AM205 HW1. Data fitting. Solution

P1. Polynomial approximation of the gamma function

(a) We consider finding a polynomial $g(x) = \sum_{k=0}^{4} p_k x^k$ that fits the data points (j, Γ(j)) for j = 1, 2, 3, 4, 6. Since here are a small number of data points, we can use the Vandermonde matrix to find the coefficients of the interpolating polynomial $g(x) = \sum_{k=0}^{4} g_k x^k$. The linear system is

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 4 & 16 & 64 & 256 \\ 1 & 6 & 36 & 216 & 1296 \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 6 \\ 120 \end{bmatrix} \tag{1}$$

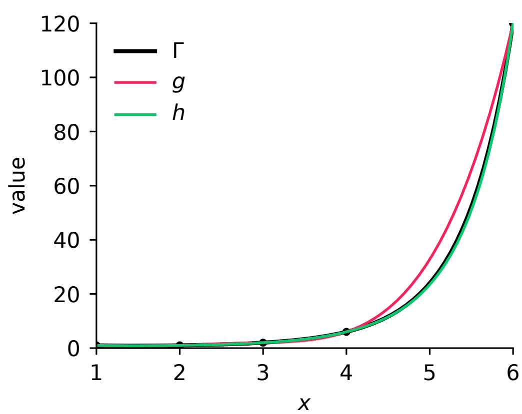
The program gamma p1_gamma.py solves this system, and shows that the coefficients are $g_0 = 17.8, g_1 = -34.917, g_2 = 24.458, g_3 = -7.0833, g_4 = 0.74167$

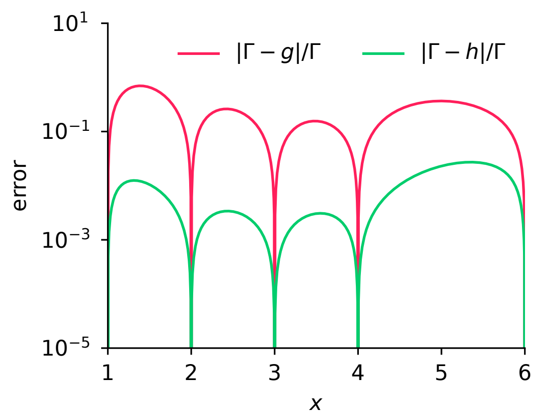
(b) We now consider finding a polynomial $p(x) = \sum_{k=0}^{4} p_k x^k$ that fits the transformed data points (j, log(j)) for j = 1, 2, 3, 4, 6. The coefficients are given by

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 4 & 16 & 64 & 256 \\ 1 & 6 & 36 & 216 & 1296 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} = \begin{bmatrix} \log 1 \\ \log 1 \\ \log 2 \\ \log 6 \\ \log 120 \end{bmatrix} \tag{2}$$

We get the coefficients as $p_0 = 1.1274, p_1 = -1.8725, p_2 = 0.848, p_3 = -0.10902, p_4 = 0.006107$

(c) The plots of three functions and relative errors are as follows

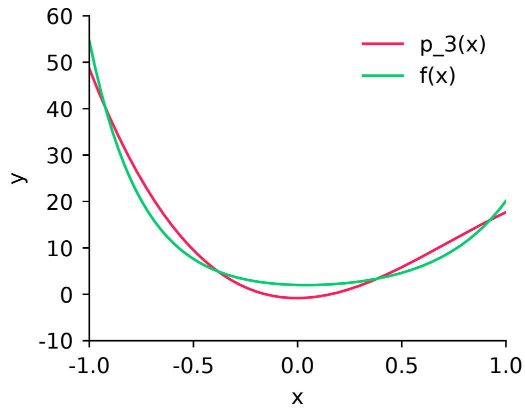




(d) Maximum relative error: 0.69894 for $g(x)$, 0.027223 for $h(x)$. The more accurate approximation is $h(x)$.

P2. Error bounds with Lagrange polynomials

(a) and (b) The following figure shows the Lagrange polynomial p_3(x) over the true function f(x) using a slightly modified version of the in-class code example. Running the code, the infinity norm of the error is approximately 6.04238.



