7 Boundary Value Problems for ODEs

Boundary value problems for ODEs are not covered in the textbook. We discuss this important subject in the scalar case (single equation) only.

7.1 Boundary Value Problems: Theory

We now consider second-order boundary value problems of the general form

$$
\begin{aligned}
y''(t) &= f(t, y(t), y'(t)) \\
a_0 y(a) + a_1 y'(a) = \alpha, &\quad b_0 y(b) + b_1 y'(b) = \beta.
\end{aligned} \tag{55}
$$

Remark

1. Note that this kind of problem can no longer be converted to a system of two first order initial value problems as we have been doing thus far.
2. Boundary value problems of this kind arise in many applications, e.g., in mechanics (bending of an elastic beam), fluids (flow through pipes, laminar flow in a channel, flow through porous media), or electrostatics.

The mathematical theory for boundary value problems is more complicated (and less well known) than for initial value problems. Therefore, we present a version of an existence and uniqueness theorem for the general problem (55).

Theorem 7.1 *Suppose f in (55) is continuous on the domain $D = \{(t, y, z) : a \le t \le b, -\infty < y < \infty, -\infty < z < \infty\}$ and that the partial derivatives f_y and f_z are also continuous on D. If*

1. *$f_y(t, y, z) > 0$ for all $(t, y, z) \in D$,*
2. *there exists a constant M such that*

$$|f_z(t, y, z)| \le M$$

for all $(t, y, z) \in D$, and

3. *$a_0 a_1 \le 0$, $b_0 b_1 \ge 0$, and $|a_0| + |b_0| > 0$, $|a_0| + |a_1| > 0$, $|b_0| + |b_1| > 0$,*

then the boundary value problem (55) has a unique solution.

Example Consider the BVP

$$
\begin{aligned}
y''(t) + e^{-ty(t)} + \sin y'(t) = 0, &\quad 1 \le t \le 2, \\
y(1) = y(2) = 0.&
\end{aligned}
$$

To apply Theorem 7.1 we identify $f(t, y, z) = -e^{-ty} - \sin z$. Then

$$f_y(t, y, z) = te^{-ty}$$

which is positive for all $t > 0$, $y, z \in \mathbb{R}$. So, in particular it is positive for $1 \le t \le 2$. Moreover, we identify $f_z(t, y, z) = -\cos z$, so that

$$|f_z(t, y, z)| = |-\cos z| \le 1 = M.$$

Obviously, all continuity requirements are satisfied. Finally, we have $a_0 = b_0 = 1$ and $a_1 = b_1 = 0$, so that the third condition is also satisfied. Therefore, the given problem has a unique solution.

If the boundary value problem (55) takes the special form

$$\begin{aligned} y''(t) &= u(t) + v(t)y(t) + w(t)y'(t) \\ y(a) &= \alpha, \quad y(b) = \beta, \end{aligned} \tag{56}$$

then it is called *linear*. In this case Theorem 7.1 simplifies considerably.

Theorem 7.2 *If u, v, w in (56) are continuous and $v(t) > 0$ on $[a, b]$, then the linear boundary value problem (56) has a unique solution.*

Remark A classical reference for the numerical solution of two-point BVPs is the book "Numerical Methods for Two-Point Boundary Value Problems" by H. B. Keller (1968). A modern reference is "Numerical Solution of Boundary Value Problems for Ordinary Differential Equations" by Ascher, Mattheij, and Russell (1995).

7.2 Boundary Value Problems: Shooting Methods

One of the most popular, and simplest strategies to apply for the solution of two-point boundary value problems is to convert them to *sequences of initial value problems*, and then use the techniques developed for those methods.

We now restrict our discussion to BVPs of the form

$$\begin{aligned} y''(t) &= f(t, y(t), y'(t)) \\ y(a) &= \alpha, \quad y(b) = \beta. \end{aligned} \tag{57}$$

With some modifications the methods discussed below can also be applied to the more general problem (55).

The fundamental idea on which the so-called *shooting methods* are based is to formulate an initial value problem associated with (57). Namely,

$$\begin{aligned} y''(t) &= f(t, y(t), y'(t)) \\ y(a) &= \alpha, \quad y'(a) = z. \end{aligned} \tag{58}$$

After rewriting this second-order initial value problem as two first-order problems we can solve this problem with our earlier methods (e.g., Runge-Kutta or s-step methods), and thus obtain a solution y_z. In order to see how well this solution matches the solution y of the two-point boundary value problem (57) we compute the difference

$$\phi(z) := y_z(b) - \beta$$

at the right end of the domain. If the initial slope z was chosen correctly, then $\phi(z) = 0$ and we have solved the problem. If $\phi(z) \neq 0$, we can use a solver for nonlinear systems of equations (such as functional iteration or Newton-Raphson iteration discussed in the previous section) to find a better slope.

