1 Numerical differentiation formulas

Suppose we want to compute the $n$th derivative $f^{(n)}(x)$ of a function $f : \mathbb{R} \to \mathbb{R}$ at a point $x \in \mathbb{R}$. A $(2d + 1)$-point approximation of $f^{(n)}(x)$ has the form

$$
\hat{f}_d^{(n)}(x) = \frac{1}{\delta^n} \sum_{k=-d}^{+d} c_k f(x + k\delta),
$$

where $c_{-d}, \ldots, c_{+d} \in \mathbb{R}$ are coefficients that must be determined, and $\delta$ is the grid size. (We assume that the function $f$ is known on a grid of equally spaced points, and $\delta$ is the distance between points where $f$ is known; the points where $f$ is known are called sample points).

(a) Suppose $f$ has a Taylor-series expansion at $x$:

$$
f(z) = \sum_{m=0}^{\infty} \frac{f^{(m)}(x)}{m!} (z - x)^m
$$

Show that we can express $\hat{f}_d^{(n)}(x)$ as

$$
\hat{f}_d^{(n)}(x) = \sum_{m=0}^{\infty} \alpha_m(n, d) \delta^{m-n} f^{(m)}(x).
$$

Give an expression for $\alpha_m(n, d)$.

(b) We need to choose $2d + 1$ coefficients: $c_{-d}, \ldots, c_{+d}$. This suggests that we can control $2d + 1$ of the coefficients $\alpha_m(n, d)$. Suppose we choose the $c_k$ such that

$$
\alpha_m(n, d) = \begin{cases} 1 & m = n, \\ 0 & m \in \{0, \ldots, 2d\} - \{n\} \end{cases}
$$

Then, we have that

$$
\hat{f}_d^{(n)}(x) = \sum_{m=0}^{\infty} \alpha_m(n, d) \delta^{m-n} f^{(m)}(x) = f^{(n)}(x) + O(\delta^{2d-n+1}).
$$

In other words, the error in our approximation of $f^{(n)}(x)$ decreases like $\delta^{2d-n+1}$, where $\delta$ is the distance between sample points. Explain how to choose the $c_k$ in order to satisfy the given conditions on the $\alpha_m(n, d)$. Report the values of $c_k$ you get with $n = 2$, and $d = 1, 2, 3$. 
(c) Consider the function

\[ f(x) = \exp(\cos(10x)) \].

Compute the second derivative of \( f(x) \). (We want you to differentiate the function using the rules of calculus, so you can compare your numerical differentiation rules to the exact value of the derivative.) For \( n = 2 \), and \( d = 1, 2, 3 \), make a plot of the log approximation error \( \log(|\hat{f}^{(n)}(1) - f^{(n)}(1)|) \) versus the negative log grid size \(-\log(\delta)\), using values of \( \delta \) ranging from \( 10^{-6} \) to \( 10^{-1} \). Briefly comment on the results.

2 Fitting a sinusoid to data

Consider the data set shown in figure 1.

![Figure 1 - a sinusoidal data set](image)

The file `fit_sinusoid_data.m` defines the data points

\[(t_1, x_1), \ldots, (t_N, x_N)\].

We want to fit a sinusoidal model to the data:

\[ \hat{x}(t) = A \cos(2\pi ft + \phi) + b, \]

where \( A \in \mathbb{R} \) is the amplitude, \( f \in \mathbb{R} \) is the frequency, \( \phi \in \mathbb{R} \) is the phase, and \( b \in \mathbb{R} \) is the offset.

(a) Explain how to use the Gauss-Newton method to choose the parameters \( A, f, \phi, \) and \( b \) in order to minimize the sum of squared errors:

\[ \sum_{i=1}^{N} (\hat{x}(t_i) - x_i)^2. \]
(b) Apply your method to the data given in `fit_sinusoid_data.m`. Explain how you chose the initial parameter estimates. Report your final estimates of the parameters, and the sums of squared errors for your initial and final parameter estimates. On a single set of axes, plot the data, the sinusoid corresponding to your initial parameter estimates, and the sinusoid corresponding to your final parameter estimates.

3 Minimum-percent-error regression

Consider the data set shown in figure 2.

![Figure 2](image-url)

Figure 2 — a data set to be fit using minimum-percent-error regression

The file `percent_error_regression_data.m` defines the data points 

\[(x_1, y_1), \ldots, (x_m, y_m).\]

We want to fit a linear model to the data:

\[\hat{y}(x) = ax + b,\]

where \(a, b \in \mathbb{R}\) are parameters.

