MATH 128A: Homework #4

Professor John Strain
Assignment: 1-10

Eric Chuu

July 21, 2016

Problem 1 (BF 3.4.11)

(a) Show that $H_{2n+1}(x)$ is the unique polynomial of least degree agreeing with f and f' at $x_0, \ldots, x_n$.

(b) Derive the error term in Theorem 3.9. Use the same method as the Lagrange error derivation, defining

$$g(t) = f(t) - H_{2n+1}(t) - \frac{(t-x_0)^2 \cdots (t-x_n)^2}{(x-x_0)^2 \cdots (x-x_n)^2} [f(x) - H_{2n+1}(x)] \tag{1}$$

and using the fact that $g'(t)$ has $(2n+2)$ distinct zeros in $[a,b]$.

Solution
(a) Suppose that $P(x)$ is another polynomial such that $P(x_k) = f(x_k)$ and $P'(x_k) = f'(x_k)0$ for $0 \leq k \leq n$, and $P(x)$ is of degree at most $(2n+1)$. Then we consider the difference

$$\begin{aligned} D(x) &:= P(x) - H_{2n+1}(x) \\ D(x_k) &= f(x_k) - f(x_k) = 0 \\ D'(x_k) &= f'(x_k) - f'(x_k) = 0 \end{aligned}$$

for $0 \leq k \leq n$. Thus $D(x)$ is a polynomial of at most $(2n+1)$, but as seen above, it has $(n+1)+(n+1) = (2n+2)$ roots, which implies that $D(x) \equiv 0$. Thus $P(x) \equiv H_{2n+1}$, and H_{2n+1} is unique. $\square$

(b) If $x = x_k$, then the error formula holds trivially, so suppose $x \neq x_k$ for $0 \leq k \leq n$. Then we define the function g as in equation (1). Note f agrees with H_{2n+1} at the interpolating points, so $g(x_k) = 0$ for $0 \leq k \leq n$. g has an additional root at x, so $g(x) = 0$. Thus, g has $(n+2)$ distinct roots, and by Rolle's Theorem, g' has $(n+1)$ distinct roots $\xi_0, \ldots \xi_n$ between $x_0, \ldots, x_n, x$. Since f' agrees with H'_{2n+1} at its interpolating points as well, we have $g'(x_k) = 0$ for $0 \leq k \leq n$. Adding the number of roots together, g' has $(n+1)+(n+1) = (2n+2)$ distinct roots. By construction g is $(2n+1)$-times differentiable, so by the Generalized Rolle's Theorem, there exists $\xi \in [a,b]$ such that $g^{(2n+2)}(\xi) = 0$. We take $(2n+2)$ derivatives:

$$g^{(2n+2)}(t) = f^{2n+2}(t) - \left(\frac{d}{dt}\right)^{2n+2} (H_{2n+1}(t)) - \frac{(2n+2)!}{(x-x_0)^2 \cdots (x-x_n)^2} \cdot [f(x) - H_{2n+1}(x)]$$

Taking $t = \xi$, we get

$$\begin{aligned} 0 = g^{(2n+2)}(\xi) &= f^{2n+2}(\xi) - \frac{(2n+2)!}{(x-x_0)^2 \cdots (x-x_n)^2} \cdot [f(x) - H_{2n+1}(x)] \\ \implies f(x) &= H_{2n+1} + \frac{(x-x_0)^2 \cdots (x-x_n)^2}{(2n+2)!} f^{(2n+2)}(\xi) \end{aligned}$$

for $\xi \in (a,b)$. $\square$

Problem 2 (BF 3.4.12)

Let $z_0 = x_0, z_1 = x_0, z_2 = x_1$, and $z_3 = x_1$. Form the following divided difference table.

