Problem P1: Calculate $257^{1/8}$ to within 10^{-5} of the exact value without any computing machinery except a pencil or pen. Prove that your answer has this accuracy.

Solution:

Recall Taylor's Theorem with remainder. If $f(x)$ has $n+1$ continuous derivatives on the interval $[x, x+h]$ then the following holds,

$$f(x+h) = f(x) + f'(x)h + \frac{1}{2}f''(x)h^2 + ... + \frac{1}{n!}f^n(x)h^n + \frac{1}{(n+1)!}f^{n+1}(\xi)h^{n+1},$$

for some $\xi \in [x, x+h]$. To compute $257^{1/8}$ consider the function $f(x) = x^{1/8}$, let $x = 256$ with $h = 1$. Computing the first few terms in the sum we get,

$$256^{1/8} = 2,$$
$$\frac{1}{2}\left(\frac{1}{2}256^{-1/2}\right) = \frac{1}{64},$$
$$\frac{1}{3!}\left(-\frac{1}{4}256^{-3/2}\right) = \frac{-1}{(24)(16^3)} = -\frac{1}{98304}.$$

Note that expanding to the next term clearly gives us a value which is smaller that 10^{-5},

$$\frac{1}{4!}\left(\frac{3}{8}256^{-5/2}\right) = \frac{1}{(64)(16^5)} < 10^{-5}.$$

Therefore $257^{1/8}$ to within 10^{-5} is given by,

$$257^{1/8} \approx 2 + \frac{1}{64} - \frac{1}{98304}.$$

Problem P2: Assume f' is continuous. Derive the remainder formula,

$$\int_0^a f(x)dx = af(0) + \frac{1}{2}a^2 f'(v) \tag{1}$$

for some unknown v between zero and a. Use two sentences to explain the meaning of (1) as an approximation to the integral. That is, answer the question 'What properties of $f(x)$ or a make the left-endpoint rule $\int_0^a f(x)dx \approx af(0)$ more inaccurate?'

Solution:

Since f' is continuous we know by Taylor's Theorem with remainder that for some $\xi \in [0, x]$,

$$f(x) = f(0) + f'(\xi)x.$$

Integrating both sides, with respect to x we get,

$$\int_0^a f(x)dx = \int_0^a f(0) + f'(\xi)xdx = xf(0) + \frac{1}{2}x^2 f'(\xi)|_{x=0}^a = af(0) + \frac{1}{2}a^2 f'(\xi).$$

Geometrically, we see that the first term $af(0)$ represents the rectangular region which approximates the integral with the initial value of the function and the second term uses the shape of $f'(x)$ and the length a to adjust the approximation. Naturally as we would expect small values of a a constant looking $f(x)$ contribute to smaller remainder.

Problem P3: Work at the command line to compute a finite sum approximation to,

$$\sum_{n=1}^{\infty} \frac{\sin n}{n^3 + 1}.$$

Compute the partial finite sums for $N = 10$ and $N = 100$ terms. Turn your command-line work into a function mysum(N), and check that it works. Turn in both the command line session and the code. Speaking informally, how close do you think the $N = 100$ partial sum is to the infinite sum?

Solution:
The following is the command line session and mysum(N) function,
Console:

```
>> sum = 0;
>> for i = 1:10
   sum = sum + sin(i)/(i^3 + i);
   end

>> sum
   0.500404325899183


>> sum = 0;

>> for i = 1:100
   sum = sum + sin(i)/(i^3 + i);
   end

>> sum
   0.499839131456909
```

Code:

```
function sum = mysum(N)
    sum = 0;
    for i = 1:N
        sum = sum + sin(i)/(i^3 + i);
    end

end
```

Very informally speaking, we can crank up the number of iterations until we reach the limit of double precision arithmetic then compute an approximate error. Note that,

$$\frac{sin(n)}{n^3+1} \leq \frac{1}{n^3+1}.$$

Solving $\frac{1}{n^3+1} < \epsilon_{mach}$ for n will give is a usable lower bound on the number of iterations needed for the next term in the series to be smaller than ϵ_{mach}. Doing so we get around 160000 iterations. The error between $N = 100$ and the best double precision approximation is on the order of 10^{-6}.

Console:

```
>> mysum(160000)
   0.499840156251049

% Verifying best double precision approximation.
>> mysum(160001)
   0.499840156251049

% Computing N = 100 partial sum.
>> sum0 = mysum(100)
   0.499839131456909

% Computing Error
>> abs(sum0 - sum)
      1.024794139437013e-06
```

Problem P4: Solve, by hand,

$$y'' + y' - 6y = 0, y(2) = 0, y'(2) = -1,$$

for the solution $y(t)$. Then find $f(4)$. Give a reasonable by-hand sketch on t, y axes which shows the initial values, the solution, and the value $y(4)$.

Solution:
Note that this is a constant-coefficient, linear, homogeneous ODE so we begin by substituting $y(t) = e^{rt}$ to form the characteristic polynomial,

$$\begin{aligned} y'' + y' - 6y &= 0, \\ r^2e^{rt} + re^{rt} - 6e^{rt} &= 0, \\ e^{rt}(r^2 + r - 6) &= 0, \\ e^{rt}(r+3)(r-2) &= 0. \end{aligned}$$

Since our characteristic polynomial has roots $r = -3, 2$ we form the general solution with,

$$y(t) = c_1e^{-3t} + c_2e^{2t}.$$

Note that $y'(t)$ would then be of the form,

$$y'(t) = -3c_1e^{-3t} + 2c_2e^{2t}.$$

Using our initial conditions we get the following system,

$$\begin{aligned} 0 &= c_1e^{-6} + c_2e^4, \\ -1 &= -3c_1e^{-6} + 2c_2e^4. \end{aligned}$$

Multiplying the first equation by three and adding it to the second equation we get the following,

$$-1 = 5c_2e^4.$$

So we get that $c_2 = \frac{-1}{5e^4}$. Solving for c_1 we get, $c_1 = \frac{e^6}{5}$. Finally we get $y(t) = \frac{e^6}{5}e^{-3t} + \frac{-1}{5e^4}e^{2t}$.

