1. (10 points) (Adapted from problem 3.2 of M. T. Heath, *Scientific Computing: an Introductory Survey*, 2nd ed., McGraw–Hill, 2002.) A common problem in surveying is to determine the altitudes of a series of points with respect to some reference point. The measurements are subject to error, and so multiple measurements are made of each altitude: one with respect to the reference point and others with respect to all of the other points. This results in an overdetermined linear system, which is solved in the least-squares sense to smooth out the errors.

Suppose there are four points whose altitudes $x_1$, $x_2$, $x_3$, and $x_4$ are to be determined, and that the following measurements are taken:

\[
\begin{align*}
  x_1 &= 2.95 & x_2 &= 2.95 & x_3 &= -1.45 & x_4 &= -1.45 \\
  x_1 - x_2 &= 1.23 & x_1 - x_3 &= 4.45 & x_1 - x_4 &= 1.61 \\
  x_2 - x_3 &= 3.21 & x_2 - x_4 &= 0.45 & x_3 - x_4 &= -2.75
\end{align*}
\]

It’s obvious that these ten equations in four unknowns can’t be satisfied exactly.

Set up the matrix $A$ and the vector $b$ for the associated least-squares problem. Form the matrix $A^T A$ and the vector $A^T b$ for the normal equation. Obtain the solution of the least-squares problem by solving the normal equation using Cholesky decomposition. (You may use MATLAB’s `chol` for this.) Determine $\kappa_2(A^T A)$, the 2-norm condition number of $A^T A$. (MATLAB’s `cond` can be used for this.) Print out and hand in $A$, $b$, $A^T A$, $A^T b$, $\kappa_2(A^T A)$, and your computed solution of the least-squares problem, together with a brief explanation of why we can safely obtain the solution using the normal equation in this case.

Remark: The easiest way to produce the solution of the least-squares problem is just to use MATLAB’s “backslash” command. On an overdetermined system $Ax = b$, typing “$A \backslash b$” produces the least-squares solution by computing the QR decomposition of $A$ using Householder transformations with column pivoting. Thus you can check the solution that you compute using the normal equation and Cholesky decomposition just by typing “$A \backslash b$.”

to fit data \((t_i, y_i), 1 \leq i \leq m\), with a function \(f(t) = \frac{(x_1 + x_2 t)}{(1 + x_3 t)}\). This can be done using the method of least-squares, but the straightforward least-squares formulation

\[
\min R(x) \equiv m \sum_{i=1}^{m} \left( y_i - \frac{x_1 + x_2 t_i}{1 + x_3 t_i} \right)^2
\]

is not linear in the unknown parameters \(x_1, x_2,\) and \(x_3\). There are excellent algorithms for solving such *nonlinear least-squares problems*; however, one can obtain similar (if not the same) results in this case by solving the linear least-squares problem

\[
\min R(x) \equiv \sum_{i=1}^{m} [(1 + x_3 t_i) y_i - (x_1 + x_2 t_i)]^2. \tag{1}
\]

Set up and solve the linear least-squares problem (1) for the following world population data, in which each \(y_i\) is the world population (in billions) in the year \(t_i\):

<table>
<thead>
<tr>
<th>(t_i)</th>
<th>1000</th>
<th>1650</th>
<th>1800</th>
<th>1900</th>
<th>1950</th>
<th>1960</th>
<th>1970</th>
<th>1980</th>
<th>1990</th>
</tr>
</thead>
<tbody>
<tr>
<td>(y_i)</td>
<td>0.340</td>
<td>0.545</td>
<td>0.907</td>
<td>1.61</td>
<td>2.51</td>
<td>3.15</td>
<td>3.65</td>
<td>4.20</td>
<td>5.30</td>
</tr>
</tbody>
</table>

Plot the resulting function \(f(t)\) at 101 equally spaced \(t\)-values in the interval \([1000, 1990]\). Plot the data \((t_i, y_i), 1 \leq i \leq m\), on the same figure. Print out and hand in your plot, together with the values \(x_1, x_2,\) and \(x_3\) that you obtain.

**Remark:** MATLAB’s `plot` command can be used for the plotting. Typing “help plot” or “doc plot” will provide instructions for using `plot` and examples of placing two or more plots on the same figure.