4 Systems of differential equations

In this tutorial, two *implicit* time-stepping methods are introduced, the implicit Euler and the Crank-Nicolson schemes. Once you are familiar with them, we will apply them to systems of first-order differential equations. We will also transform higher-order differential equations into systems of first-order equations, and analyse them as such. Of course they can also be applied two regular differential equations, and the methods from the last tutorial could be applied to systems of differential equations too.

A Implicit Euler

Consider the exponential function $y(t) = e^{-\lambda T}$, which satisfies $y'(t) = -\lambda y(t)$. Simulating it with the explicit Euler method would yield the following time-stepping scheme:

$$y(t+\delta) \approx y(t) - \lambda y(t) = (1 - \delta\lambda)y(t) \tag{1}$$

which creates oscillations if $\delta\lambda > 1$ (first-order difference equation with negative eigenvalue). This only happens due to the discretization of an otherwise continuous system, and is hence undesirable. A solution is the implicit Euler method:

$$\frac{y(t+\delta) - y(t)}{\delta} \approx y'(t+\delta) = f(y(t+\delta), t+\delta) \tag{2}$$

$$\Leftrightarrow y(t+\delta) \approx y(t) + \delta f(y(t+\delta), t+\delta) \tag{3}$$

I.e. we move along the tangent slope of the next point instead of the current. The actual time-stepping scheme is given implicitly as the solution to a fixed-point problem. In the example of the exponential function above:

$$y(t+\delta) = y(t) - \delta\lambda y(t) = (1 - \delta\lambda)y(t+\delta) \tag{4}$$

$$\Leftrightarrow y(t+\delta) = \frac{1}{1+\delta\lambda} y(t) \tag{5}$$

(Note the analogy to forward and backward solutions of exogenous sequences using lag polynomials).

B Crank-Nicolson

As you noticed in the last tutorial, the explicit Euler method creates a bias if the second derivative is non-zero. The same is true for the implicit method, but the bias is in the opposite

direction. A straightforward solution is to take the mean of both, so that the biases cancel each other out (completely, if no effects of orders higher than two are present):

$$y(t+\delta) \approx y(t) + \frac{\delta}{2}\left[f\left(y\left(t\right), t\right) + f\left(y\left(t+\delta\right), t+\delta\right)\right] \tag{6}$$

As the values at the next time step are on both sides, this is also an implicit scheme that requires finding the solution to a fixed-point equation.

B.1 Exercise

Before moving on to systems of equations, complete the functions for a single step using the implicit Euler and Crank-Nicolson methods (`cobweb_ie_1step` and `cobweb_cn_1step`). The model is the continuous-time cobweb model again: $p' = 45 - 3p$.

The code then also provides the time-stepping schemes from Tutorial 3 and the analytical solution of the model to run a comparison along the lines of the comparison in the last tutorial. If everything is correct, both Runge's method and the Crank-Nicolson scheme should exhibit second-order convergence (error ratio = 4), while the Euler methods only converge proportionally (first-order convergence, error ratio = 2).

C Systems of differential equations

Now that we know the main methods, we can apply them to systems of first-order difference equations. This step is analogous to the same extension of difference equations. We simply deal with equations in vectors and matrices instead of scalars, but everything else works in exactly the same way.

D Two goods model

Consider the model in Tutorial 4, exercise 1:

$$p_1' = -2p_1 + 4p_2 \tag{7}$$
$$p_2' = -p_1 + p_2 \tag{8}$$

Or in vector notation:

$$\begin{pmatrix} p_1' \\ p_2' \end{pmatrix} = \begin{pmatrix} -2 & 4 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \tag{9}$$
$$\Leftrightarrow \mathbf{p}' = \mathbf{A}\mathbf{p} \tag{10}$$

We are going to implement one step of the explicit and implicit Euler schemes, as well as Runge's method and the Crank-Nicolson scheme for this system. Recall the definitions:

Explicit Euler (In vector notation:)

$$\mathbf{p}_{t+\delta} = \mathbf{p}_t + \delta\mathbf{p}_t' \tag{11}$$

Which can be implement using the definition of the vector of time derivatives from the original differential equation:

$$\Rightarrow \mathbf{p}_{t+\delta} = \mathbf{p}_t + \delta \mathbf{A}\mathbf{p}_t \tag{12}$$

Implicit Euler

$$\mathbf{p}_{t+\delta} = \mathbf{p}_t + \delta \mathbf{p}'_{t+\delta} = \mathbf{p}_t + \delta \mathbf{A}\mathbf{p}_{t+\delta} \tag{13}$$

This requires a finding the solution to a fixed-point problem, i.e. we have to solve for $\mathbf{p}_{t+\delta}$:

$$\mathbf{p}_{t+\delta}(\mathbf{I} - \delta \mathbf{A}) = \mathbf{p}_t \tag{14}$$

$$\mathbf{p}_{t+\delta} = (\mathbf{I} - \delta \mathbf{A})^{-1}\mathbf{p}_t \tag{15}$$

Where $\mathbf{I}$ is the identity matrix.