$z_0 = x_0$	$f[z_0] = f(x_0)$			
		$f[z_0, z_1] = f'(x_0)$		
$z_1 = x_0$	$f[z_1] = f(x_0)$		$f[z_0, z_1, z_2]$	
		$f[z_1, z_2]$		$f[z_0, z_1, z_2, z_3]$
$z_2 = x_1$	$f[z_2] = f(x_1)$		$f[z_1, z_2, z_3]$	
		$f[z_2, z_3] = f'(x_1)$		
$z_3 = x_1$	$f[z_3] = f(x_1)$			

Show that the cubic Hermite polynomial $H_3(x)$ can also be written as

$$f[z_0] + f[z_0, z_1](x - x_0) + f[z_0, z_1, z_2](x - x_0)^2 + f[z_0, z_1, z_2, z_3](x - x_0)^2(x - x_1) \tag{2}$$

Solution
Let $H(x)$ be defined as in (2) above, so

$$H(x) = f[z_0] + f[z_0, z_1](x - x_0) + f[z_0, z_1, z_2](x - x_0)^2 + f[z_0, z_1, z_2, z_3](x - x_0)^2(x - x_1)$$

Then we show that $H(x)$ is equivalent to $H_3(x)$, thus satisfying the properties of the Hermite polynomial. Using the table above, we substitute $f[z_o] = f(x)$ and $f[z_0, z_1] = f'(x_0)$ into $H(x)$, and we evaluate $H(x)$ at $x = x_0$, we see that other than the first term, every other term in the polynomial has a factor of $(x - x_0)$ in the product, so

$$H(x_0) = f(x_0).$$

If we evaluate $H(x)$ at $x = x_1$, then the last term in the polynomial vanishes, leaving

$$H(x_1) = f(x_0) + f'(x_0)(x_1 - x_0) + [f(x_1) - f(x_0) - f'(x_0)(x_1 - x_0)] = f(x_1)$$

To check that the derivatives also satisfy the properties of the cubic Hermite polynomial, we differentiate H and evaluate at x_0 and x_1.

$$\begin{aligned} H'(x) =& f'(x_0) = 2 \cdot \frac{2f(x_1) - f(x_0) - f'(x_0)(x_1 - x_0)}{(x_1 - x_0)^2}(x - x_0) \\ &+ \frac{f'(x_1)(x_1 - x_0) - 2f(x_1) + 2f(x_0) + f'(x_0)(x_1 - x_0)}{(x_1 - x_0)^2} \cdot \left[2(x - x_0)(x - x_1) + (x - x_0)^2\right] \end{aligned}$$

The second and third terms vanish at $x = x_0$, so $H'(x_0) = f'(x_0)$. At $x = x_1$,

$$\begin{aligned} H'(x_1) &= f'(x_0) + \frac{2f(x_1) - 2f(x_0)}{x_1 - x_0} - 2f'(x_0) + f'(x_1) - \frac{2f(x_1) - 2f(x_0)}{x_1 - x_0} + f'(x_0) \\ &= f'(x_1) \end{aligned}$$

We conclude that $H_3(x)$ can be written as $H(x)$ as defined in (2). $\square$

Problem 3 (BF 4.1.13)

Use the following data and the knowledge that the first five derivatives of f are bounded on $[1,5]$ by $2, 3, 6, 12, 23$ respectively to approximate $f'(3)$ as accurately as possible. Find a bound for the error.

x	1	2	3	4	5
$f(x)$	2.4142	2.6734	2.8974	3.0976	3.2804

Solution
The formula for approximating $f'(x_0)$ is given by

$$f'(x_0) = \frac{1}{12h}\left[f(x_0-2h) - 8f(x_0-h) + 8f(x_0+h) - f(x_0+2h)\right]$$

We can approximate $f'(x_0)$ for $x_0 = 3, h = 1$ with

$$\begin{aligned} f'(3) &= \frac{1}{12}\cdot[f(3-2) - 8f(3-1) + 8f(3+1) - f(3+2)] \\ &= \frac{1}{12}\cdot[f(1) - 8f(2) + 8f(4) - f(5)] \\ &= \frac{1}{12}\cdot[2\cdot 2.4142 - 8\cdot 2.6734 + 8\cdot 3.0976 - 3.23804] \\ &= 0.21062 \end{aligned}$$

We can bound the error with

$$\left|\frac{h^4}{30}f^{(5)}(\xi)\right| \le \left|\frac{1}{30}\cdot 23\right| = 0.76667$$

where $\xi \in [1,5]$. □

Problem 4 (BF 4.1.27)

Choose your favorite function f, non-zero number x, and computer calculator. Generate approximations $f'_n(x)$ to $f'(x)$ by

$$f'_n(x) = \frac{f(x+10^{-n}) - f(x)}{10^{-n}} \tag{3}$$

for $n = 1, 2, \ldots, 20$, and describe what happens.

