Assignment 6 - Numerical Integration
COMP 350 - Numerical Computing
Prof. Chang Xiao-Wen
Fall 2018

LE, Nhat Hung
McGill ID: 260793376
Date: November 28, 2018
Due date: November 29, 2018

1. (a) (4 points) Using the recursive trapezoid rule to compute $\int_0^{2\pi} cos(2x)/e^x dx$. Stop the iteration until the difference between two consecutive computed integrals is smaller than or equal to 10^{-4}.

(b) (6 points) Using the adaptive Simpson's method to compute $\int_0^{2\pi} cos(2x)/e^x dx$ by taking $\varepsilon = 10^{-4}$ and `level_max=20`. Try to avoid redundant function evaluation.

For both methods, report the number of function evaluations and print the final results and the MATLAB codes as well.

Note: The exact integral is $(1 - e^{-2\pi})/5$. You can use this to check if your answer is reasonable.

```
Recursive trapzoid:
Number of iterations: 9
Result:   1.996390382143591e-01

Adaptive Simpson:
Number of iterations: 19
Result:   1.996245972017476e-01

Real result:   1.996265114536584e-01
```

Figure 1: Answer to (a) and (b)

ass6.m:

```
f = @(x) cos(2*x)/exp(x);

disp('Recursive trapzoid:');
I_T = recTrapezoid(f, 0, 2*pi, 100000, 10^(-4));
fprintf('Result: %23.15e\n\n', I_T);

global count
count = 0;
disp('Adaptive Simpson:');
I_S = adaptiveSimpson(f, 0, 2*pi, 10^(-4), 0, 20);
fprintf('Number of iterations: %d\n', count);
fprintf('Result: %23.15e\n\n', I_S);

fprintf('Real result: %23.15e\n', (1 - exp(-2*pi))/5);
```

recTrapezoid.m:

```
function I = recTrapezoid(fname, a, b, n, tol)
%RECTRAPEZOID Summary of this function goes here
%   fname: name of function
%   a: left endpoint of [a,b]
%   b: right endpoint of [a,b]
%   n: max num of iterations
%   tol: gap between I and previous I

m = 1;
h = b - a;
diff = realmax;
T = zeros(1,n+1);
T(1) = h*(feval(fname,a) + feval(fname,b))/2;

for i = 1:n
    m = 2 * m;
    h = h / 2;
    s = 0;
    for j = 1 : m / 2
        x = a + h * (2 * j - 1);
        s = s + feval(fname,x);
    end
    T(i + 1) = T(i) / 2 + h * s;
    I = T(i + 1);
    diff = abs(T(i) - T(i + 1));

    if diff <= tol, break; end
end

fprintf('Number of iterations: %d\n', i);
```

adaptiveSimpson.m:

```
function I = adaptiveSimpson(fname,a,b,delta,level,level_max)
%ADAPTIVESIMPSON Summary of this function goes here
%   Detailed explanation goes here

global count
count = count + 1;

h = b - a;
c = (a + b)/2;
I1 = h*(feval(fname,a) + 4*feval(fname,c) + feval(fname,b)) / 6;
level = level + 1;
d = (a + c)/2;
```

```
e = (c + b)/2;
I2 = h*(feval(fname,a) + 4*feval(fname,d) + 2*feval(fname,c) +
4*feval(fname,e) + feval(fname,b))/12;

if level >= level_max
    I = I2;
else
    if abs(I2-I1) <= 15*delta
        I = I2 + (I2-I1)/15;
    else
        I = adaptiveSimpson(fname,a,c,delta/2,level,level_max)...
            + adaptiveSimpson(fname,c,b,delta/2,level,level_max);
    end
end
```

2. (a) (6 points) Construct a rule of the form

$$\int_{-1}^{1} f(x)x^2 dx \approx af(-\alpha) + bf(0) + cf(\alpha)$$

such that it is exact for all polynomials of as high a degree as possible.

Hint: Use one of the approaches we used in class to drive the Gaussian quadrature rule for n = 2.

Let $g(x) = f(x)x^2$.

$$\int_{-1}^{1} g(x)\mathrm{d}x = ag(x_0) + bg(x_1) + cg(x_2)$$

Take

$$f(x) = x^j.$$

Then,

$$g(x) = f(x)x^2 = x^{j+2},\ j+2 = 0:m.$$

We then have

$$\int_{-1}^{1} x^{j+2} = A_0(x_0)^{j+2} + A_1(x_1)^{j+2} + A_1(x_2)^{j+2}$$

$$m+1 \text{ equations,}$$

$$2n+2 = (2)(3)+2 = 7 \text{ unknowns, with } n = 3.$$

Want

$$m+1 \le 7$$

$$\Rightarrow m \le 6 \Rightarrow j+2 = 0:6 \Rightarrow j = -2,-1,0,1,2,3$$

Knowing

$$\int_{-1}^{1} g(x)\mathrm{d}x = \int_{-1}^{1} x^{j+2}\mathrm{d}x = \frac{1}{j+3}x^{j+3}\ \Big|_{-1}^{1}\ ,$$

we have the following system of 6 equations:

$$\begin{array}{rcrcc}
j=-2: & & A_0 + A_1 + A_2 & = & 2 \\
j=-1: & & A_0x_0 + A_1x_1 + A_2x_2 & = & 0 \\
j=0: & & A_0(x_0)^2 + A_1(x_1)^2 + A_2(x_2)^2 & = & 2/3 \\
j=1: & & A_0(x_0)^3 + A_1(x_1)^3 + A_2(x_2)^3 & = & 0 \\
j=2: & & A_0(x_0)^4 + A_1(x_1)^4 + A_2(x_2)^4 & = & 2/5 \\
j=3: & & A_0(x_0)^5 + A_1(x_1)^5 + A_2(x_2)^5 & = & 0
\end{array}$$

This is the well known system for 3-point Gauss quadrature. Solving it yields

$$\begin{cases} A_0 = 5/9 \\ A_1 = 8/9 \\ A_2 = 5/9 \\ x_0 = -\sqrt{3/5} \\ x_1 = 0 \\ x_2 = \sqrt{3/5} \end{cases}$$

We then have

$$\begin{aligned}
\int_{-1}^{1} g(x)\mathrm{d}x \quad &= \frac{5}{9}g\left(-\sqrt{\frac{3}{5}}\right) + \frac{8}{9}g(0) + \frac{5}{9}g\left(\sqrt{\frac{3}{5}}\right) \\
&= \frac{5}{9}\left(-\sqrt{\frac{3}{5}}\right)^2 f\left(-\sqrt{\frac{3}{5}}\right) + \frac{8}{9}(0^2)f(0) + \frac{5}{9}\left(\sqrt{\frac{3}{5}}\right)^2 f\left(\sqrt{\frac{3}{5}}\right) \\
&= \frac{1}{3}f\left(-\sqrt{\frac{3}{5}}\right) + 0f(0) + \frac{1}{3}f\left(\sqrt{\frac{3}{5}}\right)
\end{aligned}$$

In conclusion,

$$\int_{-1}^{1} f(x)x^2dx \approx \frac{1}{3}f\left(-\sqrt{\frac{3}{5}}\right) + 0f(0) + \frac{1}{3}f\left(\sqrt{\frac{3}{5}}\right)$$

$$\text{with } a = c = \frac{1}{3},\ b = 0 \text{ and } \alpha = \sqrt{\frac{3}{5}}$$