AM205: Solutions to take-home midterm exam[1]

Problem 1 – modified Simpson's rule

Part (a)

We aim to evaluate the integral $I(f) = \int_a^b f(x)dx$ using the variation of Newton–Cotes quadrature with control points at $x_0 = a$, $x_1 = (3a+b)/4$, and $x_2 = b$. The interpolant of f in Lagrange form is

$$p_n(x) = \sum_{k=0}^{2} f(x_k) L_k(x), \tag{1}$$

where L_k is the Lagrange polynomial over the three control points. Then

$$\begin{aligned} Q_2(f) = f(x_0) \int_a^b \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}\, dx + f(x_1) \int_a^b \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}\, dx \\ + f(x_2) \int_a^b \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}\, dx, \end{aligned} \tag{2}$$

and solving the individual integrals using a symbolic system gives

$$\begin{aligned} Q_2(f) &= f(a)\left(\tfrac{1}{6}(a-b)\right) + f\left(\tfrac{3a+b}{4}\right)\left(-\tfrac{8}{9}(a-b)\right) + f(b)\left(-\tfrac{5}{18}(a-b)\right) \\ &= \frac{b-a}{3}\left(-\tfrac{1}{2} f(a) + \tfrac{8}{3} f\left(\tfrac{3a+b}{4}\right) + \tfrac{5}{6} f(b)\right). \end{aligned} \tag{3}$$

Hence $Q_2(f) = \sum_{k=0}^{2} f(x_k) w_k$ where the weights are given by $(w_0, w_1, w_2) = (-\frac{1}{6}, \frac{8}{9}, \frac{5}{18})$.

Part (b)

The composite integration rule is

$$I(f) = \int_a^b f(x) = \sum_{i=1}^{m} \int_{x_{i-1}}^{x_i} f(x)dx, \tag{4}$$

where $h = \frac{b-a}{m}$, $x_i = a + ih$, and each integration over the interval $[x_{i-1}, x_i]$ is evaluated using the formula from part (a). Let $Q_{2,i}$ denote the numerical quadrature value for the integral over the interval $[x_{i-1}, x_i]$. Then the resultant error satisfies

$$|E_{2,h}| \le \sum_{i=1}^{m} \left| \int_{x_{i-1}}^{x_i} f(x)dx - Q_{2,i} \right| = \sum_{i=1}^{m} \left| \int_{x_{i-1}}^{x_i} \left(f(x) - p_2^{[x_{i-1},x_i]}(x) \right) dx \right|, \tag{5}$$

where $p_2^{[x_{i-1},x_i]}(x)$ is the second order polynomial interpolant on the interval $[x_{i-1}, x_i]$. Recall in lecture that for quadrature points z_0, z_1, z_2 on the interval $[a, b]$,

$$f(x) - p_2^{[a,b]}(x) = \frac{f'''(\theta)}{(n+1)!}(x-z_0)(x-z_1)(x-z_2) \tag{6}$$

[1] Solutions to problems 2 and 4 were written by Kevin Chen. Solutions to problems 1 and 3 were written by Dustin Tran. Edited by Chris H. Rycroft.

for some $\theta \in [a,b]$, and hence

$$\left| f(x) - p_2^{[a,b]}(x) \right| \leq \frac{M_3^{[a,b]}}{(n+1)!} \left|(x-z_0)(x-z_1)(x-z_2)\right| \tag{7}$$

where $M_3^{[a,b]} = \max_{\theta\in[a,b]} |f'''(\theta)|$. Hence

$$\begin{aligned}
\left| \int_{x_{i-1}}^{x_i} f(x) - p_2^{[x_{i-1},x_i]}(x) dx \right| &\leq \frac{M_3^{[x_{i-1},x_i]}}{3!} \int_{x_{i-1}}^{x_i} \left| (x-x_{i-1})\left(x - \frac{3x_{i-1}+x_i}{4}\right)(x-x_i) \right| dx \\
&= \frac{M_3^{[x_{i-1},x_i]}}{3!} \int_0^h \left| x(x-\tfrac{h}{4})(x-h) \right| dx \\
&= \frac{M_3^{[x_{i-1},x_i]}}{3!} \left(\int_0^{h/4} x(x-\tfrac{h}{4})(x-h)dx - \int_{h/4}^h x(x-\tfrac{h}{4})(x-h)dx \right) \\
&= \frac{M_3^{[x_{i-1},x_i]}}{3!} \left(\frac{7h^4}{3072} + \frac{45h^4}{1024} \right) \\
&\leq \frac{M_3^{[x_{i-1},x_i]}}{6} \frac{71h^4}{1536} \qquad (8)
\end{aligned}$$

