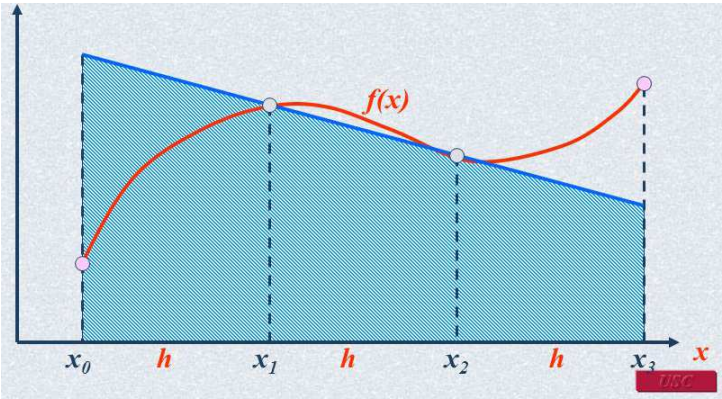
Homework – Quadrature Due Wed. 3/21/18

1. (25 pts) a. We begin with a diagram showing the Open Newton-Cotes Trapezoid Method.



Our theorem for Open Newton-Cotes quadratures with $n = 1$ gives us the formula:

$$\int_a^b f(x)dx = \sum_{i=0}^{1} a_i f(x_i) + \frac{h^3 f''(\xi)}{2!} \int_{-1}^{2} t(t-1)dt.$$

From our Lagrange interpolating polynomials, we have

$$a_0 = \int_a^b \frac{(x-x_1)}{(x_0-x_1)}dx \quad \text{and} \quad a_1 = \int_a^b \frac{(x-x_0)}{(x_1-x_0)}dx,$$

where $h = \frac{b-a}{3}$, $x_0 = a + h$, $x_1 = b - h$, and $h = x_1 - x_0$. Integrating we find

$$a_0 = \int_a^b \frac{-(x-x_1)}{h}dx = \left[\frac{-(x-x_1)^2}{2h}\right]\Big|_{x_0-h}^{x_1+h} = \left[\frac{-h^2}{2h} + \frac{(-2h)^2}{2h}\right] = \frac{3h}{2}.$$

Similarly,

$$a_1 = \frac{3h}{2}.$$

It follows that

$$\sum_{i=0}^{1} a_i f(x_i) = \frac{3h}{2}\left(f(x_0) + f(x_1)\right)$$

The error term is

$$\frac{h^3 f''(\xi)}{2!} \int_{-1}^{2} t(t-1)dt = \frac{h^3 f''(\xi)}{2}\left[\frac{t^3}{3} - \frac{t^2}{2}\right]\Big|_{-1}^{2} = \frac{h^3 f''(\xi)}{2}\left(\frac{3}{2}\right) = \frac{3h^3 f''(\xi)}{4}.$$

Combining these results gives the open Newton-Cotes Trapezoid formula of

$$\int_a^b f(x)dx = \frac{3h}{2}\left(f(x_0) + f(x_1)\right) + \frac{3h^3 f''(\xi)}{4}.$$

b. Begin by dividing the interval $[a,b]$ into $3N$ evenly spaced subintervals with $h = \frac{b-a}{3N}$. With the formula from Part a, we can write the integral as follows:

$$\int_a^b f(x)dx = \sum_{i=0}^{N} \left(\frac{3h}{2} \left(f(x_{i0}) + f(x_{i1}) \right) + \frac{3h^3 f''(\xi_i)}{4} \right),$$

where $x_{i0} = a + h(1+3i)$, $x_{i1} = a + h(2+3i)$, and $\xi_i \in [x_{i0} - h, x_{i1} + h]$. The **error term** for this integral approximation is the last term. If we assume that f is at least twice differentiable for $x \in [a,b]$, then the Extreme Value Theorem gives that $f''(x)$ has its max and min in $[a,b]$, so

$$\min_{x\in[a,b]} f''(x) \leq f''(\xi_j) \leq \max_{x\in[a,b]} f''(x),$$

so

$$N \min_{x\in[a,b]} f''(x) \leq \sum_{j=1}^{N} f''(\xi_j) \leq N \max_{x\in[a,b]} f''(x) \quad \text{or} \quad \min_{x\in[a,b]} f''(x) \leq \left(\frac{1}{N}\right) \sum_{j=1}^{N} f''(\xi_j) \leq \max_{x\in[a,b]} f''(x),$$

