CS6210: Homework 5

Christopher Mertin

November 22, 2016

1. An $n \times n$ linear system of equations $A\mathbf{x} = \mathbf{b}$ is modified in the following manner: For each $i = \{1, \ldots, n\}$, the value b_i on the right-hand side of the i^{th} equation is replaced by $b_i - x_i^3$. Obviously, the modified system of equations (for the unknowns x_i) is now nonlinear.

   (a) Find the corresponding Jacobian Matrix

   Solution:

   After making the nonlinearity change, each row in the linear system has the following format:

   $$a_{i,1}x_1 + \cdots + a_{i,n}x_n = b_i - x_i^3$$

   Where we can add a vector of $\mathbf{x}^3$ (denoting the cubing of each element independently) to each side, giving

   $$a_{i,1}x_1 + \cdots + \left(a_{i,i}x_i + x_i^3\right) + \cdots + a_{i,n}x_n = b_i$$

   Giving the Jacobian of A as being

   $$J[A] = \begin{pmatrix} a_{1,1} + 3x_1^2 & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} + 3x_2^2 & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} + 3x_n^2 \end{pmatrix}$$

   (b) Given that A is strictly diagonally dominant with positive elements on its diagonal, state whether or not it is guarenteed that the Jacobian matrix at each iterate is nonsingular.

   Solution:

   We can look at the matrix J' to determine this, with $J = J' = A + \text{diag}\left(3x_i^2\right)$, with A being the coefficient matrix and the diagonal matrix being the nonlinear elements added to J along the diagonal.

As A is diagonally dominant, and since the off-diagonals of $\operatorname{diag}(3x_i^2)$ are zero, making it diagonally dominant as well, we can say that J' (and J) are also diagonally dominant since the sum of two diagonally dominant matrices are diagonally dominant themselves. There is also a theorem which guarentees that every diagonally dominant matrix is non-singular, meaning that J is guarenteed to be non-singular.

(c) Suppose that A is symmetric positive definition (not necessarily diagonally dominant) and that Newton's method is applied to solve the nonlinear system. Is it guarenteed to converge?

Solution:

The definition of a matrix to be symmetric positive definite, then with $\mathbf{z}$ being a positive vector, we should get

$$\begin{aligned}\mathbf{z}^T J\mathbf{z} &= \mathbf{z}^T \left(A + \operatorname{diag}\left(3x_i^2\right)\right) \\ &= \mathbf{z}^T A\mathbf{z} + \mathbf{z}^T \operatorname{diag}\left(3x_i^2\right)\mathbf{z}\end{aligned}$$

with $\mathbf{z}^T A\mathbf{z}$ being positive since A is symmetric positive definite. $\operatorname{diag}(3x_i^2)$ is always positive along the diagonal, so $\mathbf{z}^T\operatorname{diag}(3x_i^2)\mathbf{z}$ is also always positive, and symmetric (due to the off diagonals being zero). Therefore, J is symmetric positive definite and Newton's Method will converge.

2.

(a) Suppose Newton's method is applied to a linear system $A\mathbf{x} = \mathbf{b}$. How does the iterative formula look and how many iterations does it take to converge?

Solution:

The equation for Newton's Method is

$$J[\mathbf{x}]p_k = -f(\mathbf{x})$$

So we have to calculate the Jacobian of $f(x) = A\mathbf{x} - \mathbf{b}$, which is $J[\mathbf{x}] = A$, giving

$$Ap_k = -f(\mathbf{x})$$

with a given $\mathbf{x}_0$ of an initial guess, it gives

$$Ap_0 = -\left(A\mathbf{x}_0 - \mathbf{b}\right)$$

Where we can multiply by the interverse of A, giving

$$p_0 = -\mathbf{x}_0 + A^{-1}\mathbf{b}$$

where we can finally show that it will require only one iteration for convergence

$$\mathbf{x}_1 = \mathbf{x}_0 + p_0 = \mathbf{x}_0 + \left(-\mathbf{x}_0 + A^{-1}\mathbf{b}\right) = A^{-1}\mathbf{b}$$

(b) Suppose the Jacobian matrix is singular at the solution of a nonlinear system of equations. Speculate what can occur in terms of convergence and the rate of convergence. Specifically, is it possible to have a situation where the newton iteration converges but convergence is not quadratic?