(a) Fit a line to the data using least-squares: that is, find \(a_{ls}\) and \(b_{ls}\) that minimize the sum of squared errors

\[\sum_{i=1}^{m} (\hat{y}(x_i) - y_i)^2.\]

Report your values of \(a_{ls}\) and \(b_{ls}\). Submit a plot with the data points \((x_i, y_i)\), and the least-squares line \(y = a_{ls}x + b_{ls}\).
(b) The data points seem to follow a line closely for small values of $y$, but there seem to be large deviations from the linear trend for large values of $y$. In this case, it may be reasonable to choose parameter estimates $a_{\text{mpe}}$ and $b_{\text{mpe}}$ that minimize the total percent error:

$$\sum_{i=1}^{m} 100 \times \frac{|\hat{y}(x_i) - y_i|}{|y_i|}.$$ 

Explain how to use iteratively reweighted least squares to compute $a_{\text{mpe}}$ and $b_{\text{mpe}}$. In particular, what is the weight function, and what is the update equation? Apply your method to the example data; use $a_{\text{ls}}$ and $b_{\text{ls}}$ as your initial values. Report your final estimates of $a_{\text{mpe}}$ and $b_{\text{mpe}}$. Submit a plot with the data points $(x_i, y_i)$, and the minimum-percent-error line $y = a_{\text{mpe}}x + b_{\text{mpe}}$.

(c) The data points were generated using the model

$$y_i = (1 + \epsilon_i)(ax_i + b), \quad i = 1, \ldots, m,$$

where $\epsilon_i$ is uniformly distributed on $[-0.25, +0.25]$, $a = 2$, and $b = 8$. In light of this revelation, briefly compare the least-squares and minimum-percent-error lines. In particular, give the root mean squared errors in $(a_{\text{ls}}, b_{\text{ls}})$ and $(a_{\text{mpe}}, b_{\text{mpe}})$ as approximations of $(a, b)$.

4 Surface imaging using a multimode fiber

Suppose we want to use an optical fiber to image a surface. In particular, we want to determine the profile of the surface, which we can describe by a vector $z \in \mathbb{R}^m$, where $z_i$ is the elevation of the $i$th reference point of the surface relative to some baseline value. An example of a surface is shown in figure 3.

![Figure 3](image)

**Figure 3** – representation of a surface

The fiber has $n$ different modes, which correspond to different patterns of light intensity that appear at the imaging end of the fiber. We can describe the pattern of light intensity corresponding to the $i$th mode using a vector $\phi_i \in \mathbb{R}^m$, which is a discretized version of the pattern of light intensity, as shown in figure 4.
Each of the modes can be excited independently, and we let $x_i$ denote the excitation applied to the $i$th mode. The pattern of light intensity at the imaging end of the fiber is a superposition of the modes:

$$w = x_1\phi_1 + \cdots + x_n\phi_n.$$ 

If we shine the light coming out of the imaging end of the fiber onto a surface, then some amount of light is reflected back into the fiber. We assume that the amount of light reflected back into the fiber is

$$r = w_1z_1 + \cdots + w_mz_m.$$ 

We get a noisy measurement $y = r + \epsilon$ of $r$ at the end of the fiber opposite the imaging surface. The complete experiment setup is shown in figure 5.

The file `multimode_fiber_data.m` defines the following variables.

- **Phi**, an $m \times n$ matrix whose $j$th column is $\phi_j$
- **Y**, an $m \times k$ matrix giving measurements of $y$ (more details are given below)
- **ztrue**, a vector of length $m$ giving the true surface profile

(a) **Spot formation.** If we can choose the excitation vector $x$ in such a way that $w = e_i$, then the resulting measurement $y$ is a noisy measurement of $z_i$. This process is called forming a spot at location $i$. The total energy used to excite the fiber is $\|x\|^2$. In practice, the amount of energy required to form an exact spot may be prohibitively large (for example, applying excitations that are too large may damage the fiber). Thus, we want to find a set of excitation vectors $x^{(1)}, \ldots, x^{(m)} \in \mathbb{R}^n$ such that $x^{(i)}$ gives
an approximate spot at location $i$, while also using as little input energy as possible. We can balance these objectives by minimizing the cost

$$J = \sum_{i=1}^{m} \| w(x^{(i)}) - e_i \|^2 + \lambda \sum_{i=1}^{m} \| x^{(i)} \|^2,$$

where $\lambda > 0$ is a tradeoff parameter. For a given value of $\lambda$, explain how to choose $x^{(1)}, \ldots, x^{(m)}$ in order to minimize $J$. Apply your method to the example data, and make a plot of the tradeoff curve for the costs

$$J_1 = \sum_{i=1}^{m} \| w(x^{(i)}) - e_i \|^2, \quad \text{and} \quad J_2 = \sum_{i=1}^{m} \| x^{(i)} \|^2.$$

For $\lambda = 10$, report the optimal value of $J$, and make a plot of $w(x^{(15)})$. Additionally, mark the point corresponding to $\lambda = 10$ on your tradeoff curve.