By the Intermediate Value Theorem there exists $\mu \in (a,b)$ so that

$$f''(\mu) = \frac{1}{N} \sum_{j=1}^{N} f''(\xi_j) \quad \text{or} \quad N f''(\mu) = \sum_{j=1}^{N} f''(\xi_j).$$

However, $N = \frac{b-a}{3h}$, so the error term becomes

$$\frac{3h^3}{4} \sum_{i=0}^{N} f''(\xi_i) = \frac{3h^3}{4} N f''(\mu) = \frac{h^2(b-a)}{4} f''(\mu),$$

which implies this method is $\mathcal{O}\left(h^2\right)$.

c. We begin with the exact value, so

$$\int_0^6 e^x \, dx = 402.42879.$$

Below is a composite open Newton-Cotes Trapezoid method MatLab code:

```
function T = comptrap1c(a,b,N)
% Composite Trapezoid Rule for function f(x)
% on [a,b] using 3*N steps
f = @(x) exp(x);
h = (b-a)/(3*N);
i = 0:N-1;
xi0 = a+h*(1 + 3*i);
xi1 = a+h*(2 + 3*i);
T = (3*h/2)*sum(f(xi0)+f(xi1));
end
```

h	Trap	$abs-err$	err/h	err/h^2	err/h^3
1	319.67797	82.75082	82.75082	82.75082	82.75082
0.5	378.56793	23.86086	47.72172	95.44343	190.88687
0.25	396.22483	6.20396	24.81585	99.26342	397.05367
0.125	400.86211	1.56668	12.53347	100.26778	802.14221

From this table above, we see that this numerical integration method is $\mathcal{O}\left(h^2\right)$. The column with err/h^2 is roughly constant.

d. Adapting the code above for the Trapezoid Method to include a `while` loop to test for the difference between successive iterates, we obtain the following MatLab code:

```
function [T,h] = comptrap1ctol(a,b,tol)
% Composite Trapezoid Rule for function f(x)
% on [a,b] using 3*N steps with a tolerance
f = @(x) exp(x);
T0 = 0;
T = 1;
N = 1;
j = 0;
while (abs(T-T0) > tol)
    T0 = T;
    h = (b-a)/(3*N);
    i = 0:N-1;
    xi0 = a+h*(1 + 3*i);
    xi1 = a+h*(2 + 3*i);
    T = (3*h/2)*sum(f(xi0)+f(xi1));
    j = j + 1;
    N = 2^j;
end
```

With this code, we find that starting with $h = 2$ the stepsize through halving must be reduced to $h = 3.0517578125 \times 10^{-5}$, yielding an approximation to the original integral of $T = 402.428793399$.

2. (20 pts) a. We use our techniques from Calculus to write

$$\int_{-1}^{1} \frac{2}{1+x^2}\,dx = 2\arctan(x)\Big|_{-1}^{1} = 2\left(\frac{\pi}{4} - \frac{-\pi}{4}\right) = \pi.$$

b. **Composite midpoint** with $h = 0.5$:

$$A_M = \frac{1}{2}\left(f\left(-\frac{3}{4}\right) + f\left(-\frac{1}{4}\right) + f\left(\frac{1}{4}\right) + f\left(\frac{3}{4}\right)\right) = 3.162352941.$$

Absolute error $= 0.020760288$.

Trapezoid with $h = 0.5$:

$$A_T = \frac{1}{4}\left(f(-1) + 2f(-0.5) + 2f(0) + 2f(0.5) + f(1)\right) = 3.1.$$

Absolute error $= 0.041592654$.

Simpson's with $h = 0.5$:

$$A_S = \frac{1}{6}\left(f(-1) + 4f(-0.5) + 2f(0) + 4f(0.5) + f(1)\right) = 3.13333333.$$

Absolute error $= 0.00825932$.

c. We use our Legendre program to obtain the points and weights for the Gaussian quadrature. Below are our approximations and absolute errors:

Gaussian 4 point:

$$\begin{aligned} A_{G4} &= 0.3478548451 f(-0.8611363116) + 0.6521451549 f(-0.3399810436) \\ &\quad + + 0.6521451549 f(0.3399810436) + 0.3478548451 f(0.8611363116) \\ &= 3.1372549020. \end{aligned}$$

Absolute error = 0.004337752 .

