MATH/CS 5466 · NUMERICAL ANALYSIS

Problem Set 1 · Solutions

Posted Friday 29 January 2016. Due Monday 8 February 2016 (5pm).
Students should complete any 5 problems (total of 100 points).
Students are welcome to attempt more problems if they wish, but they will not count for extra points.

1. [20 points]

   This problem addresses the $\xi = \xi(x)$ term that appears in the formula

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}\prod_{j=0}^{n}(x - x_j)$$

   given in Theorem 1.3 in the course notes, and Section 2.2.2 of Gautschi's book.

   (a) Write down the linear interpolant $p_1(x)$ for the function $f(x) = x^3$ at the interpolation points $x_0 = 0$ and $x_1 = b$. Show that $\xi(x)$ takes the unique value $\xi(x) = (x+b)/3$.

   (b) Write down the linear interpolant $p_1(x)$ for the function $f(x) = 1/x$ at the interpolation points $x_0 = 1$ and $x_1 = 2$. Explicitly write down the function $\xi(x)$ for this case, and find the extreme values $\min_{1\le x\le 2}\xi(x)$ and $\max_{1\le x\le 2}\xi(x)$.

   [Süli and Mayers, Gautschi]

Solution.

(a) The linear interpolant p_1 to $f(x) = x^3$ at $x = 0$ and $x = b$ is the line that passes through $(0, 0)$ and (b, b^3), i.e.,

$$p_1(x) = b^2 x.$$

The interpolation error formula with $n = 1$, $f''(x) = 6x$, $x_0 = 0$ and $x_1 = b$ reduces to

$$f(x) - p_1(x) = \frac{f''(\xi(x))}{2}(x - x_0)(x - x_1) = \frac{6\xi(x)}{2}x(x - b).$$

On the other hand, from the formulas $f(x) = x^3$ and $p_1(x) = b^2x$, we can directly compute

$$f(x) - p_1(x) = x^3 - b^2x = x(x^2 - b^2) = x(x - b)(x + b).$$

Equate these two expressions for $f(x) - p_1(x)$, cancel the common factor $x(x - b)$, and simplify to get

$$\xi(x) = \frac{x + b}{3}.$$

You are not asked to verify this, but notice that $(x+b)/3 \in [b/3, 2b/3] \subset [0, b]$, i.e., $\xi(x) \in [x_0, x_1]$.

(b) The linear interpolant to $f(x) = 1/x$ at $x_0 = 1$ and $x_1 = 1$ is the line through $(1, 1)$ and $(2, 1/2)$, i.e.,

$$p_1(x) = \frac{3 - x}{2}.$$

The interpolation error formula with $n = 1$, $f''(x) = -1/x^3$, $x_0 = 1$ and $x_1 = 2$ reduces to

$$f(x) - p_1(x) = \frac{f''(\xi(x))}{2}(x - x_0)(x - x_1) = \frac{(x-1)(x-2)}{\xi(x)^3}.$$

On the other hand, from the formulas $f(x) = 1/x$ and $p_1(x) = (3-x)/2$ we can directly compute

$$f(x) - p_1(x) = \frac{1}{x} - \frac{3-x}{2} = \frac{1}{2x}(x-2)(x-1).$$

Equate these two expressions for $f(x) - p_1(x)$, cancel the common factor $(x-2)(x-1)$, and simplify to get

$$\xi(x) = (2x)^{1/3}.$$

This function takes the extrema

$$\min_{1\le x\le 2} \xi(x) = 2^{1/3} = 1.259921\ldots, \qquad \max_{1\le x\le 2} \xi(x) = 4^{1/3} = 1.587401\ldots,$$

which confirms that requirement that $\xi(x) \in [1, 2]$.

---

2. [20 points]

Recall that for $\mathbf{A} \in \mathbb{C}^{n\times n}$, the linear system $\mathbf{Ac} = \mathbf{f}$ has a unique solution for any $\mathbf{f}$ provided $\mathrm{Ker}(\mathbf{A}) = \{\mathbf{0}\}$, where $\mathrm{Ker}(\mathbf{A})$ denotes the kernel (null space) of $\mathbf{A}$.

If the kernel of $\mathbf{A}$ is larger, i.e., if there is a nonzero vector $\mathbf{z} \in \mathrm{Ker}(\mathbf{A})$, then there are two possibilities:

- If $\mathbf{f} \notin \mathrm{Ran}(\mathbf{A})$, then there is *no solution* $\mathbf{c}$ to the linear system $\mathbf{Ac} = \mathbf{f}$.
- If $\mathbf{f} \in \mathrm{Ran}(\mathbf{A})$, then there are *infinitely many solutions* to the linear system $\mathbf{Ac} = \mathbf{f}$. In particular, if $\widehat{\mathbf{c}}$ satisfies $\mathbf{A}\widehat{\mathbf{c}} = \mathbf{f}$, then any $\mathbf{c}$ of the form $\mathbf{c} = \widehat{\mathbf{c}} + \gamma\mathbf{z}$ is also a solution, where γ is an arbitrary constant.

With these facts in mind, please answer the following questions.

(a) Suppose we wish to construct a polynomial $p_5 \in \mathcal{P}_5$ that interpolates a function $f \in C^2[-1,1]$ in the following (somewhat unusual) manner: $p_5(-1) = f(-1)$; $p_5'(-1) = f'(-1)$; $p_5(0) = f(0)$; $p_5''(0) = f''(0)$; $p_5(1) = f(1)$; $p_5'(1) = f'(1)$. Write down a linear system to determine the coefficients $c_0, \ldots, c_5$ for p in the monomial basis: $p_5(x) = c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4 + c_5x^5$.

(b) What is the kernel of the matrix $\mathbf{A}$ constructed in part (a)?
(You may use the MATLAB command `null(A,'r')`.)

What does your answer imply about the existence and uniqueness of the interpolant p_5?

(c) Consider the data: $f(-1) = -1$, $f'(-1) = 0$, $f(0) = 1$, $f''(0) = -2$, $f(1) = 3$, $f'(1) = 4$. Show that there are infinitely many choices for the polynomial p_5 that interpolate this data. Plot six of them. (Superimpose all on the same plot.)

Solution.

