

6.4 Stability of Time-stepping Methods

Stability: Does the approximating (discretized) solution converge to the true solution?

Let us just consider the simple example of exponential decay:

$$\frac{dy}{dt} = -\lambda y, \quad \lambda > 0 \quad \Rightarrow \quad y(t) = y_0 e^{-\lambda t}$$

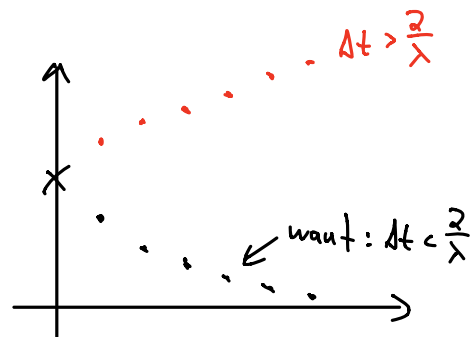
There are several ways to discretize the time derivative; we discuss the two most important ones:

• **Explicit Euler Method:** $\frac{y^{j+1} - y^j}{\Delta t} = -\lambda y^j$ (rhs evaluated at j)

$$\Rightarrow y^{j+1} = y^j - \lambda \Delta t y^j = (1 - \lambda \Delta t) y^j$$

$$\Rightarrow y^M = (1 - \lambda \Delta t)^M y_0$$

$$\left(\text{check: } \lim_{M \rightarrow \infty} \underbrace{\left(1 - \lambda \frac{t}{M}\right)}_{\Delta t}^M y_0 = e^{-\lambda t} y_0 \right)$$



In particular: Given y_0 , we want the y^j to be decreasing:

So we need that $|1 - \lambda \Delta t| < 1$ i.e.,

$$\bullet \quad 1 - \lambda \Delta t < 1 \Rightarrow \lambda \Delta t > 0 \text{ holds anyway}$$

$$\bullet \quad -(1 - \lambda \Delta t) < 1 \Rightarrow \lambda \Delta t < 2 \Rightarrow \Delta t < \frac{2}{\lambda}$$

this is a condition on stability

\Rightarrow Here, we need to choose Δt small enough: s.t. $\Delta t < \frac{2}{\lambda}$

General conclusion: • easy to write down and to solve for γ^{j+1} (in terms of γ^j)
• there are usually stability conditions, e.g., Δt must be chosen small enough (or later, Δt must be small enough compared to Δs or Δx)

• **Implicit Euler Method:** $\frac{\gamma^{j+1} - \gamma^j}{\Delta t} = -\lambda \gamma^{j+1}$ (rhs is evaluated at $j+1$)

$$\Rightarrow \gamma^{j+1} - \gamma^j = -\lambda \Delta t \gamma^{j+1} \Rightarrow \gamma^{j+1} = \frac{1}{1 + \lambda \Delta t} \gamma^j$$

$$\Rightarrow \gamma^M = \left(\frac{1}{1 + \lambda \Delta t} \right)^M \gamma_0$$

So here, we need that $\left| \frac{1}{1 + \lambda \Delta t} \right| < 1$, which always holds for $\lambda > 0$.

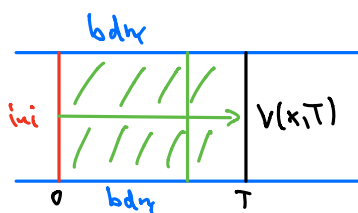
So here the scheme converges unconditionally.

General conclusion: • might not be so easy to solve for γ^{j+1} (in terms of γ^j); and computationally less efficient than explicit schemes
• implicit schemes are usually unconditionally stable

4.5 Application to the Heat Equation

Heat eq.: $\frac{\partial V}{\partial t} = \frac{\partial^2 V}{\partial x^2}$, with initial value $V(x, 0) = V_{in}(x)$

say, with boundary conditions $V(x_{01}, t) = V_0$, $V(x_{max}, t) = V_{max}$



(for B.-S. eq., it is the other way around: given $V(x, T)$, solve for $V(x, 0)$)

$$\frac{\partial v}{\partial t}(x_{i,t_j}) = \frac{V(x_{i,t_{j+1}}) - V(x_{i,t_j})}{\Delta t} + O(\Delta t) \quad \text{explicit}$$

$$\frac{\partial v}{\partial t}(x_{i,t_{j+1}}) = \frac{V(x_{i,t_{j+1}}) - V(x_{i,t_j})}{\Delta t} + O(\Delta t) \quad \text{implicit}$$

$$\text{eq.: } \frac{\partial v}{\partial t}(x,t) = \frac{\partial^2 v}{\partial x^2}(x,t)$$

$$\frac{\partial^2 v}{\partial x^2}(x_{i,t}) = \frac{V(x_{i+1,t}) - 2V(x_{i,t}) + V(x_{i-1,t}))}{(\Delta x)^2} + O(\Delta x^2)$$

