta \cos(2kh)} \quad (2.15)$$

As previously, we can do the Taylor development series of this expression to get different schemes:

$$\begin{aligned} \text{Taylor of order 1:} \quad & 1 + \alpha + \beta = a + b + c \implies \text{Error of order 2} \\ \text{Taylor of order 2:} \quad & 2 \cdot 3(\alpha + 2^2 \beta) = a + 2^2 b + 2^2 c \implies \text{Error of order 4} \\ \text{Taylor of order 3:} \quad & 3 \cdot 5(\alpha + 2^4 \beta) = a + 2^4 b + 2^4 c \implies \text{Error of order 6} \\ \text{Taylor of order 4:} \quad & 4 \cdot 7(\alpha + 2^6 \beta) = a + 2^6 b + 2^6 c \implies \text{Error of order 8} \\ \text{Taylor of order 5:} \quad & 5 \cdot 9(\alpha + 2^8 \beta) = a + 2^8 b + 2^8 c \implies \text{Error of order 10} \end{aligned} \quad (2.16)$$

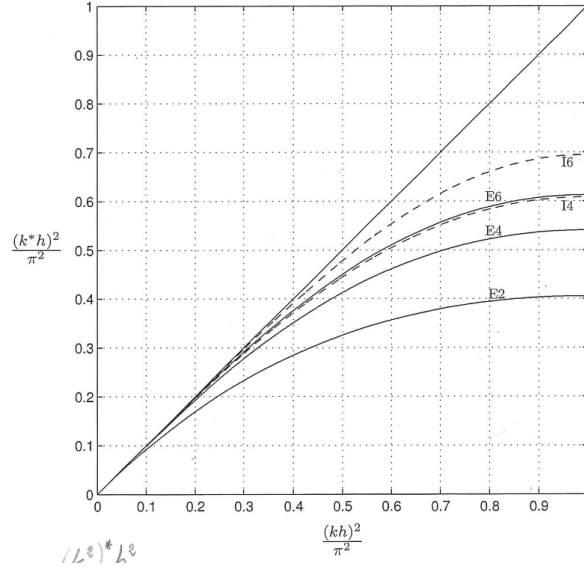


Figure 2.2: Evolution of the modified k^2 with k^2 .

2.3 One-sided finite differences

Let us consider the backward differences for the convection equation:

$$\frac{d}{dt} u_i = -c \frac{u_i - u_{i-1}}{h} \implies \frac{d}{dt} A_k = -\frac{c}{h} (1 - e^{jkh}) \quad (2.17)$$

Then, $\lambda_k = -\frac{c}{h} (1 - e^{-jkh})$ is no longer purely imaginary. Its real part gives an amplitude error (numerical diffusion) and its imaginary part gives a (numerical) dispersion error.

→ Note: for $c < 0$, we need to use forward differences instead because otherwise the real part of λ_k becomes positive and thus unstable.

Let us compute the term of numerical diffusion:

$$\frac{d}{dt}u_i = -c \frac{u_i - u_{i-1}}{h} = -c \frac{u_{i+1} - u_{i-1}}{2h} + c \frac{u_{i+1} - 2u_i + u_{i-1}}{2h} = -c \frac{u_{i+1} - u_{i-1}}{2h} + \frac{ch}{2} \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \quad (2.18)$$

The modified equation becomes

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = \frac{ch}{2} \frac{\partial^2 u}{\partial x^2} + \dots \quad (2.19)$$

and so the diffusivity parameter is thus $\epsilon = \frac{ch}{2}$.

For this scheme, from (2.17), $\lambda_k \Delta t = -\frac{c\Delta t}{h} ((1 - \cos(kh)) + i \sin(kh))$ and the region of stability of the method is a circle centered in $(-1, 0)$ in the complex plane of $\lambda_k \Delta t$.