(a) We seek the coefficients $c_0, \ldots, c_5$ to the polynomial

$$p_5(x) = c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4 + c_5x^5,$$

which will determined by the five constraints

$$\begin{aligned}
c_0 + c_1x_0 + c_2x_0^2 + c_3x_0^3 + c_4x_0^4 + c_5x_0^5 &= f(x_0)\\
c_1 + 2c_2x_0 + 3c_3x_0^2 + 4c_4x_0^3 + 5c_5x_0^4 &= f'(x_0)\\
c_0 + c_1x_1 + c_2x_1^2 + c_3x_1^3 + c_4x_1^4 + c_5x_1^5 &= f(x_1)\\
2c_2 + 6c_3x_0 + 12c_4x_0^2 + 20c_5x_0^3 &= f''(x_1)\\
c_0 + c_1x_2 + c_2x_2^2 + c_3x_2^3 + c_4x_2^4 + c_5x_2^5 &= f(x_2)\\
c_1 + 2c_2x_2 + 3c_3x_2^2 + 4c_4x_2^3 + 5c_5x_2^4 &= f'(x_2).
\end{aligned}$$

These equations can be written in the matrix form

$$\begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 & x_0^4 & x_0^5\\ 0 & 1 & 2x_0 & 3x_0^2 & 4x_0^3 & 5x_0^4\\ 1 & x_1 & x_1^2 & x_1^3 & x_1^4 & x_1^5\\ 0 & 0 & 2 & 6x_1 & 12x_1^2 & 20x_1^3\\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & x_2^5\\ 0 & 1 & 2x_2 & 3x_2^2 & 4x_2^3 & 5x_2^4 \end{bmatrix} \begin{bmatrix} c_0\\ c_1\\ c_2\\ c_3\\ c_4\\ c_5 \end{bmatrix} = \begin{bmatrix} f(x_0)\\ f'(x_0)\\ f(x_1)\\ f''(x_1)\\ f(x_2)\\ f'(x_2) \end{bmatrix}$$

With our values for the nodes $x_0 = -1$, $x_1 = 0$, $x_2 = 1$, we have

$$\begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 \\ 0 & 1 & -2 & 3 & -4 & 5 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f'(x_0) \\ f(x_1) \\ f''(x_1) \\ f(x_2) \\ f'(x_2) \end{bmatrix}.$$

(b) Using the `null` command or otherwise, one finds that the kernel is the span of the vector

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ -2 \\ 0 \\ 1 \end{bmatrix}.$$

Since the kernel is non-trivial, there are two possible situations: there will either be *no polynomial*, or there will be *infinitely many polynomials* that satisfy the six interpolation conditions. Which of the two depends on the actual interpolation conditions.

(c) Note that the vector

$$\mathbf{f} = \begin{bmatrix} f(x_0) \\ f'(x_0) \\ f(x_1) \\ f''(x_1) \\ f(x_2) \\ f'(x_2) \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \\ -2 \\ 3 \\ 4 \end{bmatrix}$$

is in the range of the coefficient matrix, since we we can write

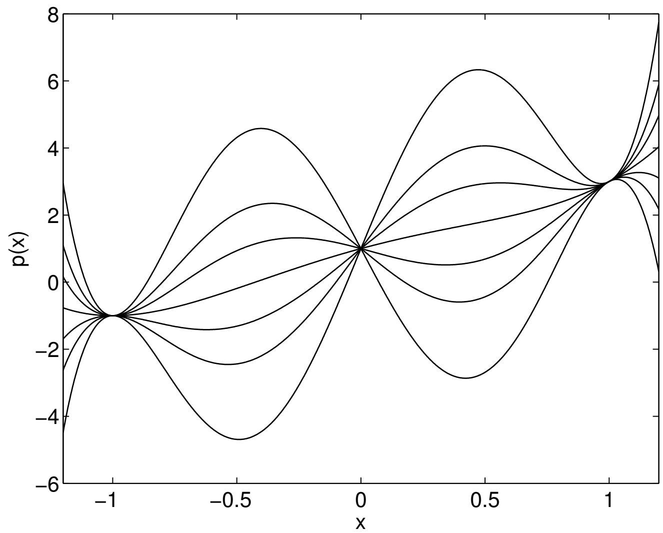
$$\begin{bmatrix} -1 \\ 0 \\ 1 \\ -2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 \\ 0 & 1 & -2 & 3 & -4 & 5 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ -1 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

Since $\mathbf{f}$ is in $\mathrm{Ran}(A)$, there are infinitely many choices for the coefficients $c_0, \ldots, c_5$ that will satisfy the six constraints. All solutions have the form

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ -1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \gamma \begin{bmatrix} 0 \\ 1 \\ 0 \\ -2 \\ 0 \\ 1 \end{bmatrix}$$

for arbitrary γ.

The plot below shows polynomials for the above coefficients chosen with $\gamma = -16, -8, -4, 0, 4, 8, 16$.



3. [20 points]

The *Hermite interpolant* $h_n \in \mathcal{P}_{2n+1}$ of $f \in C^1[a,b]$ at the points $\{x_j\}_{j=0}^n$ can be written in the form

$$h_n(x) = \sum_{j=0}^{n} \Big(A_j(x) f(x_j) + B_j(x) f'(x_j) \Big),$$

where the functions A_j and B_j generalize the Lagrange basis functions:

$$\begin{aligned} A_j(x) &= \big(1 - 2\,\ell_j'(x_j)(x - x_j)\big)\,\ell_j^2(x) \\ B_j(x) &= (x - x_j)\,\ell_j^2(x), \end{aligned}$$

with $\ell_j(x) = \prod_{k=0,k\neq j}^{n} (x - x_k)/(x_j - x_k)$.

(a) Verify that

$$A_j(x_k) = \begin{cases} 1 & j = k \\ 0 & j \neq k, \end{cases} \qquad A_j'(x_k) = 0, \qquad B_j(x_k) = 0, \qquad B_j'(x_k) = \begin{cases} 1 & j = k \\ 0 & j \neq k. \end{cases}$$

(b) The above expression for the Hermite interpolating polynomial mimics the *Lagrange form* of the the standard interpolating polynomial. Devise a scheme for constructing Hermite interpolants that generalizes the *Newton form*. What are your new Newton-like basis functions for $\mathcal{P}_{2n+1}$?

Solution.

(a) First consider $A_j(x_k)$. If $k \neq j$, then

$$A_j(x_k) = \big(1 - 2\ell_j'(x_j)(x_k - x_j)\big)\ell_j^2(x_k) = 0$$

since $\ell_j(x_k) = 0$ by construction. For $k = j$,

$$A_j(x_j) = \big(1 - 2\ell_j'(x_j)(x_j - x_j)\big)\ell_j^2(x_j) = \ell_j^2(x_j) = 1,$$

since $\ell_j(x_j) = 1$ by construction.

