Sample Solutions for Assignment 3.

Reading: Sec. 2.16-2.17.

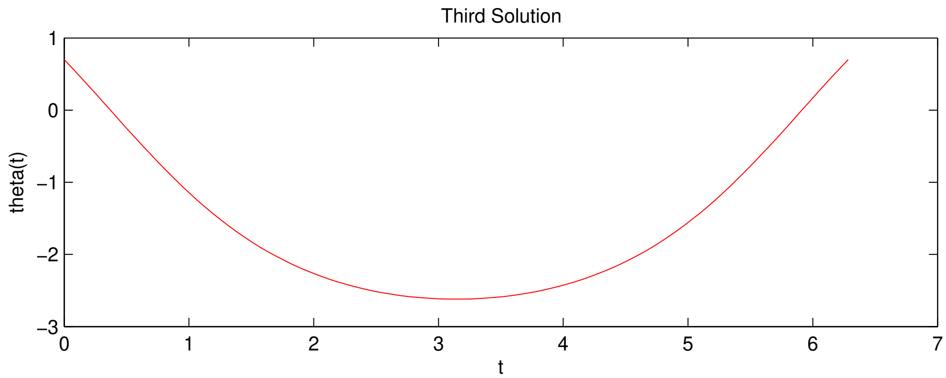
1. (nonlinear pendulum)

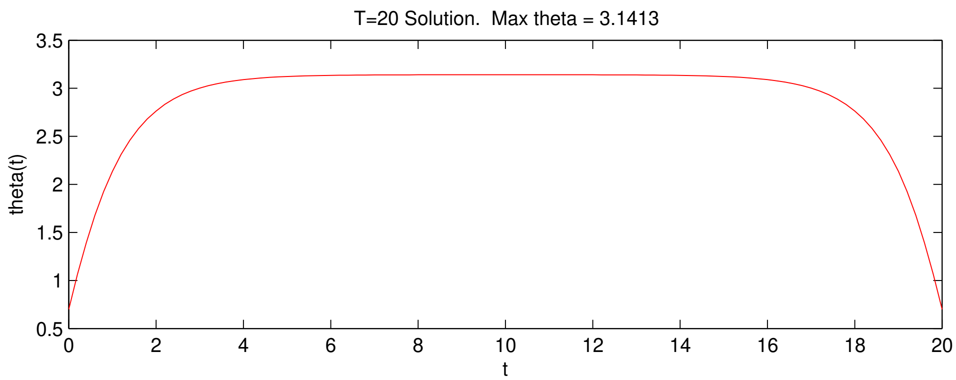
   (a) Write a program to solve the boundary value problem for the nonlinear pendulum as discussed in the text. See if you can find yet another solution for the boundary conditions illustrated in Figures 2.4 and 2.5.

   > See Matlab code `nonlinear_pendulum.m`. To see another solution, enter `3` when it asks which initial guess. (Initial guesses 1 and 2 lead to the results in Figures 2.4 and 2.5.) On the next page is a plot of the third solution that I found, along with the solution in part (b).

   (b) Find a numerical solution to this BVP with the same general behavior as seen in Figure 2.5 for the case of a longer time interval, say, $T = 20$, again with $\alpha = \beta = 0.7$. Try larger values of T. What does $\max_i \theta_i$ approach as T is increased? Note that for large T this solution exhibits "boundary layers".

   > I was able to get Newton's method to converge with $T = 20$ by first solving the problem with $T = 2\pi$ and then using that as the initial guess for the $T = 20$ problem. (In the code `nonlinear_pendulum.m`, you can first enter `2` when it asks which initial guess. When the code has run, type `thetasave=theta;`. Then run the code again and enter `0` when it asks which initial guess. Then type `thetasave` when it asks for theta.) I then used the solution from the $T = 20$ problem as the initial guess for problems with larger T. For each of these problems I used 100 subintervals. The maximum value of θ approached π as T was increased. For $T = 50$, the maximum value of θ was *surprisingly* close to π: 3.14159265349159.





2. (Gerschgorin's theorem and stability) Consider the boundary value problem

$$-u_{xx} + (1 + x^2)u = f, \quad 0 \le x \le 1,$$

$$u(0) = 0, \quad u(1) = 0.$$

On a uniform grid with spacing $h = 1/(m+1)$, the following set of difference equations has local truncation error $O(h^2)$:

$$\frac{2u_i - u_{i+1} - u_{i-1}}{h^2} + (1 + x_i^2)u_i = f(x_i), \quad i = 1, \ldots, m.$$

(a) Use Gerschgorin's theorem to determine upper and lower bounds on the eigenvalues of the coefficient matrix for this set of difference equations.