2.3.1 One-sided finite differences of higher order

This type of scheme of higher order would give for example

$$\begin{aligned} \frac{d}{dt}u_i &= \frac{3u_i - 4u_{i-1} + u_{i-2}}{2h} \\ &= \frac{11u_i - 18u_{i-1} + 9u_{i-2} - 2u_{i-3}}{6h} \end{aligned} \quad (2.20)$$

and all those expressions give different stability regions. The general form for the spatial derivative is

$$\left. \frac{\partial u}{\partial x} \right|_i = a_1 \frac{u_i - u_{i-1}}{h} + a_2 \frac{u_i - u_{i-2}}{2h} + a_3 \frac{u_i - u_{i-3}}{3h} \quad (2.21)$$

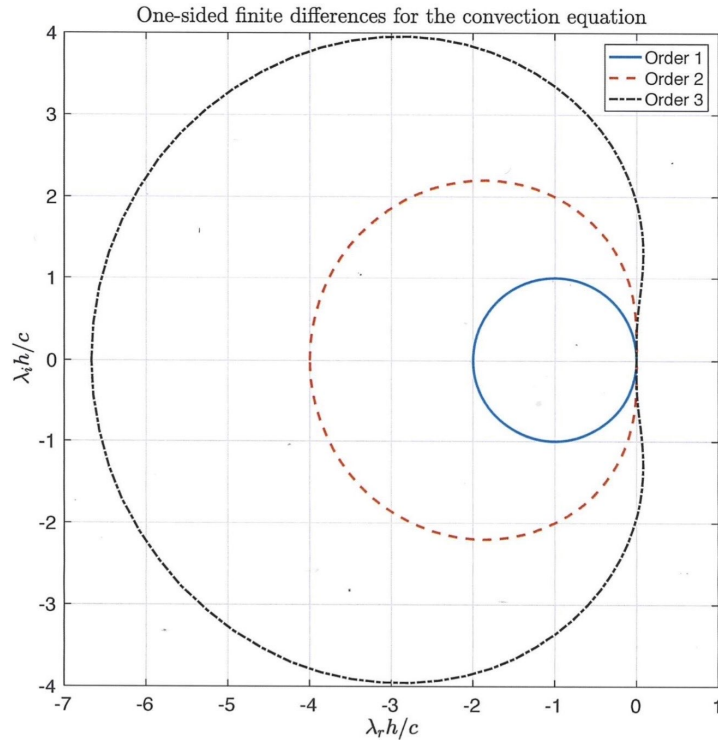


Figure 2.3: Stability regions for one-sided schemes.

This means that if our system has a high eigenvalue λ , we need to decrease the time step to have convergence properties. This means that those one-sided schemes are not much helpful, because they are very costly. Moreover, at order 3, we can see low wave numbers with positive real part, meaning that they blow up (positive exponent in the exponential).

2.4 Convection-diffusion equation

The convection-diffusion equation is

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = \epsilon \frac{\partial^2 u}{\partial x^2} \quad (2.22)$$

→ Note: numerical diffusion can appear and pollute the physical diffusion. This is not a problem if we have $\epsilon_{num} \ll \epsilon_{phys}$.

2.4.1 Centered finite differences

The scheme here is

$$\frac{d}{dt}u_i = -c \frac{u_{i+1} - u_{i-1}}{2h} + \epsilon \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \quad (2.23)$$

This gives

$$\lambda_k = -\frac{2\epsilon}{h^2}(1 - \cos(kh)) - j \frac{c}{h} \sin(kh) \quad (2.24)$$

This expression in the complex plane of $\lambda_k \Delta t$ is an ellipse with vertical axis $b = \beta$ and horizontal axis $a = 2r$, $\beta = c\Delta t/h$ being the CFL number, and $r = \epsilon\Delta t/h^2$ being the Fourier number. We also define the mesh Peclet number $R = |c|h/\epsilon$, giving the identity $\beta = rR$.

This scheme, using a Euler scheme for integration, is stable if the ellipse is fully contained in the circle centered in $(-1, 0)$.

- If $b \leq a$, the scheme is stable if $a \leq 1$, or equivalently, if $R \leq 2$, the scheme is stable if $r \leq 1/2$;
- If $b > a$, the scheme is stable if the largest radius of curvature of the ellipse is $\leq 1^1$, i.e. $b^2/a \leq 1 \iff c^2\Delta t/\epsilon \leq 2$, or equivalently, if $R > 2$, this scheme is stable for $r \leq 2/R^2$.

¹This value is the radius of curvature of the circle.