Solution For the function f, where $f(x) = x^2$, the approximation f'_n's initially approximate the derivative, $f'(3)$, reasonably accurately, but then go to 0, since the difference in the numerator goes to 0. This behavior is shown in the MATLAB code output below. After the 15th approximation of the derivative, $f'_n(3) \to 0$. □

Iteration n	$f'_n(3)$
1	6.100000000000012
2	6.009999999999849
3	6.000999999999479
4	6.000100000012054
5	6.000009999951318
6	6.000001000927568
7	6.000000087880153
8	5.999999963535174
9	6.000000496442226
10	6.000000496442226
11	6.000000496442225
12	6.000533403494046
13	6.004086117172847
14	6.217248937900877
15	5.329070518200751
16	0
17	0
18	0
19	0
20	0

```
% approximate f'(x) with f_n'(x) for n = 1,2,...,20

function f_prime = approximate_derivative(x)
    f_prime = zeros(20,1);
    for n = 1:20
        h = 10^(-n);
        f_prime(n,1) = (f(x + h) - f(x)) / h;
    end

    function y = f(x)
        y = x^2;
```

Problem 5 (BF 4.2.13)

Suppose the following extrapolation table has been constructed to approximate the number M with $M = N_1(h) + K_1h^2 + K_2h^4 + K_3h^6$:

$N_1(h)$		
$N_1\left(\frac{h}{2}\right)$	$N_2(h)$	
$N_1\left(\frac{h}{4}\right)$	$N_2\left(\frac{h}{2}\right)$	$N_3(h)$

(a) Show that the linear interpolating polynomial $P_{0,1}(h)$ through $(h^2, N_1(h))$ and $(h^2/4, N_1(h/2))$ satisfies $P_{0,1}(0) = N_2(h)$. Similarly, show that $P_{1,2}(0) = N_2(h/2)$.

(b) Show that the linear interpolating polynomial $P_{0,2}(h)0$ through $(h^4, N_2(h))$ and $(h^4/16, N_2(h/2))$ satisfies $P_{0,2}(0) = N_3(h)$.

Solution
(a) The linear interpolating polynomial $P_{0,1}$ is given by

$$\begin{aligned}
P_{0,1}(x) &= \frac{(x-h^2)N_1(h/2) - (x-h^2/4)(N_1(h))}{h^2/4 - h^2} \\
P_{0,1}(0) &= \frac{-h^2N_1(h/2) + h^2/4 \cdot N_1(h)}{-3h^2/4} \\
&= \frac{4N_1(h/2) - N_1(h)}{3} \\
&= N_2(h)
\end{aligned}$$

If we consider the linear interpolating polynomial $P_{1,2}$, we see that

$$\begin{aligned}
P_{1,2}(x) &= \frac{(x-h^2/4)N_1(h/4) - (x-h^2/16)N_1(h/2)}{h^2/16 - h^2/4} \\
P_{1,2}(0) &= \frac{-h^2/4N_1(h/4) + h^2/16 \cdot N_1(h/2)}{-3h^2/16} \\
&= \frac{4N_1(h/4) - N_1(h/2)}{3} \\
&= N_2(h/2)
\end{aligned}$$

and we see that $P_{1,2}(0) = N_2(h/2)$. □

(b) We consider the linear interpolating polynomial $P_{0,2}$ and check that it satisfies $P_{0,2}(0) = N_3(h)$

$$\begin{aligned}
P_{0,2}(x) &= \frac{(x-h^4)N_2(h/2) - (x-h^4/16)N_2(h/2)}{h^4/16 - h^4} \\
P_{0,2}(0) &= \frac{-h^4N_2(h/2) + h^4/16 \cdot N_2(h)}{-15h^4/16} \\
&= \frac{16N_2(h/2) - N_2(h)}{15} \\
&= N_3(h)
\end{aligned}$$

□

Problem 6 (BF 4.3.15)

Find the degree of precision of the quadrature formula

$$\int_{-1}^{1} f(x)dx = f\left(-\frac{\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right)$$

Solution
For $n = 1$,

$$\int_{-1}^{1} xdx = \frac{1}{2}x^2\Big|_{-1}^{1} = 0$$
$$\left(-\frac{\sqrt{3}}{3}\right) + \left(\frac{\sqrt{3}}{3}\right) = 0$$