and substituting this expression into Eq. 5 gives

$$\begin{aligned}
|E_{2,h}| &\leq \sum_{i=1}^{m} \left| \int_{x_{i-1}}^{x_i} f(x) - p_2^{[x_{i-1},x_i]}(x)dx \right| \\
&\leq \frac{71h^4}{9216} \sum_{i=1}^{m} M_3^{[x_{i-1},x_i]} \\
&\leq \frac{71h^4}{9216} m \|f'''\|_\infty = \frac{71h^3}{9216}(b-a)\|f'''\|_\infty. \qquad (9)
\end{aligned}$$

This error bound is larger than for the usual Simpson's rule, where the error is bounded by $h^3(b-a)\|f'''\|_\infty/192$.[2] This is expected, since the unevenly spaced control points create the possibility that some parts of the function are not well-approximated.

Problem 2 – a class of Runge–Kutta methods

Part (a) – value of γ

The given Butcher tableau corresponds to computing the three intermediate steps

$$k_1 = f(t_n, y_n), \tag{10}$$
$$k_2 = f(t_n + \beta h, y_n + \beta h k_1), \tag{11}$$
$$k_3 = f(t_n + \gamma h, y_n + \gamma h k_2), \tag{12}$$

[2] In fact, as discussed in class and on Piazza, a more detailed analysis of the Simpson's rule error shows that it is actually $O(h^4)$, due to cancellation of the $O(h^3)$ terms.

after which the next step is given by

$$y_{n+1} = y_n + k_3 h. \tag{13}$$

Analytically, the Taylor expansion of $y(t_{n+1})$ is

$$y(t_{n+1}) = y(t_n) + hf(t_n, y_n) + \frac{h^2}{2} f'(t_n, y_n) + O(h^3), \tag{14}$$

where $f'(t_n, y_n) = f_t(t_n, y_n) + f(t_n, y_n) f_y(t_n, y_n)$. Taylor expanding k_1, k_2, and k_3 gives

$$\begin{aligned}
k_1 &= f(t_n, y_n), && (15)\\
k_2 &= f(t_n, y_n) + \beta h f_t(t_n, y_n) + \beta h f(t_n, y_n) f_y(t_n, y_n), && (16)\\
k_3 &= f(t_n, y_n) + \gamma h f_t(t_n, y_n) + \gamma h f_y(t_n, y_n) f(t_n, y_n) \\
&\quad + \gamma h f_y(t_n, y_n) \beta h f_t(t_n, y_n) + \gamma h f_y(t_n, y_n) \beta h f(t_n, y_n) f_y(t_n, y_n). && (17)
\end{aligned}$$

Substituting the expression for k_3 into Eq. 13 gives

$$\begin{aligned}
y_{n+1} &= y_n + hf(t_n, y_n) + \gamma h^2 \left(f_t(t_n, y_n) + f_y(t_n, y_n) f(t_n, y_n) \right) \\
&\quad + h^3 \left(\gamma f_y(t_n, y_n) \beta f_t(t_n, y_n) + \gamma f_y(t_n, y_n) \beta f(t_n, y_n) f_y(t_n, y_n) \right). && (18)
\end{aligned}$$

Comparing Eq. 18 to the Taylor expansion in Eq. 14 shows that $\gamma = \frac{1}{2}$ in order for second-order terms to agree. The terms involving β are all $O(h^3)$ and hence there is no restriction on this parameter.

Part (b)

Consider the special case of solving the differential equation $y' = pt + qy$ for some constants p and q; the aim is to find β so the scheme is third-order accurate for this differential equation. Expanding to third order, the Taylor series of $y(t_{n+1})$ is

$$y(t_{n+1}) = y(t_n) + hf(t_n, y_n) + \frac{h^2}{2} f'(t_n, y_n) + \frac{h^3}{6} f''(t_n, y_n) + O(h^4), \tag{19}$$

where the derivatives of f are given by

$$\begin{aligned}
f &= pt + qy, \\
f' &= p + q(pt + qy) = p + pqt + q^2 y, \\
f'' &= pq + q^2(pt + qy) = pq + pq^2 t + q^3 y. && (20)
\end{aligned}$$

From part (a) the third order terms are

$$\gamma f_y(t_n, y_n) \beta f_t(t_n, y_n) + \gamma f_y(t_n, y_n) \beta f(t_n, y_n) f_y(t_n, y_n) = \gamma \beta pq + \gamma \beta (pt + qy). \tag{21}$$

Equating these with the third-order terms in Eq. 19 gives

$$\gamma \beta pq + \gamma \beta (pt + qy) = \frac{1}{6}(pq + pq^2 t + q^3 y) \tag{22}$$

and hence $\beta = \frac{1}{3}$ in order for the Runge–Kutta scheme to be third-order accurate for this restricted class of differential equations.