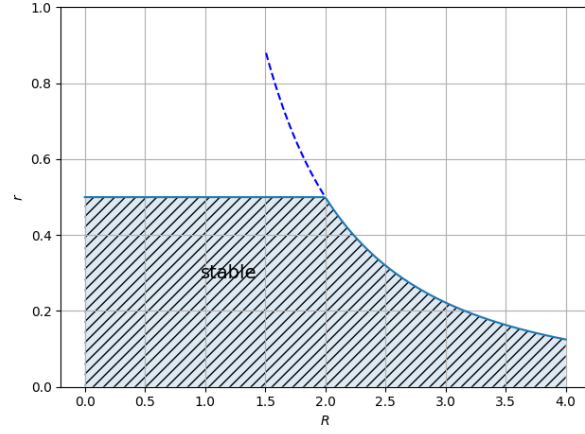


Figure 2.4: Stability region for the parameters of the convection-diffusion equation.

If using backward finite differences, i.e. $c > 0$, the scheme is

$$\frac{d}{dt}u_i = -c \frac{u_i - u_{i-1}}{h} + \epsilon \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = -c \frac{u_{i+1} - u_{i-1}}{2h} + \left(\epsilon + \frac{ch}{2} \right) \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \quad (2.25)$$

Then,

$$\lambda_k = - \left(\frac{c}{h} + \frac{2\epsilon}{h^2} \right) (1 - \cos(kh)) - j \frac{c}{h} \sin(kh) \quad (2.26)$$

This is also an ellipse, with axes $a = \beta + 2r$ and $b = \beta$. Here, $b \leq a$ because $r > 0$ always, and thus

$$r \leq \frac{1}{1 + R} \quad (2.27)$$

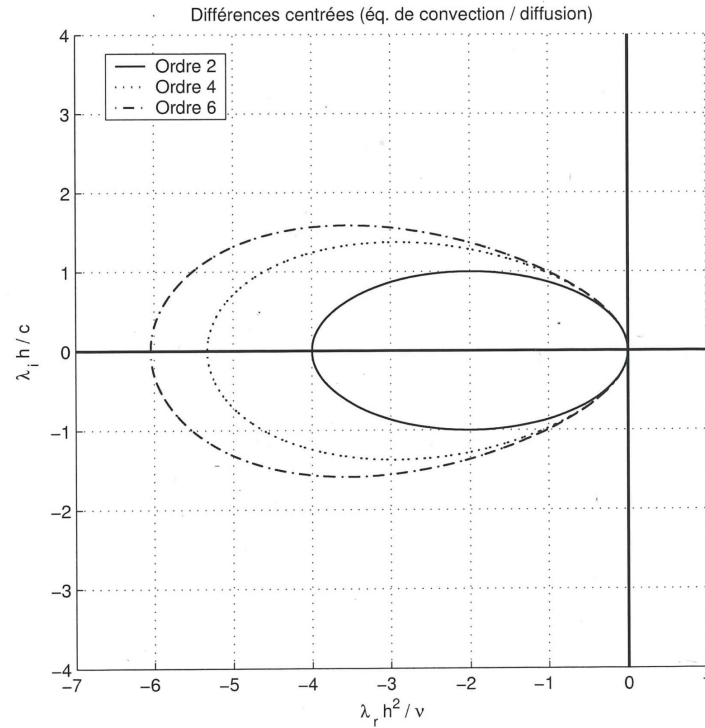


Figure 2.5: Stability region of the convection-diffusion equation.

2.5 Convection equation in 2D

The convection equation in 2D has two different velocity values:

$$\frac{\partial u}{\partial t} + u_0 \frac{\partial u}{\partial x} + v_0 \frac{\partial u}{\partial y} = 0 \quad (2.28)$$

We consider here a periodic problem to be able to do a modal analysis. The E2 scheme gives

$$\frac{du_{ij}(t)}{dt} = - \left(u_0 \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + v_0 \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} \right) \quad (2.29)$$

Which gives a system of ODEs for the $u_{i,j}(t)$.

In the continuous problem, the solution is

$$u(x, y, t) = \sum_{k,\ell} A_{k\ell}(t) e^{j(kx + \ell y)} \quad (2.30)$$

Our ODE gives

$$A_{k\ell}(t) = A_{k\ell}(0) e^{-j(u_0 k + v_0 \ell)t} \quad (2.31)$$

For the discrete problem (in space but continuous time), the ODE for $A_{k\ell}$ is

$$\frac{dA_{k\ell}}{dt} = -j \left(\frac{u_0}{\Delta x} \sin(k\Delta x) + \frac{v_0}{\Delta y} \sin(\ell\Delta y) \right) A_{k\ell} \quad (2.32)$$

and so the eigenvalues $\lambda_{k\ell}$ are purely imaginary, with $\lambda_{k\ell} = \lambda_k + \lambda_\ell$. We can also define the global CFL number

$$\beta = \beta_x + \beta_y = \frac{|u_0|\Delta t}{\Delta x} + \frac{|v_0|\Delta t}{\Delta y} \quad (2.33)$$

and

$$|\lambda_{k\ell}|_{\max} \Delta t = \beta \quad (2.34)$$

2.6 Convection-diffusion equation in 2D

$$\frac{\partial u}{\partial t} + u_0 \frac{\partial u}{\partial x} + v_0 \frac{\partial u}{\partial y} = \epsilon \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (2.35)$$