For $n = 2$,

$$\int_{-1}^{1} x^2dx = \frac{1}{3}x^3\Big|_{-1}^{1} = \frac{2}{3}$$
$$\left(-\frac{\sqrt{3}}{3}\right)^2 + \left(\frac{\sqrt{3}}{3}\right)^2 = \frac{2}{3}$$

For $n = 3$,

$$\int_{-1}^{1} x^3dx = \frac{1}{4}x^4\Big|_{-1}^{1} = 0$$
$$\left(-\frac{\sqrt{3}}{3}\right)^3 + \left(\frac{\sqrt{3}}{3}\right)^3 = 0$$

For $n = 4$,

$$\int_{-1}^{1} x^4dx = \frac{1}{5}x^5\Big|_{-1}^{1} = \frac{2}{5}$$
$$\left(-\frac{\sqrt{3}}{3}\right)^4 + \left(\frac{\sqrt{3}}{3}\right) = \frac{2}{9}$$

so we see that for $n = 4$, the quadrature formula does not hold, and we conclude that it has precision of degree 3. $\square$

Problem 7 (BF 4.3.21)

Approximate the following integrals using formulas 4.25 through 4.32. Are the accuracies of the approximations consistent with the error formulas? Which of parts (d) and (e) give the better approximation?

$$(a) \quad \int_0^{0.1} \sqrt{1+x}dx \qquad (b) \quad \int_0^{\pi/2} (\sin x)^2 dx$$

$$(c) \quad \int_{1.1}^{1.5} e^x dx \qquad (d) \quad \int_1^{10} \frac{1}{x} dx$$

$$(e) \quad \int_1^{5.5} \frac{1}{x} dx + \int_{5.5}^{10} \frac{1}{x} dx \qquad (f) \quad \int_0^1 x^{1/3} dx$$

Solution

We evaluate each of the integrals 8 different ways:
(a)

$$\begin{aligned}
\int_0^{0.1} \sqrt{1+x} &= \frac{0.1}{2}(1+\sqrt{1.1}) = 0.1024404 \\
&= \frac{0.05}{3}(1+4\sqrt{1.05}+\sqrt{1.1}) = 0.1024598 \\
&= \frac{3\cdot 0.03}{8}(1+3\sqrt{1.0333}+3\sqrt{1.0666}+\sqrt{1.1}) = 1.024598 \\
&= \frac{2\cdot 0.025}{45}(7\cdot 1+32\sqrt{1.025}+12\sqrt{1.05}+32\sqrt{1.075}+7\sqrt{1.1}) = 0.1024598 \\
&= 2\cdot 0.05\sqrt{1.5} = 0.1024695 \\
&= \frac{3\cdot 0.0333}{2}(\sqrt{1.0333}+\sqrt{1.0666}) = 0.1023638 \\
&= \frac{4\cdot 0.025}{3}(2\sqrt{1.025}-\sqrt{1.05}+2\sqrt{1.075}) = 0.1024598 \\
&= \frac{5\cdot 0.02}{24}(11\cdot\sqrt{1.02}+\sqrt{1.04}+\sqrt{1.06}+11\sqrt{1.08}) = 0.1024598
\end{aligned}$$

For the integrals in (b) - (f), we use the MATLAB code on the following page to generate the approximations for each of the integrals. The results for each part are shown below. Note that (e) more accurately approximates the integral than does (d), as its actual value is ≈ 2.3026. The method for approximation runs down the first column. □

Table 1: Integral Approximations for (b) - (f)

	(b)	(c)	(d)	(e)	(f)
4.25	0.785398163397448	1.497171018856899	4.950000000000000	3.293181818181818	0.500000000000000
4.26	0.785398163397448	1.477536117650765	2.740909090909091	2.407900969997744	0.695800350656066
4.27	0.785398163397448	1.477528858911821	2.563392857142857	2.359771294881589	0.712603152157600
4.28	0.785398163397448	1.477523049502318	2.385700428603654	2.314754053014805	0.730634139949349
4.29	0.785398163397448	1.467718667047697	1.636363636363636	1.965260545905707	0.793700525984100
4.30	0.785398163397448	1.470981472263461	1.767857142857143	2.048634453781513	0.783470869543467
4.31	0.785398163397448	1.477511614872427	2.116378555332526	2.233250500654733	0.761113705580971
4.32	0.785398163397448	1.477515101121390	2.116378555332526	2.249000604210563	0.759357225960260