Remark

1. Changing the "aim" of the initial value problem by adjusting the initial slope to "hit" the target value $y(b) = \beta$ is what gave the name to this numerical method.

2. Even though the shooting method is fairly simple to implement, making use of standard code for initial value problems, and a nonlinear equation solver, it inherits the stability issues encountered earlier for IVP solvers. For boundary value problems the situation is even worse, since even for a stable boundary value problem, the associated initial value problem can be unstable, and thus hopeless to solve.

We illustrate the last remark with

Example For $\lambda < 0$ the (decoupled) boundary value problem

$$\begin{aligned} y_1'(t) &= \lambda y_1(t) \\ y_2'(t) &= -\lambda y_2(t) \\ y_1(0) = 1, &\quad y_2(a) = 1 \end{aligned}$$

for $t \in [0, a]$ is stable since the solution $y_1(t) = e^{\lambda t}$, $y_2(t) = e^{a\lambda}e^{-\lambda t}$ remains bounded for $t \to \infty$ even for large values of a. On the other hand, the initial value problem

$$\begin{aligned} y_1'(t) &= \lambda y_1(t) \\ y_2'(t) &= -\lambda y_2(t) \\ y_1(0) = \alpha, &\quad y_2(0) = \beta \end{aligned}$$

is unstable for any $\lambda \neq 0$ since always one of the components of the solution $y_1(t) = \alpha e^{\lambda t}$, $y_2(t) = \beta e^{-\lambda t}$ will grow exponentially.

Remark A convergence analysis for the shooting method is very difficult since two types of errors are now involved. On the one hand there is the error due to the IVP solver, and on the other hand there is the error due to the discrepancy of the solution at the right boundary.

We now explain how we can use Newton iteration as part of the shooting method. Newton's method for solving the nonlinear equation $\phi(z) = 0$ is

$$z^{[i+1]} = z^{[i]} - \frac{\phi(z^{[i]})}{\phi'(z^{[i]})}, \quad i \geq 0.$$

Now the problem is to obtain the value $\phi'(z^{[i]})$. Note that this is anything but obvious, since we do not even have an expression for the function ϕ — only for the value $\phi(z^{[i]})$.

In order to obtain an expression for $\phi'(z^{[i]})$ we consider the initial value problem (58) in the form

$$\begin{aligned} y''(t, z) &= f(t, y(t, z), y'(t, z)) \\ y(a, z) = \alpha, &\quad y'(a, z) = z. \end{aligned} \tag{59}$$

We now look at the change of the solution y with respect to the initial slope z, i.e.,

$$\begin{aligned} \frac{\partial y''(t, z)}{\partial z} &= \frac{\partial}{\partial z} f(t, y(t, z), y'(t, z)) \\ &= \frac{\partial f}{\partial y}\frac{\partial y}{\partial z} + \frac{\partial f}{\partial y'}\frac{\partial y'}{\partial z}, \end{aligned} \tag{60}$$

where we have omitted the arguments of f, y, and y' in the second line. The initial conditions become

$$\frac{\partial y}{\partial z}(a,z) = 0, \quad \text{and} \quad \frac{\partial y'}{\partial z}(a,z) = 1.$$

If we introduce the notation $v(t) = \frac{\partial y}{\partial z}(t,z)$, then (60) becomes

$$\begin{aligned} v''(t) = \tfrac{\partial f}{\partial y}(t, y(t), y'(t))v(t) + \tfrac{\partial f}{\partial y'}(t, y(t), y'(t))v'(t) \\ v(a) = 0, \quad v'(a) = 1. \end{aligned} \tag{61}$$

Equation (61) is called the *first variational equation* of (**??**). We can recognize this as another initial value problem for the function v.

Now,

$$\phi(z) = y(b,z) - \beta,$$

so that

$$\phi'(z) = \frac{\partial y}{\partial z}(b,z) = v(b).$$

Therefore, we can obtain the value $\phi'(z^{[i]})$ required in Newton's method by solving the initial value problem (61) up to $t = b$.