Gaussian 5 point:

$$\begin{aligned} A_{G5} &= 0.2369268851 f(-0.9061798459) + 0.4786286705 f(-0.5384693101) \\ &\quad +0.5688888889 f(0) + 0.4786286705 f(0.5384693101) \\ &\quad +0.2369268851 f(0.9061798459) = 3.1423423423. \end{aligned}$$

Absolute error = 0.000749689 .

Since the Gaussian quadrature is optimal it does substantially better than the other routines. We see that the ordering of accuracy from worst to best is Trapezoid rule, Midpoint rule (which is a composite Gaussian with 1 point), Simpson's rule, then Gaussian with 4 point and 5 point being the best. None of the methods is doing particularly well, but we have over an order of magnitude improvement with the Gaussian routines.

d. We decrease the stepsize in half and the Table below shows us the results.

Method	Midpoint	Trapezoid	Simpson's	Gaussian (9)
Integral	3.146800518	3.131176471	3.141568627	3.14159331186
Error	0.005207865	0.010416183	2.40261×10^{-5}	6.58268×10^{-7}

Halving the stepsize quarters the error in the Midpoint and Trapezoid rules, while Simpson's rule improves by over 2 orders of magnitude with only doubling the points, which is quite substantial. However, the Gaussian quadrature increased its accuracy over 3 orders of magnitude with less than doubling the points. Thus, it proves to be an excellent method of approximate integration.

3. (10 pts) a. Students can modify the `AdaptiveCSR.m` in a number of ways. The `ACSR.m` is not changed. Below shows a number of lines of the `AdaptiveCSR.m` that would satisfy this problem.

```
%
% Adaptive CSR
%
% This is the setup/driver part...
% Play with the tolerance for different results.
%

tol = 10^(-6);
a   = 0;
b   = 1;
```

```
11
12 f = @(x) 1-((x-pi/2/exp(1)).^2).^(1/3);
13
14 figure(1); xv = 0:0.001:1;
15 plot(xv,f(xv),'-','linewidth',3)
16 title('Adaptive CSR --- The Function','fontweight','bold','fontsize',14)
17 axis([0 1 0 1])
18 xlabel('x','fontweight','bold','fontsize',14)
19 ylabel('f(x)','fontweight','bold','fontsize',14)
```

b. Students may choose any number of possible integrals. Probably the easiest would be

$$\int_0^1 \frac{dx}{x^p},$$

where $0 < p < 1$. For example, if $p = \frac{1}{2}$, then this integral is 2, but it is singular at $a = 0$.

WeBWorK problems in Quad

1. No write-up for this problem.

2. (8 pts) b. Consider the function:

$$f(x) = x^2 \cos(1.2x) - 2.7x.$$

This has the linear interpolating polynomial through $x_0 = 0.1$ and $x_1 = 0.2$ giving

$$P_1(x) = -2.4107x - 0.018997.$$

At $x = 0.18$, the absolute error is 0.00142. This has the quadratic interpolating polynomial through $x_0 = 0.1$, $x_1 = 0.2$, and $x_2 = 0.3$ giving

$$P_2(x) = 0.82259x^2 - 2.6575x - 0.0025456.$$

At $x = 0.18$, the absolute error is 0.0001052.

The error for the linear polynomial is

$$E_1(x) = \frac{f''(\xi)}{2}(x - 0.1)(x - 0.2), \quad \xi \in [0.1, 0.2],$$

and for the quadratic polynomial is

$$E_2(x) = \frac{f'''(\xi)}{6}(x - 0.1)(x - 0.2)(x - 0.3), \quad \xi \in [0.1, 0.3].$$

It can be shown for $\xi \in [0.1, 0.2]$ that $|f''(\xi)| \leq 1.914$. In this interval, $|(x - 0.1)(x - 0.2)| \leq 0.0025$, so we have

$$\left| \frac{f''(\xi)}{2}(x - 0.1)(x - 0.2) \right| \leq 0.0024,$$

which is roughly twice the absolute error for $|f(x) - P_1(x)|$ at $x = 0.18$.

It can be shown for $\xi \in [0.1, 0.3]$ that $|f'''(\xi)| \leq 4.91$. In this interval, $|(x-0.1)(x-0.2)(x-0.3)| \leq 0.000385$, so we have

$$\left| \frac{f'''(\xi)}{6}(x-0.1)(x-0.2)(x-0.3) \right| \leq 0.00032,$$

which is roughly three times the absolute error for $|f(x) - P_2(x)|$ at $x = 0.18$.