We can compute the eigenvalues by using a scheme, e.g. E2, to get

$$\frac{dA_{k\ell}}{dt} = \lambda_{k\ell} A_{k\ell} \quad (2.36)$$

The parameters of the scheme are

$$\begin{cases} r_x = \frac{\epsilon \Delta t}{(\Delta x)^2} & R_x = \frac{|u_0| \Delta x}{\epsilon} \implies \beta_x = r_x R_x \\ r_y = \frac{\epsilon \Delta t}{(\Delta y)^2} & R_y = \frac{|v_0| \Delta y}{\epsilon} \implies \beta_y = r_y R_y \end{cases} \quad (2.37)$$

In the case $\Delta x = \Delta y = h$, $r_x = r_y = r = \epsilon \Delta t / h^2$ and

$$\begin{cases} \beta_x = \frac{|u_0| \Delta t}{h} \\ \beta_y = \frac{|v_0| \Delta t}{h} \end{cases} \implies \begin{cases} \beta = \beta_x + \beta_y = \frac{(|u_0| + |v_0|) \Delta t}{h} \\ R = R_x + R_y = \frac{(|u_0| + |v_0|) h}{\epsilon} \end{cases} \quad (2.38)$$

Numerical time-integration schemes for ODEs

The general formulation of first order ODEs is

$$\frac{du}{dt} = f(u, t) \quad (3.1)$$

Until now, we have only had a simple system with $f(u, t) = \lambda u$.

We can do a Taylor development of $f(\cdot)$ to analyze the local behaviour of the system:

$$\frac{du}{dt} = f(u, t) = \underbrace{f(u_0, t_0)}_{:=\beta} + (u - u_0) \underbrace{\frac{\partial f}{\partial u}}_{=:\lambda} + (t - t_0) \underbrace{\frac{\partial f}{\partial t}(u_0, t_0)}_{=:\gamma} + \dots \quad (3.2)$$

and defining $u' = u - u_0$ and $t' = t - t_0$. The solution of this is

$$u'(t) = u'_p + u'_g = -\frac{1}{\lambda} \left(\gamma t' + \left(\frac{\gamma}{\lambda} + \beta \right) \right) + A e^{\lambda t'} = \frac{1}{\lambda} \left[\left(\frac{\gamma}{\lambda} + \beta \right) (e^{\lambda t'} - 1) - \gamma t' \right] \quad (3.3)$$

The stability of the scheme used to integrate that ODE is governed by the homogeneous part of the equation, as it is the only one that can blow up (exponential term).

For our special case $\frac{du}{dt} = \lambda u$, the solution is $u(t) = u(0)e^{\lambda t}$. As we use a discretized space, let $t = t^n = n\Delta t$. Then, $u^n = u(t^n) = (e^{\lambda \Delta t})^n u^0 = (\rho_e)^n u^0$, where we call ρ_e the amplification factor of the exact solution. The ODE is (physically) stable if $|\rho_e| \leq 1$. In the $\lambda \Delta t$ complex plane, it represents the left half-plane.

3.1 Explicit Euler scheme

$$\frac{u^{n+1} - u^n}{\Delta t} = \lambda u^n \quad (3.4)$$

Then,

$$u^{n+1} = (1 + \lambda \Delta t) u^n = \rho u^n \implies u^n = (\rho)^n u^0 \quad (3.5)$$

The scheme is stable if

$$|\rho|^2 = (1 + \lambda_r \Delta t)^2 + (\lambda_i \Delta t)^2 \leq 1 \quad (3.6)$$

This is a circle centered in $(-1, 0)$.

At each time step, the Euler scheme produces an $\mathcal{O}((\Delta t)^2)$ error. To arrive at $t = T$, it takes $N = T/\Delta t$ time steps. The error at $t = T$ is thus $\mathcal{O}((\Delta t)^2/\Delta t) = \mathcal{O}(\Delta t)$. The Euler scheme is a first-order scheme.