Algorithm

1. Provide an initial guess z_0 and a tolerance δ.
2. Solve the initial value problems (58) and (61) with initial conditions

   $$y(a) = \alpha, \ y'(a) = z_0, \quad \text{and} \quad v(a) = 0, \ v'(a) = 1,$$

   respectively. Let $i = 0$. This provides us with $\phi(z^{[i]}) = y_{z^{[i]}}(b) - \beta$ and $\phi'(z^{[i]}) = v(b)$.
3. Apply Newton's method, i.e., compute

   $$z^{[i+1]} = z^{[i]} - \frac{\phi(z^{[i]})}{\phi'(z^{[i]})}.$$

4. Check if $|\phi(z^{[i+1]})| < \delta$. If yes, stop. Otherwise, increment i and repeat from 3.

Remark 1. Note that the computation of $\phi(z^{[i+1]})$ in Step 4 requires solution of an IVP (58).

2. The initial value problems (58) and (61) can be solved simultaneously using a vectorized IVP solver.
3. If the boundary value problem (57) is linear, then the function ϕ will also be linear, and therefore a single step of the Newton method will provide the "correct" initial slope.

4. It is also possible to subdivide the interval $[a,b]$, and then apply the shooting method from both ends. This means that additional (internal boundary) conditions need to be formulated that ensure that the solutions match up at the subdivision points. This leads to a *system of nonlinear equations* (even in the scalar case!) which can then be solved using a (modified) Newton-Raphson method. This approach is known as a *multiple shooting method*. More details can be found in the book "Introduction to Numerical Analysis" by Stoer and Bulirsch (1980).

7.3 Boundary Value Problems: Finite Differences

Again we consider the boundary value problem

$$\begin{aligned} y''(t) &= f(t, y(t), y'(t)) \\ y(a) &= \alpha, \quad y(b) = \beta. \end{aligned} \tag{62}$$

Now we create a uniform partition of the interval $[a,b]$ into $m+1$ subintervals $[t_k, t_{k+1}]$, $k = 0, 1, \ldots, m$, where

$$t_k = a + kh, \; k = 0, 1, \ldots, m+1, \quad \text{and} \quad h = \frac{b-a}{m+1}.$$

The basic idea is to *discretize* the differential equation (62) on the given partition.

Before we attempt to solve the BVP (62) we first review approximation of (continuous) derivatives by (discrete) differences.

7.3.1 Numerical Differentiation

From calculus we know that the value of the derivative of a given function f at some point x in its domain can be approximated via

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}, \tag{63}$$

where h is small. In order to get an error estimate for this approximation we use a Taylor expansion of f

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(\eta), \qquad \eta \in (x, x+h).$$

This implies

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2} f''(\eta),$$

i.e., the truncation error for the standard difference approximation of the first derivative is $\mathcal{O}(h)$.

We now consider a more accurate approximation. To this end we take *two* Taylor expansions

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(x) + \frac{h^3}{6} f'''(\eta_1), \tag{64}$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2} f''(x) - \frac{h^3}{6} f'''(\eta_2). \tag{65}$$

Subtracting (65) from (64) yields

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{h^3}{6}\left[f'''(\eta_1) + f'''(\eta_2)\right]$$

or the following formula

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + \mathcal{O}(h^2) \tag{66}$$

for the first derivative which is more accurate than (63).

Similarly, by adding (64) and (65) we obtain the following formula for the second derivative

$$f''(x) = \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} + \mathcal{O}(h^2). \tag{67}$$

As with the basic numerical integration methods, there is again a close connection between numerical differentiation methods and polynomial interpolation. If we have information of f at $n+1$ points, then we can find an interpolating polynomial p of degree n. We then differentiate p to get an estimate for the derivative of f.

Consider the error formula for the Lagrange form of the interpolating polynomial (see (7) in Chapter 1)

$$f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) w(x)$$

or

$$f(x) = \sum_{j=0}^{n} f(x_j) p_j(x) + \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) w(x)$$

where $w(x) = \prod_{i=0}^{n}(x - x_i)$, and the p_j are the Lagrange basis polynomials as studied in Chapter 1. It is important for the next step to note that the point ξ in the error formula depends on the evaluation point x. This explains the use of the notation ξ_x.