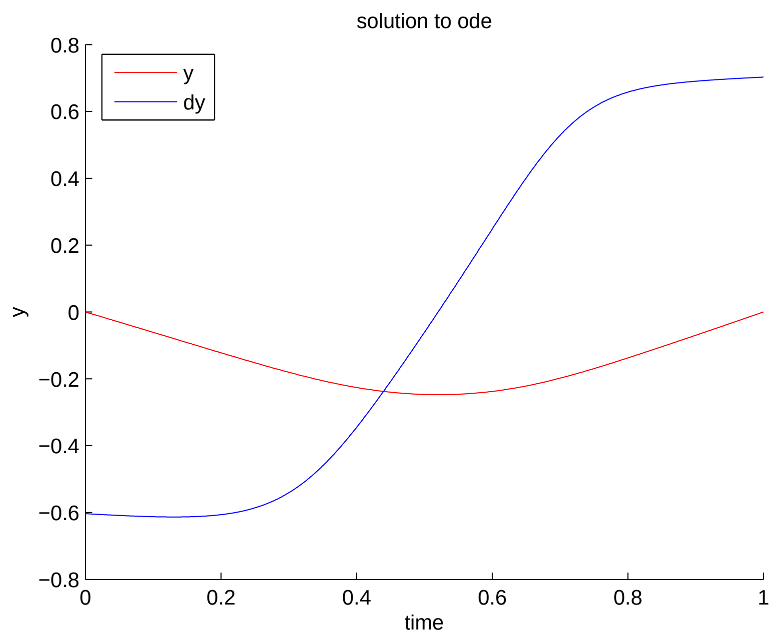

Figure 1: $y(t)$ and $y'(t)$ when $g = -0.603205$.

Part (c)

We use the methods derived in (a) and (b) to solve this problem. Since this ODE is second order, we can re-write it as a first-order system,

$$y' = q, \tag{23}$$
$$q' = 2 + 2t - 16q^4, \tag{24}$$

and then solve the system using $\gamma = \frac{1}{2}$ and $\beta = \frac{1}{3}$. Figure 1 shows y and y' as functions of t. We find $g = -0.603205$.

Problem 3 – differentiation on unequal grids

Part (a)

We first write the Taylor expansion of the two additional evaluations around x:

$$f(x-c) = f(x) - cf'(x) + \frac{f''(x)}{2}c^2 - \cdots$$
$$f(x+b) = f(x) + bf'(x) + \frac{f''(x)}{2}b^2 - \cdots$$

We aim to cancel out the second order term by using a linear combination of $f(x-c)$ and $f(x+b)$. We can then normalize in order to obtain precisely $f'(x)$ with constant 1, and add an arbitrary multiple of $f(x)$ to cancel out its own term. For $C_1, C_2 \in \mathbb{R}$, consider the expression

$$\frac{c^2 f(x+b) - b^2 f(x-c)}{C_1} + C_2 f(x) = \frac{1}{C_1}\Big((c^2-b^2)f(x) + (bc^2+cb^2)f'(x) + 0f''(x) - \cdots\Big) + C_2 f(x) \tag{25}$$

Setting $C_1 = bc^2 + cb^2 = bc(b+c)$ and $C_2 = -(c^2-b^2)/C_1$ yields the numerical formula

$$\begin{aligned} f'_{\text{(a)}}(x) &= \frac{c^2 f(x+b) - b^2 f(x-c)}{bc(b+c)} - \frac{c^2-b^2}{bc(b+c)} f(x) \\ &= \frac{-b^2 f(x-c) - (c^2-b^2) f(x) + c^2 f(x+b)}{bc(b+c)} \\ &= \frac{(b-c)f(x)}{bc} + \frac{cf(x+b)}{b(b+c)} - \frac{bf(x-c)}{c(b+c)}, \end{aligned} \tag{26}$$

which is second-order accurate in b and c. As a check, note that if $c = b$ then

$$f'_{\text{(a)}}(x) = \frac{(b-b)f(x)}{b^2} + \frac{bf(x+b)}{b(b+b)} - \frac{bf(x-c)}{b(b+b)}, \qquad = \frac{f(x+b) - f(x-b)}{2b}, \tag{27}$$

which is the usual second-order centered-difference formula, as expected.