Solution:

The text states that the Jacobian matrix should have a bounded inverse and also a continuous derivative to guarentee that Newton's Method can converge quadratically. However, the question states that the Jacobian is singular, so it won't converge quadratically.

3. Consider minimizing the function $\phi(\mathbf{x}) = \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T H\mathbf{x}$, where $\mathbf{c} = (5.04, -59.4, 146.4, -96.6)^T$ and

$$H = \begin{pmatrix} 0.16 & -1.2 & 2.4 & -1.4 \\ -1.2 & 12.0 & -27.0 & 16.8 \\ 2.4 & -27.0 & 64.8 & -42.0 \\ -1.4 & 16.8 & -42.0 & 28.0 \end{pmatrix}$$

Try both Newton and BFGS methods, starting from $\mathbf{x}_0 = (-1, 3, 3, 0)^T$. Explain why the BFGS method requires significantly more iterations than Newton's.

Solution:

- Broyden-Fletcher-Goldfarb-Shanno (BFGS)

  The necessary condition for minimization at $\widetilde{\mathbf{x}}$ is $\nabla\phi(\widetilde{\mathbf{x}}) = 0$, so in finding $\nabla\phi(\mathbf{x})$

$$\begin{aligned} \phi(\mathbf{x}) &= \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T H\mathbf{x} \\ \nabla\phi(\mathbf{x}) &= H\mathbf{x} + \mathbf{c} \end{aligned}$$

  By setting $\nabla\phi(\mathbf{x}) = 0$

$$H\mathbf{x} + \mathbf{c} = 0$$

$$\mathbf{x} = -H^{-1}\mathbf{c} = \begin{pmatrix} 2212.33 \\ 2119.56 \\ 1912.67 \\ 1711.33 \end{pmatrix}$$

- Newton

  For Newton's Method, the requirement is

$$\begin{aligned} \nabla^2\phi(\mathbf{x}_k)p_k &= -\nabla\phi(\mathbf{x}_k) \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + p_k \end{aligned}$$

So, in solving for $\nabla^2\phi$

$$\begin{aligned}\nabla\phi(\mathbf{x}) &= H\mathbf{x} + \mathbf{c}\\ \nabla^2\phi(\mathbf{x}) &= H\end{aligned}$$

And in solving for p_k

$$\begin{aligned}p_k &= -\left(\nabla^2\phi(\mathbf{x}_k)\right)^{-1}\nabla\phi(\mathbf{x}_k)\\ &= -H^{-1}\left(H\mathbf{x}_k + \mathbf{c}\right) = -\mathbf{x}_k - H^{-1}\mathbf{c}\end{aligned}$$

Giving us $\mathbf{x}_{k+1}$ as being

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k - \mathbf{x}_k - H^{-1}\mathbf{c}\\ &= -H^{-1}\mathbf{c}\end{aligned}$$

So the initial guess was independent of the convergence. Newton's Method is faster at converging than BFGS since it uses the second derivative. In the given question, the second derivative is independent of x, and would thus only need one iteration.

4. With the notation of Exercise 21, for $p = 2$ the problem can be solved as in Example 9.15 using SVD. But it can also be solved using the techniques of constrained optimization:

   (a) Explain why it is justified to replace the objective function by $\frac{1}{2}\left|\left|y\right|\right|_2^2$.

   Solution:

   Multiplying a function by a constant will not change the minimization problem, just the final value will be shifted by that multiplicative factor.

   (b) Form the KKT conditions of the resulting constrained optimization problem, obtaining a linear system of equations.

   Solution:

   The conditions for KKT in the constrained minimization problem are

$$\begin{aligned}1 - \nabla_y L\left(\widetilde{y}, \widetilde{\lambda}\right) &= 0\\ 2 - C_i\left(\widetilde{y}\right) &= 0\\ 3 - \widetilde{\lambda}_i C_i\left(\widetilde{y}\right) &= \nabla\phi(y)\end{aligned}$$

   for every Lagrangian Multiplier $\widetilde{\lambda_i}$.

   (c) Devise an example and solve it using the method developed as well as the one from Example 9.15. Compare and discuss.

   Solution:

5. Given the four data points $(-1,1), (0,1), (1,2), (2,0)$, determine the interpolating cubic polynomial

   - Using the Monomial basis
   - Using the Lagrange basis
   - Using the Newton basis

   Show that the three representations give the same polynomial.