→ Note: always check the order of a scheme using a fixed time interval, and not a single time step, as we want a global estimation.

3.2 Implicit Euler scheme

$$\frac{u^{n+1} - u^n}{\Delta t} = \lambda u^{n+1} \quad (3.7)$$

Then,

$$u^{n+1} = \frac{1}{1 - \lambda \Delta t} u^n = \rho u^n \quad (3.8)$$

This scheme is unconditionally stable, as it is stable for all problems with $\lambda_r \leq 0$ and we do not consider the problems that are not physically stable.

3.3 Trapezoid rule

$$\frac{u^{n+1} - u^n}{\Delta t} = \lambda \frac{u^{n+1} + u^n}{2} \quad (3.9)$$

Then,

$$u^{n+1} = \frac{1 + \frac{\lambda \Delta t}{2}}{1 - \frac{\lambda \Delta t}{2}} u^n = \rho u^n \quad (3.10)$$

This scheme is also unconditionally stable for physically stable problems. Moreover, it is of second order, because it is centered around $t^{n+1/2}$.

3.4 Explicit Runge-Kutta schemes

We will work here on the integration of ODEs of the form

$$\frac{du}{dt} = f(u, t) \quad (3.11)$$

The notation is $f^n = f(u^n, t^n)$ for the value of the function at the n -th time step. The general form of a Runge-Kutta scheme of order m is the following:

$$u^{n+1} = u^n + \sum_{i=1}^m \gamma_i k_i \quad \text{s.t.} \quad \begin{cases} k_1 = \Delta t f(u^n, t^n) \\ k_i = \Delta t f(u^n + \sum_{j=1}^{i-1} \beta_{ji} k_j, t^n + \sum_{j=1}^{i-1} \alpha_j \Delta t) \end{cases} \quad \forall i \geq 2 \quad (3.12)$$

3.4.1 RK2

The formulation of order two is

$$u^{n+1} = u^n + \gamma_1 k_1 + \gamma_2 k_2 \quad \text{s.t.} \quad \begin{cases} k_1 = \Delta t f(u^n, t^n) \\ k_2 = \Delta t f(u^n + \beta k_1, t^n + \alpha \Delta t) \end{cases} \quad (3.13)$$

The explicit Runge-Kutta schemes work with Taylor series:

$$u^{n+1} = u^n + \Delta t \left(\frac{du}{dt} \right)^n + \frac{(\Delta t)^2}{2} \left(\frac{d^2 u}{dt^2} \right)^n + \mathcal{O}((\Delta t)^3) \quad (3.14)$$

and

$$\begin{aligned} k_2 &= \Delta t \left[f^n + \beta k_1 \left(\frac{\partial f}{\partial u} \right)^n + \alpha \Delta t \left(\frac{\partial f}{\partial t} \right)^n + \dots \right] \\ \implies u^{n+1} &= u^n + \gamma_1 \Delta t f^n + \gamma_2 \Delta t \left[f^n + \beta \Delta t f^n \left(\frac{\partial f}{\partial u} \right)^n + \alpha \Delta t \left(\frac{\partial f}{\partial t} \right)^n + \dots \right] \end{aligned} \quad (3.15)$$

By comparing those developments, we find a system of 3 equations:

$$\begin{aligned} \gamma_1 + \gamma_2 &= 1 \\ \gamma_2 \beta &= 1/2 \\ \gamma_2 \alpha &= 1/2 \end{aligned} \quad (3.16)$$

There remains one degree of freedom, meaning we get a family of schemes. The usual methods are

- RK2 classic: Heun's method

$$\begin{aligned} \gamma_1 &= \gamma_2 = 1/2 \\ \alpha &= \beta = 1 \end{aligned} \quad (3.17)$$

$$\begin{cases} k_1 = \Delta t f(u^n, t^n) \\ k_2 = \Delta t f(u^n + k_1, t^n + \Delta t) \end{cases} \implies u^{n+1} = u^n + \frac{1}{2} k_1 + \frac{1}{2} k_2 \quad (3.18)$$

- RK2 classic: midpoint method

$$\begin{aligned} \gamma_1 &= 0 \quad \gamma_2 = 1 \\ \alpha &= \beta = 1/2 \end{aligned} \quad (3.19)$$

$$\begin{cases} k_1 = \Delta t f(u^n, t^n) \\ k_2 = \Delta t f(u^n + \frac{1}{2} k_1, t^n + \frac{1}{2} \Delta t) \end{cases} \implies u^{n+1} = u^n + k_2 \quad (3.20)$$