Next consider $A_j'(x_k)$. To begin with,

$$A_j'(x) = -2\ell_j'(x_j)\ell_j^2(x) + 2\big(1 - 2\ell_j'(x_j)(x - x_j)\big)\ell_j(x)\ell_j'(x).$$

For $k \neq j$, we have

$$A_j'(x_k) = -2\ell_j'(x_j)\ell_j^2(x_k) + 2\big(1 - 2\ell_j'(x_j)(x_k - x_j)\big)\ell_j(x_k)\ell_j'(x_k) = 0,$$

since both terms in this sum have $\ell_j(x_k)$ terms. For $j = k$,

$$\begin{aligned} A_j'(x_j) &= -2\ell_j'(x_j)\ell_j^2(x_j) + 2\big(1 - 2\ell_j'(x_j)(x_j - x_j)\big)\ell_j(x_j)\ell_j'(x_j) \\ &= -2\ell_j'(x_j)\ell_j^2(x_j) + 2\ell_j'(x_j) \\ &= 0, \end{aligned}$$

since $\ell_j(x_j) = 1$. Thus A_j and A_j' both perform as required.

It is simple to see that $B_j(x_k) = 0$ since $\ell_j(x_k) = 0$ if $k \neq j$, and $(x_k - x_j) = 0$ if $k = j$. Note that

$$B_j'(x) = \ell_j^2(x) + 2(x - x_j)\ell_j(x)\ell_j'(x).$$

If $k \neq j$, then $\ell_j(x_k) = 0$ and so $B_j'(x_k) = 0$. If $k = j$,

$$B_j'(x_j) = \ell_j^2(x_j) + 2(x_j - x_j)\ell_j(x_j)\ell_j'(x_j) = 1,$$

since $\ell_j(x_j) = 1$.

(b) The principle behind the Newton basis for standard polynomial interpolation is: find some constant function that interpolates at x_0. Thus use this to find a linear function that interpolates at x_0 and x_1, and so on. The Newton basis functions are $\{1, x-x_0, (x-x_0)(x-x_1), \ldots, \prod_{j=0}^{n-1}(x-x_j)\}$. For Hermite interpolation, we will attempt to follow the same methodology. First, find $p_0 \in \mathcal{P}_0$ such that $p_0(x_0) = f(x_0)$:

$$p_0(x) = c_0 = f(x_0).$$

Thus, $p_0 = c_0 q_0(x)$, where the basis function $q_0(x) \equiv 1$.

Next, find a linear polynomial that interpolates both f and f' at x_0: i.e., find $p_1 \in \mathcal{P}_1$ such that $p_1(x_0) = f(x_0)$ *and* $p_1'(x_0) = f'(x_0)$. In keeping with the Newtonian spirit, write p_1 in the form

$$p_1(x) = p_0(x) + c_1 q_1(x)$$

for some $q_1 \in \mathcal{P}_1$. Our challenge is to find c_1 and q_1 to satisfy the interpolation conditions. Since $p_0(x_0) = f(x_0)$, the interpolation condition $p_1(x_0) = f(x_0)$ implies $q_1(x_0) = 0$. Therefore, we conclude that q_1 has a root at x_0; this completely determines q_1, since it is a linear polynomial and we are not concerned about scaling factors (which are absorbed by c_1):

$$q_1(x) = x - x_0.$$

Now determine c_1 so that $p_1'(x_0) = c_0 q_0'(x_0) + c_1 q_1'(x_0) = f'(x_0)$:

$$c_1 = \frac{f'(x_0) - c_0 q_0'(x_0)}{q_1'(x_0)} = f'(x_0).$$

So far, this basis is the same as the usual Newton basis. The next step introduces the critical difference. We want to construct $p_2 \in \mathcal{P}_2$ of the form

$$p_2(x) = p_1(x) + c_2 q_2(x)$$

for some $c_2 \in \mathbb{R}$ and $q_2 \in \mathcal{P}_2$ such that

$$p_2(x_0) = f(x_0), \quad p_2'(x_0) = f'(x_0), \quad p_2(x_1) = f(x_1).$$

The first two conditions are satisfied by p_1 itself, so we conclude that

$$q_2(x_0) = q_2'(x_0) = 0.$$

Thus $q_2 \in \mathcal{P}_2$ has a root at x_0, and so does it's first derivative. Hence

$$q_2(x) = (x - x_0)^2,$$

and the interpolation condition $p_2(x_1) = f(x_1)$ determines c_2:

$$c_2 = \frac{f(x_1) - c_0 q_0(x_1) - c_1 q_1(x_1)}{q_2(x_1)}.$$

A pattern is emerging, and the general procedure should be clear from one more step. Now we seek $p_3 \in \mathcal{P}_3$ of the form

$$p_3(x) = p_2(x) + c_3 q_3(x),$$

for some $c_3 \in \mathbb{R}$ and $q_3 \in \mathcal{P}_3$ such that

$$p_3(x_0) = f(x_0), \quad p_3'(x_0) = f'(x_0), \quad p_3(x_1) = f(x_1), \quad p_3'(x_1) = f'(x_1).$$

The first three conditions demand that $q_3(x_0) = q_3'(x_0) = q_3(x_1) = 0$. Hence q_3 has a double root at x_0 and a single root at x_1:

$$q_3(x) = (x - x_0)^2 (x - x_1).$$

Similarly, $q_4(x) = (x - x_0)^2 (x - x_1)^2$ and, in general,

$$q_j(x) = \begin{cases} 1, & j = 0; \\ \displaystyle\prod_{k=0}^{j/2-1} (x - x_k)^2, & j > 0 \text{ even}; \\ (x - x_{\lfloor j/2 \rfloor}) \displaystyle\prod_{k=0}^{\lfloor j/2 \rfloor - 1} (x - x_k)^2, & j \text{ odd}. \end{cases}$$

The expansion coefficients are thus available as the solution of the *triangular* linear system

$$\begin{bmatrix} q_0(x_0) & & & & & \\ q_0'(x_0) & q_1'(x_0) & & & & \\ q_0(x_1) & q_1(x_1) & q_2(x_1) & & & \\ q_0'(x_1) & q_1'(x_1) & q_2'(x_1) & q_3'(x_1) & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ q_0'(x_n) & q_1'(x_n) & q_2'(x_n) & q_3'(x_n) & \cdots & q_{2n+1}'(x_n) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{2n+1} \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f'(x_0) \\ f(x_1) \\ f'(x_1) \\ \vdots \\ f'(x_n) \end{bmatrix},$$

where derivatives are applied in the even rows.