(c) The difference between f(x) and and the interpolating polynomial $p_{n-1}(x)$ can be expressed as

$$f(x) - p_{n-1}(x) = \frac{f^{(n)}(\theta)}{n!} \prod_{i=1}^{n} (x - x_i) \tag{3}$$

where θ is a specific value within the interval from -1 to 1. To obtain a bound on $\|f - p_{n-1}\|_\infty$, we consider the magnitude of the terms on the right hand side. Since the x_i are at the roots of the n-th Chebyshev polynomial $T_n(x)$, it follows that the product is a scalar multiple of this polynomial

$$\prod_{i=1}^{n} (x - x_i) = \lambda T_n(x) \tag{4}$$

where λ is some scaling constant. The coefficient in front of x^n on the left hand side is 1. Using properties of Chebyshev polynomials, the coefficient of x^n in $T_n(x)$ is 2^{n-1}. Hence $\lambda = 2^{-(n-1)}$. The Chebyshev polynomials satisfy $|T_n(x)| \leq 1$ for $x \in [-1, 1]$ and hence

$$\left| \prod_{i=1}^{n} (x - x_i) \right| \leq \frac{1}{2^{n-1}} \tag{5}$$

for $x \in [-1, 1]$.

Now consider the n-th derivative of f , which is given by

$$f^{(n)}(\theta) = (-4)^n e^{-4\theta} + (3)^n e^{3\theta} \tag{6}$$

The maximum value of $|f(n)(\theta)|$ can occur at two places: (i) at an internal maximum, or (ii) at one of the end points of the interval, $\theta = \pm 1$. Consider case (i) first. If n is odd, then

$$f^{(n+1)}(\theta) = 4^{n+1} e^{-4\theta} + 3^{n+1} e^{3\theta} \tag{7}$$

and since both terms are positive, there is no value of θ where $f^{n+1}(\theta) = 0$. If n is even, then

$$f^{(n+1)}(\theta) = -4^{n+1} e^{-4\theta} + 3^{n+1} e^{3\theta} \tag{8}$$

Setting $f^{n+1}(\theta) = 0$ gives

$$4^{n+1} e^{-4\theta} = 3^{n+1} e^{3\theta} \tag{9}$$

and hence $(4/3)^{n+1} = e^{7\theta}$, so

$$\theta = \frac{(n+1)\log(4/3)}{7} \tag{10}$$

is a single solution. However, since

$$f^{(n+2)}(\theta) = \left|(-4)^{n+2} e^{-4\theta} + 2^{n+2} e^{2\theta}\right| > 0 \tag{11}$$

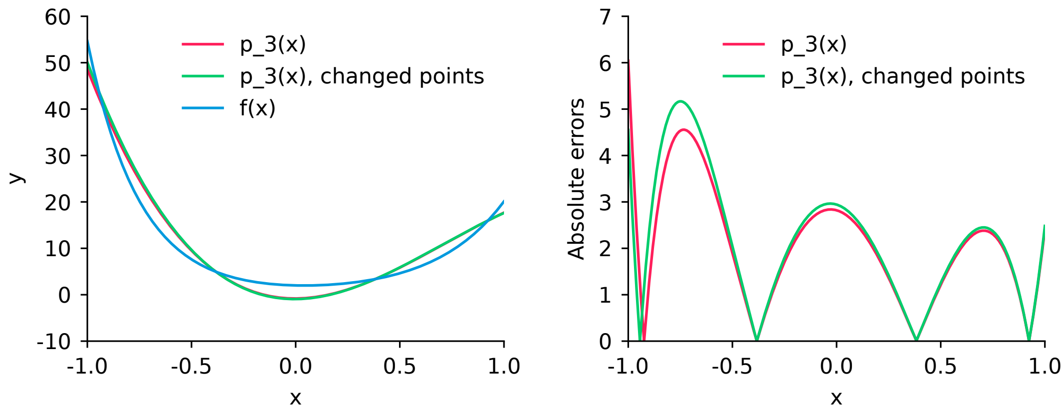
it follows that this must be a minimum of $f^(n)$. Since $f^(n) > 0$, it must be a minimum of $\left|f^{(n)}\right|$ also. Hence, for all values of n there is no possibility that the maximum of $\left|f^{(n)}\right|$ occurs in the interior of the interval. Thus the only remaining possibilities are at the endpoints. Since the factor of $(-3)^n$ grows more rapidly in magnitude, the maximum will occur at $\theta = -1$, and hence

$$\left|f^{(n)}(\theta)\right| \le \left|(-4)^n e^4 + 3^n e^{-3}\right| \tag{12}$$

Combining the results from above equations establishes that

$$\|f - p_{n-1}\|_\infty \le \frac{\left|(-4)^n e^4 + 3^n e^{-3}\right|}{n! 2^{n-1}} \tag{13}$$

(d) There are many ways to find better control points, and this problem illustrates that while the Chebyshev points are a good set of points at which to interpolate a general unknown function, they are usually not optimal for a specific function. One simple method is to examine where the maximum interpolation error is achieved. This is happens near x = -1. Hence if we move the first control point to the left, it will result in a better approximation of f (x) within this region. In this case, we shift the first control point by -0.02, which leads to an infinity norm of 5.16790. The following are the fitting plots after changing tha control points and the corresponding errors.



