CS/SE 4X03 — Assignment 3

Ned Nedialkov

3 March 2019

Due date: 13 March, 1:30pm, in class.

- Without MSAF, no assignments will be accepted after 1:35pm on the 13th.
- With MSAF, no assignments will be accepted after 1:35pm on the 18th.

The SVN submissions must be under subdirectory with name `A3` of your main directory.

Problem 1 (4 points) Consider the region $-1 \le x \le 1$, $-1 \le y \le 1$ in 2D. If there is a uniform charge distribution in this region, the electrostatic potential at a point $(\widehat{x}, \widehat{y})$ outside this region is given by

$$\Phi(\widehat{x}, \widehat{y}) = \int_{-1}^{1} \int_{-1}^{1} \frac{dxdy}{\sqrt{(\widehat{x} - x)^2 + (\widehat{y} - y)^2}}. \tag{1}$$

One can easily evaluate this integral in Matlab using `dblquad`. For this problem, you are not allowed to use `dblquad`.

Write a function

```
function f = phi(xhat, yhat)
%Computes electrostatic potential at xhat, yhat
```

that computes (1). This function must call adaptive Simpson integration (you can use the function I posted or a modification of it). To make this work, study `https://www.mathworks.com/help/matlab/math/parameterizing-functions.html`

Then execute the script

```
close all; clear
%Plots the surface of phi
[X,Y] = meshgrid(2:.5:12);
Z = phi(X,Y);
surface(X,Y,Z)
view(135,30);
xlabel('x')
ylabel('y')
zlabel('\phi(x,y)')
```

Submit

- hard copy: the produced plot and `phi.m`
- SVN: `phi.m`

Problem 2 (11 points)

(a) (3 points) Implement the composite midpoint, trapezoid and Simpson in functions

```
function [Q] = midpoint( f, a, b, n )
%Approximates the integral of f on [a,b] using the
%composite midpoint rule and n equal subintervals.
%The approximated value is returned in Q.
```

```
function [Q] = trapezoid( f, a, b, n )
%Approximates the integral of f on [a,b] using the
%composite trapezoid rule and n equal subintervals.
%The approximated value is returned in Q.
```

```
function [Q] = simpson( f, a, b, n )
%Approximates the integral of f on [a,b] using the
%composite simpson rule and n equal subintervals.
%n must be even
```

You **must not use any loops** in these functions.

(b) (4 points) Consider $\int_1^2 e^x/x\,\mathrm{d}x$. Apply each of the above methods to evaluate this integral for $n = 2^4, 2^5, \ldots, 2^{10}$. Plot in the same figure using **loglog** the error versus $h = 1/n$ for each of the three methods, where $n = 2^4, 2^5, \ldots, 2^{10}$. What conclusions can you make from this plot?

(c) (4 points) The error in each of these methods is of the form ch^p. Describe how you can determine the constants c and p. Then implement the function

```
function [c,p] = findconstants(rule,f,a,b)
%Computes the constants in the error term c*h^p
%of an integration rule.
%Input
%rule function implementing a composite rule
%f function to be integrated on [a,b]
%Output
% c, p the constants in c*h^p
```

It should work as e.g. in

```
f = @(x) exp(x)./x;
a = 1; b = 2;
[c,p] = findconstants(@midpoint, f, a,b);
fprintf('midpoint  %f  %f\n', c, p);
[c,p] = findconstants(@trapezoid, f, a,b);
fprintf('trapezoid %f  %f\n', c, p);
[c,p] = findconstants(@simpson, f, a,b);
fprintf('Simpson   %f  %f\n', c, p);
```

Submit

- hard copy:
  (a) `midpoint.m`, `trapezoid.m`, `simpson.m`
  (b) plot and discussion
  (c) method description, `findconstants.m`, and the output of `findconstants` for the three rules on $\int_1^2 e^x/x\,\mathrm{d}x$

- SVN: `findconstants.m`

Problem 3 (5 points) The error function is defined as

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{2}$$

(see e.g. `https://en.wikipedia.org/wiki/Error_function`). Determine the smallest number of subintervals n such that if you apply the composite midpoint rule to compute erf(1) the error is

$$|Q - \operatorname{erf}(1)| \leq 10^{-10},$$

where Q denotes the value computed by the composite midpoint rule.

The matlab function **erf** computes (2). Verify numerically that your n is correct. That is check the output of

```
f = @(x)   2/(sqrt(pi))*exp(-x.^2);
a = 0; b = 1;
n = 100;  %put your value here
Q = midpoint(f,a,b,n)
err = abs(Q-erf(1))
```

Submit

- hard copy: your calculation of n and the computed error
- SVN: nothing

Problem 4 (10 points) You are given the data file
`http://www.cas.mcmaster.ca/~nedialk/COURSES/4X03/nbody.dat`
with positions of Jupiter, Saturn, Uranus, Neptune and Pluto.

- The first column is time. Each time unit is 100 days. This file contains data up to time 5000, which gives $5000 \times 100/365 \approx 1369,86$ years.
- Columns 2, 3, 4 contain the coordinates (x, y, z) of Jupiter, then next three columns contain the coordinates of Saturn, and so on. Distance is measured from the sun in astronomical units (AU), where 1 AU is the mean radius of the earth's orbit.

A general quadratic curve can be written as

$$ay^2 + bxy + cx + dy + e = x^2. \tag{3}$$

A planet's orbit is an ellipse in the (x, y) plane.

(a) (5 points) For each of the planets, determine the coefficients a, b, c, d, e in (3) and report them in a table

planet	a	b	c	d	e
Jupiter					
Saturn					
Uranus					
Neptune					
Pluto					

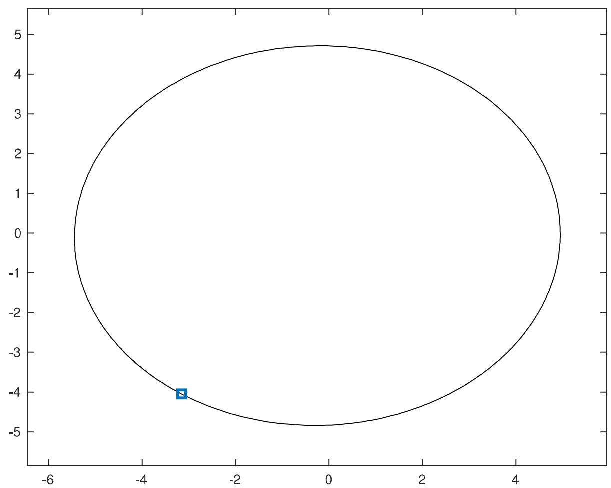
(b) (5 points) For a given planet, if `x` contains the x values, and `y` contains the y values from the data file, then you can plot the (x, y) satisfying (3) using

```
[xs, ys] = meshgrid(min(x)-1:0.1:max(x)+1, min(y)-1:0.1:max(y)+1);
contour(xs, ys, a*ys.^2+b*xs.*ys+c*xs+d*ys+e-xs.^2, [0, 0], 'k-');
```

where `a`, `b`, `c`, `d`, `e` are your computed values.

Describe how you can determine (accurately) the position of a planet at time $t \in [0, 5000]$, where the value for t is not in the data file.

Then, after executing the above, plot the position of a planet at time $t = 500$ by putting a maker:



For all the planets, plot in the same figure the ellipses and positions at time $t = 500$. That is, you should have 5 ellipses and 5 markers.

Submit

- hard copy: the above table and plot.
- SVN: nothing