Boundary value problems: shooting method

So far we have been using time-stepping algorithms to compute approximate solutions of *initial value problems*, which consist of an ODE system together with *initial conditions*—the value(s) of the dependent variable(s) specified at a particular "initial" time t_0. However, mathematical models of physical processes often give rise to *boundary value problems*, where values of the dependent variable(s) are given at points on the boundary of the domain over which the problem is defined.

For example, models describing the deflection of a beam subject to loading, waves on a string with fixed ends (e.g. on a musical instrument), and the temperature distribution in a metal rod whose ends are insulated or held at certain temperatures all give rise to boundary value problems.

The simplest example of a boundary value problem is the second-order ODE

$$y'' = f(x, y, y') \qquad \text{defined on the interval } a \leq x \leq b$$

and subject to the boundary conditions

$$y(a) = \alpha, \qquad y(b) = \beta$$

where α and β are given numbers. Note that because the independent variable usually represents a spatial coordinate, it is common to denote it by x (so $' \equiv \frac{d}{dx}$ here).

We shall consider a couple of ways of computing solutions to a second-order boundary value problem like the one above. The first method makes use of time-stepping algorithms (except that here "time" is the x variable); the procedure is as follows:

- Rewrite $y'' = f(x, y, y')$ as a first-order system of two ODEs: $\mathbf{u}' = f(x, \mathbf{u})$ where $\mathbf{u} = \begin{pmatrix} y \\ y' \end{pmatrix}$.
- Pick a value S for the initial slope $y'(a)$ and use a time-stepping algorithm to compute the solution of the initial value problem
$$\mathbf{u}' = f(x, \mathbf{u}), \qquad \mathbf{u}(0) = \begin{pmatrix} y(a) \\ y'(a) \end{pmatrix} = \begin{pmatrix} \alpha \\ S \end{pmatrix}$$
from $x = a$ to $x = b$. Check the value of the computed solution at $x = b$:
if it is greater than the specified $y(b) = \beta$, then this value of S has led to an "overshoot";
if it is less than the specified $y(b) = \beta$, then this value of S has led to an "undershoot".
- Next, find another value of S that yields the opposite over/undershoot effect at $x = b$.

Now we have two values of the slope at $x = a$, call them S_1 and S_2, one of which yields an overshoot and the other an undershoot at $x = b$. So, to obtain a solution that equals the target value β at $x = b$, we should take a slope at $x = a$ that is somewhere between S_1 and S_2. Let $\tilde{y}(x\,; S)$ denote the approximate solution computed using an initial slope S; then essentially we are looking for a zero S^* of the expression $\tilde{y}(b\,; S) - \beta$, and we know that S^* must be bracketed between S_1 and S_2. Therefore:

- Use a root-finding algorithm (such as one of those developed earlier in the course) to find S^* that makes $\tilde{y}(b\,; S) - \beta$ zero. Since we usually don't have an explicit formula for $\tilde{y}(b\,; S) - \beta$ but do know two initial approximations S_1 and S_2 of the root, the bisection method or the secant method would be most suitable.

Examples

$\boxed{1}$ Consider the linear second-order boundary value problem

$$y'' = 5\,(\sinh x)(\cosh^2 x)\,y, \qquad y(-2) = 0.5, \quad y(1) = 1$$

Solve this problem with the shooting method, using `ode45` for time-stepping and the bisection method for root-finding.

$\boxed{2}$ Sometimes, the value of y' rather than y is specified at one or both of the endpoints, e.g. $y'(b) = \gamma$. In this case, we need to find the zero of $\tilde{y}'(b\,;S) - \beta$. It is most common to have $\gamma = 0$, such as in the following modification of Example 1:

$$y'' = 5\,(\sinh x)(\cosh^2 x)\,y, \qquad y(-2) = -0.5, \quad y'(1) = 0$$

$\boxed{3}$ Nonlinear problems such as

$$y'' = -3yy', \qquad y(0) = 0, \quad y(2) = 1$$

can be solved in the same way.