P3. Condition number of a matrix

(a) Throughout this problem, $\|\cdot\|$ is taken to mean the Euclidean norm. The first two parts of this problem can be solved using diagonal matrices only. Consider first

$$B = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \tag{14}$$

Then $\|B\| = 2$, $\|B^{-1}\| = 1$ and hence $\kappa(B) = 2$. Similarly, $\kappa(C) = 2$. Adding the two matrices together gives

$$B + C = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} = 3I \tag{15}$$

and hence $\kappa(B+C) = \|3I\| \, \|\frac{1}{3}I\| = 3 \times \dfrac{1}{3} = 1$. For these choices of matrices,$\kappa(B+C) < \kappa(B) + \kappa(C)$.

(b) If

$$B = \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{16}$$

then $\kappa(B) = 2$. Similarly, $\kappa(C) = 1$. Adding the two matrices together gives

$$B + C = \begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix} \tag{17}$$

and hence $\kappa(B+C) = 5$. For these choices of matrices,$\kappa(B+C) > \kappa(B) + \kappa(C)$.

(c)

$$\kappa(\alpha A) = \|\alpha A\| \times \|(\alpha A)^{-1}\| = \|\alpha A\| \times \|\frac{1}{\alpha}A^{-1}\| = \|\alpha\| \times \|\frac{1}{\alpha}\| \times \|A\| \times \|A^{-1}\|$$

$$\kappa(\alpha A) = \|A\| \times \|A^{-1}\| = \kappa(A)$$

$$\|QA\| = \sup_{x \neq 0} \frac{\|QAx\|}{\|x\|}.$$

But, multiplying by an orthogonal matrix Q does not change the 2-norm. Therefore,

$$\sup_{x \neq 0} \frac{\|QAx\|}{\|x\|} = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \|A\|.$$

One can also show this by

$$\|QAx\| = \sqrt{\langle QAx, QAx \rangle} = \sqrt{\langle Ax, Q^T QAx \rangle} = \sqrt{\langle Ax, Ax \rangle} = \|Ax\|.$$

Similary, one may also find

$$\|(QA)^{-1}\| = \|A^{-1}\|$$

Finally,

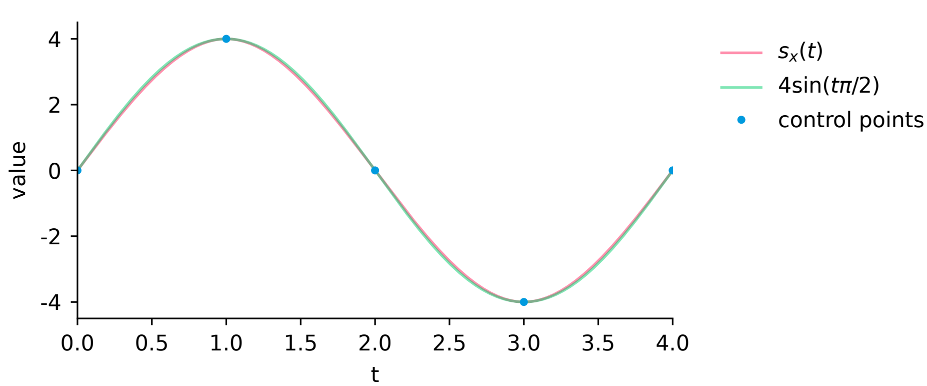
$$\kappa(QA) = \|QA\| \times \|(QA)^{-1}\| = \|A\| \times \|A^{-1}\| = \kappa(A)$$

P4. Periodic cubic splines

(a)

$$s_x(t) = \begin{cases} 6t - 2t^3 & 0 \leq t < 1 \\ -4 + 18t - 12t^2 + 2t^3 & 1 \leq t < 2 \\ -4 + 18t - 12t^2 + 2t^3 & 2 \leq t < 3 \\ 104 - 90t + 24t^2 - 2t^3 & 3 \leq t \leq 4 \end{cases}$$

(b)