3. (5 pts) For the integral

$$\int_3^5 \frac{1.9x}{x^2+1}\,dx$$

the exact value is 0.907735872776064.

With the MatLab Midpoint rule program below, we find

```
function M = compmidpt4(a,b,N)
% Composite Midpoint Rule for function f(x)
% on [a,b] using N steps
f = @(x) 1.9*x./(x.^2+1);
h = (b-a)/N;
i = 1:N;
ci = a+0.5*(2*i-1)*h;
M = h*sum(f(ci));
end
```

h	M	$abs-err$	err/h	err/h^2	err/h^3	err/h^4
0.5	0.9068568	0.0008791	0.001758	0.003516	0.007033	0.01407
0.25	0.9075158	0.0002201	0.0008803	0.003521	0.01408	0.05634
0.125	0.9076808	5.5073E-05	0.00044058	0.003525	0.02820	0.2256
0.0625	0.9077221	1.3773E-05	0.0002204	0.003526	0.05641	0.90266

4. (5 pts) For the integral

$$\int_1^3 \frac{2.7x}{x^2+25}\,dx$$

the exact value is 0.362156381902817.

With the MatLab Trapezoid rule program below, we find

```
function T = comptrap5(a,b,N)
% Composite Trapezoid Rule for function f(x)
% on [a,b] using 3*N steps
f = @(x) 2.7*x./(x.^2+25);
h = (b-a)/N;
i = 0:N-1;
xi0 = a+h*i;
xi1 = a+h*(i+1);
T = (h/2)*sum(f(xi0)+f(xi1));
end
```

h	T	$abs-err$	err/h	err/h^2	err/h^3	err/h^4
0.5	0.3609357	0.0012207	0.0024414	0.004883	0.009765	0.01953
0.25	0.3618516	0.0003048	0.0012191	0.004877	0.019506	0.07802
0.125	0.3620802	7.618E-05	0.0006095	0.004876	0.039005	0.3120
0.0625	0.3621373	1.908E-05	0.0003053	0.004885	0.078159	1.2506

5. (5 pts) For the integral

$$\int_1^3 \frac{2.1x}{x^2+4}\,dx$$

the exact value is 1.003287017278808.

With the MatLab Simpson's rule program below, we find

```
function S = compsimp8(a,b,N)
% Composite Simpson's Rule for function f(x)
% on [a,b] using 2N steps
f = @(x) 2.1*x./(x.^2 + 4);
h = (b-a)/(2*N);
i = 0:N-1;
xi = a+2*i*h;
xi1 = a+2*(i+0.5)*h;
xi2 = a+2*(i+1)*h;
S = (h/3)*sum(f(xi)+4*f(xi1)+f(xi2));
end
```

h	S	$abs-err$	err/h^2	err/h^3	err/h^4	err/h^5
0.5	1.00328476	2.260E-06	9.040E-06	1.808E-05	3.616E-05	7.232E-05
0.25	1.00328689	1.253E-07	2.005E-06	8.019E-06	3.207E-05	1.283E-04
0.125	1.00328701	7.591E-09	4.858E-07	3.887E-06	3.109E-05	2.488E-04
0.0625	1.00328702	4.708E-10	1.205E-07	1.928E-06	3.085E-05	4.936E-04

6. (8 pts) For the integral

$$\int_0^8 28e^{-0.58x}\sin(0.39x)\,dx$$

the exact value is 22.56305245.

We can use the Composite Midpoint rule including a `while` loop to check against a tolerance level.

```
function [M,h] = compmid9tol(a,b,tol)
% Composite Trapezoid Rule for function f(x)
% on [a,b] using N steps with a tolerance
f = @(x) 28*exp(-0.58*x).*sin(0.39*x);
M0 = 0;
M = 1;
N = 1;
j = 0;
while (abs(M-M0) > tol)
M0 = M;
h = (b-a)/N;
```

```
i = 1:N;
ci = a+0.5*(2*i-1)*h;
M = h*sum(f(ci));
j = j + 1;
N = 2^j;
end
```

