ESC407 Lab 2

Maggie Wang

October 16, 2023

1. Central Differences for Numerical Differentiation

   (a) The derivative of $f(x) = e^{2x}$ was calculated numerically at $x = 0$, using central differences. Table 1 lists the numerically computed value of $f'(0)$ for different step sizes h between 10^{-16} and 1.

h	$f'(0)$
10^{-16}	1.1102230246251565
10^{-15}	2.1094237467877974
10^{-14}	1.9984014443252818
10^{-13}	1.9995116673499070
10^{-12}	2.0000667788622195
10^{-11}	2.0000001654807420
10^{-10}	2.0000001654807420
10^{-9}	2.0000000544584395
10^{-8}	1.9999999878450580
10^{-7}	1.9999999989472883
10^{-6}	1.9999999999464890
10^{-5}	2.0000000000242046
10^{-4}	2.0000000033337795
10^{-3}	2.0000003333333627
10^{-2}	2.0000333334999842
10^{-1}	2.0033350003968820
1	2.3504023872876028

Table 1: Derivative of $f(x) = e^{2x}$ at $x = 0$ for different h values

The raw output from the code is

```
1 a) errors
    h       f'(0)
    1e-16: 1.1102230246251565
    1e-15: 2.1094237467877974
    1e-14: 1.9984014443252818
    1e-13: 1.999511667349907
    1e-12: 2.0000667788622195
    1e-11: 2.000000165480742
    1e-10: 2.000000165480742
    1e-09: 2.0000000544584395
    1e-08: 1.999999987845058
    1e-07: 1.9999999989472883
    1e-06: 1.999999999946489
```

```
1e-05: 2.0000000000242046
0.0001: 2.0000000033337795
0.001: 2.0000003333333627
0.01: 2.0000333334999842
0.1: 2.003335000396882
1.0: 2.3504023872876028
```

(b) The expected optimal step size h^* and corresponding error are calculated using equations (5.99) and (5.100) from the textbook, using $C = 10^{-16}$. We find

$$\begin{aligned} h^* &= \left(24C\left|\frac{f(x)}{f'''(x)}\right|\right)^{1/3} \\ &= \left(\frac{24Ce^{2x}}{2^3e^{2x}}\right)^{1/3} \\ &= \left(\frac{24C}{2^3}\right)^{1/3} \\ &\approx 6.7 \times 10^{-6} \end{aligned}$$

and

$$\begin{aligned} \epsilon &= \frac{2C|f(x)|}{h^*} + \frac{1}{24}h^2|f'''(x)| \\ &= 4.48 \times 10^{-11} \end{aligned}$$

The actual error is plotted against h in fig 1. We see that the expected optimal step size and minimum in error are very close to their actual values.

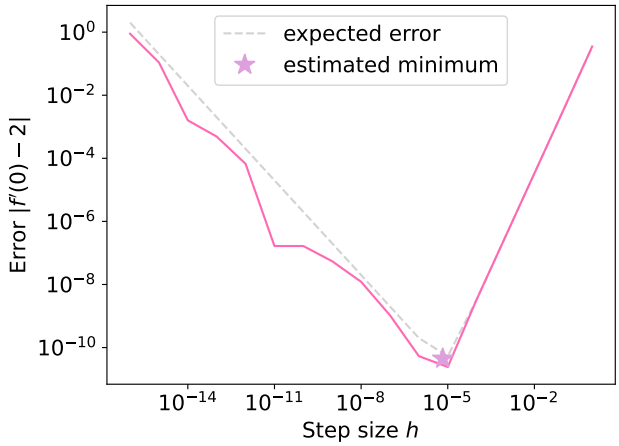

Figure 1: Calculated error of $f'(0)$ using central difference approximation vs step size h

(c) The first 10 derivatives of $f'(x)$ at $x = 0$ calculated using central differences with the optimal step size from part b) (which is the same for all m), and with $h - 0.05$, are listed in table 2. However, for $m > 3$, rounding error from repeated approximations accumulate, and calculated values of $f^m(0)$ diverges. Using a larger h reduces the roundoff error and leads to more accurate calculations the higher order derivatives.

m	Analytical value	$h \approx 6.7 \times 10^{-6}$	$h = 0.05$
1	2	2.0000000000234888	2.000833437506202
2	4	4.000001933007014	4.003334444642892
3	8	7.77155789571232	8.010005418359611
4	16	-221126.08039335813	16.02668667564089
5	32	8258210993.557659	32.06673059601428
6	64	2.2204420456714464e+16	64.16018684518576
7	128	-3.68559504405105e+20	128.37384304020816
8	256	-2.1195518363153336e+27	256.8546079828593
9	514	2.05607099203305e+31	513.9293079992058
10:	1028	1.971716534777767e+38	1028.4099971613614

Table 2: Approximate values for $f^m(0)$ computed using central differences, with step size $h = 6.7 \times 10^{-6}$, and $h = 0.05$.

The raw output from the code is:

```
1 c) derivatives using optimal h from part b:
    m       optimal h                 h=0.05
    1:      2.0000000000234888        2.000833437506202
    2:      4.000001933007014         4.003334444642892
    3:      7.77155789571232          8.010005418359611
    4:      -221126.08039335813       6.02668667564089
    5:      8258210993.557659         32.06673059601428
    6:      2.2204420456714464e+16    64.16018684518576
    7:      -3.68559504405105e+20     128.37384304020816
    8:      -2.1195518363153336e+27   256.8546079828593
    9:      2.05607099203305e+31      513.9293079992058
    10:     1.971716534777767e+38     1028.4099971613614
```

2. Integrating the Dawson Function Again

   (a) Fig 2 shows the Dawson function at $x = 4$ evaluated using trapezoidal rule, Simpson's rule and gaussian quadrature for varying integration slices N. The function calculated using gaussian quadrature converges much more quickly than integration schemes using Newton-Cotes formulae.