   Solution:

   - MONOMIAL BASIS

     We should use a polynomial interpolant of degree 4, of the form

     $$p(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3$$

     Where we can start by putting the points in $p(x)$, giving

     $$\begin{aligned} 1 &= c_0 - c_1 - c_2 - c_3 \\ 1 &= c_0 \\ 2 &= c_0 + c_1 + c_2 + c_3 \\ 0 &= c_0 + 2c_1 + 4c_2 + 8c_3 \end{aligned}$$

     where solving this system of equations gives

     $$\begin{aligned} c_0 &= 1 \\ c_1 &= \frac{7}{6} \\ c_2 &= \frac{1}{2} \\ c_3 &= -\frac{2}{3} \end{aligned}$$

     where putting it back in $p(x)$ gives

     $$p(x) = 1 + \frac{7}{6}x + \frac{1}{2}x^2 - \frac{2}{3}x^3$$

   - LAGRANGE BASIS The Lagrange Interpolant is

     $$p(x) = \sum_{j=0}^{n} y_j L_j(x)$$

and the Lagrange Polynomials being

$$L_j(x) = \prod_{i=0}^{n} \frac{x - x_i}{x_j - x_i}$$

for $i \neq j$, so solving for the polynomials gives

$$\begin{aligned}
L_0(x) &= \frac{x(x+1)(x-2)}{(-1-0)(-1-1)(-1-2)} = -\frac{x(x+1)(x-2)}{6} \\
L_1(x) &= \frac{(x+1)(x-1)(x-2)}{1(-1)(-2)} = \frac{(x+1)(x-1)(x-2)}{2} \\
L_2(x) &= \frac{(x+1)x(x-2)}{(1+1)1(1-2)} = -\frac{(x+1)x(x-2)}{2} \\
L_3(x) &= \frac{(x+1)x(x-1)}{(2+1)2(2-1)} = \frac{(x+1)x(x-1)}{6}
\end{aligned}$$

and plugging back into $p(x)$ gives

$$\begin{aligned}
p(x) &= L_0(x) + L_1(x) + 2L_3(x) + 0 \\
&= 1 + \frac{7}{6}x + \frac{1}{2}x^2 - \frac{2}{3}x^3
\end{aligned}$$

- Newton Basis
  The Newton Interpolant is

$$p(x) = \sum_{j=0}^{3} c_j \prod_{i=0}^{j-1} (x - x_j)$$
$$p(x) = c_0 + c_1(x+1) + c_2(x+1)x + c_3(x+1)x(x-1)$$

Where we can input the points

$$\begin{aligned}
p(-1) &= c_0 = 1 \\
p(0) &= c_0 + c_1 = 1 \\
p(1) &= c_0 + 2c_1 + 2c_2 = 2 \\
p(2) &= c_0 + 3c_1 + 6c_2 + 6c_3 = 0
\end{aligned}$$

And after solving this system of equations gives

$$p(x) = 1 + \frac{7}{6}x + \frac{1}{2}x^2 - \frac{2}{3}x^3$$

6. Suppose we are given 137 uniformly spaced data pairs at distinct abscissae: $(x_i, y_i), i = \{0, 1, \ldots, 136\}$. These data are thought to represent a function which is piecewise

smooth; that is, the unknown function $f(x)$ which gave rise to these data values has many bounded derivatives everywhere expect for a few points where it jups discontinuously. (Imagine drawing a curve smoothly from left to right, mixed with lifting the pen and moving it vertically a few times.) For each sub interval $[x_{i-1}, x_i]$ we wanit to pass the hopefully best cubic $p_3(x)$ for an accurate interpolation of $f(x)$ at points $x_{i-1} < x < x_i$. This involves choosing good neighbors to interpolate at.

Propose an algorithm for this task. Justify.

Solution:

A cubic interpolation requires four points for each subinterval. Two of the points are trivial and can be used as the end points. The other two we can choose neighbor points of the subinterval endpoints.

---

if Sub Interval is (x_0, x_1) **then**
    Find a cubic interpolation by using $\{x_0, x_1, x_2, x_3\}$
else if Sub Interval is (x_i, x_{i+1}) **then**
    Find a cubic interpolation by using $\{x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$
else if Sub Interval is (x_{n-2}, x_{n-1}) **then**
    Find a cubic interpolation by using $\{x_{n-4}, x_{n-3}, x_{n-2}, x_{n-1}\}$
end if

Since we are using the exact values of (x_i, y_i), it should give an "accurate" interpolation as the question asked. If we wanted an approximation, we could instead of used midpoints such as $\left(\frac{x_i+x_{i+1}}{2}, \frac{y_i+y_{i+1}}{2}\right)$ and $\left(\frac{x_i+x_{i-1}}{2}, \frac{y_i+y_{i-1}}{2}\right)$ as the points, except for the endpoints.

