The nonlinear boundary value problem

The general nonlinear boundary value problem is

$$y''(x) = f(x, y(x), y'(x))$$

$$y(a) = \alpha$$

$$y(b) = \beta$$

In the method of finite differences we seek an approximate solution to this boundary value problem by setting up a grid of $N + 2$ equally spaced points x_i with $x_0 = a$ and $x_{N+1} = b$:

$$h = \frac{b-a}{N+1}$$

$$x_i = a + i\,h$$

The method seeks to compute estimates for $y(x_i)$ at each of the interior points x_i for i ranging from 1 to N by replacing the derivative terms with finite difference estimates and solving a set of equations. Specifically, we replace the term $y''(x_i)$ with an $O(h^2)$ centered difference formula

$$y''(x_i) = \frac{y(x_{i+1}) - 2\,y(x_i) + y(x_{i-1})}{h^2} - \frac{h^2}{12}\,y^{(4)}(\xi_i)$$

and we replace the term $y'(x_i)$ with an $O(h^2)$ centered difference formula

$$y'(x_i) = \frac{y(x_{i+1}) - y(x_{i-1})}{2\,h} - \frac{h^2}{6}\,y^{(3)}(\eta_i)$$

Making these substitutions gives

$$\frac{y(x_{i+1}) - 2\,y(x_i) + y(x_{i-1})}{h^2} = f\left(x_i,\, y(x_i),\, \frac{y(x_{i+1}) - y(x_{i-1})}{2\,h}\right)$$

If $f(x,y,y')$ is a nonlinear function, this is a coupled, nonlinear system of equations in the N unknowns $y(x_i)$. If we let w_i be the solution of this equation for $y(x_i)$ for each of these i values with $w_0 = \alpha$ and $w_{N+1} = \beta$, we get a coupled system of nonlinear equations in w_1 through w_N:

$$-\frac{w_{i+1} - 2\,w_i + w_{i-1}}{h^2} + f\left(x_i,\, w_i,\, \frac{w_{i+1} - w_{i-1}}{2\,h}\right) = 0$$

Our problem has now degenerated to a root-finding problem. We seek the root of a function

$$\mathbf{g}(\mathbf{w}) = \begin{bmatrix} g_1(w_1, w_2, \ldots, w_N) \\ g_2(w_1, w_2, \ldots, w_N) \\ \vdots \\ g_N(w_1, w_2, \ldots, w_N) \end{bmatrix}$$

where

$$g_k(w_1, w_2, \ldots, w_N) = -\frac{w_{k+1} - 2\,w_k + w_{k-1}}{h^2} + f\left(x_k, w_k, \frac{w_{k+1} - w_{k-1}}{2\,h}\right)$$

and $w_0 = \alpha$ and $w_{N+1} = \beta$.

Solving the system of equations

To solve the nonlinear root-finding problem we employ Newton's method for functions from $\mathbb{R}^n$ to $\mathbb{R}^n$.

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - J^{-1}(\mathbf{w}^{(k-1)})\,\mathbf{g}(\mathbf{w}^{(k-1)})$$

where the Jacobian matrix is

$$J(\mathbf{x}) = \begin{bmatrix} \frac{\partial g_1(\mathbf{w})}{\partial w_1} & \frac{\partial g_1(\mathbf{w})}{\partial w_2} & \ldots & \frac{\partial g_1(\mathbf{w})}{\partial w_N} \\ \frac{\partial g_2(\mathbf{w})}{\partial w_1} & \frac{\partial g_2(\mathbf{w})}{\partial w_2} & \ldots & \frac{\partial g_2(\mathbf{w})}{\partial w_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_N(\mathbf{w})}{\partial w_1} & \frac{\partial g_N(\mathbf{w})}{\partial w_2} & \ldots & \frac{\partial g_N(\mathbf{w})}{\partial w_N} \end{bmatrix}$$

One fact working to our advantage here is that $g_j(\mathbf{w})$ is independent of w_k for $k < j\text{-}1$ and $k > j\text{+}1$. This means that the Jacobian matrix is tri-diagonal.

Another optimization we can make here is to note that we don't have to compute the inverse of the Jacobian to compute the term

$$J^{-1}(\mathbf{w}^{(k-1)})\,\mathbf{g}(\mathbf{w}^{(k-1)})$$

instead, we can solve the equation

$$J\,(\mathbf{w}^{(k-1)})\,\mathbf{z} = \mathbf{g}(\mathbf{w}^{(k-1)})$$

for **z** and then compute

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \mathbf{z}$$

Since the Jacobian matrix is tri-diagonal, we can apply a Crout factorization to it to get

$$J\,(\mathbf{w}^{(k-1)}) = L\,U$$

and then solve

$$L\,\mathbf{y} = \mathbf{g}(\mathbf{w}^{(k-1)})$$

and

$$U\,\mathbf{z} = \mathbf{y}$$

Summary of the method

Fix $w_0 = \alpha$ and $w_{N+1} = \beta$ and set w_1 through w_N to starting values. (Interpolating linearly between α and β is a good choice.) This generates $\mathbf{w}^{(0)}$.

Now repeat the following steps until $\|\mathbf{g}(\mathbf{w}^{(k)})\|$ drops below a desired tolerance.

1. Compute the tridiagonal Jacobian matrix $J\,(\mathbf{w}^{(k-1)})$.

2. Use Crout factorization to obtain

$$J\,(\mathbf{w}^{(k-1)}) = L\,U$$

3. Use back substitution to solve for $\mathbf{y}$:

$$L\,\mathbf{y} = \mathbf{g}(\mathbf{w}^{(k-1)})$$

4. Use back substitution to solve for $\mathbf{z}$:

$$U\,\mathbf{z} = \mathbf{y}$$

5. Compute

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \mathbf{z}$$

Extrapolation

For the same reasons as applied in the linear case, these results can be improved by extrapolation. The technique is exactly the same: we compute a set of w_i values for a given step size h, and then compute a second set for a step size of $h/2$ and throw away every other w_i. We then form extrapolated w_i values

$$y(x_i) = \frac{4\,w_i(h/2) - w_i(h)}{3} + O(h^4)$$