All of the eigenvalues lie in

$$\left\{z : \left|z - \left(\frac{2}{h^2} + 1 + h^2\right)\right| \le \frac{1}{h^2}\right\} \cup \bigcup_{i=2}^{m-1} \left\{z : \left|z - \left(\frac{2}{h^2} + 1 + x_i^2\right)\right| \le \frac{2}{h^2}\right\} \cup$$

$$\left\{z : \left|z - \left(\frac{2}{h^2} + 1 + (mh)^2\right)\right| \le \frac{1}{h^2}\right\}.$$

Therefore the smallest eigenvalue is at least $\min\{\frac{1}{h^2} + 1 + h^2,\ 1 + (2h)^2\}$, which is greater than 1. The largest eigenvalue is at most $\max\{\frac{3}{h^2} + 1 + (mh)^2,\ \frac{4}{h^2} + 1 + ((m-1)h)^2\}$, which is less than $\frac{4}{h^2} + 2$.

(b) Show that the L_2-norm of the *global error* is of the same order as the local truncation error.

If τ denotes the local truncation error vector and $\mathbf{e}$ the global error vector, then $A\mathbf{e} = \tau$, or, $\mathbf{e} = A^{-1}\tau$. It follows that $\|\mathbf{e}\|_{L_2} \leq \|A^{-1}\|_2 \cdot \|\tau\|_{L_2}$, and $\|\mathbf{e}\|_{L_2}$ will be of the same order in h as $\|\tau\|_{L_2}$ if $\|A^{-1}\|_2$ is bounded above by a constant independent of h. Since A is symmetric, the 2-norm of A^{-1} is the largest absolute value of an eigenvalue of A^{-1}, which is the reciprocal of the smallest absolute value of an eigenvalue of A. From (a), all eigenvalues of A are greater than 1, and hence $\|A^{-1}\| < 1$.

3. (Richardson extrapolation) Use your code from problem 6 in assignment 1, or download the code from the course web page to do the following exercise. Run the code with $h = .1$ (10 subintervals) and with $h = .05$ (20 subintervals) and apply Richardson extrapolation to obtain more accurate solution values on the coarser grid. Record the L_2-norm or the ∞-norm of the error in the approximation obtained with each h value and in that obtained with extrapolation.

   Using $h = .1$, I got .0018 for the L_2-norm of the error. Using $h = .05$, I got $4.2635e-4$ for the L_2-norm of the error – a reduction of about a factor of 4, indicating second order accuracy. Therefore to do the extrapolation, I took (4/3) times the fine grid solution (at the even nodes) minus (1/3) times the course grid solution. I obtained an approximate solution on the course grid for which the L_2-norm of the error was $5.4607e-6$.

   Suppose you assume that the coarse grid approximation is piecewise linear, so that the approximation at the midpoint of each subinterval is the average of the values at the two endpoints. Can one use Richardson extrapolation with the fine grid approximation and these interpolated values on the coarse grid to obtain a more accurate approximation at these points? Explain why or why not?

   Although the error in the approximation at the midpoint of each subinterval is $O(h^2)$, the constant C_I multiplying this error is *different* from the constant C multiplying the $(h/2)^2$ error term in the fine grid approximation. Hence if one tries to take 4/3 times the fine grid approximation minus 1/3 times the interpolated coarse grid approximation, the $O(h^2)$ error terms will not cancel. If one used a higher order interpolation method, such as quadratic interpolation, where the interpolation error is $O(h^3)$, then one should be able to use Richardson extrapolation to eliminate the $O(h^2)$ error terms. Alternatively, one could try to estimate the constant C_I and relate it to C. Then one could eliminate the $O(h^2)$ error by taking $(4C_I)/(4C_I - C)$ times the fine grid approximation minus $C/(4C_I - C)$ times the interpolated value from the coarse grid.

4. Write down the Jacobian matrix associated with Example 2.2 and the nonlinear difference equations (2.106) on p. 49. Write a code to solve these difference equations when $a = 0$, $b = 1$, $\alpha = -1$, $\beta = 1.5$, and $\epsilon = 0.01$. Use an initial guess of the sort suggested

in the text. Try, say, $h = 1/20$, $h = 1/40$, $h = 1/80$, and $h = 1/160$, and turn in a plot of your results.

The Jacobian $J_G|_U$ is an m by m tridiagonal matrix whose nonzero entries are:

$$J_G(i,i) = -\frac{2\epsilon}{h^2} + \frac{U_{i+1} - U_{i-1}}{2h} - 1,$$

$$J_G(i,i+1) = \frac{\epsilon}{h^2} + \frac{U_i}{2h}, \quad J_G(i,i-1) = \frac{\epsilon}{h^2} - \frac{U_i}{2h}.$$

See Matlab code `ex2_2.m`, which produced the plot below.