Differentiation then leads to

$$f'(x) = \sum_{j=0}^{n} f(x_j) p_j'(x) + \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) w'(x) + \frac{1}{(n+1)!} \frac{d}{dx}\left[f^{(n+1)}(\xi_x)\right] w(x).$$

Let us now assume that the evaluation point x is located at one of the interpolation nodes, x_k say, i.e., we know f at certain points, and want to estimate f' at (some of) those same points. Then $w(x_k) = 0$ and

$$f'(x_k) = \sum_{j=0}^{n} f(x_j) p_j'(x_k) + \frac{1}{(n+1)!} f^{(n+1)}(\xi_{x_k}) w'(x_k).$$

One can simplify this expression to

$$f'(x_k) = \sum_{j=0}^{n} f(x_j) p_j'(x_k) + \frac{1}{(n+1)!} f^{(n+1)}(\xi_{x_k}) \prod_{\substack{i=0 \\ i \neq k}}^{n} (x_k - x_i). \tag{68}$$

Remark 1. If all nodes are equally spaced with spacing h, then (68) in an $\mathcal{O}(h^n)$ formula.

2. The values $p_j'(x_k)$ in (68) are called the *coefficients* of the derivative formula.

Example 1. Using linear interpolation at two equally spaced points, $x_0 = x$ and $x_1 = x + h$, leads to the estimate (63).

2. (66) is obtained by performing quadratic interpolation at $x_0 = x - h$, $x_1 = x$, and $x_2 = x + h$.

3. (67) is obtained by performing quadratic interpolation at $x_0 = x - h$, $x_1 = x$, and $x_2 = x + h$.

4. These and other examples are illustrated in the Maple worksheet `478578_DerivativeEstimates.mws`.

Remark The discussion in Section 7.1 of the Iserles textbook employs a more abstract framework based on discrete *finite difference operators* and formal Taylor expansions of these operators.

We now return the the finite difference approximation of the BVP (62) and introduce the following formulas for the first and second derivatives:

$$\begin{aligned} y'(t) &= \frac{y(t+h) - y(t-h)}{2h} - \frac{h^2}{6} y^{(3)}(\eta) \\ y''(t) &= \frac{y(t+h) - 2y(t) + y(t-h)}{h^2} - \frac{h^2}{12} y^{(4)}(\tau). \end{aligned} \tag{69}$$

If we use the notation $y_k = y(t_k)$ along with the finite difference approximations (69), then the boundary value problem (62) becomes

$$\begin{aligned} y_0 &= \alpha \\ \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} &= f\left(t_k, y_k, \frac{y_{k+1} - y_{k-1}}{2h}\right), \quad k = 1, \ldots, m, \\ y_{m+1} &= \beta. \end{aligned} \tag{70}$$

7.3.2 Linear Finite Differences

We now first discuss the case in which f is a *linear* function of y and y', i.e.,

$$f(t, y(t), y'(t)) = u(t) + v(t)y(t) + w(t)y'(t).$$

Then (70) becomes

$$\begin{aligned} y_0 &= \alpha \\ \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} &= u_k + v_k y_k + w_k \frac{y_{k+1} - y_{k-1}}{2h}, \quad k = 1, \ldots, m, \\ y_{m+1} &= \beta, \end{aligned} \tag{71}$$

where we have used the notation $u_k = u(t_k)$, $v_k = v(t_k)$, and $w_k = w(t_k)$. This is a system of m linear equations for the m unknowns y_k, $k = 1, \ldots, m$. In fact, the system is *tridiagonal*. This can be seen if we rewrite (71) as

$$y_0 = \alpha$$

$$
\begin{aligned}
\left(-1-\frac{w_k}{2}h\right)y_{k-1}+\left(2+h^2v_k\right)y_k+\left(-1+\frac{w_k}{2}h\right)y_{k+1} &= -h^2u_k, \quad k=1,\ldots,m,\\
y_{m+1} &= \beta,
\end{aligned}
$$

or in matrix form

$$
\begin{bmatrix}
2+h^2v_1 & -1+\frac{w_1}{2}h & 0 & \ldots & 0\\
-1-\frac{w_2}{2}h & 2+h^2v_2 & -1+\frac{w_2}{2}h & & \vdots\\
\vdots & \ddots & \ddots & \ddots & \vdots\\
\vdots & & -1-\frac{w_{m-1}}{2}h & 2+h^2v_{m-1} & -1+\frac{w_{m-1}}{2}h\\
0 & \ldots & 0 & -1-\frac{w_m}{2}h & 2+h^2v_m
\end{bmatrix}\times
$$

$$
\times\begin{bmatrix} y_1\\ y_2\\ \vdots\\ y_{m-1}\\ y_m\end{bmatrix}
=\begin{bmatrix} -h^2u_1-\alpha\left(-1-\frac{w_1}{2}h\right)\\ -h^2u_2\\ \vdots\\ -h^2u_{m-1}\\ -h^2u_m-\beta\left(-1+\frac{w_m}{2}h\right)\end{bmatrix}.
$$

