Applied Numerical Methods - Math 151B

Homework 7

Erik Lamb

May 17, 2016

In this assignment we solved the following nonlinear system of equations,

$$\begin{aligned} 15x_1 + x^2 - 4x_3 &= 13 \\ x_1^2 + 10x_2 - x_3 &= 11 \\ x_2^3 - 25x_3 &= -22 \end{aligned} \tag{1}$$

with the initial approximation $\mathbf{x}^{(0)} = \mathbf{0}$. From equation (1) we see that our vector of nonlinear root $F(\mathbf{x})$ equations is,

$$F(\mathbf{x}) = \begin{pmatrix} 15x_1 + x^2 - 4x_3 - 13 \\ x_1^2 + 10x_2 - x_3 - 11 \\ x_2^3 - 25x_3 + 22 \end{pmatrix} \tag{2}$$

Knowing $F(\mathbf{x})$, we can its corresponding Jacobian matrix $J(\mathbf{x})$, given by

$$J(\mathbf{x}) = \begin{pmatrix} 15 & 2x_2 & -4 \\ 2x_1 & 10 & -1 \\ 0 & 3x_2^2 & -25 \end{pmatrix} \tag{3}$$

Thus our desired root or fixed-point solution in this multidimensional case is a vector $\mathbf{x} = (x_1, x_2, x_3)^T$. $F(\mathbf{x})$ and the Jacobian matrix $J(\mathbf{x})$ are very important for finding a approximation of the the solution to (1), and in fact are used in all of the methods we consider in this assignment. These methods include the Improved Broyden's Method, where approximations are made with the equation $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + s_{k+1}$ and $s_{k+1} = -B_k F(\mathbf{x}^{(k)})$, where B_k is an approximation of the Jacobian matrix (3); the Method of Steepest Descent, which makes the approximation $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha \nabla g(\mathbf{x}^{(k)})$ where α minimizes the characteristic equation $P(\alpha) = h(\alpha_1) + h_{1,2}\alpha + h_{1,2,3}\alpha(\alpha - \alpha_2)$, $h(\alpha) = g(\mathbf{x}^{(k)} - \alpha \nabla g(\mathbf{x}^{(k)})$, and $\nabla g(\mathbf{x}^{(k)}) = 2J(\mathbf{x})^T F(\mathbf{x})$; the Homotopy method, which maps $F(\mathbf{x})$ onto the domain $\lambda \in [0, 1]$ thereby transforming the nonlinear system into a linear system where the increment function Φ is approximated by $\Phi(\lambda, \mathbf{x}) = -J(\mathbf{x}(\lambda))^{-1} F(\mathbf{x}(0))$, which can then be solved by any IVP, where we used the Midpoint and Runge-Kutta of order four (RK4) to solve the system of IVPs. Finally, all of our approximations were compared with the solution, $\mathbf{x}^*$, found using Newton's Method for solving a nonlinear system of equations, given by $\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} - J(\mathbf{x}^{(k-1)})^{-1} F(\mathbf{x}^{(k-1)})$, with a tolerance of $||\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}||_\infty < \epsilon$, where ϵ is known as "machine epsilon," given in MATLAB by `eps` and on the order of 2^{-52}. All methods, except for the Homotopy method which iterates for N sub-intervals, iterates until the infinite norm of the difference of the last approximation from the previous approximation, $||\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}||_\infty = max|x_i^{(k+1)} - x_i^{(k)}|$, is less than a given tolerance.

Having briefly discussed the nonlinear numerical methods we will use, we now present their results. This is best accomplished by presenting all of our data in a table so they can be easily compared, given on the following page. All errors in the last column are found against the "actual" solution $\mathbf{x}^*$ given by Newton's method for a tolerance $\epsilon = 2^{-52}$. The solution is

$$\mathbf{x}^* = \begin{pmatrix} 1.036400470329211 \\ 1.085706550741678 \\ 0.931191442315390 \end{pmatrix} \tag{4}$$

and was found in five iterations with a run-time of 9.2882×10^{-4} seconds.

In the table below, the name of the method used is given in the first column, the tolerance for the approximation in the second, the number of iteration in the third, the number of sub-intervals N in the fourth, the run-time in seconds in the fifth, the approximation $\mathbf{x}^{(k)}$ in the sixth, and the error from (4) in the seventh. For practical reasons, only the first five digits are given in the approximations in column six. Note that for the Homotopy method the RK4 or Midpoint indicates which of the two IVP solvers was used for the following approximations.