MATLAB Code for approximating the integral

```
function approx = approx_integral()
    result = zeros(8,1); % store each of the 8 approximations
    lower = 0; upper = 1;
    result(1,1) = m1(lower, upper);
    result(2,1) = m2(lower, upper);
    result(3,1) = m3(lower, upper);
    result(4,1) = m4(lower, upper);
    result(5,1) = m5(lower, upper);
    result(6,1) = m6(lower, upper);
    result(7,1) = m7(lower, upper);
    result(8,1) = m8(lower, upper);
    approx = result;

    function y1 = m1(lower, upper)
        h = upper - lower;
        y1 = h / 2 * (f(lower) + f(upper));
    function y2 = m2(lower, upper)
        h = (upper - lower) / 2;
        y2 = h / 3 * (f(lower) + 4 * f(lower + h) + f(upper));
    function y3 = m3(lower, upper)
        h = (upper - lower) / 3;
        y3 = 3 * h / 8 * (f(lower) + 3 * f(lower + h) + 3 * f(lower + 2*h) + f
            (upper));
    function y4 = m4(lower, upper)
        h = (upper - lower) / 4;
        y4 = 2 * h / 45 * (7*f(lower) + 32*f(lower + h) + 12*f(lower + 2*h) +
            32*f(lower + 3*h) + 7*f(upper));
    function y5 = m5(lower, upper)
        h = (upper - lower) / 2;
        y5 = 2 * h * f(lower + h);
    function y6 = m6(lower, upper)
        h = (upper - lower) / 3;
        y6 = 3 * h / 2 * (f(lower + h) + f(lower + 2*h));
    function y7 = m7(lower, upper)
        h = (upper - lower) / 4;
        y7 = 4 * h / 3 * (2 * f(lower + h) - f(lower + 2*h) + 2 * f(lower + 3*
            h));
    function y8 = m8(lower, upper)
        h = (upper - lower) / 5;
        y8 = 5 * h / 24 * (11 * f(lower + h) + f(lower + 2*h) + f(lower + 3*h)
             + 11 * f(lower + 4*h));
    function y = f(x)
        % y = (sin(x))^2; % (b)
        % y = exp(x);      % (c)
        % y = 1 / x;       % (d)
        y = x^(1/3);       % (f)
```

Problem 8

(a) Show that

$$\int_0^1 x^{-x} dx = \sum_{n=1}^{\infty} n^{-n} \tag{4}$$

(b) Use the sum in (a) to evaluate the integral in (a) to 12-digit accuracy.

(c) Evaluate the integral in (a) by Romberg integration to 1,2, and 3-digit accuracy. Estimate how many function evaluations Romberg integration will require to achieve 12-digit accuracy. Explain the agreement or disagreement of your results with theory.

Solution
(a) We first rewrite x^{-x} into something that is more easily integrated:

$$x^{-x} = e^{\ln x^{-x}} = e^{-x \ln x}$$

Integrating this, using the Taylor series expansion of e^x, we get

$$\int_0^1 x^{-x} dx = \int_0^1 e^{-x \ln x} dx = \int_0^1 \sum_{n=0}^{\infty} \frac{(-x \ln x)^n}{n!} dx.$$

We can then use Fubini's Theorem to change the order of summation and integration

$$\int_0^1 \sum_{n=0}^{\infty} \frac{(-x \ln x)^n}{n!} = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \int_0^1 x^n (\ln x)^n dx$$

Using a change of variable by letting $u := e^u$ and changing the limits of integration accordingly, we get

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \int_{-\infty}^{0} e^{un} u^n e^u du = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \int_{-\infty}^{0} e^{u(n+1) u^n du}$$

We can then use another change of variable by letting $u := \frac{-y}{n+1}$ and change the limits of integration accordingly (note when $u = -\infty$, $y = \infty$) to get