Remark As with our earlier solvers for initial value problems (which were also used for the shooting method) the numerical solution is obtained only as a set of discrete values $\{y_k : k=0,1,\ldots,m+1\}$. However, all values are *obtained simultaneously* once the linear system is solved.

The tridiagonal system above can be solved most efficiently if we can ensure that it is *diagonally dominant*, since then a tridiagonal Gauss solver without pivoting (such as `tridiag.m`; compare MATLAB 6 vs. MATLAB 7) can be applied. Diagonal dominance for the above system means that we need to ensure

$$
|2+h^2v_k| > |1+\frac{h}{2}w_k| + |1-\frac{h}{2}w_k|.
$$

This inequality will be satisfied if we assume $v_k > 0$, and that the discretization is so fine that $|\frac{h}{2}w_k| < 1$. Under these assumptions we get

$$
2+h^2v_k > 1+\frac{h}{2}w_k+1-\frac{h}{2}w_k = 2 \quad\Longleftrightarrow\quad h^2v_k > 0
$$

which is obviously true.

Remark

1. The assumption $v_k > 0$ is no real restriction since this is also a condition for the Existence and Uniqueness Theorem 7.2.
2. The assumption $|\frac{h}{2}w_k| < 1$ on the mesh size h is a little more difficult to verify.

For the linear finite difference method one can give error bounds.

Theorem 7.3 *The maximum pointwise error of the linear finite difference method is given by*

$$\max_{k=1,\ldots,m} |y(t_k) - y_k| \leq Ch^2, \quad \textit{as } h \to 0,$$

where $y(t_k)$ is the exact solution at t_k, and y_k is the corresponding approximate solution obtained by the finite difference method.

Proof For the exact solution we have for any $k = 1, \ldots, m$

$$y''(t_k) = u(t_k) + v(t_k)y(t_k) + w(t_k)y'(t_k)$$

or

$$\frac{y(t_k+h) - 2y(t_k) + y(t_k-h)}{h^2} - \frac{h^2}{12}y^{(4)}(\tau_k) = u_k + v_k y(t_k) + w_k\left[\frac{y(t_k+h) - y(t_k-h)}{2h} - \frac{h^2}{6}y^{(3)}(\eta_k)\right], \tag{72}$$

whereas for the approximate solution we have the relation

$$\frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} = u_k + v_k y_k + w_k \frac{y_{k+1} - y_{k-1}}{2h}$$

(cf. (71)). Subtracting (71) from equation (72) yields

$$\frac{e_{k+1} - 2e_k + e_{k-1}}{h^2} = v_k e_k + w_k \frac{e_{k+1} - e_k}{2h} + h^2 g_k, \tag{73}$$

where

$$e_k = y(t_k) - y_k$$

and

$$g_k = \frac{1}{12}y^{(4)}(\tau_k) - \frac{1}{6}y^{(3)}(\eta_k).$$

Since (73) is analogous to (71) it can be rewritten as

$$\left(-1 - \frac{w_k}{2}h\right)e_{k-1} + \left(2 + h^2 v_k\right)e_k + \left(-1 + \frac{w_k}{2}h\right)e_{k+1} = -h^4 g_k.$$

Then we get

$$\left|\left(2 + h^2 v_k\right)e_k\right| = \left|-\left(-1 - \frac{w_k}{2}h\right)e_{k-1} - \left(-1 + \frac{w_k}{2}h\right)e_{k+1} - h^4 g_k\right|$$

and using the triangle inequality

$$\left|\left(2 + h^2 v_k\right)e_k\right| \leq \left|\left(-1 - \frac{w_k}{2}h\right)e_{k-1}\right| + \left|\left(-1 + \frac{w_k}{2}h\right)e_{k+1}\right| + \left|h^4 g_k\right|.$$