---

4. [20 points]

The one-dimensional interpolation scheme studied in class can be adapted to higher dimensions. For example, suppose we are given a scalar-valued function $f(x, y)$, such as

$$f(x, y) = \mathrm{e}^x \sin y,$$

and wish to construct a function of the form

$$p(x, y) = c_0 + c_1 x + c_2 y + c_3 xy + c_4 x^2 + c_5 y^2$$

that interpolates $f(x, y)$ at (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5).

(a) Set up a linear system $\mathbf{Ac} = \mathbf{f}$ to determine the coefficients $c_0, \dots, c_5$.

(b) Write a MATLAB code to determine $\mathbf{c}$ when $f(x, y) = \mathrm{e}^x \sin y$ and the (x_j, y_j) pairs take the values listed in the following table.

j	0	1	2	3	4	5
x_j	0	0	1	1	2	2
y_j	0	2	0	2	1	3

Report your value for $\mathbf{c}$.

(c) Plot your model function $p(x, y)$ over $x \in [-1, 3]$, $y \in [-1, 3]$ using MATLAB's `surf` command. Compare this plot to the similar plot for $f(x, y)$, which can be obtained in the following manner.

```
f = inline('exp(x).*sin(y)','x','y');
[xx,yy] = meshgrid(linspace(-1,3,25),linspace(-1,3,25));
zz = f(xx,yy);
figure(1), clf
surf(xx,yy,zz)
```

Please submit plots of both $p(x, y)$ and $f(x, y)$.

Solution.

(a) The six interpolation conditions are

$$\begin{aligned}
c_0 + c_1 x_0 + c_2 y_0 + c_3 x_0 y_0 + c_4 x_0^2 + c_5 y_0^2 &= \mathrm{e}^{x_0} \sin y_0 \\
c_0 + c_1 x_1 + c_2 y_1 + c_3 x_1 y_1 + c_4 x_1^2 + c_5 y_1^2 &= \mathrm{e}^{x_1} \sin y_1 \\
c_0 + c_1 x_2 + c_2 y_2 + c_3 x_2 y_2 + c_4 x_2^2 + c_5 y_2^2 &= \mathrm{e}^{x_2} \sin y_2 \\
c_0 + c_1 x_3 + c_2 y_3 + c_3 x_3 y_3 + c_4 x_3^2 + c_5 y_3^2 &= \mathrm{e}^{x_3} \sin y_3 \\
c_0 + c_1 x_4 + c_2 y_4 + c_3 x_4 y_4 + c_4 x_4^2 + c_5 y_4^2 &= \mathrm{e}^{x_4} \sin y_4 \\
c_0 + c_1 x_5 + c_2 y_5 + c_3 x_5 y_5 + c_4 x_5^2 + c_5 y_5^2 &= \mathrm{e}^{x_5} \sin y_5.
\end{aligned}$$

These equations can be arranged into the matrix equation $\mathbf{Ac} = \mathbf{f}$:

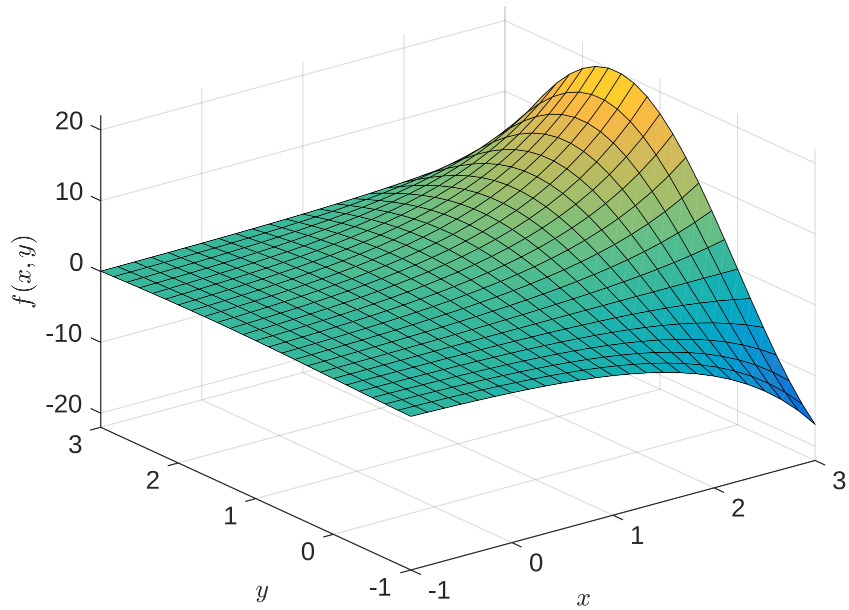
$$\begin{bmatrix} 1 & x_0 & y_0 & x_0 y_0 & x_0^2 & y_0^2 \\ 1 & x_1 & y_1 & x_1 y_1 & x_1^2 & y_1^2 \\ 1 & x_2 & y_2 & x_2 y_2 & x_2^2 & y_2^2 \\ 1 & x_3 & y_3 & x_3 y_3 & x_3^2 & y_3^2 \\ 1 & x_4 & y_4 & x_4 y_4 & x_4^2 & y_4^2 \\ 1 & x_5 & y_5 & x_5 y_5 & x_5^2 & y_5^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} \mathrm{e}^{x_0} \sin y_0 \\ \mathrm{e}^{x_1} \sin y_1 \\ \mathrm{e}^{x_2} \sin y_2 \\ \mathrm{e}^{x_3} \sin y_3 \\ \mathrm{e}^{x_4} \sin y_4 \\ \mathrm{e}^{x_5} \sin y_5 \end{bmatrix}.$$

(b),(c) The following MATLAB code solves the system and produces the desired plots.