Parts (b) and (c)

The Python program `unequal.py` calculates derivative of the function $f(x) = \sin 3x$ using the irregular grid $x_k = (1+\alpha)^k$ for $k = -N, -N+1, \ldots, N$. To ensure that $x_N = 3$, the constant α is chosen so that

$$\alpha = 3^{1/n} - 1. \tag{28}$$

This choice of α automatically ensures that $x_{-N} = \frac{1}{3}$. At interior grid points where $|k| < N$, the program uses the finite-difference formula in Eq. 26, where

$$b = x_{k+1} - x_k = \alpha(1+\alpha)^k, \qquad c = x_k - x_{k-1} = \alpha(1+\alpha)^{k-1} \tag{29}$$

At $k = N$, the program uses

$$b = x_{k-2} - x_k, \qquad c = x_k - x_{k-1}, \tag{30}$$

and note that $b < 0$ in this case. Similarly, at $k = -N$, the program uses

$$b = x_{k+1} - x_k, \qquad c = x_k - x_{k+2}, \tag{31}$$

so that $c < 0$ in this case. Figure 2(a) shows a plot of the numerically computed derivative of $f(x)$ in comparison to the exact solution $f'(x) = 3\cos 3x$ for the case of $N = 80$; on these axes, the curves are near-identical. Figure 2(b) shows the difference between these two curves, showing a high level of agreement. The largest error is at x_N since the choices of b and c in Eq. 30 lead to a slightly less accurate solution.

Figure 3 shows a log–log plot of the infinity norm between the numerically computed derivative and the exact answer, as a function of α, for $N = 10, 20, 40, 80, 160, 320, 640$. The plot also shows a best fit line, which has slope 1.94. As expected, this is close to two, since the program makes use of a second-order method.

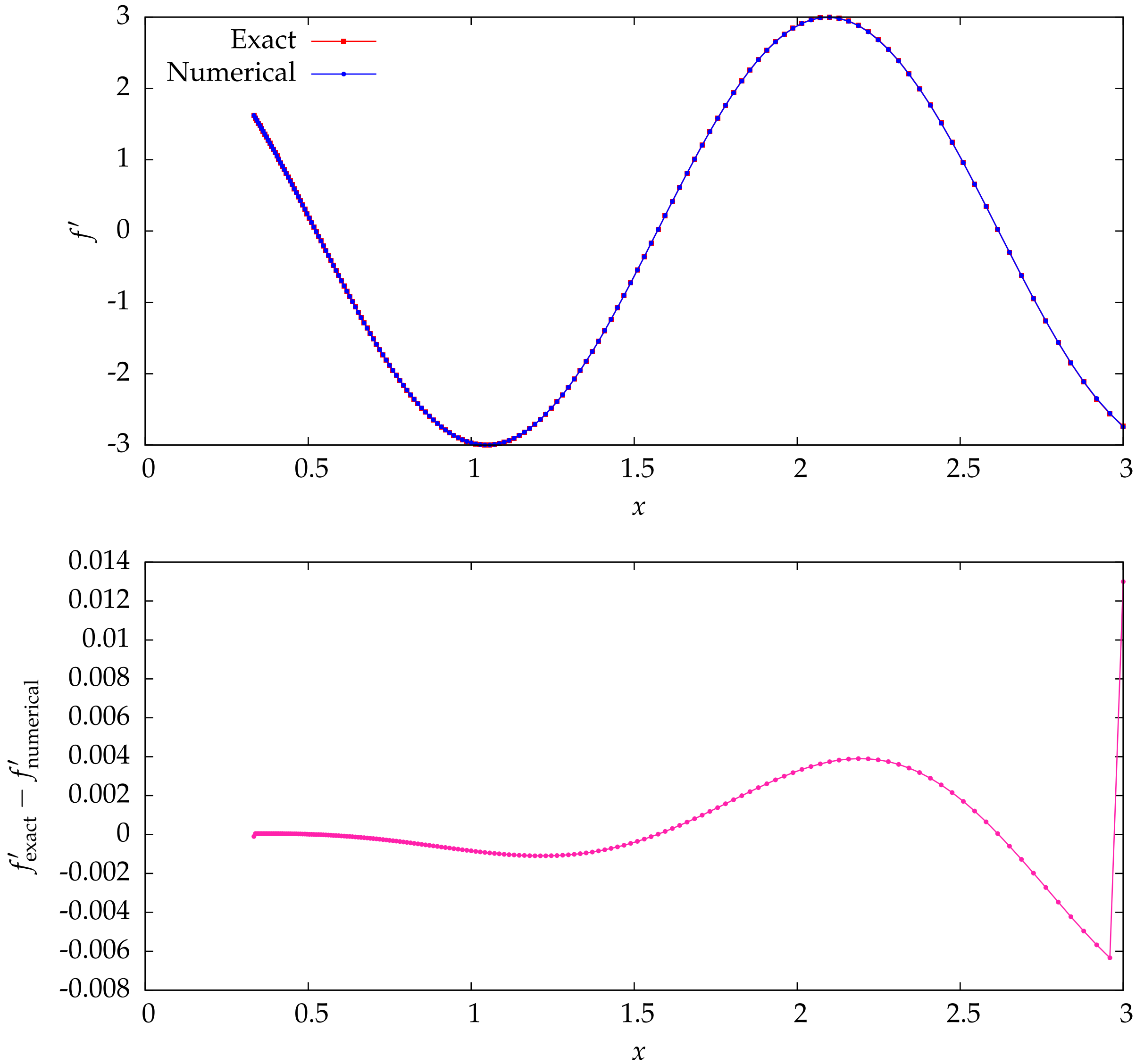
3
2
1
0
-1
-2
-3
f'
Exact
Numerical
0
0.5
1
1.5
2
2.5
3
x
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
-0.002
-0.004
-0.006
-0.008
$f'_{\text{exact}} - f'_{\text{numerical}}$
0
0.5
1
1.5
2
2.5
3
x