Now we let $\lambda = \|e\|_\infty = \max_{j=1,\ldots,m}|e_j|$, and pick the index k such that

$$|e_k| = \|e\|_\infty = \lambda,$$

i.e., we look at the largest of the errors. Therefore

$$|2 + h^2 v_k|\underbrace{|e_k|}_{=\lambda} \leq h^4|g_k| + |-1 + \frac{w_k}{2}h|\underbrace{|e_{k+1}|}_{\leq\lambda} + |-1 - \frac{w_k}{2}h|\underbrace{|e_{k-1}|}_{\leq\lambda}.$$

Using the definition of λ, and bounding $|g_k|$ by its maximum we have

$$\lambda\left(|2+h^2v_k| - |-1+\frac{w_k}{2}h| - |-1-\frac{w_k}{2}h|\right) \le h^4\|g\|_\infty.$$

Using the same assumptions and arguments as in the diagonal dominance discussion above, the expression in parentheses is equal to h^2v_k, and therefore we have

$$\lambda h^2 v_k \le h^4\|g\|_\infty \quad \Longleftrightarrow \quad \lambda v_k \le h^2\|g\|_\infty,$$

or, since $\lambda = \|e\|_\infty$,

$$\max_{k=1,\ldots,m} |y(t_k) - y_k| \le Ch^2,$$

where

$$C = \frac{\|g\|_\infty}{\min_{a\le t\le b} v(t)}.$$

■

Remark The error bound in Theorem 7.3 holds only for C^4 functions y, whereas for the solution to exist only C^2 continuity is required.

7.3.3 Nonlinear Finite Differences

We now return to the original discretization

$$\begin{aligned} y_0 &= \alpha \\ \frac{y_{k+1}-2y_k+y_{k-1}}{h^2} &= f\left(t_k, y_k, \frac{y_{k+1}-y_{k-1}}{2h}\right), \quad k=1,\ldots,m, \\ y_{m+1} &= \beta \end{aligned}$$

of the boundary value problem (62). However, now we allow f to be a nonlinear function. This leads to the following system of *nonlinear* equations:

$$\begin{aligned} 2y_1 - y_2 + h^2 f\left(t_1, y_1, \frac{y_2-\alpha}{2h}\right) - \alpha &= 0 \\ -y_{k-1} + 2y_k - y_{k+1} + h^2 f\left(t_k, y_k, \frac{y_{k+1}-y_{k-1}}{2h}\right) &= 0, \quad k=2,\ldots,m-1, \\ -y_{m-1} + 2y_m + h^2 f\left(t_m, y_m, \frac{\beta - y_{m-1}}{2h}\right) - \beta &= 0. \end{aligned} \tag{74}$$

One can show that this system has a unique solution provided

$$h < \frac{2}{M},$$

where M is the same as in the Existence and Uniqueness Theorem 7.1.

To solve the system we need to apply Newton iteration for nonlinear systems. This is done by solving the linear system

$$J(\mathbf{y}^{[i]})\mathbf{u} = -F(\mathbf{y}^{[i]})$$

for $\mathbf{u}$, and then updating

$$\mathbf{y}^{[i+1]} = \mathbf{y}^{[i]} + \mathbf{u},$$

where $\mathbf{y}^{[i]}$ is the i-th iterate of the vector of grid values $y_0, y_1, \ldots, y_{m+1}$, and J is the tridiagonal Jacobian matrix defined by

$$J(\mathbf{y})_{k\ell} = \begin{cases} -1 + \frac{h}{2} f_z \left(t_k, y_k, \frac{y_{k+1}-y_{k-1}}{2h} \right), & k = \ell - 1, \ \ell = 2, \ldots, m, \\ 2 + h^2 f_y \left(t_k, y_k, \frac{y_{k+1}-y_{k-1}}{2h} \right), & k = \ell, \ \ell = 1, \ldots, m, \\ -1 - \frac{h}{2} f_z \left(t_k, y_k, \frac{y_{k+1}-y_{k-1}}{2h} \right), & k = \ell + 1, \ \ell = 1, \ldots, m-1. \end{cases}$$

Here $f = f(t, y, z)$ and $F(\mathbf{y})$ is given by the left-hand side of the equations in (74).

Remark 1. As always, Newton iteration requires a "good" initial guess $y_0, \ldots, y_{m+1}$.

2. One can show that the nonlinear finite difference method also has $\mathcal{O}(h^2)$ convergence order.