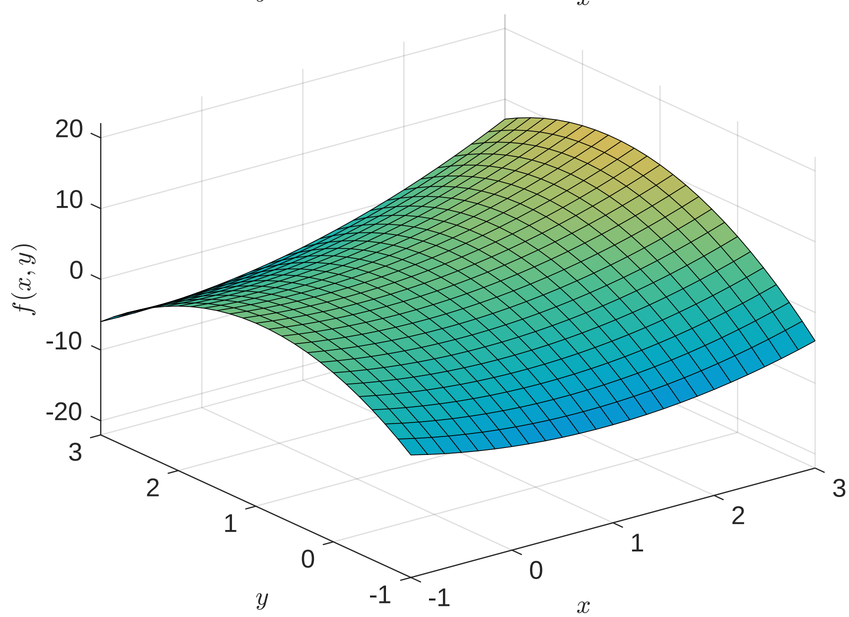
```
 x = [0;0;1;1;2;2];             % x interpolation points
 y = [0;2;0;2;1;3];             % y interpolation points

% set up coefficient matrix
 A = [ones(6,1) x y x.*y x.*x y.*y];

% set up a grid in two dimensions and evaluate the function there
 f = inline('exp(x).*sin(y)','x','y');
 [xx,yy] = meshgrid(linspace(-1,3,25),linspace(-1,3,25));
 zz = f(xx,yy);

% plot the function f on the grid
 figure(1), clf
 surf(xx,yy,zz)
 zlim([-22 22])                 % set z axis limits
 caxis(zlim)                    % color limits
 set(gca,'fontsize',16)
 xlabel('$x$','interpreter','latex','fontsize',16)
 ylabel('$y$','interpreter','latex','fontsize',16)
 zlabel('$f(x,y)$','interpreter','latex','fontsize',16)

% find the coefficients c and plot the polynomial
 figure(2), clf
 c = A\f(x,y);
 surf(xx,yy,c(1)+c(2)*xx+c(3)*yy+c(4)*xx.*yy+c(5)*xx.^2+c(6)*yy.^2)
 zlim([-22 22])                 % same z axis limits as first plot
 caxis(zlim)                    % same color limits as first plot
 set(gca,'fontsize',16)
 xlabel('$x$','interpreter','latex','fontsize',16)
 ylabel('$y$','interpreter','latex','fontsize',16)
 zlabel('$f(x,y)$','interpreter','latex','fontsize',16)

% output coefficients
 format long
 fprintf('coefficients, c:\n')
 disp(c)

% check error at interpolation points
  pxy = c(1)+c(2)*x+c(3)*y+c(4)*x.*y+c(5)*x.^2+c(6)*y.^2;
  fxy = f(x,y);
  fprintf('\nmaximum error at interpolation points = %10.7e\n', max(abs(fxy-pxy)))

  print -dpdf twodim2
```

The resulting output follows.

```
coefficients, c:
                  0
  -0.94916310522349
   5.05919300007085
   0.78121462258957
   0.94916310522349
  -2.30227214332900

maximum error at interpolation points = 1.7763568e-15
```



20
10
0
-10
-20
$f(x,y)$
3
2
1
0
-1
y
-1
0
1
2
3
x



20
10
0
-10
-20
$f(x,y)$
3
2
1
0
-1
y
-1
0
1
2
3
x

5. [20 points]

Suppose the complex-valued function $f(z)$ of the variable $z \in \mathbb{C}$ is analytic in a region D of the complex plane whose boundary C is a simple closed contour. Furthermore, suppose the interpolation points $x_0, \ldots, x_n$ ($n \geq 1$) and the point x all lie in D.

(a) Let $p_n \in \mathcal{P}_n$ denote the polynomial that interpolates f at $x_0, \ldots, x_n$.
For any $x \in D$, confirm the identity

$$f(x) - p_n(x) \;=\; \frac{1}{2\pi \mathrm{i}} \int_C \frac{f(z)}{z - x} \prod_{j=0}^{n} \frac{x - x_j}{z - x_j} \,\mathrm{d}z$$

by computing the integral on the right. (Hint: Consider the poles of the integrand, and use the Cauchy integral formula.)

For the rest of the problem, suppose that the real number x and the interpolation points $x_0, \ldots, x_n$ all lie in the real interval $[a, b]$, and define, for constant $K > 0$,

$$D = \{z \in \mathbb{C} : |z - t| < K \text{ for some } t \in [a, b]\}.$$

(b) Plot (or draw) the boundary C of D for $[a, b] = [-1, 1]$ and $K = 1$.

(c) Show that the length of the contour C is $2(b - a) + 2\pi K$, and that the integral formula in (a) leads to the bound

$$|f(x) - p_n(x)| < \frac{(b - a + \pi K)M}{\pi K} \Big(\frac{b - a}{K}\Big)^{n+1},$$

where M is such that $|f(z)| \leq M$ on C.

(d) Deduce that if f is analytic on D for some $K > |b - a|$, then the sequence $\{p_n\}$ converges to f uniformly on $[a, b]$ as $n \to \infty$.

(e) Show that the requirements for the conclusion in (d) *are not satisfied* by Runge's function, $f(x) = 1/(1 + x^2)$ over $[a, b] = [-5, 5]$. For what values of α are the conditions satisfied by this f over $[a, b] = [-\alpha, \alpha]$?

[Süli and Mayers, Problem 6.11]

Solution.