7. A popular technique arising in methods for minimizing functions in several variables involves a *weak line search*, where an approximate minimum $\widetilde{x}$ is found for a function in one variable, $f(x)$, for whcih the values of $f(0)$, $f'(0)$, and $f(1)$ are given. The function $f(x)$ is defined for all nonnegative x, has a continuous second derivative, and satisfies $f(0) < f(1)$ and $f'(0) < 0$. We then interpolate the given values by a quadratic polynomial and set $\widetilde{x}$ as the minimum of the interpolant.

   (a) Find $\widetilde{x}$ for the values of $f(0) = 1,\ f'(0) = -1,\ f(1) = 2$

   Solution:

   The interpolant is

$$p(x) = c_0 + c_1 x + c_2 x^2$$

   Where we can utilize 3 equations to get the coefficients

$$\begin{aligned} p(0) &= c_0 = 1 \\ p(1) &= c_0 + c_1 x + c_2 x^2 = 1 + c_1 x + c_2 x^2 \\ p'(x) &= c_1 + 2c_2 x \\ &\Rightarrow p'(0) = c_1 = -1 \end{aligned}$$

Plugging in the coefficients gives

$$p(x) = 1 - x + 2x^2$$

Where finding the minimum gives

$$\begin{aligned} p'(x) &= -1 + 4x = 0 \\ x &= \frac{1}{4} \end{aligned}$$

(b) Show that the quadratic interpolant has a unique minimum satisfying $0 < \widetilde{x} < 1$. Can you show the same for the function f itself?

Solution:

Since the interpolant is of degree one, it has a single root. So the minimum has to be unique. Since the point $(1/4)$, is between 0 and 1, it is satisfied.

For f, we have the constraints of $f(0) < f(1)$ and $f'(0) < 0$. The second constraint states that the slope of the function at $x = 0$ is decreasing, and the first states that the first point is strictly less than the last, and must have a positive slope at some point in the interval such that it can reach $f(1)$ since it is continuous. This point where the slope switches is the minimum in the interval.

Rolle's Theorem can also be used to show that at some point c in $(0, 1)$ we have $f'(c) = 0$, the minimum of f.

8. Let $f \in C^3[a, b]$ be given at equidistant points $x_i = a + ih$, $i = \{0, 1, \ldots, n\}$, where $nh = b - a$. Assume further that $f'(a)$ is given as well.

(a) Construct an algorithm for C^1 piecewise quadratic interpolation of the given values. Thus, the interpolating function is written as

$$v(x) = s_i(s) = a_i + b_i\left(x - x_i\right) + c_i\left(x - x_i\right)^2, \quad x_i \le x \le x_{i+1}$$

for $i = \{0, \ldots, n-1\}$, and your job is to specify an algorithm for determining the $3n$ coefficients a_i, b_i, and c_i.

Solution:

With $3n$ coefficients, we need $3n$ equations. We have n equations from $x_i\left(x_i\right) = a_i + 0 + 0 = f\left(x_i\right)$, and another n from $s_i\left(x_{i+1}\right) = f\left(x_{i+1}\right)$. The third will come from being C^1 continuous, thus the s_i's need to be differentiable in the interval (x_i, x_{i+1}). At each point of x_i, $d/dx\left(s_i\right) = d/dx\left(s_{i+1}\right)$. This gives

$$\begin{aligned} \frac{d}{dx}s_{i-1}\left(x_{i-1}\right) &= \frac{d}{dx}s_i\left(x_{i-1}\right) \\ s_i(x) &= a_i + b_i(x - x_i) + c_i(x - x_i)^2 \\ s_{i-1}(x) &= a_{i-1} + b_{i-1}(x - x_{i-1}) + c_{i-1}(x - x_{i-1})^2 \end{aligned}$$

Now, in taking the derivative of the last two

$$s_i'(x) = b_i + 2c_i(x - x_i)$$
$$s_{i-1}'(x) = b_{i-1} + 2c_{i-1}(x - x_{i-1})$$