3.4.2 RK3-4

Runge-Kutta 3 and 4 are to be derived in the same way. They both have two degrees of freedom and thus it is possible to have many different approaches to get the coefficients. The most common way is to do a diagonal scheme, i.e. k_j only depends on the

value of k_{j-1} . This gives the usual RK4 method

$$\begin{cases} k_1 = \Delta t f(u^n, t^n) \\ k_2 = \Delta t f(u^n + \frac{1}{2}k_1, t^n + \frac{1}{2}\Delta t) \\ k_3 = \Delta t f(u^n + \frac{1}{2}k_2, t^n + \frac{1}{2}\Delta t) \\ k_4 = \Delta t f(u^n + k_3, t^n + \Delta t) \end{cases} \quad (3.21)$$

$$u^{n+1} = u^n + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4)$$

3.4.3 Runge Kutta schemes in stability analysis

The analysis of stability is for the equation

$$\frac{du}{dt} = \lambda u \quad (3.22)$$

Stability only considers the linearized equation, because it analyses the local behaviour. In the case of this equation, we can express u^{n+1} as a function of u^n :

$$u^{n+1} = \rho u^n \quad (3.23)$$

where $\rho = \sum_{i=1}^m \frac{(\lambda \Delta t)^i}{i!}$ for a Runge-Kutta scheme of order m . The method is stable if $|\rho| \leq 1$.

→ Note: the intersection of the marginal stability curve ($|\rho| = 1$) with the imaginary axis is particularly relevant, as it is necessary to determine the maximum value of the CFL for simple spatial-integration schemes such as Euler explicit, because the λ values of those schemes are purely imaginary. In short, the stability region of the spatial scheme must be inside the stability region of the stability region of the time scheme to be fully stable.

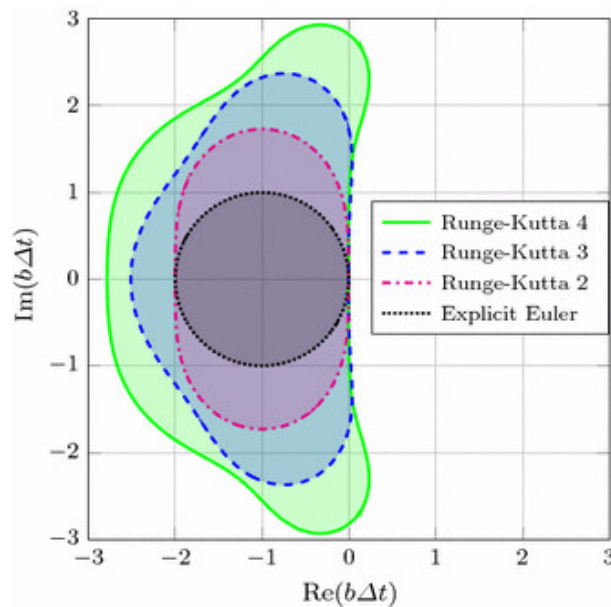


Figure 3.1: Stability regions of Runge-Kutta schemes.

3.4.4 General Explicit Runge-Kutta schemes

A Butcher tableau is a representation of the RK coefficients meant for them to be easy to read.

				γ_1
α_2	β_{21}			γ_2
α_3	β_{31}	β_{32}		γ_3
α_4	β_{41}	β_{42}	β_{43}	γ_4

The first column contains the time weights, the β are the space weights, and the γ are the weights in the linear combination of k_i .

We can check that we always have $\alpha_i = \sum_j \beta_{ji}$ and that $\sum_i \gamma_i = 1$.

→ Note: the diagonal schemes mentioned earlier give a diagonal matrix for the parameters β .

In an Explicit Runge-Kutta scheme (ERK), the number of substeps (number of coefficients k) is not necessarily the order of the scheme: it is possible to do ERK54, which has 5 substeps, but only an order 4. This is because the expression of the amplification parameter ρ changes, and does not correspond exactly to the Taylor development terms.