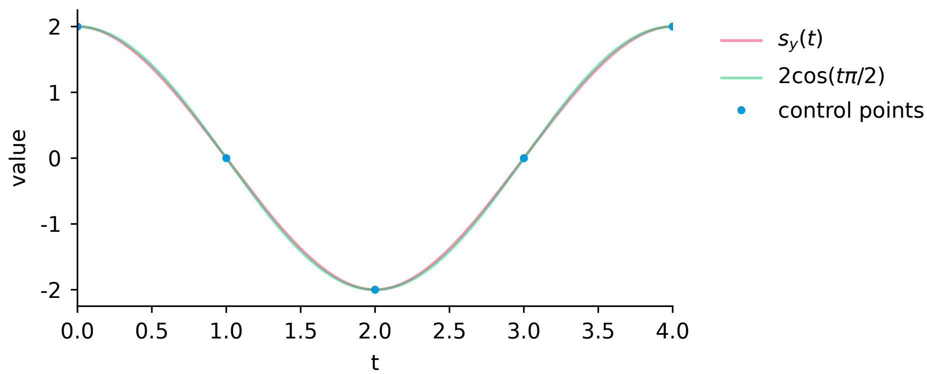


(c)

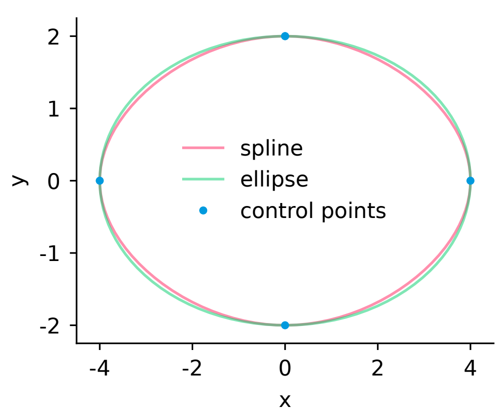
$$s_y(t) = \frac{1}{2}s_x(t+1) = \begin{cases} 3(t+1) - (t+1)^3 & 0 \le t < 1 \\ -2 + 9(t+1) - 6(t+1)^2 + (t+1)^3 & 1 \le t < 2 \\ -2 + 9(t+1) - 6(t+1)^2 + (t+1)^3 & 2 \le t < 3 \\ 52 - 45(t+1) + 12(t+1)^2 - (t+1)^3 & 3 \le t \le 4 \end{cases}$$

simplifies to

$$s_y(t) = \begin{cases} 2 - 3t^2 - t^3 & 0 \le t < 1 \\ 2 - 3t^2 + t^3 & 1 \le t < 2 \\ 2 - 3t^2 + t^3 & 2 \le t < 3 \\ 18 - 24t + 9t^2 - t^3 & 3 \le t \le 4 \end{cases}$$



(d)



The estimated π value is 3.05000.

P5. Image reconstruction from low light

(a) Reconstruction of the regular-light photo `0927` from the three low-light photos `0258`, `0646`, `0704`. Using fragments 0 and 1 for training, and fragments 2 and 3 for testing. The program `p5_reconstruction.py` implements the algorithm.

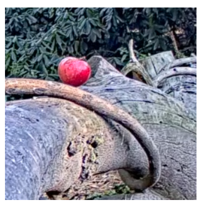
The fitted matrices are

$$F^A = \begin{bmatrix} 0.01344 & 0.01344 & 0.01344 \\ 0.04126 & 0.04126 & 0.04126 \\ 0.05247 & 0.05247 & 0.05247 \end{bmatrix}$$
$$F^B = \begin{bmatrix} -0.54727 & 0.34517 & -0.3521 \\ -1.34219 & 1.13862 & -0.20986 \\ -1.33221 & 0.21761 & 0.65449 \end{bmatrix}$$
$$F^C = \begin{bmatrix} 1.57982 & -0.61108 & 0.40334 \\ 0.07498 & 1.07357 & 0.2709 \\ 0.08154 & -0.85494 & 2.11257 \end{bmatrix}$$
$$\mathbf{p}_{\text{const}} = \begin{bmatrix} 4.54571 \\ -12.72907 \\ -5.966 \end{bmatrix}$$

The error for each fragment

$$S_{ABC}(K_0) = 0.0598315$$
$$S_{ABC}(K_1) = 0.0719668$$
$$S_{ABC}(K_2) = 0.0795461$$
$$S_{ABC}(K_3) = 0.150443$$

fragment 2
reconstructed `0927`



fragment 2
actual `0927`

fragment 3
reconstructed `0927`

fragment 3
actual `0927`

(b) Reconstruction of the regular-light photo `0927` from one low-light photo `0646`. Using fragments 0 and 1 for training, and fragments 2 and 3 for testing.

$$F^B = \begin{bmatrix} 4.3031 & -3.56403 & 1.56626 \\ -1.11043 & 2.64979 & 0.97944 \\ -1.57174 & -4.69737 & 8.40347 \end{bmatrix}$$

