Problem Set 5 - ECON 880

Spring 2022 - University of Kansas

Gunawan, Minh Cao

March 10, 2022

Question 1

We have given $f(x) = (x^{1/2} + 1)^{2/3}$, with a starting value $x_0 = 1$. The first derivative is given by

$$f'(x) = \frac{1}{3x^{1/2}(x^{1/2} + 1)^{1/3}},$$

while the second one is given by

$$f''(x) = \frac{-1}{18x(x^{1/2} + 1)^{4/3}} - \frac{1}{6x^{3/2}(x^{1/2} + 1)^{1/3}}$$

1(a) Taylor Series Approximation

The first-order Taylor series approximation is given by:

$$\begin{aligned} f(x) &= f(x_0) + f'(x_0)(x - x_0) \\ &= 1.5874 + 0.2646(x - 1) \end{aligned}$$

The second-order Taylor series approximation is given by:

$$\begin{aligned} f(x) &= f(x_0) + f'(x_0)(x - x_0) + \frac{(x - x_0)^2}{2} f''(x_0) \\ &= 1.5874 + 0.2646(x - 1) - 0.0772(x - 1)^2 \end{aligned}$$

1(b) Padé Approximation

We want to compute Padé Approximation (1,1)

$$P_{M=1}^{N=1}(x) = \frac{a_0 + a_1(x - x_0)}{b_0 + b_1(x - x_0)},$$

where $b_0 = 1$ (normalized). The Padé coefficients are normally best found from an $(M + N)$-th order Taylor series expansion of $f(x)$:

$$\begin{aligned} T_2(x) &= \frac{a_0 + a_1(x - 1)}{1 + b_1(x - 1)} \\ 1.5874 + 0.2646(x - 1) - 0.0772(x - 1)^2 &= \frac{a_0 + a_1(x - 1)}{1 + b_1(x - 1)} \end{aligned}$$

Multiplying up the denominator of the RHS with the LHS gives the following equivalent set of coefficient relations:

$$\begin{aligned} a_0 &= 1.5874 && (1)\\ a_1 &= 1.5874b_1 + 0.2646 && (2)\\ 0 &= 0.2646b_1 - 0.0772 && (3) \end{aligned}$$

From equation 3, we obtain $b_1 = 0.2917$. Then, we plug it in into equation 2 to obtain $a_1 = 0.7276$. Finally, the Padé Approximation (1,1) for $f(x)$ is given by

$$P^{N=1}_{M=1}(x) = \frac{1.5874 + 0.7276(x-1)}{1 + 0.2917(x-1)},$$

Question 2

Consider the function

$$f(x) = e^{4x-2}$$

over $[0, 2]$ interval, and construct the following approximations to it:

(a) Chebyshev polynomials of degree 4; choose 5 points to use as nodes.

(b) Cubic spline over 5 equally spaced points in $[0, 2]$.

Then evaluate the approximations over 101 equally spaced points in $[0, 2]$ and plot them along with the true function.

2(a) Chebyshev Interpolation

We use the following algorithm† to construct the Chebyshev interpolation of order 4. The code is saved as `chebyshev.m`. The result is shown on Figure 1.

Step 1. Compute five Chebyshev interpolation nodes on $[0, 2]$ using the following formula

$$z_k = -\cos\left(\frac{2k-1}{2m}\pi\right), \quad k = 1, \ldots, m$$

Step 2. Adjust the nodes to the $[0, 2]$ interval:

$$x_k = (z_k + 1)\left(\frac{2-0}{2}\right) + 0 = z_k + 1, \quad k = 1, \ldots, m$$

Step 3. Evaluate f at the approximation nodes

$$y_k = f(x_k), \quad k = 1, \ldots, m$$

Step 4. Compute Chebyshev coefficients $\{a_i\}_{i=0}^n$

$$a_i = \frac{\sum_{k=1}^m y_k T_i(z_k)}{\sum_{k=1}^m T_i(z_k)^2},$$

where $T_i(x) = 2xT_{i-1}(x) - T_{i-2}(x)$, with $T_0(x) = 1$ and $T_1(x) = x$, for $i = 2, \ldots, n$. We coded this recursion relation for $T_i(x)$ separately in `Tn.m`.

† Kenneth L. Judd, 1998. "Numerical Methods in Economics," MIT Press Books, The MIT Press, p.223

Step 5. The approximation for $f(x)$, $x \in [0,2]$ is given by:

$$\hat{f}(x) = \sum_{i=0}^{n} a_i T_i\,(x-1)\,.$$

2(a) Cubic Spline Interpolation

We use the following algorithm[†] to construct the cubic spline interpolation. The result is shown on Figure 1.

Step 1. Suppose we have the dataset $\{(x_i, y_i) | i = 1, \ldots, n\}$. We start the index i from 1 (instead of 0) in order to synchronize with our coding in Matlab. First, do some precalculations:

- $h_i = x_{i+1} - x_i$, for $i = 1, \ldots, n-1$
- $v_i = 2(h_{i-1} + h_i)$, for $i = 2, \ldots, n-1$
- $f_i = 6\left(\frac{y_{i+1}-y_i}{h_{i-1}} - \frac{y_i - y_{i-1}}{h_{i-1}}\right)$, for $i = 2, \ldots, n-1$
- $s_1 = s_n = 0$. This represents boundary conditions for natural cubic spline.