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \int_{\infty}^{0} e^{-y} \cdot (-1)^{n+1} \frac{y^n}{(n+1)^{n+1}} dy = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \cdot \frac{(-1)^{n+1}}{(n+1)^{n+1}} \int_{\infty}^{0} e^{-y} \cdot y^n dy \tag{5}$$

$$= \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \cdot \frac{(-1)^{n+1}}{(n+1)^{n+1}} \cdot (-1) \int_0^{\infty} e^{-y} \cdot y^n dy \tag{6}$$

$$= \sum_{n=0}^{\infty} \frac{-1^{2n+2}}{n! \cdot (n+1)^{n+1}} \cdot \Gamma(n+1) \tag{7}$$

$$= \sum_{n=0}^{\infty} \frac{1}{n! \cdot (n+1)^{n+1}} \cdot n! \tag{8}$$

$$= \sum_{n=0}^{\infty} \frac{1}{(n+1)^{n+1}} \tag{9}$$

Finally, letting $k := n + 1$, we can rewrite the last sum as

$$\sum_{k=1}^{\infty} \frac{1}{k^k} = \sum_{k=1}^{\infty} k^{-k}$$

so the equality in (4) holds. Note that in (6) we swap the limits of integration, thus giving an extra factor of -1, so that we can then use the Γ-function (7) to simplify the summation. $\square$

(b) We can then use the sum in (4) to evaluate the integral to 12-digit accuracy. The MATLAB code below can be used to obtain the result. As seen from the 11th iteration, the sum evaluated to 12 digits of accuracy is 1.291285997062.

The last three iterations of output are as follows:
Iteration: 9
Current sum: 1.29128599705904

Iteration: 10
Current sum: 1.29128599706255

Iteration: 11
Current sum: 1.29128599706266

```
% Approximate the sum to 12 digits of accuracy
function sigma = sum_n()
    TOL = 10^(-12);
    currSum = 1^(-1)
    sum = 0;
    n = 2;
    while abs(currSum - sum) > TOL
        sum = currSum;
        currSum = sum + n^(-n);
        fprintf('Iteration: %d %s', n-1, 10);
        fprintf('Previous sum: %.14f %s', sum, 10);
        fprintf('Current sum: %.14f %s %s', currSum, 10, 10);
        n = n + 1;
    end
    sigma = currSum;
```

(c) Romberg Integration suggests if we want to approximate the integral to 12-digit accuracy, then $h \approx 2^{-k}$ implies that the error is $O(h^{2k+2}) = O(2^{-k(2k+2)}) = 10^{-12}$, so we need approximately $k = 5$ steps, but our results imply otherwise. The reason there is this discrepancy is because Romberg Integration also requires that the function have $(2k+2)$-times differentiable, so in this case we need the function to be 12-times differentiable, but the function we are integrating begins to have issues with differentiability after taking one derivative, so the hypotheses of Romberg's Integration are not met, so faster convergence is not guaranteed.

Table 2: Romberg Integration for $1, 2, 3$-digit accuracy

Approximation	# Function Evals	Total Function Evaluations
1.0000	2	2
1.2071	1	3
1.2673	4	7
1.2845	8	15
1.2894	16	31
1.2908	32	63
1.2911	64	127

As seen in the Table 2 above, each new calculation of $R_{k,1}$ requires 2^{k-1} function evaluations, for $k \geq 2$. Based on the above data, it would take just over 4 million function evaluations to achieve 12-digit accuracy. The MATLAB code used for Romberg Integeration is on the following page. □

MATLAB Code for Romberg Integration

```
% romberg integration

function R = romberg(n)
    a = 0;
    b = 1;
    h = b - a;
    TOL = 10^(-n);
    R(1,1) = h/2 * (f(a) + f(b));
    R(1,2) = 2; % 2 function evals
    R(1,3) = 2; % Total: 2 function evals
    R(2,1) = 1/2 * (R(1,1) + h*f(a + h/2));
    R(2,2) = 1; % 1 function eval
    R(2,3) = R(2,2) + R(1,3); % Total: 3 function evals
    k = 2;
    % each iteration => 2^(k-2) function evals
    while abs(R(k,1) - R(k-1,1)) > TOL
        k = k + 1;
        sum_k = 0;
        for i = 1:2^(k-2)
            sum_k = sum_k + f(a + (2 * i - 1) * h / 2^(k-1));
        end
        R(k,1) = 1/2 * (R(k-1, 1) + h/2^(k-2) * sum_k);
        R(k,2) = 2^(k-1); % update function eval
        R(k,3) = R(k,2) + R(k-1,3);
    end

    function y = f(x)
        y = x^(-x);
```

Problem 9

(a) Given the Euler-Maclaurin summation formula

$$\int_0^1 f(x)dx = \frac{1}{2}(f(0)+f(1)) + \sum_{m=1}^{\infty} b_m \left(f^{(2m-1)}(1) - f^{(2m-1)}(0)\right) \tag{10}$$

for some known constants b_m independent of f, find a formula for b_m by evaluating both sides for $f(x) = e^x$ where λ is a parameter.