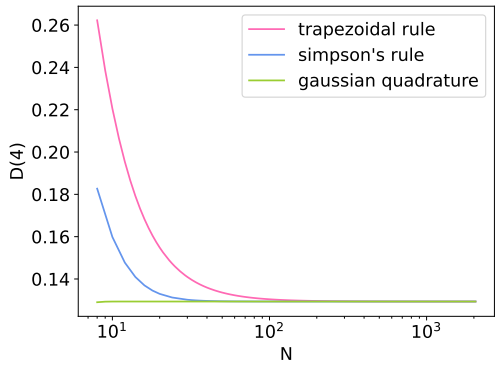

Figure 2: Dawson function $D(x)$ at $x = 4$ calculated numerically vs number of integration slices N

(b) The error on $D(4)$ using gaussian quadrature, $I_{2N} - I_N$ for the values of N in 2a) are plotted in fig 3. The error decreases quickly from $N = 8$ to around $N = 20$, and then remains around the machine precision limit, which suggests the accuracy is limited by roundoff error instead of estimations errors from using Gaussian Quadrature.

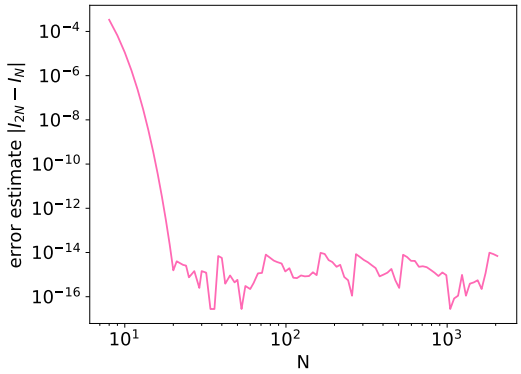

Figure 3: Error on $D(4)$ using gaussian quadrature vs number of integration slices

(c) Fig 4 shows the relative error on $D(4)$ calculated using gaussian quadrature compared to the true value (from Scipy) as a function of the number of integration slices. The behaviour of the relative error is the same as the estimated error from fig 3, while being around an order or magnitude larger than the estimated error.

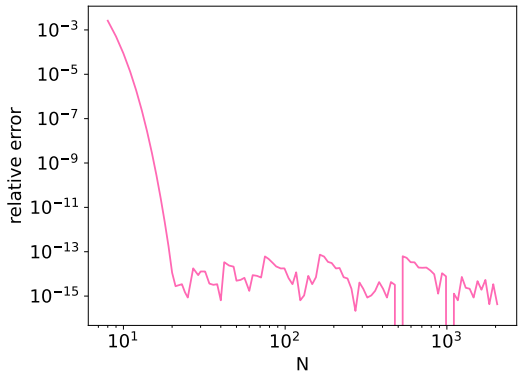

Figure 4: Relative error on $D(4)$ vs number of integration slices

3. Calculating potential energy of QM harmonic oscillator

(a) A function $H(n, x)$ was written to calculate the n^{th} degree Hermite polynomial for a given x based on its recursion relation $H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x)$.

(b) Fig 5 shows the wave functions for the first 4 energy levels of a 1D quantum harmonic oscillator.

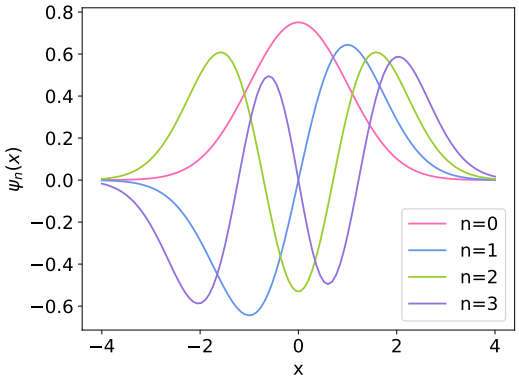

Figure 5: Harmonic oscillator wave functions

(c) The potential energy U for n from 0 to 10 computed using Gaussian quadrature using 100 points, from the formula

$$U(n) = \frac{\langle x^2 \rangle}{2} = \int_{-\infty}^{\infty} x^2 |\psi_n(x)|^2 dx$$

The improper integral is treated by applying the following transformation

$$\int_{-\infty}^{\infty} f(x)dx = \int_{-\pi/2}^{\pi/2} \frac{f(\tan z)}{\cos^2 z} dz$$

Table 3 lists the calculated potential energies of the particle for each n

n	0	1	2	3	4	5	6	7	8	9	10
$U(n)$	0.25	0.75	1.25	1.75	2.25	2.75	3.25	3.75	4.25	4.75	5.25

Table 3: Potential energies of the quantum harmonic oscillator

The raw output from the code is

```
0 0.24999999999999997
1 0.7500000000000119
2 1.2500000000003055
3 1.7499999999981126
4 2.249999999916142
5 2.7499999996902913
6 3.250000005587624
7 3.750000044358483
8 4.249999920351134
9 4.749998195042198
10 5.24999677662962
```