Figure 2: Q3.

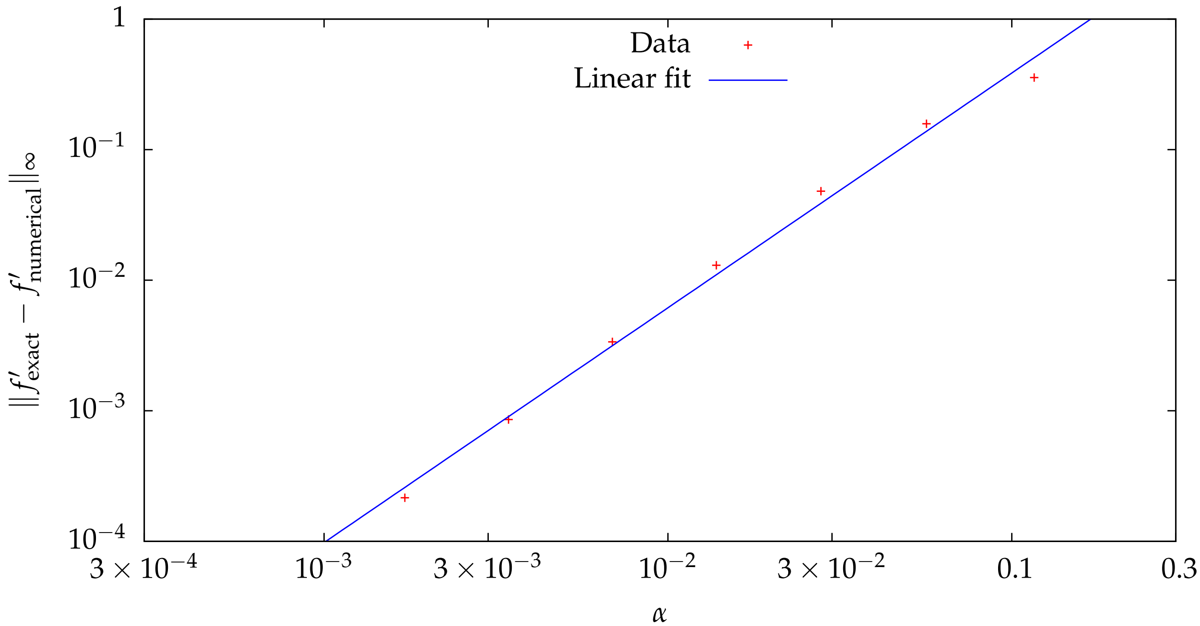

Figure 3: Q3.