We can use the Composite Trapezoid rule including a `while` loop to check against a tolerance level.

```
function [T,h] = comptrap9tol(a,b,tol)
% Composite Trapezoid Rule for function f(x)
% on [a,b] using 3*N steps with a tolerance
f = @(x) 28*exp(-0.58*x).*sin(0.39*x);
T0 = 0;
T = 1;
N = 1;
j = 0;
while (abs(T-T0) > tol)
T0 = T;
h = (b-a)/(3*N);
i = 0:N-1;
xi0 = a+h*(1 + 3*i);
xi1 = a+h*(2 + 3*i);
T = (3*h/2)*sum(f(xi0)+f(xi1));
j = j + 1;
N = 2^j;
end
```

We can use the Composite Simpson's rule including a `while` loop to check against a tolerance level.

```
function [h,S] = compsimp9tol(a,b,tol)
% Composite Simpson's Rule for function f(x)
% on [a,b] doubling steps til within tolerance
f = @(x) 28*exp(-0.58*x).*sin(0.39*x);
S0 = 0;
N = 1;
j = 0;
h = (b-a)/2;
S = f(h)*(b-a);
while (abs(S-S0)>tol)
S0 = S;
h = (b-a)/(2*N);
i = 0:N-1;
xi = a+2*i*h;
xi1 = a+2*(i+0.5)*h;
xi2 = a+2*(i+1)*h;
S = (h/3)*sum(f(xi)+4*f(xi1)+f(xi2));
j = j + 1;
N = 2^j;
end
```

7. (6 pts) For the integral

$$\int_{-1}^{3} 5x^2 \cos(0.64x)\, dx$$

the exact value is 6.631210452129920.

We can use Gaussian Quadrature rule with N points

```
function S = GQww10(N)
% using N points for the Gaussian Quadrature
[L,Lroots,GQw] = LegendrePolynomials(N);
gr = Lroots{N};
wt = GQw{N};
a = -1; b = 3;
f = @(x) 5*x.^2.*cos(0.64.*x);
gx = ((b-a)*gr + (b+a))/2;
wx = wt*(b-a)/2;
S = sum(wx.*f(gx))
end
```

which calls the LegendrePolynomials program given from the class.

We can use the Composite Simpson's rule

```
function S = compsimp10b(a,b,N)
% Composite Simpson's Rule for function f(x)
% on [a,b] using 2N steps
f = @(x) 5.0*x.^2.*cos(0.64*x);
h = (b-a)/(2*N);
i = 0:N-1;
xi = a+2*i*h;
xi1 = a+2*(i+0.5)*h;
xi2 = a+2*(i+1)*h;
S = (h/3)*sum(f(xi)+4*f(xi1)+f(xi2));
end
```

Comparing the Gaussian Quadrature error and that for the Composite Simpson rule, we see that the optimal accuracy of the Gaussian quadrature is superior. It only takes a 3 point evaluation to exceed the Composite Simpson's rule using 5 points. The 5 point Gaussian quadrature is more than 2 orders of magnitude improved over the Composite Simpson's rule using 5 points.

8. (6 pts) For the integral

$$\int_{3}^{7} 6.8x \ln(x^2 + 12.96)\, dx$$

the exact value is 6.631210452129920.

We can use Gaussian Quadrature rule with N points

```
function S = GQeg2(N)
[L,Lroots,GQw] = LegendrePolynomials(N);
gr = Lroots{N};
wt = GQw{N};
```

```
a = 3; b = 7;
f = @(x) 6.8*x.*log(x.^2 + 12.96);
gx = ((b-a)*gr + (b+a))/2;
wx = wt*(b-a)/2;
S = sum(wx.*f(gx))
end
```

which calls the LegendrePolynomials program given from the class.

We can use the Composite Simpson's rule

```
function S = compsimp11b(a,b,N)
% Composite Simpson's Rule for function f(x)
% on [a,b] using 2N steps
f = @(x) 6.8*x.*log(x.^2 + 12.96);
h = (b-a)/(2*N);
i = 0:N-1;
xi = a+2*i*h;
xi1 = a+2*(i+0.5)*h;
xi2 = a+2*(i+1)*h;
S = (h/3)*sum(f(xi)+4*f(xi1)+f(xi2));
end
```

Comparing the Gaussian Quadrature error and that for the Composite Simpson rule, we see that the optimal accuracy of the Gaussian quadrature is superior. It only takes a 3 point evaluation to exceed the Composite Simpson's rule using 5 points. The 5 point Gaussian quadrature is more than 2 orders of magnitude improved over the Composite Simpson's rule using 5 points.