Previous Next

12.3.3 Backward Euler for Predator-Prey System

Bookmark this page

MO2.3 MO2.7 MO2.8

For the $M = 2$ predator-prey model, recall that the forcing $\underline{f}(\underline{u}, t)$ (see Section 8.3.5) is,

$$f_0 = a u_0 - b u_0 u_1 \tag{12.40}$$

$$f_1 = -m u_1 + c b u_0 u_1 \tag{12.41}$$

Then, to calculate the Jacobian, we need to find $\nabla \underline{f}(\underline{u}, t^{n+1})$,

$$\nabla \underline{f}(\underline{u}, t^{n+1}) = \begin{pmatrix} \frac{\partial f_0}{\partial u_0} & \frac{\partial f_0}{\partial u_1} \\ \frac{\partial f_1}{\partial u_0} & \frac{\partial f_1}{\partial u_1} \end{pmatrix} \tag{12.42}$$

And, the partial derivatives for this predator-prey model are,

$$\frac{\partial f_0}{\partial u_0} = a - b u_1 \qquad \frac{\partial f_0}{\partial u_1} = -b u_0 \tag{12.43}$$

$$\frac{\partial f_1}{\partial u_0} = c b u_1 \qquad \frac{\partial f_1}{\partial u_1} = -m + c b u_0 \tag{12.44}$$

An example Python implementation is shown below. In it, we make use of the `SciPy` Python module. SciPy is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. In fact, many of the algorithms we study in this class are already implemented in SciPy. In the example program below, we use the SciPy root finding method, `scipy.optimize.root` to solve the implicit Backward Euler iteration for $\underline{v}^{n+1}$. The root finding algorithm being used is an advanced version of the Newton-Raphson algorithm we have discussed (known as the modified Powell algorithm). The specific call to the root-finding method is: `vn1 = optimize.root(evalr, vn, jac=evalj)` where

- `eval` is the function that evaluates the Backward Euler residual
- `vn` is being used as the initial guess for the root-finding method. Since we expect a gradual evolution of $\underline{v}$, then using the current value $\underline{v}^n$ should be a good initial guess for $\underline{v}^{n+1}$.
- `evalj` is the function that evaluates the Jacobian of the Backward Euler residual. The `optimize.root` method does not actually require that the user provide the Jacobian. So, the `jac=evalj` argument

Discussions

All posts sorted by recent activity

‹ Previous Next ›

in the following zip file.