(a) The integrand

$$g(z) = \frac{f(z)}{z - x_j} \prod_{j=0}^{n} \frac{x - x_j}{z - x_j}$$

has poles at $z = x, x_0, x_1, \ldots, x_n$. If $x = x_j$ for any $j \in \{0, \ldots, n\}$, then $f(x) - p_n(x) = 0$ by construction of the interpolant p_n, so the ultimate result (part (c)) is trivial. If x is not equal to any of the interpolation points, then all the poles of the integrand g are simple. Hence, the integral reduces to a straightforward residue calculation:

$$\begin{aligned} \operatorname{res}(g, x_k) &= \lim_{z \to x_k} g(z)(z - x_k) \\ &= -f(x_k) \prod_{j=0, j \neq k}^{n} \frac{x - x_j}{x_k - x_j} \end{aligned}$$

and

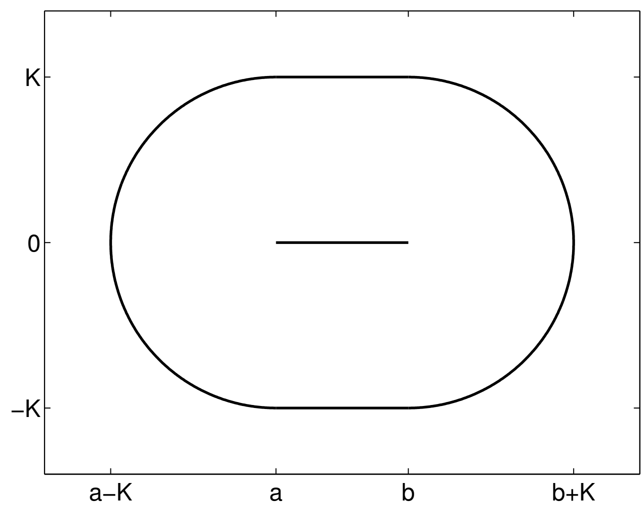
$$\begin{aligned} \operatorname{res}(g, x) &= \lim_{z \to x} g(z)(z - x) \\ &= f(x). \end{aligned}$$

Cauchy's integral formula then gives that

$$\begin{aligned}
\frac{1}{2\pi \mathrm{i}} \int_C g(z)\,\mathrm{d}z &= \operatorname{res}(g,x) + \sum_{k=0}^{n} \operatorname{res}(g,x_k) \\
&= f(x) - \sum_{k=0}^{n} f(x_k) \prod_{j=0, j\neq k}^{n} \frac{x-x_j}{x_k-x_j} \\
&= f(x) - \sum_{k=0}^{n} f(x_k)\ell_k(x) \\
&= f(x) - p_n(x),
\end{aligned}$$

where ℓ_k denotes the kth Lagrange basis polynomial.

(b) The specified region is an oval whose boundary is a curve C that consists of two half-circles of radius K centered at a and b on the left and right, with line segments connecting them; see the figure below.



(c) The arc-length of C is the sum of the perimeter of the circular arcs and line segments: $2\pi K + 2(b-a)$.

We can now bound the error by coarsely approximating the integral for $f - p_n$. (Note that for any $z \in C$, $|x - z| \geq K$ and $|x_k - z| \geq K$, and that $|x - x_k| \leq b - a$ for all $x \in [a, b]$.) Letting L_C denote the arc-length of C, we have

$$\begin{aligned}
|f(x) - p_n(x)| &= \Big| \frac{1}{2\pi \mathrm{i}} \int_C \frac{f(z)}{z-x} \prod_{j=0}^{n} \frac{x-x_j}{z-x_j}\,\mathrm{d}z \Big| \\
&\leq \frac{L_C}{2\pi} \max_{z\in C} \frac{|f(z)|}{|z-x|} \prod_{j=0}^{n} \frac{|x-x_j|}{|z-x_j|} \\
&\leq \frac{L_C}{2\pi} \frac{M}{K} \frac{(b-a)^{n+1}}{K^{n+1}}.
\end{aligned}$$

Substituting $L_C = 2\pi K + 2(b-a)$, this formula simplifies to

$$|f(x) - p_n(x)| \;=\; \frac{(b-a+\pi K)M}{\pi K} \left(\frac{b-a}{K}\right)^{n+1},$$

and, since $K > b - a$, this bound goes to zero as $n \to \infty$ independent of our choice of $x \in [a, b]$.

(d) If $[a, b] = [-5, 5]$, the result requires that f be analytic on D with $k > |b - a| = 10$. Note that D contains the imaginary segment $(-10\mathrm{i}, 10\mathrm{i})$, but $f() = (1 + x^2)^{-1}$ has poles at $z = \pm\mathrm{i}$, so f is not analytic on D, and we cannot apply the bound in (c).

Now we seek an interval $[-\alpha, \alpha]$ over which we can apply the bound in (c). That is, we must select the interval $[-\alpha, \alpha]$ such that the contour C does not touch or enclose either pole $\pm\mathrm{i}$, i.e., we must have $K < 1$. Recalling that $K > b - a = 2\alpha$ (to ensure that the $((b - a)/K)^{n+1}$ term goes to zero), we must have $\alpha < 1/2$.

In conclusion, polynomial interpolants to Runge's function *for any interpolation points* on the interval $[-1/2, 1/2]$ must always converge.

For further details, see: P. J. Davis, *Interpolation and Approximation*, Dover, 1975 (page 82) and the treatise: J. L. Walsh, *Interpolation and Approximation by Rational Functions in the Complex Domain*, 5th ed., American Mathematical Society, 1969.

---

6. [20 points]

The standard Lagrange interpolation formula for the polynomial $p_n \in \mathcal{P}_n$ that interpolates $f \in C[a,b]$ at the distinct points $\{x_j\}$,

$$p_n(x) = \sum_{j=0}^{n} \ell_j(x) f(x_j), \qquad \text{where} \qquad \ell_j(x) = \prod_{k=0,k\neq j}^{n} \frac{(x-x_k)}{(x_j-x_k)},$$

requires $O(n^2)$ floating point operations to evaluate for each point x. In this exercise, we construct an alternative Lagrange interpolation formula, known as the *barycentric interpolant*, that can evaluated more efficiently and also has superior numerical stability.

Let $w(x) = \prod_{k=0}^{n}(x-x_k)$ and define the *barycentric weight* as

$$\beta_j = \frac{1}{\prod_{k=0,k\neq j}^{n}(x_j-x_k)}, \qquad j = 0,\dots,n.$$

(a) Show that the Lagrange form for p_n can be rewritten as

$$p_n(x) = w(x)\sum_{j=0}^{n} \frac{\beta_j}{x-x_j} f(x_j).$$

(b) Verify that

$$1 = w(x)\sum_{j=0}^{n} \frac{\beta_j}{x-x_j}.$$

(Hint: This follows from part (a) with a special choice of f.)

(c) Dividing the result of part (a) by the result of part (b) yields the *barycentric interpolation formula*

$$p_n(x) = \frac{\displaystyle\sum_{j=0}^{n} \frac{\beta_j}{x-x_j} f(x_j)}{\displaystyle\sum_{j=0}^{n} \frac{\beta_j}{x-x_j}}.$$

Assuming the β_j values are already known, how many floating point operations are required to evaluate $p_n(x)$ for some point x?