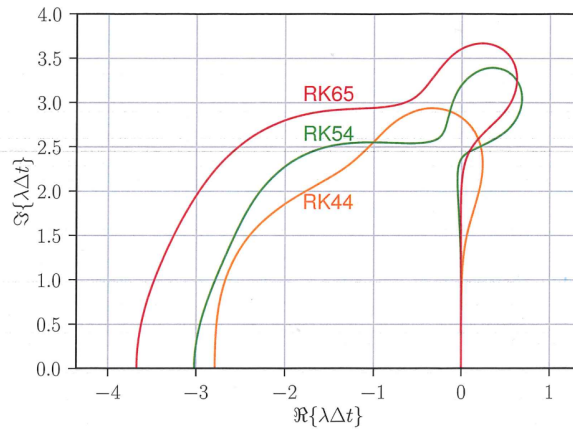


Figure 3.2: Stability regions for Explicit Runge-Kutta schemes.

You can see on 3.2 that the scheme RK54 goes to the right on the imaginary axis. This is a bad behaviour for stability, as there is a small physically stable zone that is not numerically stable.

3.4.5 RK4 pseudocode

Algorithm 1 RK4 pseudocode

```

1:  $(t, u) = \text{time } t^n \text{ and solution } u(t^n) = u^n;$ 
2: Save to avoid overwriting:  $u_{safe} = u, t_{safe} = t;$ 
3: for  $k = 1, 2, 3, 4$  do
4:    $u_\ell = u_{safe} + \alpha_k \Delta u$ 
5:    $t_\ell = t_{safe} + \alpha_k \Delta t$ 
6:   Evaluate:  $\Delta u = \Delta t f(u_\ell, t_\ell)$ 
7:   And update:  $\begin{cases} u = u + \gamma_k \Delta u \\ t = t + \gamma_k \Delta t \end{cases}$ 
8: end for
9: return  $(t, u)$ 

```

At the end of the loop, (t, u) are the time $t^{n+1} = t^n + \Delta t$ and the solution $u(t^{n+1}) = u^{n+1}$. This algorithm needs 4 arrays to be stored. For big simulations, this is a lot of data, and this is why there exists some low-storage RK methods that only require 3 arrays. They use different parameter values, e.g. for RK3:

			5/30
1/3	1/3		9/30
3/4	-3/16	15/16	16/30

3.5 Multistep time schemes

Multisteps time schemes use more than one previous step to compute the value t^n . They are generally cheap, but not self starting: t^1 cannot be computed through it, as it would need to know t^{-1} . Moreover, they produce multiple solutions even though only one is physically correct.

3.5.1 Leap-frog

The scheme is

$$\frac{u^{n+1} - u^{n-1}}{2\Delta t} = f(u^n, t^n) \iff u^{n+1} = u^{n-1} + 2\Delta t f(u^n, t^n) \quad (3.24)$$

The scheme is centered around t^n , meaning it is of order 2.
To analyse its stability region, we have

$$\frac{du}{dt} = \lambda u \implies \begin{cases} u^n = \rho u^{n-1} \\ u^{n+1} = \rho^2 u^{n-1} \end{cases} \quad (3.25)$$

Then,

$$\rho^2 = 1 + 2\lambda\Delta t\rho \iff \begin{cases} \rho_1 + \rho_2 = 2\lambda\Delta t \\ \rho_1\rho_2 = -1 \end{cases} \implies \rho = \lambda\Delta t \pm \sqrt{1 + (\lambda\Delta t)^2} \quad (3.26)$$

By Taylor series,

$$\begin{cases} \rho_1 = 1 + \lambda\Delta t + \frac{(\lambda\Delta t)^2}{2} - \frac{(\lambda\Delta t)^4}{8} + \dots \approx e^{\lambda\Delta t} \\ \rho_2 = -1 + \lambda\Delta t - \frac{(\lambda\Delta t)^2}{2} + \frac{(\lambda\Delta t)^4}{8} + \dots \approx e^{-\lambda\Delta t} \end{cases} \quad (3.27)$$

ρ_1 is the physical root and ρ_2 is called the parasitic root. Because of the condition $\rho_1\rho_2 = -1$ and the stability condition $|\rho_1|, |\rho_2| \leq 1$, we need $|\rho_1| = |\rho_2| = 1$.

Then, the solution of the scheme is

$$\rho = c_1\rho_1 + c_2\rho_2 \implies u^n = (c_1\rho_1 + c_2\rho_2)u^{n-1} \quad (3.28)$$

where u^1 must be computed using another scheme with less steps, e.g. RK2.

Fourier document

TODO.