Math 4NA3 - Assignment 2

February 8, 2021

NOTICE: This assignment is unfinished as I had to prioritize but I thought I'd hand in some rough work for partial credit.

Question 1:

a) Below is code for the first derivative approximation of $\sin(x)/x$ as well as the plots, but first a few notes. The stencils for i, ii, and iv are $1/2\{-3,2,-1\}, 1/2\{-1,1\}$, and $1/12\{1,-8,0,8,-1\}$. We also have for iii the following calculation.

$$\Im\left(\frac{\sin(x+ih)}{x+ih}\right)/h$$

$$= \frac{\Im(\sin(x+ih))\Re(x+ih) - \Im(x+ih)\Re(\sin(x+ih))}{h|x+ih|^2}$$

$$= \frac{x\cos(x)\sinh(h) - h\sin(x)\cosh(h)}{h(x^2+h^2)}$$

Lastly, the exact derivative is given by

$$\frac{x\cos(x) - \sin(x)}{x^2}$$

Now the code.

```
% setup
h = 0.2;
xs = -pi:h:pi;
ys  = arrayfun(@(x) sin(x)/x, xs);
n = length(xs);
forward2 = zeros(1,n-2);
center2 = zeros(1,n-2);
center4 = zeros(1,n-4);

% Exact derivative
exacts = arrayfun(@(x) (x*cos(x) - sin(x)) / (x^2), xs);

% Second-order forward
for i = 1:n-2
    forward2(i) = (-3*ys(i) + 4*ys(i+1) - ys(i+2))/(2*h);
end

% Second-order center
for i = 2:n-1
    center2(i-1) = (ys(i+1) - ys(i-1))/(2*h);
end

```

```
% Complex step
complex = arrayfun(@(x) (1/(h*x^2+h^3))*(x*cos(x)*sinh(h)-h*sin(x)*cosh(h)),xs);

% Fourth-order center
for i = 3:n-2
    center4(i-2) = (ys(i-2) - 8*ys(i-1) + 8*ys(i+1) - ys(i+2))/(12*h);
end

grid on
hold on
plot(xs,exacts,"k--",'DisplayName',"Exact")
plot(xs(1:n-2),forward2,"g",'DisplayName',"Forward 2nd")
plot(xs(2:n-1),center2,"b",'DisplayName',"Center 2nd")
plot(xs,complex,"c",'DisplayName',"Complex")
plot(xs(3:n-2),center4,"r",'DisplayName',"Center 4th")
hold off
lgd = legend;
```