(b) *Surface imaging.* Suppose you decide to use the values of $x^{(1)}, \ldots, x^{(m)}$ corresponding to $\lambda = 10$. You take $k$ measurements of $y$ with each of the $x^{(i)}$. The data file defines a matrix of noisy measurements $Y \in \mathbb{R}^{m \times k}$, where $Y_{ij}$ is the $j$th measurement taken with excitation vector $x^{(i)}$. Explain how to choose an estimate $\hat{z} \in \mathbb{R}^m$ that minimizes the sum of squared errors:

$$E = \sum_{i=1}^{m} \sum_{j=1}^{k} (\hat{y}_i - Y_{ij})^2,$$

where we define the vector $y \in \mathbb{R}^m$ such that

$$\hat{y}_i = w_1(x^{(i)}) \hat{z}_1 + \cdots + w_m(x^{(i)}) \hat{z}_m.$$

Apply your method to the example data. Make a plot of the estimated surface profile $\hat{z}$.

(c) Explain how to choose $\hat{z}^{(1)}$ in order to minimize

$$E_1 = \sum_{i=1}^{m} (\hat{y}_i^{(1)} - Y_{i1})^2,$$

where

$$\hat{y}_i^{(1)} = w_1(x^{(i)}) \hat{z}_1^{(1)} + \cdots + w_m(x^{(i)}) \hat{z}_m^{(1)}.$$

(Intuitively, we only use the first measurement taken with each excitation vector $x^{(i)}$.) Make a plot of the estimated surface profile $\hat{z}^{(1)}$. Report the RMS errors in $\hat{z}$ and $\hat{z}^{(1)}$ as approximations of the true $z$:

$$E_{\text{rms}} = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (z_i - \hat{z}_i)^2}, \quad \text{and} \quad E_{1\text{rms}} = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (z_i - \hat{z}_i^{(1)})^2}.$$

Compare $\hat{z}$ and $\hat{z}^{(1)}$ to the true surface profile. Briefly comment on your results.
5 Designing an equalizer for backwards-compatible wireless transceivers

You want to design the equalizer for a new line of wireless handheld transceivers (more
commonly called walkie-talkies). The transmitter for the new line of transceivers has already
been designed (and cannot be changed) – if the input signal is $x \in \mathbb{R}^n$, then the transmitted
signal is $y = A_{\text{new}}x \in \mathbb{R}^m$, where $A_{\text{new}} \in \mathbb{R}^{m \times n}$ is known. An equalizer for $A_{\text{new}}$ is a matrix
$B \in \mathbb{R}^{n \times m}$ such that $By = x$ for every $x \in \mathbb{R}^n$.

The new line of transceivers will replace an older model. Given an input signal $x \in \mathbb{R}^n$,
the old line of transceivers transmit a signal $y_{\text{old}} = A_{\text{old}}x \in \mathbb{R}^m$, where $A_{\text{old}} \in \mathbb{R}^{m \times n}$ is
known. In addition to providing exact equalization for the new line of transceivers, you want
your equalizer to be able to at least partially equalize signals transmitted using the old line
of transceivers. In other words, to the extent that it is possible, you want the new line of
transceivers to be backwards compatible with the old line of transceivers.

(a) Explain how to find an equalizer $B$ that minimizes

$$J = \|BA_{\text{old}} - I\|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} (BA_{\text{old}} - I)_{ij}^2$$

among all $B$ that exactly equalize $A_{\text{new}}$. Such a $B$ is an exact equalizer for $A_{\text{new}}$, and
an approximate equalizer for $A_{\text{old}}$. State any assumptions that are needed for your
method to work.

(b) The file `backwards_compatible_transceiver_data.m` defines the following variables.

- $A_{\text{new}}$, the $m \times n$ matrix that describes the transmitter used in the new line of
  transceivers
- $A_{\text{old}}$, the $m \times n$ matrix that describes the transmitter used in the old line of
  transceivers
- $x$, a vector of length $n$ that serves as an example input signal

Apply your method to this example data. Report the optimal value of $J$. The pseudo-
doinverse $A_{\text{new}}^\dagger$ is another exact equalizer for $A_{\text{new}}$. Compare the optimal value of $J$, and
the value of $J$ achieved by $A_{\text{new}}^\dagger$.

(c) The example signal $x$ defined in the data file is a binary signal. Form the signal
$y_{\text{old}} = A_{\text{old}}x$ transmitted by the old line of transceivers, and construct an estimate of
$x$ by equalizing $y_{\text{old}}$ using $B$, and then rounding the result to a binary signal. More
concretely, compute the estimate $\hat{x} \in \mathbb{R}^n$, where

$$\hat{x}_i = \begin{cases} 1 & (By_{\text{old}})_i > \frac{1}{2}, \\ 0 & \text{otherwise}. \end{cases}$$

Report the bit error rate of your estimate, which is defined as

$$\frac{1}{n} \sum_{i=1}^{n} I(x_i \neq \hat{x}_i),$$
where $I(x_i \neq \hat{x}_i)$ is an indicator function:

$$I(x_i \neq \hat{x}_i) = \begin{cases} 
1 & x_i \neq \hat{x}_i, \\
0 & \text{otherwise}.
\end{cases}$$

Similarly, report the bit error rate if $A_{\text{new}}^t$ is used as the equalizer.