(b) Compute $b_1, b_2, \ldots, b_{10}$.

Solution
(a) We calculate the left and right sides accordingly and get the initial equality below

$$\frac{e^\lambda - 1}{\lambda} = \frac{1}{2}(1+e^\lambda) + \sum_{m=1}^{\infty} b_m \lambda^{2m-1}(e^\lambda - 1)$$
$$1 = \frac{\lambda}{2}\left(\frac{e^\lambda+1}{e^\lambda-1}\right) + \sum_{m=1}^{\infty} b_m \lambda^{2m}$$

We can then use the hyperbolic function identities to rewrite this as

$$1 - \frac{\lambda}{2} \cdot \frac{e^\lambda+1}{e^\lambda-1} = \sum_{m=1}^{\infty} b_m \lambda^{2m}$$
$$1 - \coth\left(\frac{\lambda}{2}\right) = \sum_{m=1}^{\infty} b_m \lambda^{2m}$$

$$\implies \boxed{b_m = \frac{1}{m!}\left(\frac{d}{d\lambda}\right)^m \left[\frac{\sqrt{\lambda}}{2}\coth\left(\frac{\sqrt{\lambda}}{2}\right)\right]\Bigg|_{\lambda=0}} \tag{11}$$

(b) Calculating the first 10 coefficients, we get

$$b_1 = 1$$
$$b_2 = \frac{-1}{4}$$
$$b_3 = \frac{1}{6 \cdot 3!}$$
$$b_4 = 0$$
$$b_5 = \frac{-1}{30 \cdot 5!}$$
$$b_6 = 0$$
$$b_7 = \frac{1}{42 \cdot 6!}$$
$$b_8 = 0$$
$$b_9 = \frac{-1}{30 \cdot 9!}$$
$$b_{10} = 0$$

□

Problem 10

(a) Use the Euler-Maclaurin formula to show that

$$\sum_{j=1}^{n} j^k = P_{k+1}(n) \tag{12}$$

is a degree-$(k+1)$ polynomial in n. Example:

$$\sum_{j=1}^{n} j = \frac{n(n+1)}{2}.$$

(b) Use $b_2, \ldots, b_{10}$ to find P_{k+1} for $2 \leq k \leq 10$.

(c) Use polynomial interpolation to find P_{k+1} for $2 \leq k \leq 10$ and compare with the results from (b).

Solution
(a) Using the Euler-Maclaurin formula, we can express the (12) as

$$\sum_{j=1}^{n} j^k = \int_0^n f(x)dx + \frac{n^k}{2} + \sum_{m=1}^{\infty} b_m \left(f^{(2m-1)}(n) - f^{(2m-1)(0)} \right)$$

where $f(x) = x^k$. If we consider the terms of the sum on the right, we see that terms $2j - 1 > k$ vanish, so the degree of the sum is no greater than k in n. The second term is a polynomial of degree k in n. Finally, the integral give us a polynomial of degree $(k+1)$, evaluated at n. Thus, the right hand side of the equality is a polynomial of degree $(k+1)$ in n. $\square$

(b) We can use numbers calculated from the previous part to find P_{k+1} to get

$$\sum_{j=1}^{10} j^k = \frac{b_{k+1}(n+1) - b_{k+1}}{k+1}$$

where $b_1, \ldots, b_{10}$, are the calculated numbers from 9(b). The first few polynomials are as follows

$$\begin{aligned}
P_3 &= \frac{1}{4}n^4 + \frac{1}{2}n^3 + \frac{1}{6}n^2 \\
P_4 &= \frac{1}{5}n^5 + \frac{1}{2}n^4 + \frac{1}{3}n^3 + \frac{1}{30}n \\
P_5 &= \frac{1}{6}n^6 + \frac{1}{2}n^5 + \frac{5}{12}n^4 - \frac{1}{12}n^2 \\
P_6 &= \frac{1}{7}n^7 + \frac{1}{2}n^6 + \frac{1}{5}n^5 - \frac{1}{6}n^3 + \frac{1}{42}n
\end{aligned}$$

(c) We can then use polynomial interpolation to find P_{k+1} for $2 \leq k \leq 10$. Using MATLAB to find the coefficients of this interpolating polynomial, we find that it agrees at the roots with the result from (b). $\square$