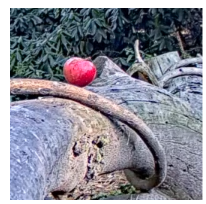
$$\mathbf{p}_{\text{const}} = \begin{bmatrix} 36.76942 \\ 18.52413 \\ 24.98479 \end{bmatrix}$$

The error for each fragment

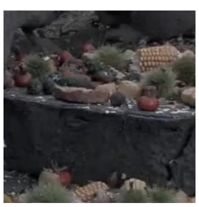
$$S_B(K_0) = 0.0845624$$
$$S_B(K_1) = 0.081768$$
$$S_B(K_2) = 0.106275$$
$$S_B(K_3) = 0.165774$$

fragment 2
input `0646`

fragment 2
reconstructed `0927`



fragment 2
actual `0927`



fragment 3
input `0646`

fragment 3
reconstructed `0927`

fragment 3
actual `0927`

(c) The fitting error S_{ABC} is smaller than S_B for all fragments. The reconstructed fragments 2 and 3 from part **(a)** appear more similar to the actual images. Therefore, including more light levels improves the quality of the fit.

P6. Determining hidden chemical sources

(a) The time derivative of ρ_c is

$$\frac{\partial \rho_c}{\partial t} = \frac{x^2 + y^2 - 4bt}{16\pi b^2 t^3} \exp(-\frac{x^2 + y^2}{4bt}) \tag{18}$$

The x derivative of ρ_c is

$$\frac{\partial \rho_c}{\partial x} = \frac{-2x}{16\pi b^2 t^2} \exp(-\frac{x^2+y^2}{4bt}) \tag{19}$$

and the second x derivative is

$$\frac{\partial^2 \rho_c}{\partial^2 x} = \frac{x^2-2bt}{16\pi b^3 t^3} \exp(-\frac{x^2+y^2}{4bt}) \tag{20}$$

By symmetry the second y derivative is

$$\frac{\partial^2 \rho_c}{\partial^2 y} = \frac{y^2-2bt}{16\pi b^3 t^3} \exp(-\frac{x^2+y^2}{4bt}) \tag{21}$$

and hence

$$\nabla^2 \rho_c = \frac{x^2+y^2-4bt}{16\pi b^3 t^3} \exp(-\frac{x^2+y^2}{4bt}) \tag{22}$$

(b) We now consider the case when b = 1 and 49 point sources of chemicals are introduced at t = 0 with different strengths, on a 7×7 regular lattice covering the coordinates x = -3,-2,...,3 and y = -3,-2,...,3. The concentration satisfies

$$\rho(\mathbf{x}, t) = \sum_{k=0}^{48} \lambda_k \rho_c(\mathbf{x} - \mathbf{v}_k, t) \tag{23}$$

where $\mathbf{v}_k$ is the kth lattice site and λ_k is the strength of the chemical introduced at that site. Two hundred measurements, $\rho_M(\mathbf{x}_i, t)$, at locations $\mathbf{x}_i$ and at t = 4 are provided. Estimating the concentrations can be viewed as a linear least squares problem, finding the λ_k such that

$$S = \sum_{i=0}^{199} \left| \rho_M(\mathbf{x}_i, t) - \sum_{k=0}^{48} \lambda_k \rho_c(\mathbf{x}_i - \mathbf{v}_k, t) \right| \tag{24}$$

Even though Eq. 24 is quite complicated and involves the the expression for ρ_c, the parameters λ_k still enter linearly, and hence it can be solved using the linear least squares approach. The function `part_b()` in `p6_diffusion.py` computes the λ_k and prints them. They are all positive, with a maximum value of approximately 24.

(c) Suppose that the measurements have some experimental error,so that the measured values $\tilde{\rho}_i$ in the file are related to the true values ρ_i according to

$$\tilde{\rho}_i = \rho_i + e_i \tag{25}$$

The function `part_c()` in `p6_diffusion.py` performs a sample of N computations of the λ_k when each of the ρ_M are perturbed by a small normally distributed shift with mean 0 and variance 10^{-8}. The obtained standard deviations for the λ_k at four lattice sites are: 22268 at $(0,0)$, 14034 at $(1,1)$, 2868 at $(2,2)$, and 117 at $(3,3)$. They show much larger variations than the actual λ_k values that were measured in part **(b)**. The largest errors are at the central (0, 0) lattice site, which is reasonable since it is furthest away from any of the measurements in the file, thus making it most difficult to estimate.

(d) A common mistake here is that the floating point values λ_k are not rounded (e.g. using `round()`) as requested but rather truncated (e.g. using `int()`), which leads to incorrect images. The encoded message is "AM205"