(d) Suppose $[a,b] = [0,1]$ and $x_j = j/n$ for $j = 0,\dots,n$. Derive a simple formula for β_j in terms of j and n. For which j values is β_j largest (in absolute value)? (These terms will be favored in the formula in part (c).)

[Berrut and Trefethen]

Solution.

This problem was intended to introduce you to the idea of barycentric interpolation, the subject of a recent article by J.-P. Berrut and L. N. Trefethen ('Barycentric Lagrange Interpolation', *SIAM Review* 46 (2004) 501–517; `http://epubs.siam.org/sam-bin/dbq/article/41771`). You are encouraged to review this article.

(a) Note that ℓ_j can be written as

$$\begin{aligned}\ell_j(x) &= \prod_{k=0,k\neq j}^{n} \frac{(x-x_k)}{(x_j-x_k)} \\ &= \beta_j \prod_{k=0,k\neq j}^{n}(x-x_k) \;=\; \beta_j \frac{w(x)}{x-x_j},\end{aligned}$$

and hence

$$
\begin{aligned}
p_n(x) &= \sum_{j=0}^{n} \ell_j(x) f(x_j) \\
&= \sum_{j=0}^{n} \beta_j \frac{w(x)}{x - x_j} f(x_j) \;=\; w(x) \sum_{j=0}^{n} \frac{\beta_j}{x - x_j} f(x_j).
\end{aligned}
$$

It may seem unappealing that this formula suggests that $p_n(x_j) = 0/0$, but notice that the offending zero terms in the numerator and denominator cancel.

(b) The formula

$$
1 = w(x) \sum_{j=0}^{n} \frac{\beta_j}{x - x_j}
$$

follows from part (a) by taking $f(x) = 1$. Recall that the interpolating polynomial $p_n \in \mathcal{P}_n$ is *unique*. If f is a polynomial of degree n or less, then we must have $p_n = f$. In this case, $f(x) = 1$ for all x implies that $p_n(x) = 1$ for all x.

(c) First consider the numerator, which is the sum of $n + 1$ terms, each of which is computed via 3 floating point operations. To sum these $n + 1$ terms, an additional n flops are required, giving a total of $4n + 3$ flops for the numerator.

The denominator is also the sum of $n+1$ terms. We can organize the computation so that we do not need to recompute $\beta_j/(x - x_j)$, which was a component of the numerator. Hence, we only have an additional n floating point operations for the sum.

Finally, the main division adds one more floating point operation, for a total of $5n + 4$ floating point operations.

The key advantage of the barycentric formula is that the β_j factors need only be computed once independent of the point x where p_n is evaluated. Note that $O(n^2)$ operations are required to compute all of the β_j terms, but beyond this initial cost (which can often be allayed, as compact formulas for β_j for canonical choices of $\{x_j\}$ are known; see, e.g., part (d)), $p_n(x)$ can be evaluated in $O(n)$ operations. Contrast this with the standard Lagrange formulation, which requires $O(n^2)$ operations for every evaluation of $p_n(x)$.

(d) Suppose $x_j = j/n$. Then

$$
\begin{aligned}
\beta_j^{-1} = \prod_{k=0, k\neq j}^{n} (x_j - x_k) &= \prod_{k=0, k\neq j}^{n} (j/n - k/n) \\
&= n^{-n} \prod_{k=0, k=j}^{n} (j - k) \\
&= n^{-n} \Big((j)(j-1)(j-2)\cdots(1)(-1)(-2)\cdots(j-n) \Big) \\
&= n^{-n} (-1)^{n-j} j!\,(n-j)!.
\end{aligned}
$$

Hence,

$$
\beta_j = \frac{n^n (-1)^{n-j}}{j!\,(j-n)!}.
$$

We conclude that β_j terms are largest (in magnitude) for when $j \approx n/2$. For example, when $n = 5$ we have $\beta_n = -\beta_0 = 5^5/5! = 3125/120 \approx 26.40$, while $\beta_3 = -\beta_2 = 5^5/(3!2!) = 3125/12 \approx 260.4$. Better choices of interpolation points (e.g., Chebyshev points) give more uniform values for β_j.

7. [20 points]

As mentioned in class, the Weierstrass Approximation Theorem states that for any $f \in C[a,b]$ and any $\varepsilon > 0$, there exists some polynomial (of unspecified degree) such that $\max_{x\in[a,b]} |f(x) - p(x)| < \varepsilon$. The most common proof of this fact is *constructive*: one can use for the approximating polynomial the *Bernstein polynomial* of appropriate degree. When $[a,b] = [0,1]$, the degree-n Bernstein polynomial is defined as

$$B_n(x) = \sum_{k=0}^{n} f(k/n)\binom{n}{k} x^k(1-x)^{n-k},$$

where $\displaystyle\binom{n}{k} = \frac{n!}{k!(n-k)!}$ can be obtained in MATLAB via the `nchoosek` command.

Remarkably, it turns out that for any $f \in C[a,b]$, we have $\max_{x\in[a,b]} |f(x) - B_n(x)| \to 0$ as $n \to \infty$. In this exercise you shall explore the rate at which this convergence occurs.

(a) Confirm that $B_n(x) \to f(x)$ for $x \in [0,1]$ and $f(x) = \sin(3\pi x)$ by producing a MATLAB plot that compares $f(x)$ to $B_n(x)$ on $x \in [0,1]$ for $n = 5, 10, 20$. (Please label the plot clearly!)

(b) Describe how to modify the definition of B_n so as to work for a general interval $[a,b] \neq [0,1]$.