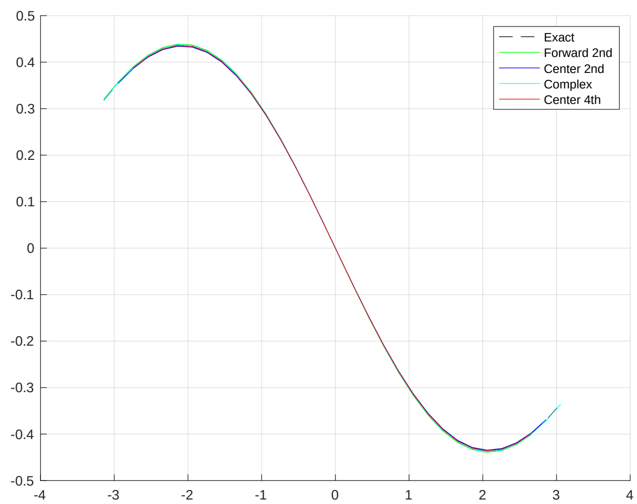

Figure 1: Zoomed-out view of all derivatives

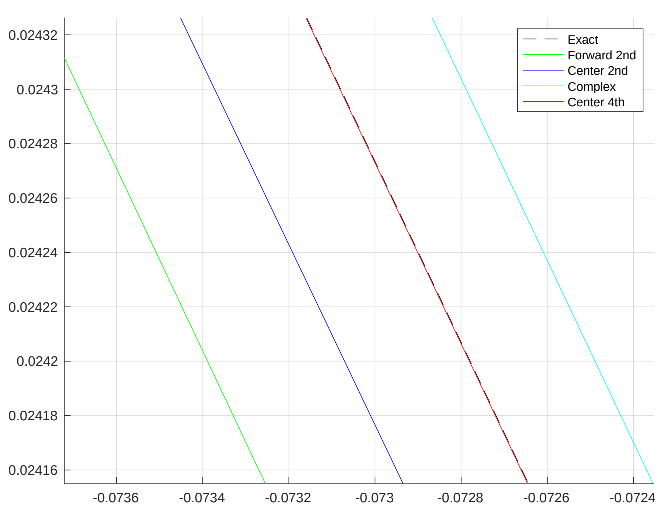

Figure 2: Zoomed-in view of all derivatives

From the first plot, we see that each approximation roughly matches the exact derivative. However, the second plot shows that the second-order forward difference is about twice the distance away from the exact curve as the complex step derivative and second-order center difference. In particular, the complex step derivative is about the same distance above the true curve as the second-order center difference is below the true curve. Lastly, the fourth-order center difference is the most accurate lying exactly on the curve at this particular level of zoom.

b) Here is the code for the error plot.

```
x0=pi/12;
h=logspace(-10,0,10);
exact = (x0*cos(x0) - sin(x0)) / (x0^2);
f2_error = zeros(1,10);
c2_error = zeros(1,10);
comp_error = zeros(1,10);
c4_error = zeros(1,10);

for i = 1:10
    f2 = (-3*cw(x0)+4*cw(x0+h(i))-cw(x0+2*h(i)))/(2*h(i));
    f2_error(i) = abs((f2-exact)/exact);

    c2  = (cw(x0+h(i))-cw(x0-h(i)))/(2*h(i));
    c2_error(i) = abs((c2-exact)/exact);

    comp = (1/(h(i)*x0^2+h(i)^3))*(x0*cos(x0)*sinh(h(i))-h(i)*sin(x0)*
        cosh(h(i)));
    comp_error(i) = abs((comp-exact)/exact);

    c4 = (cw(x0-2*h(i))-8*cw(x0-h(i))+8*cw(x0+h(i))-cw(x0+2*h(i)))/(12*
        h(i));
    c4_error(i) = abs((c4-exact)/exact);
end

loglog(h,f2_error,'DisplayName',"Forward2");
hold on
grid on;
loglog(h,c2_error,'DisplayName',"Center2");
loglog(h,comp_error,'DisplayName',"Comp");
loglog(h,c4_error,'DisplayName',"Center4");
xlabel('Step size','fontsize',16);
ylabel('Relative error','fontsize',16);
legend
hold off

function y = cw(x)
```

```
y = sin(x)/x;
end
```

The error plot shows that the second-order forward and center differences achieve relative errors of about 10^{-10} at a step size just above 10^{-4}. Below that, we see the fourth-order center difference performing better with a relative error of about 10^{-12} at a step size just below 10^{-4}. Lastly, the lowest relative error of about machine precision or 10^{-16} is achieved by the complex step method at a step size of 10^{-8}. As the step size decreases, second-order forward and center and complex step reduce error at about the same rate, however, the complex step error continues to reduce whereas the others suffer from roundoff error and eventually increase. The fourth-order center also eventually increases, however, its rate is much faster than the others.

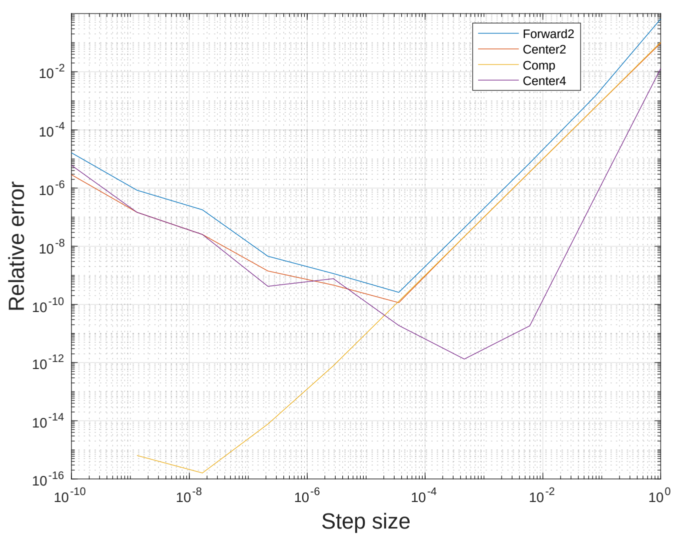

Figure 3: Relative error v.s. Step size

Question 2:

Here is a function that performs central difference Richardson extrapolation to approximate the second derivative of $f(x) = \text{sech}^2(x)$ at $x = 1$; it was altered from the one on avenue for the required purposes. Note the second derivative is $f''(x) = 4\text{sech}^2(x)\text{tanh}^2(x) - 2\text{sech}^4(x)$ and its value at $x = 1$ is about 0.621627.

```matlab
function Dnum = CentRichExtrappp(x0, maxRich, h, tol)
% Evaluates Richardson extrapolations of central difference approximation
% for the second derivative of y = sech(x)^2 up to order maxRich while
% error between successive approximations is less than tol and plots
% the error.
%
% Input variables:
% x0      = point at which to approximate derivative
% maxRich = maximum order of Richardson extrapolation (determines size of grid)
% h       = step size
% tol     = tolerance
%
% Output variable:
% Dnum - row-vector of central Richardson extrapolations
%
x    = x0 + h*2.^(0:maxRich-1);
xNeg = x0 - h*2.^(0:maxRich-1);
xs = x0 + zeros(1,length(x));
y    = sech(x).^2;
yNeg = sech(xNeg).^2;
y0 = sech(x0)^2;
ys = y0 + zeros(1,length(y));

yprime = 4*y0*tanh(x0)^2-2*y0^2;
D = ones(maxRich, maxRich);
D(:,1) = (y' + yNeg' - 2.*ys')./(((x' - xNeg')./2).^2);

for k = 2 : maxRich
    for kk = 1 : (maxRich-k+1)
        D(kk,k) = D(kk,k-1)+(D(kk,k-1) - D(kk+1,k-1))/(4^(k-1)-1);
    end
    if abs(real(D(1,k))-real(D(1,k-1))) < tol
        i = k
        break
    end
end
```

```
i
Dnum = real(D(1,1:i));
loglog(abs(Dnum-yprime),'o-');grid on;
xlabel('Iteration');ylabel('Absolute Error');
end
```

We now apply the function.

```
Dnum = CentRichExtrappp(1,10,5e-2, 10^-8)
```

```
i =
     5

Dnum =
    0.6205    0.6216    0.6216    0.6216    0.6216
```

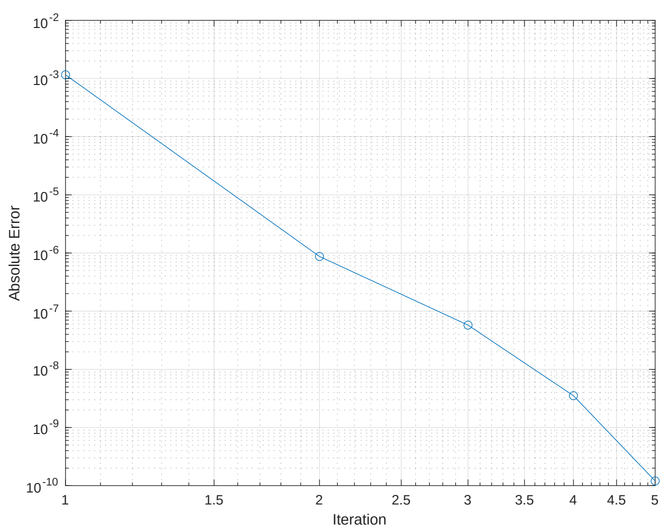

Figure 4: Center difference Richardson extrapolation

Question 3:

Here is the code I used to do the integrations, plot the error, and the power fit.

```
exact = 0.4;
n = 1:100;
for i = n
    h = 1/i;
    hs(i) = h;
    xs = 0:h:1;

    % Trap
    tys = arrayfun(@(x) x^(3/2), xs);
    trap_error(i) = abs(trapz(xs,tys) - exact);

    % Mid
    for k = 1:length(xs)-1
        xms(k) = (xs(k)+xs(k+1))/2;
    end
    mys = arrayfun(@(x) x^(3/2), xms);
    mid_error(i) = abs(h*sum(mys)-exact);

    % Simp
    sys = ys(2:end-1);
    odds = sum(sys(1:2:end));
    evens = sum(sys(2:2:end));
    (h/3)*(4*odds+2*evens);
    simp_error(i) = abs((h/3)*(4*odds+2*evens) - exact);
end


grid on
hold on
plot(hs,trap_error,"k*",'DisplayName',"Trap error")
ft = polyfit(log(hs),log(trap_error),1);
pt = ft(1);
Ct = exp(ft(2));
fplot(@(x) Ct*x.^pt,[hs(end) hs(1)],'DisplayName',"Trap fit")


plot(hs,mid_error,"c*",'DisplayName',"Mid error")
fm = polyfit(log(hs),log(mid_error),1);
pm = fm(1);
Cm = exp(fm(2));
fplot(@(x) Cm*x.^pm,[hs(end) hs(1)],'DisplayName',"Mid fit")
```

```
plot(hs,simp_error,"r*",'DisplayName',"Simp error")
fs = polyfit(log(hs),log(simp_error),1);
ps = fs(1);
Cs = exp(fs(2));
fplot(@(x) Cs*x.^ps,[hs(end) hs(1)],'DisplayName',"Simp fit")

hold off
lgd = legend;
```

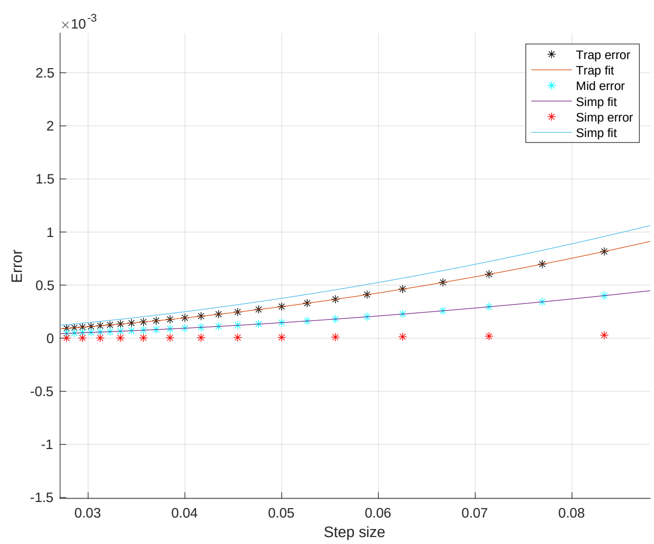

Figure 5: Function $x^{3/2}$

Question 4:

a) We have

$$\int_0^1 e^{-x}/x^{3/4}dx$$

For a change of variables, we set $x^{3/4} = t^3 \implies x = t^4$. This removes the singularity giving us

$$4\int_0^1 e^{-t^4}dt$$

```
 exact = 3.37935437902841;
n = 2:100;
for i = n
     h = 1/i;
     hs(i-1) = h;
     xs = 0:h:1;

     ys = arrayfun(@(x) 4*exp(-x^4),xs);

     trap_error(i) = abs(trapz(xs,ys) - exact);

end
grid on
hold on
loglog(hs,trap_error, 'DisplayName',"Trap")
xlabel('Step size');
ylabel('Error');
lg = legend
```

Question 5:

```
function Dnum = CentRichExtrappp(x0, maxRich, h)
x    = x0 + h*2.^(0:maxRich-1);
xNeg = x0 - h*2.^(0:maxRich-1);

y    = x-x.^3+2.*x.^5-x.^7;
yNeg = xNeg-xNeg.^3+2.*xNeg.^5-xNeg.^7.^2;

yprime = 1-3*x0^2+10*x0^4-7*x0^6;
D = ones(maxRich, maxRich);
D(:,1) = (y' - yNeg')./(x' - xNeg');

for k = 2 : maxRich
    for kk = 1 : (maxRich-k+1)
        D(kk,k) = D(kk,k-1)+(D(kk,k-1) - D(kk+1,k-1))/(4^(k-1)-1);
    end
end
i
Dnum = real(D(1,1:i));
loglog(abs(Dnum-yprime),'o-');grid on;
xlabel('Iteration');ylabel('Absolute Error');
end
```