Solving for $y(4) = \frac{1}{5e^6} - \frac{e^4}{5}$.

Problem P5: Using Euler's method for approximately solving ODEs, write your own program to solve the initial value problem in Problem 4 to find $y(4)$. A first step is to convert the second-order ODE into a system of two first-order ODEs. Use a few different step sizes, decreasing as needed so that you get apparent three-digit accuracy.

Solution:
As stated to proceed in converting the second-order ODE into a system of two first-order ODEs we will perform the following substitution,

$$y_1(t) = y(t)$$
$$y_2(t) = y'(t)$$

Note that differentiating both sides gives,

$$y_1' = y'$$
$$y_2' = y''$$

Finally by substitution we get the following system of first-order ODEs,

$$y_1' = y_2$$
$$y_2' = 6y_1 - y_2$$

with initial value of $y_1(2) = 0$ and $y_2(2) = -1$.

The following function uses Euler's method to approximate $y(t)$ and $y'(t)$. Experimenting with step sizes and comparing with the ode45() function in Matlab we get that to achieve an accuracy of three digits we need a step size on the order of 10^{-6}.

Code:

```
function [x,y,yp] = eulers2by2(dx,i,x0, xf)
% This function takes in a step size dx, an initial value vector i
% and a range of x0, xf values and uses Euler's Method to approximate
% y(t) and y'(t) defined by the second order IVP in Problem 4.

    %Allocating space for solution
    x = [x0:dx:xf]';
    [n, m] = size(x);
    y = zeros(n, 1);
    yp = zeros(n, 1);

    %Loading in initial values.
    y(1) = i(1);
    yp(1) = i(2);


    for i = 2:n
        %Compute x1' and x2'
```

```
        x1p = yp(i - 1);
        x2p = 6*y(i - 1) - yp(i - 1);
        %Euler's method: Use linear approximation to get next function value
        y(i) = y(i - 1) + dx*x1p;
        yp(i) = yp(i - 1) + dx*x2p;
    end

end
```

Console:

```
function dydt = vdp1(t,y)
    dydt = [y(2); 6*y(1) - y(2)];
end

>> [t,yblackbox] = ode45(@vdp1,[2 4],[0; -1]);

>> yblackbox(end,:)
 -10.919187009895570  -21.840853512402525


>> hist = [];
>> for i = 1:7
    [x,y,yp] = eulers2by2(10^(-i),[0, -1],2, 4);
    hist = [hist; y(end), yp(end)];
end

hist =

  -7.667360400362357  -15.335518723387690  % dx = .1
 -10.496527227539721  -20.995315696089442  % dx = .01
 -10.875605346271703  -21.753667191722364  % dx = .001
 -10.914768305768790  -21.832013133394916  % dx = .0001
 -10.918697530168801  -21.839873589432013  % dx = .00001
 -10.919090582280564  -21.840659894428672  % dx = .000001 <- Three digits
 -10.919129888788637  -21.840738527522866  % dx = .0000001
```

Problem P6: Solve, by hand, the ODE boundary value problem

$$y'' + 2y' - 3y = 0, y(0) = \alpha, y(\tau) = \beta$$

for the solution $y(t)$. Note that α, β and τ are the data fo the problem, so the solution will have these parameters in it.

Solution:

We proceed as before by substituting $y(t) = e^{rt}$ to form the characteristic polynomial,

$$\begin{aligned} y'' + 2y' - 3y &= 0 \\ r^2e^{rt} + 2re^{rt} - 3e^{rt} &= 0, \\ e^{rt}(r^2 + 2 - 3) &= 0, \\ e^{rt}(r+3)(r-1) &= 0. \end{aligned}$$

Since our characteristic polynomial has roots $r = -3, 1$ we form the general solution with,

$$y(t) = c_1e^{-3t} + c_2e^t.$$

With two equations and two unknowns we solve the system for c_1 and c_2,

$$\begin{aligned} \alpha &= c_1e^{-3(0)} + c_2e^{(0)}. \\ \tau &= c_1e^{-3(\beta)} + c_2e^{(\beta)}. \end{aligned}$$

The first equation simplifies to $\alpha = c_1 + c_2$, which gives $c_2 = \alpha - c_1$. By substitution we get

$$\begin{aligned} \tau &= c_1e^{-3(\beta)} + (\alpha - c_1)e^{(\beta)}, \\ \tau &= c_1e^{-3(\beta)} - c_1e^{(\beta)} + \alpha e^{(\beta)}, \\ \tau &= c_1(e^{-3(\beta)} - e^{(\beta)}) + \alpha e^{(\beta)}, \\ c_1 &= \frac{\tau - \alpha e^{(\beta)}}{e^{-3(\beta)} - e^{(\beta)}}, \end{aligned}$$

and,

$$c_2 = \alpha - \frac{\tau - \alpha e^{(\beta)}}{e^{-3(\beta)} - e^{(\beta)}}.$$

So our solution $y(t)$ is,

$$y(t) = \left(\frac{\tau - \alpha e^{(\beta)}}{e^{-3(\beta)} - e^{(\beta)}}\right)e^{-3t} + \left(\alpha - \frac{\tau - \alpha e^{(\beta)}}{e^{-3(\beta)} - e^{(\beta)}}\right)e^t.$$