Part (d)

Let $F(y) = f(e^y) = f(x)$. Then

$$F'(y) = f'(e^y)e^y = f'(x)x. \tag{32}$$

The standard second-order centered-difference formulae for $F'(y_k)$ at $k = 0$ is

$$F'(y_0) = \frac{F(y_0 + h) - F(y_0 - h)}{2h} = \frac{F(y_1) - F(y_{-1})}{2h}, \tag{33}$$

and by using Eq. 32 one obtains

$$f'(x_0) = \frac{f(x_1) - f(x_{-1})}{2hx_0} \tag{34}$$

Since $h = \log(1 + \alpha)$ this implies that

$$f'_{(\mathrm{d})}(x_0) = \frac{f(x_1) - f(x_{-1})}{2x_0 \log(1 + \alpha)} \tag{35}$$

is a formula

Part (d)

Given that $k = 0$, and in part (b) $x_k = (1 + \alpha)^k$, we therefore have $x_0 = 1$, $x_1 = 1 + \alpha$ and $x_{-1} = (1 + \alpha)^{-1}$. From the deriviation in part (a), it is known that

$$b = x_1 - x_0 = \alpha, \qquad c = x_0 - x_{-1} = \frac{\alpha}{1 + \alpha}. \tag{36}$$

By construction, the finite-difference formulae Thus the formula for part (a) is

$$\begin{aligned}
f'_{(a)}(x_0) &= \frac{-\alpha^2 f(x_{-1}) - ((\frac{\alpha}{1+\alpha})^2 - \alpha^2) f(x_0) + (\frac{\alpha}{1+\alpha})^2 f(x_1)}{\alpha \frac{\alpha}{1+\alpha}\left(\alpha + \frac{\alpha}{1+\alpha}\right)} \\
&= \frac{-\alpha^2}{\frac{\alpha^3(\alpha+2)}{(1+\alpha)^2}} f(x_{-1}) - \frac{(\frac{\alpha}{1+\alpha})^2 - \alpha^2}{\alpha \frac{\alpha}{1+\alpha}\left(\alpha + \frac{\alpha}{1+\alpha}\right)} f(x_0) + \frac{(\frac{\alpha}{1+\alpha})^2}{\frac{\alpha^3(\alpha+2)}{(1+\alpha)^2}} f(x_1) \\
&= \frac{-(1+\alpha)^2}{\alpha(\alpha+2)} f(x_{-1}) - \frac{\frac{\alpha}{1+\alpha} - \alpha}{\alpha \frac{\alpha}{1+\alpha}} f(x_0) + \frac{1}{\alpha(\alpha+2)} f(x_1) \\
&= \frac{-(1+\alpha)^2}{\alpha(\alpha+2)} f(x_{-1}) - \left(\frac{1}{\alpha} - \frac{1+\alpha}{\alpha}\right) f(x_0) + \frac{1}{\alpha(\alpha+2)} f(x_1) \\
&= \frac{-(1+\alpha)^2}{\alpha(\alpha+2)} f(x_{-1}) + f(x_0) + \frac{1}{\alpha(\alpha+2)} f(x_1)
\end{aligned}$$

Recall that for $|\alpha| < 1$, the Maclaurin series expansion for $\log(1+\alpha)$ is

$$\log(1+\alpha) = \alpha - \frac{\alpha^2}{2} + \mathcal{O}(\alpha^3) \tag{37}$$

We plug in x_0 and this expansion into the formula for part (d):

$$\begin{aligned}
f'_{(d)}(x_0) &= \frac{1}{2\log(1+\alpha)} f(x_1) - \frac{1}{2\log(1+\alpha)} f(x_{-1}) \\
&\approx \frac{1}{2(\alpha - \alpha^2/2)} f(x_1) - \frac{1}{2(\alpha - \alpha^2/2)} f(x_{-1}) \\
&\approx \frac{1}{\alpha(2-\alpha)} f(x_1) - \frac{1}{\alpha(2-\alpha)} f(x_{-1})
\end{aligned}$$