Of all of the methods, the Improved Broyden method is the fastest with a run-time of 0.007 seconds and produces a highly accurate result on the order of 10^{-11}. It is also tied with the Steepest Descent Method in number of iterations, requiring only 7 iterations to achieve a much, much better approximation than the steepest descent method. The steepest descent method, when given a rather large tolerance of 0.05, produces the most inaccurate result that is on the order of 10^{-1}. Additionally, it requires more time to than any other method to reach this poor approximation, likely due to the double nested while loops and many computations involved in implementing the steepest descent method in the MATLAB function `steepest.m`. Thus, in order for the steepest descent method to achieve the same accuracy as the Broyden method or even a homotopy method, it would have to have a very small tolerance and a large run-time. The homotopy methods, for both IVP solvers provide substantially better approximations than the steepest decline method. Their approximations achieve better and better accuracy for more sub-intervals N. The homotopy method using the midpoint solver, however, only improves in accuracy by 10^{-1} when the number of sub-intervals is increased by five, from 10 to 50. This is totally unsatisfactory compared to the homotopy method with the RK4 solver, which improves in accuracy by 10^{-3}, from 10^{-8} to 10^{-11}, when the sub-intervals are increased by five. Thus the homotopy method using RK4 is superior to the homotopy method using the midpoint solver. In fact, the homotopy method using RK4 for $N = 50$ sub-intervals has the smallest error of any of the approximations in Table 1. The homotopy method has a couple of drawbacks, however. First is that it requires 50 iterations to achieve the same degree of accuracy as the Broyden method, which only required seven. Because of this, the homotopy method using RK4 takes almost twice as long as the Broyden method. Overall, then, the Improved Broyden Method is the most superior of those in Table 1, since it requires the least amount of iterations, finds a solution in the smallest amount of time, and finds a solution accurate to 10^{-11}, the highest order of accuracy exhibited in Table 1. Improving accuracy by about 5^{-12} is not worth the computational cost of using the homotopy method with an RK4 solver versus the Improved Broyden method. The Broyden method is the most superior due to its super-linear convergence.

All methods in Table 1, however, pale in performance when compared to the Newton method result given in equation (4). The Newton method is the most accurate and fastest result, requiring only 9×10^{-4} seconds, to achieve accuracy on the order of 2^{-52}. This is because all methods in Table 1 make some form of an approximation for either the Jacobian matrix or the increment function whereas the Newton method, though it requires more preliminary computation, provides the most information and therefore the best approximation of the five methods considered.

Method	$\|\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|\|_\infty$	Iterations	N	Run-time (s)	$\mathbf{x}^{(k)}$	$\|\|\mathbf{x}^{(k)} - \mathbf{x}^*\|\|_\infty$
`broyden.m`	10^{-6}	7	N/A	0.007	$\begin{pmatrix} 1.0364 \\ 1.0857 \\ 0.9312 \end{pmatrix}$	3.1293×10^{-11}
`steepest.m`	0.05	7	N/A	0.0664	$\begin{pmatrix} 1.0723 \\ 0.9409 \\ 0.9283 \end{pmatrix}$	0.1448
`homotopy.m`/RK4	N/A	N/A	10	0.0068	$\begin{pmatrix} 1.0364 \\ 1.0857 \\ 0.9312 \end{pmatrix}$	1.5904×10^{-8}
`homotopy.m`/Midpoint	N/A	N/A	10	0.0031	$\begin{pmatrix} 1.0363 \\ 1.0857 \\ 0.9311 \end{pmatrix}$	8.2858×10^{-5}
`homotopy.m`/RK4	N/A	N/A	20	0.0126	$\begin{pmatrix} 1.0364 \\ 1.0857 \\ 0.9312 \end{pmatrix}$	9.8267×10^{-10}
`homotopy.m`/Midpoint	N/A	N/A	20	0.0061	$\begin{pmatrix} 1.0364 \\ 1.0857 \\ 0.9312 \end{pmatrix}$	2.0655×10^{-5}
`homotopy.m`/RK4	N/A	N/A	50	0.0311	$\begin{pmatrix} 1.0364 \\ 1.0857 \\ 0.9312 \end{pmatrix}$	2.4976×10^{-11}
`homotopy.m`/Midpoint	N/A	N/A	50	0.0153	$\begin{pmatrix} 1.0364 \\ 1.0857 \\ 0.9312 \end{pmatrix}$	3.2989×10^{-6}

Table 1: Our data for all eight approximations. Of all of the methods, the Improved Broyden method is the fastest with a run-time of 0.007 seconds and produces a highly accurate result on the order of 10^{-11}. The steepest descent method is the slowest and has the worst accuracy, requiring a long time to achieve the same level of accuracy as the Broyden or homotopy methods. The homotopy methods improve in accuracy with a larger number of sub-intervals, but the midpoint solver barely increases in accuracy by only 10^{-1} when the number of sub-intervals is increased from 10 to 50, maxing out at accuracy on the order of 10^{-6}. The RK4 solver, on the other hand, increases in accuracy by 10^{-3} to maximum level of accuracy on the order of 10^{-11}. The RK4 solver is clearly the superior homotopy method, especially since the run-times of both methods are comparable. The homotopy method with the RK4 solver for 50 sub-intervals provides the most accurate approximation in Table 1, but it is still on the order of the Broyden method, and the lacks the Broyden method's speed. The Broyden method is the most superior in Table 1, because it provides the same level of accuracy as the homotopy method with the RK4 solver for 50 sub-intervals in less than half of the time and less than a fifth of the iterations. This is due to the fact that the Broyden method can achieve super-linear convergence.