We can impose $x = x_{i-1}$, resulting in

$$s_i'(x) = b_i + 2c_i(x_{i-1} - x_i)$$
$$s_{i-1}'(x) = b_{i-1}$$

Finally

$$b_{i-1} = b_i + 2c_i(x_{i-1} - x_i)$$

(b) How accurate do you expect this approximation to be as a function of h? Justify.
Solution:
Page 336 of the text gives the boundary errors as being:

- CONSTANT

$$\frac{h}{2} \max |f'(\epsilon)|$$

- LINEAR

$$\frac{h^2}{2!2^2} \max |f''(\epsilon)|$$

- CUBIC

$$\frac{h^4}{4!2^4} \max \left|f^{(4)}(\epsilon)\right|$$

So the quadratic should be

$$\frac{h^3}{3!2^3} \max \left|f^{(3)}(\epsilon)\right|$$

This can be proven with the same method on page 334, expect we'd use

$$|f(x) - v(x)| \leq \frac{h^3}{3!2^3} \max \left|f^{(3)}(\epsilon)\right|$$

since we need to use three points instead of two.

9. Derive a B-spline basis representation for piecewise linear interplation and for piecewise Hermite cubic interpolation.

Solution:

- Piecewise Linear Interpolation

  Based on "Spline Methods Draft" by Tom Lyche and Knut Morken, page 38. Since piecewise linear interpolation, it just needs to be C^0 continuous, which it is. It is not required to be everywhere differentiable, so the sequence of nodes are

  $$(x_0, x_1, \ldots, x_n) = (t_0, t_0, t_1, t_2, \ldots, t_{n-1}, t_n, t_n)$$

  Which is a B-spline of order 1. From "Spline Methods Draft," polynomials of degree 1 are

  $$B_{j,1}(x) = \begin{cases} \frac{x-t_j}{t_{j+1}-t_j} & t_j \leq x < t_{j+1} \\ \frac{t_{j+2}-x}{t_{j+2}-t_{j+1}} & t_{j+1} \leq x < t_{j+2} \\ 0 & \text{Else} \end{cases}$$

  Finally, the linear spline is $\sum_{j=1}^{n} B_{j,1}(x)$, for which y_j is the original value at x_j.

- Piecewise Hermite Cubic Interpolation

  Since it is a cubic interpolation, we need four points, where we can use four points of x_1 and four points of x_n. The knot vector is

  $$(x_1, x_1, x_1, x_1, x_2, x_2, \ldots, x_{n-1}, x_{n-1}, x_n, x_n, x_n, x_n)$$

  Which is a B-spline of degree $\ell = 3$. Page 108 of "Spline Methods Draft" states that the B-spline basis representation for Piecewise Hermite Cubic Interpolation is

  $$\sum_{i=1}^{2n} c_i B_{i,3}$$

  for which

  $$\begin{aligned} c_{2i-1} &= f(x_i) - \frac{1}{3}\Delta x_{i-1} f'(x_i) \\ c_{2i} &= f(x_i) - \frac{1}{3}\Delta x_i f'(x_i) \\ \Delta x_j &= x_{j+1} - x_j \end{aligned}$$

10. Consider interpolating the data $(x_0, y_0), \ldots, (x_6, y_6)$ given by

x	0.1	0.15	0.2	0.3	0.35	0.5	0.75
y	3.0	2.0	1.2	2.1	2.0	2.5	2.5

Construct the five interpolants specified below (you may use available software for this), evaluate them at the points $\{0.05, 0.06, \ldots, 0.80\}$, plot, and comment on their respective properties:

(a) Polynomial Interpolant

Solution:

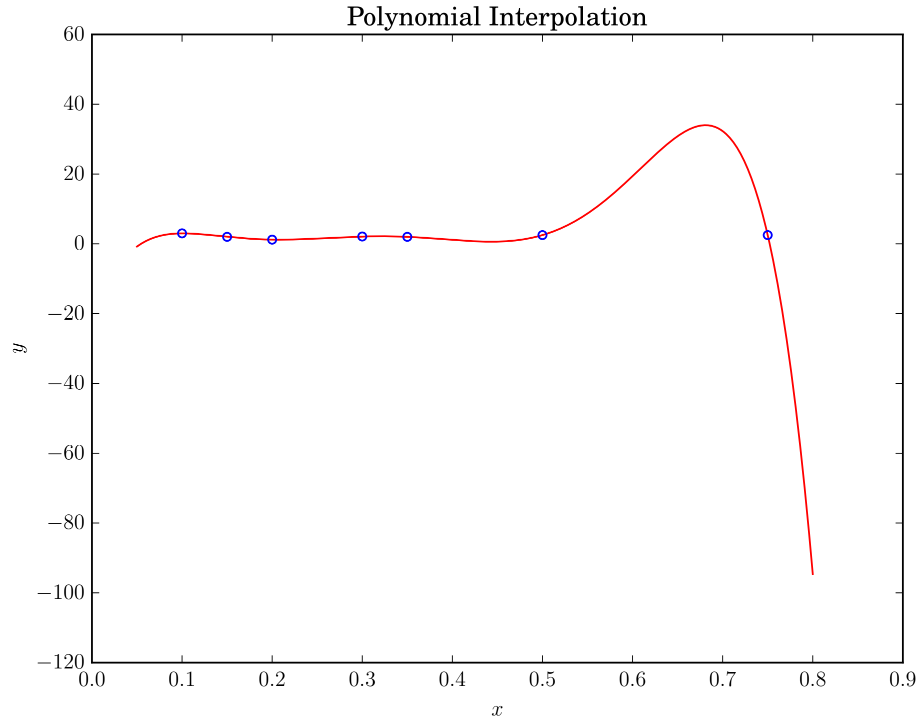

Figure 1: Polynomial Interpolation. Chose $n = 6$ since there were 7 points

(b) Cubic Spline Interpolant

Solution:

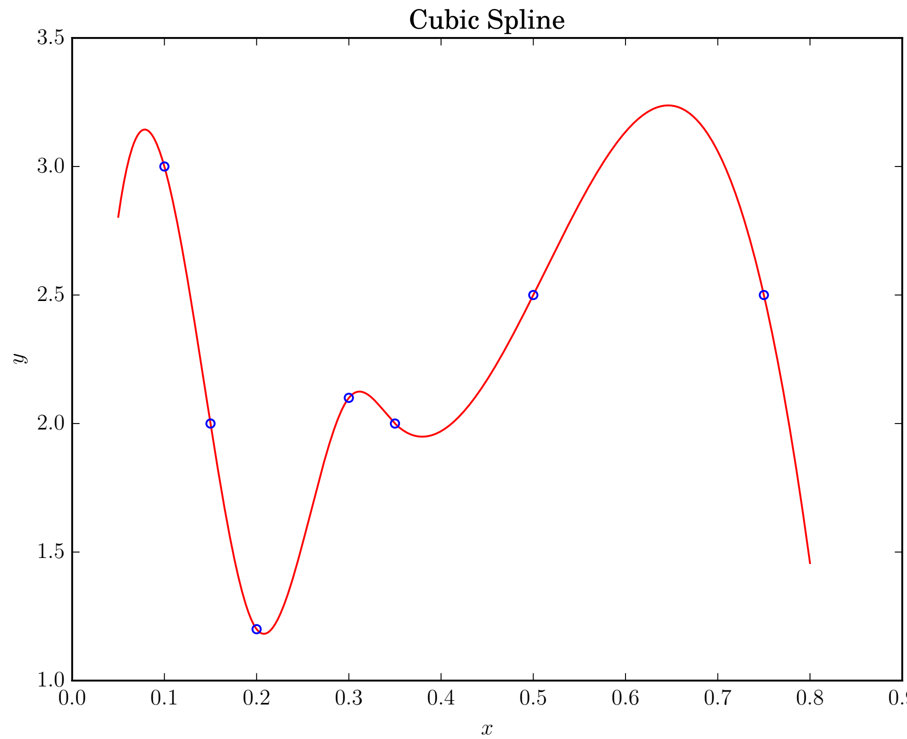

Figure 2: Cubic Spline Interpolation using `scipy.interpolate.UnivariateSpline`

(c) The Interpolant

$$v(x) = \sum_{j=0}^{n} c_j \phi_j(x)$$

where $n = 7$, $\phi_0(x) = 1$, and

$$\phi_j(x) = \sqrt{(x - x_{j-1})^2 + \epsilon^2} - \epsilon$$

In addition to the n interpolation requirements, the condition $c_0 = -\sum_{j=1}^{n} c_j$ is imposed. Construct this interpolant with

i. $\epsilon = 0.1$

ii. $\epsilon = 0.01$

iii. $\epsilon = 0.001$

Make as many observations as you can. What will happen if we let $\epsilon \to 0$?

