Integral Estimations

Colton Williams

September 20, 2017

1 Estimation of $\sin(x)$ from 0 to 2

Method used	Integral approximation	Error
Midpoint rule	1.419842	0.003695
Trapezoidal rule	1.408763	0.007384
Simpson's rule	1.416149	0.000002
Gaussian quadrature (3)	1.416198	0.000051
Gaussian quadrature (4)	1.416146	0.000001
Gaussian quadrature (5)	1.416146	0.000001

2 Program listing

```
/*
        COLTON WILLIAMS 2017
        NUMERICAL ANALYSIS
        SIN ESTIMATION
*/
#include <stdio.h>
#include <math.h>

float sinmidpoint()
{
        float result = 0.0;
        for (float i = 0.0;i < 2.0; i = i + 0.25)
        {
                result += sin((i + i + .25)/2);
        }
        result *= 0.25;
        return result;
}
```

```c

float sintrapezoid()
{
        float result = 0.0;
        for (float i = 0.0; i <= 2.0; i = i + 0.25) // include 2.0 in loop in order to have 9 points
        {
                result += 2 * sin(i);
        }
        result -= sin(0.0) + sin(2.0);
        result *= 2.0/16.0;
        return result;
}

float sinsimpsons()
{
        float result = 0.0;
        for (float i = 0.0; i < 2.0; i = i + 0.25)
        {
                result = result + (0.25/6)*(sin(i) + 4*sin((i + i + 0.25)/2) + sin(i + 0.25));
        }
        return result;
}

float singaussian(int n)
{
        float result = 0.0;
        switch(n)
        {
        case 3:
                result += 0.555555 * sin(0.774596 + 1.0);
                result += 0.888888 * sin(1.0);
                result += 0.555555 * sin(-0.774596 + 1.0);
                break;
        case 4:
                result += 0.347854 * sin(0.861136 + 1.0);
                result += 0.652145 * sin(0.339981 + 1.0);
                result += 0.347854 * sin(-0.861136 + 1.0);
                result += 0.652145 * sin(-0.339981 + 1.0);
                break;
        case 5:
```

```
                        result += 0.236926 * sin(0.906179 + 1.0);
                        result += 0.478628 * sin(0.538469 + 1.0);
                        result += 0.568888 * sin(1.0);
                        result += 0.236926 * sin(-0.906179 + 1.0);
                        result += 0.478628 * sin(-0.538469 + 1.0);
                        break;
        }
        return result;
}

int main()
{
        printf("Midpoint estimation: %f\n", sinmidpoint());
        printf("Trapezoidal estimation: %f\n", sintrapezoid());
        printf("Simpson's rule estimation: %f\n", sinsimpsons());
        printf("Gaussian estimation (3): %f\n", singaussian(3));
        printf("Gaussian estimation (4): %f\n", singaussian(4));
        printf("Gaussian estimation (5): %f\n", singaussian(5));
        return 0;
}
```