Step 2. Solve the triagonal system $H \cdot \vec{s} = \vec{f}$, where

$$H = \begin{bmatrix} 1 & 0 & 0 & & \cdots & \cdots & \cdots & 0 \\ 0 & v_1 & h_1 & & & & & \vdots \\ 0 & h_1 & v_2 & h_2 & & & & \vdots \\ \vdots & & h_2 & v_3 & h_3 & & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & & \ddots & \ddots & h_{n-1} & 0 \\ \vdots & & & & & h_{n-1} & v_{n-1} & 0 \\ 0 & 0 & 0 & & \cdots & \cdots & 0 & 1 \end{bmatrix}, \quad \vec{s} = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ \vdots \\ s_{n-1} \\ s_n \end{pmatrix}, \quad \vec{f} = \begin{pmatrix} 0 \\ f_2 \\ f_3 \\ \vdots \\ \vdots \\ f_{n-1} \\ 0 \end{pmatrix}$$

Note that the matrix H is a tridiagonal symmetric matrix. It is diagonally dominant, because

$$|v_i| > |h_i| + |h_{i-1}|.$$

Therefore, the vector $\vec{s}$ can be determined as the unique solution of the system.

Step 3. For $i = 1, \ldots, n-1$, compute

$$\begin{aligned} a_i &= (s_{i+1} - s_i)/6h_i \\ b_i &= s_i/2 \\ c_i &= (y_{i+1} - y_i)/h_i - (2h_i s_i + h_i s_{i+1})/6 \\ d_i &= y_i \end{aligned}$$

Step 1-3 is performed by calling `splinecoefs.m`.

[†]Wen Shen, 2016. "An Introduction to Numerical Computation," World Scientific Publishing, p.52-55

Step 4. Calculate the cubic spline approximation function

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i,$$

for $i = 1, \ldots, n - 1$. This step is performed by calling `evspline.m`.

Comment: Cubic spline works better because the approximated curve is relatively much closer to the true function. Higher order Chebyshev coefficients seem to be smaller because the curve looks more like a second order polynomial curve. Both approximation methods have a matching slope and curvature with the true function only for the interval $x > 1$. For $x < 1$, cubic spline has more accurate curvature approximation to the true function.

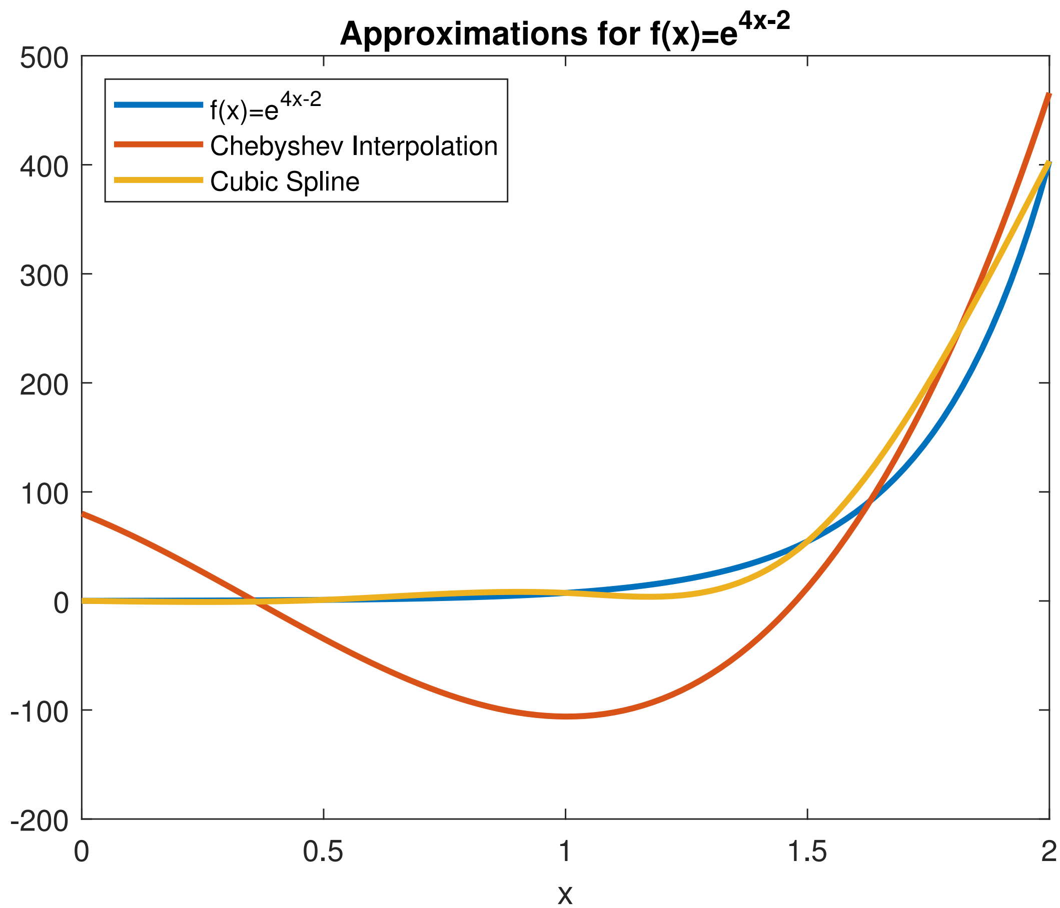

Figure 1: Approximations for $f(x) = e^{4x-2}$