\rightarrow explicit: $t = t_j$
 \rightarrow implicit: $t = t_{j+1}$

$$V(x_{i,t_j}) =: V_i^j$$

Explicit scheme: $\frac{V_i^{j+1} - V_i^j}{\Delta t} = \frac{V_{i+1}^j - 2V_i^j + V_{i-1}^j}{(\Delta x)^2}$ (here: given V_i^j , go to V_i^{j+1})

$$\Rightarrow V_i^{j+1} = \frac{\Delta t}{(\Delta x)^2} V_{i+1}^j + \left(1 - 2 \frac{\Delta t}{(\Delta x)^2}\right) V_i^j + \frac{\Delta t}{(\Delta x)^2} V_{i-1}^j$$

we will see numerically that here $\frac{\Delta t}{(\Delta x)^2} < \text{const}$ is a condition for stability

Implicit scheme: $\frac{V_i^{j+1} - V_i^j}{\Delta t} = \frac{V_{i+1}^{j+1} - 2V_i^{j+1} + V_{i-1}^{j+1}}{(\Delta x)^2}$

$$V_i^j = - \underbrace{\frac{\Delta t}{(\Delta x)^2}}_{=: a} V_{i+1}^{j+1} + \left(1 + 2 \frac{\Delta t}{(\Delta x)^2}\right) V_i^{j+1} - \frac{\Delta t}{(\Delta x)^2} V_{i-1}^{j+1}$$

Want to write this as $\vec{V}^j = A \vec{V}^{j+1}$
 \downarrow
 vector in i

In matrix notation:

← boundary condition still missing

$$\underbrace{\begin{pmatrix} (1+2\alpha) & -\alpha & & & 0 \\ -\alpha & (1+2\alpha) & -\alpha & & \\ & -\alpha & \ddots & \ddots & \\ 0 & & & -\alpha & (1+2\alpha) \end{pmatrix}}_A \begin{pmatrix} V_1^{j+1} \\ V_2^{j+1} \\ \vdots \\ V_n^{j+1} \end{pmatrix} = \begin{pmatrix} V_1^j \\ V_2^j \\ \vdots \\ V_n^j \end{pmatrix}$$

tridiagonal matrix

⇒ write solution as $\vec{V}^{j+1} = A^{-1} \vec{V}^j$, i.e., need to invert A ;
 or equivalently, solve the system of linear equations $\vec{V}^j = A \vec{V}^{j+1}$, e.g., by
 Gaussian elimination

Boundary condition:

• $V(x_{0,t}) = V_0$, i.e., $V_0 = V_0^{j+1} \forall j$:

$$V_1^j = -\alpha V_2^{j+1} + (1+2\alpha) V_1^{j+1} - \alpha \underbrace{V_0^{j+1}}_{\text{given boundary condition}}$$

• $V(x_{max,t}) = V_{max}$, i.e., $V_{n+1}^{j+1} = V_{max} \forall j$:

$$V_n^j = -\alpha \underbrace{V_{n+1}^{j+1}}_{\text{given boundary condition}} + (1+2\alpha) V_n^{j+1} - \alpha V_{n-1}^{j+1}$$

So the right equation including the boundary conditions is:

$$\begin{pmatrix} (1+2a) & -a & & & \\ -a & (1+2a) & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & -a & (1+2a) \end{pmatrix} \underbrace{\begin{pmatrix} V_1^{j+1} \\ V_2^{j+1} \\ \vdots \\ V_n^{j+1} \end{pmatrix}}_{\text{what I used to solve for}} = \underbrace{\begin{pmatrix} V_1^j + a V_0^{j+1} \\ V_2^j \\ \vdots \\ V_n^j + a V_{n+1}^{j+1} \end{pmatrix}}_{\text{given}} \begin{matrix} \swarrow \\ \searrow \end{matrix} \begin{matrix} \text{given body conditions} \\ \text{given body conditions} \end{matrix}$$

or
$$A \vec{V}^{j+1} = \vec{V}^j + \begin{pmatrix} a V_0^{j+1} \\ 0 \\ \vdots \\ 0 \\ a V_{n+1}^{j+1} \end{pmatrix}$$