Hence their difference up to third-order accuracy is

$$\begin{aligned}
f'_{(a)}(x_0) - f'_{(d)}(x_0) &= \left(\frac{1}{\alpha(\alpha+2)} - \frac{1}{\alpha(2-\alpha)}\right) f(x_1) + f(x_0) + \left(\frac{-(1+\alpha)^2}{\alpha(\alpha+2)} - \frac{1}{\alpha(2-\alpha)}\right) f(x_{-1}) \\
&= \left(\frac{2}{\alpha^2-4}\right) f(x_1) + f(x_0) + \left(\frac{-(1+\alpha)^2 - (\alpha+2)}{\alpha(4-\alpha^2)}\right) f(x_{-1})
\end{aligned}$$

Problem 4 – a hidden charge distribution

While this may ostensibly look like a numerical PDE problem, it is actually a linear least squares problem, where the data in the file `efield.txt` can be be used to find the best fit to the unknown charges. For a single point charge q located at $\mathbf{x}_k$, the electric potential is

$$\phi(\mathbf{x}) = \frac{q}{4\pi|\mathbf{x} - \mathbf{x}_k|} \tag{38}$$

and hence the electric field is

$$\mathbf{E}(\mathbf{x}) = -\nabla\phi(\mathbf{x}) = \frac{(\mathbf{x} - \mathbf{x}_k)q}{4\pi|\mathbf{x} - \mathbf{x}_k|^3}. \tag{39}$$

5	27	27	27	21
21	4	10	0	4
5	17	31	16	4
5	0	10	4	20
21	26	26	26	4

Table 1: The values of the hidden charge distribution q_k.

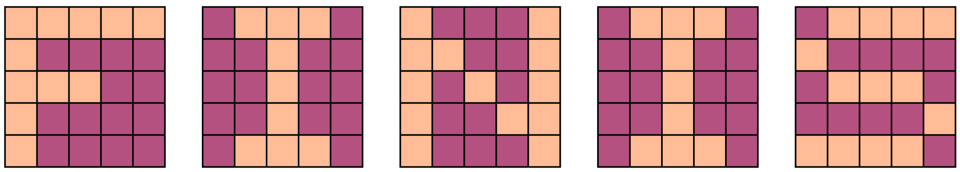

Figure 4: Components of the charges q_k for binary digits representing 1, 2, 4, 8, 16. Yellow corresponds to a one and mauve corresponds to a zero.

Hence, for the given problem with twenty five charges q_k at locations $\mathbf{x}_k$, the electric field will be

$$\mathbf{E}(\mathbf{x}) = \frac{1}{4\pi} \sum_{k=0}^{24} \frac{(\mathbf{x} - \mathbf{x}_k) q_k}{|\mathbf{x} - \mathbf{x}_k|^3}. \tag{40}$$

The file `efield.txt` contains the electric field $\mathbf{E}(\mathbf{p}_i)$ at 500 points $\mathbf{p}_i$ for $i = 0, \ldots, 499$. For this to be consistent with Eq. 40, the matrix system

$$\begin{pmatrix} \mathbf{E}_{0,0} & \mathbf{E}_{0,1} & \cdots & \mathbf{E}_{0,24} \\ \mathbf{E}_{1,0} & \mathbf{E}_{1,1} & \cdots & \mathbf{E}_{1,24} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{E}_{499,0} & \mathbf{E}_{499,1} & \cdots & \mathbf{E}_{499,24} \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{24} \end{pmatrix} = \begin{pmatrix} \mathbf{E}(\mathbf{p}_0) \\ \mathbf{E}(\mathbf{p}_1) \\ \ddots \\ \mathbf{E}(\mathbf{p}_{499}) \end{pmatrix} \tag{41}$$

should be satisfied, where

$$\mathbf{E}_{i,k} = \frac{(\mathbf{x}_i - \mathbf{x}_k)}{4\pi |\mathbf{x}_i - \mathbf{x}_k|^3}. \tag{42}$$

Here, each vector $\mathbf{E}_{i,k}$ and data value $\mathbf{E}(\mathbf{p}_i)$ in Eq. 41 is interpreted as covering two rows of the matrix. This is therefore an overdetermined linear system, with 1000 data points and 25 constraints. It can be solved using Python's `lstsq` function, or using Matlab's backslash operator. Table 1 shows the computed grid of charges.

As stated in the question, the charges are all integers to within numerical precision, over the range from 0 to 31. If each charge is written as a 5-bit binary number, then five separate grids can be constructed showing whether for each bit is zero or one. These are shown in Fig. 4 and spell the word *finis*, sometimes used to denote the ending of a book or movie.