(c) Let $f(x) = \mathrm{e}^x$ and $[a,b] = [-1,1]$. Write MATLAB code to compute $B_n(x)$ as well as the polynomial $p_n(x)$ that interpolates f at the Chebyshev points

$$x_k = \cos(k\pi/n), \qquad k = 0,\ldots,n.$$

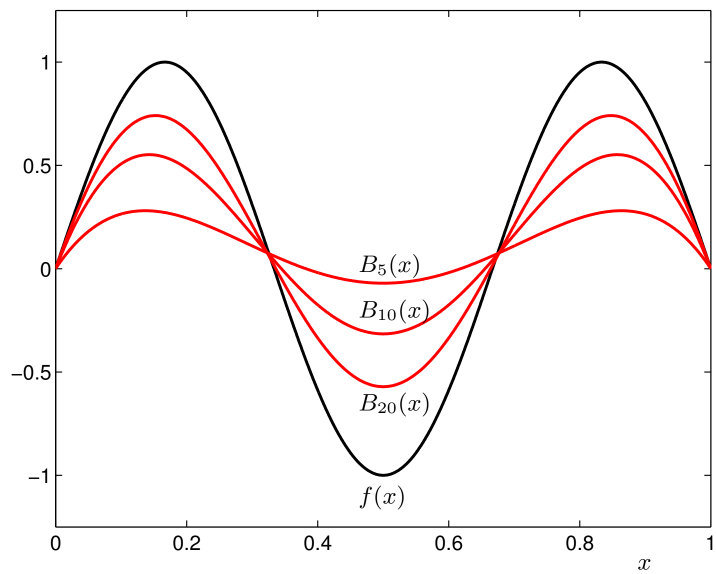
(You may use the monomial, Newton, or Lagrange basis.) Turn in a `semilogy` plot that compares $\max_{x\in[-1,1]} |f(x) - B_n(x)|$ with $\max_{x\in[-1,1]} |f(x) - p_n(x)|$ for $n = 1,\ldots,40$. (For purposes of this problem, you may ignore any warnings issued by `nchoosek` for large n.)

(d) Repeat part (c) with $f(x) = x^2 - 1$ and $[a,b] = [-1,1]$.

Solution.

(a) The following MATLAB code produces the required plot.

```
f = @(x) sin(3*pi*x);
x = linspace(0,1,1000)';
fx = f(x);

figure(1), clf
plot(x,fx,'k-','linewidth',2), hold on
for n=[5 10 20]
  xx = [0:n]/n;
  Bx = zeros(size(x));
  for k=0:n
     Bx = Bx+f(xx(k+1))*nchoosek(n,k)*(x.^k).*(1-x).^(n-k);
  end
  plot(x,Bx,'r-','linewidth',2)
end
ylim([-1.25 1.25])
set(gca,'fontsize',14)
```



(b) We obtain Bernstein polynomials on the interval $[a, b]$ by changing variables in the previous definition. In the formula for B_n given in part (a), the $f(k/n)$ term evaluates f at a point $k/n \in [0, 1]$. The change of variable replaces $x \in [0, 1]$ with $a + x(b - a)$, so that the $f(k/n)$ term becomes

$$f(a + (k/n)(b - a)).$$

The variable $x \in [0, 1]$ in the x^k and $(1 - x)^{n-k}$ terms in definition of B_n is replaced by $(x - a)/(b - a)$, which maps $[a, b] \mapsto [0, 1]$. Altogether, we have

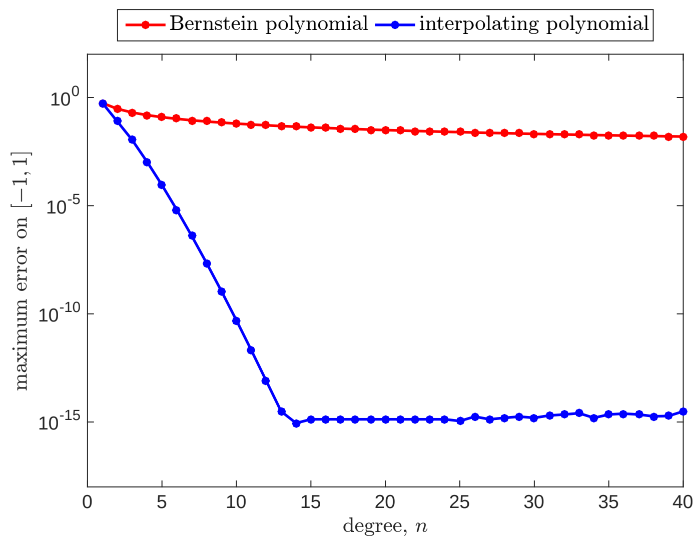
$$B_n(x) = \sum_{k=0}^{n} f(a + (k/n)(b - a)) \binom{n}{k} \left(\frac{x - a}{b - a}\right)^k \left(\frac{b - x}{b - a}\right)^{n-k}.$$

(c) The code below produces the desired plot. For simplicity, polynomial interpolation is performed using the `polyfit` command in MATLAB, which uses the monomial basis. If this approach exhibited significant instability, we could replace it with a Newton or barycentric Lagrange approach.

Notice that the convergence of the Bernstein polynomials is incredibly slow compared to the interpolant at Chebyshev points. Though a theoretically useful tool, Bernstein polynomials are not well-suited to practical approximations. [The original problem asked for a plot with $n = 0, \ldots, 40$, but the Bernstein polynomial is not well defined at $n = 0$, so we have just used $n = 1, \ldots, 40$.]

```
a = -1;
b = 1;
f = @(x) exp(x);
x = linspace(a,b,1000)';
fx = f(x);

nmax = 40;
err = zeros(nmax,2);
for n=1:nmax
  xx = a+[0:n]*(b-a)/n;
  Bx = zeros(size(x));
  for k=0:n
     Bx = Bx+f(xx(k+1))*nchoosek(n,k)*(((x-a)/(b-a)).^k).*((b-x)/(b-a)).^(n-k);
  end
  err(n,1) = max(abs(fx-Bx));
  yy = cos([0:n]*pi/n);
  p = polyfit(yy,f(yy),n);
  err(n,2) = max(abs(fx-polyval(p,x)));
end

figure(1), clf
```

```
semilogy([1:nmax],err(:,1),'r.-','linewidth',2,'markersize',22)
hold on
semilogy([1:nmax],err(:,2),'b.-','linewidth',2,'markersize',22)
ylim([1e-18 1e2])
set(gca,'fontsize',14)
xlabel('degree, $n$','interpreter','latex','fontsize',16)
ylabel('maximum error on $[-1,1]$','interpreter','latex','fontsize',16)
ylim([1e-18 1e2])
set(gca,'fontsize',14)
xlabel('degree, $n$','interpreter','latex','fontsize',16)
ylabel('maximum error on $[-1,1]$','interpreter','latex','fontsize',16)
leg = legend('Bernstein polynomial','interpolating polynomial',...
      'orientation','horizontal','location','northoutside');
set(leg,'fontsize',16,'interpreter','latex')
```



(d) This problem requires only a minor modification to the previous code (replace the definition of the function `f`), but the results are even more extreme: the Chebyshev interpolant is *exact* (up to rounding error) for $n \geq 2$, given that the interpolating polynomial is unique and $f(x) = x^2 - 1$ is a polynomial interpolant of itself. The Bernstein polynomial is not exact: it converges slowly, as before.