Solution:

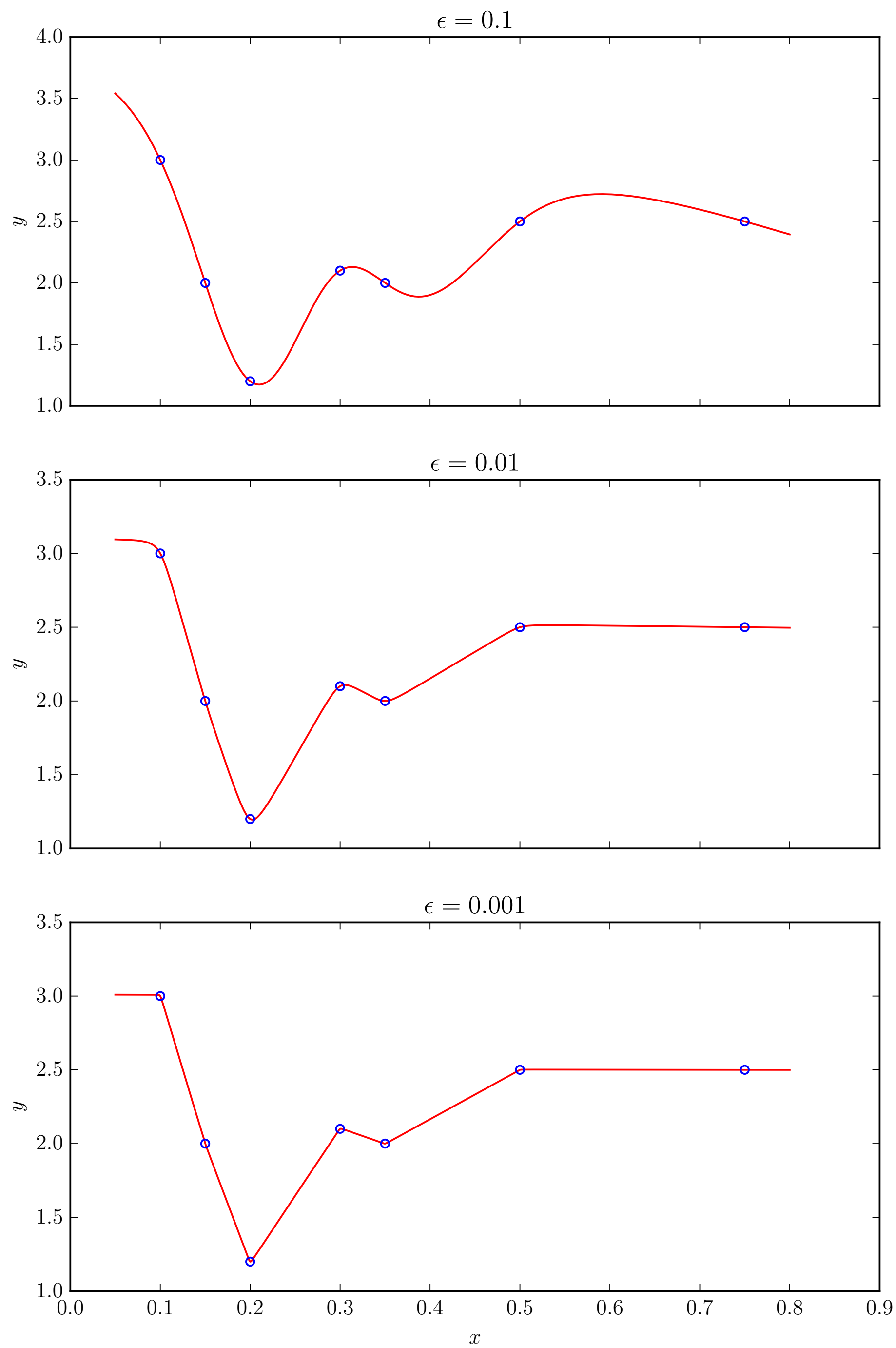

Figure 3: Interpolation based on the given function for different values of ϵ

As the value of ϵ decreased, the interpolation function became more linear. As you increase ϵ, the interpolation function smooths out.