Runge's central difference quotient

This one does not change much, it is essentially an extension of the explicit Euler scheme with an intermediate step:

$$\mathbf{p}_{t+\delta} = \mathbf{p}_t + \delta \mathbf{A}\mathbf{p}_{t+\delta/2}\,; \quad \text{with} \tag{16}$$

$$\mathbf{p}_{t+\delta/2} = \mathbf{p}_t + \frac{\delta}{2}\mathbf{A}\mathbf{p}_t \tag{17}$$

Crank-Nicolson

$$\mathbf{p}_{t+\delta} = \mathbf{p}_t + \frac{\delta}{2}(\mathbf{p}'_{t+\delta} + \mathbf{p}'_t) = (\mathbf{I} + \frac{\delta}{2}\mathbf{A})\mathbf{p}_t + \frac{\delta}{2}\mathbf{A}\mathbf{p}_{t+\delta} \tag{18}$$

$$\Leftrightarrow \mathbf{p}_{t+\delta} = (\mathbf{I} - \frac{\delta}{2}\mathbf{A})^{-1}(\mathbf{I} + \frac{\delta}{2}\mathbf{A})\mathbf{p}_t \tag{19}$$

D.1 Exercise

Implement each of these methods in the respective functions in the script: `twogoods_ee`, `twogoods_ie`, `twogoods_runge`, and `twogoods_cn`. They should take as inputs the current price vector p, the coefficient matrix A, and the size of the step δ.

D.2 Exercise

Implement the loop in the function `twogoods_simulate`. The function simulates the behaviour of the model over time for 1 unit time step, which is divided into $1/\delta$ incremental time steps. The particular time-stepping method is passed as an input as well, so we can exchange the methods flexibly and compare behaviour.

Finally, you can use the codes in the script to plot your results, using any preferred time-stepping scheme of your choice. In the left subplot, the behaviour of both prices is plotted over time, while the left one shows the behaviour in the phase space.

D.3 Exercise

Now consider the second-order differential equation $y'' + 2y = 0$. Instead of extending the methods we have learned to higher-order systems, re-state this equation as a 2-dimensional system of first-order differential equations and simulate it. You only have to define the coefficient matrix A, then we can re-use the single-step functions above. Finally, we will compare the three methods with this particular equation.

You can verify that the eigenvalues of this system are both purely imaginary, so that the system neither converges to a steady state, nor breaks out to $\pm\infty$. Instead, it remains at a constant distance to the steady state (in the phase space it is circling around the fixed point). As the Euler schemes are first-order Taylor approximations of the "true" behaviour of the system, they create small errors, when second-order effects are present. Namely, in the case of a convex function ($\frac{\partial^2 f}{\partial x^2} > 0$), the explicit Euler scheme will create a positive bias, and the implicit Euler scheme a negative one. This can easily be proven with Jensen's inequality. The opposite holds for concave functions. The circle being convex, we can verify that the explicit Euler scheme diverges outward, while the implicit one turns inwards (see Figure 1). Taking the mean of both will hence always provide much more accurate solutions! (Note: in physical applications, this is often related to the conservation of energy.) The Runge method would in this case yield practically the same result as the Crank-Nicolson method, which is why only one of them is plotted.

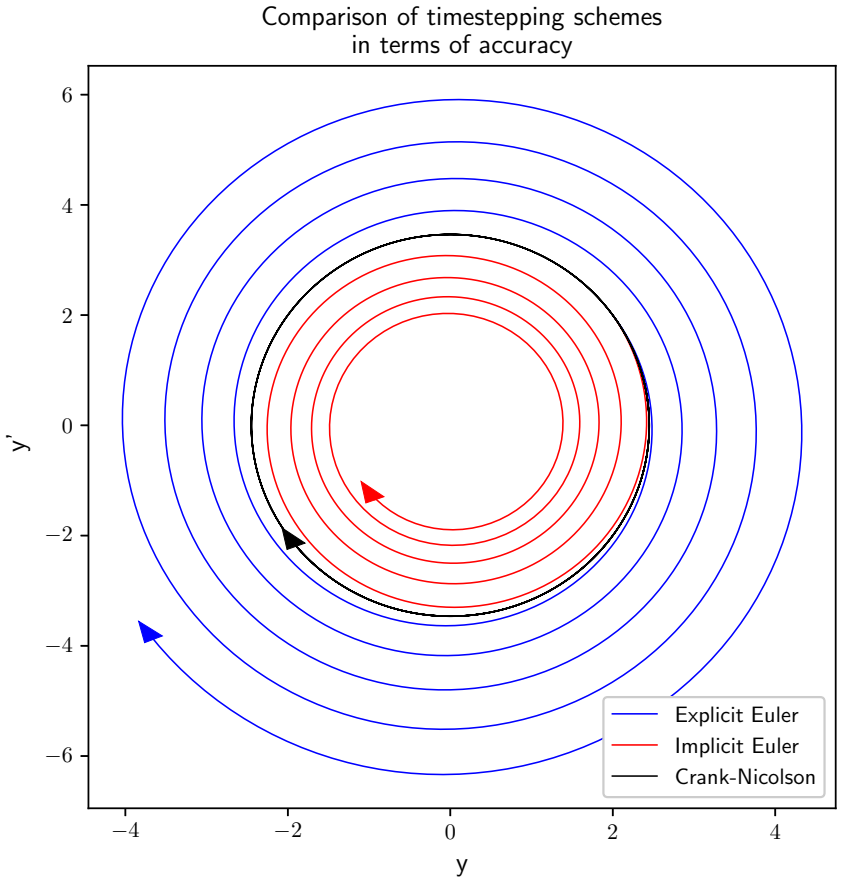

Figure 1: Second-order biases of the explicit and implicit Euler schemes cancel each other out if we apply the Crank-Nicolson method.