Homework 0 – Problem 3

An accurate estimate of the heat capacity (C_P) for a particular inlet mixture is critical to an efficient separation, so you decide to carry out a series of heat capacity measurements at different temperatures in the operating range and fit a polynomial model to compute specific heat capacity as a function of temperature. To find the polynomial that best fits the data, we seek the values of the weights $\{d_i\}_{i=0}^{n_d}$ that give the best fit (in a least squares sense) for the linear system of n_T equations, where n_T is the number of T values at which $C_P(T)$ was measured:

$$d_0 + d_1 T_j + d_2 T_j^2 + \cdots + d_{n_d} T_j^{n_d} = C_P(T_j), \ \ j = 1, \ldots, n_T$$

The measured values of the heat capacity have a relative error bound of 0.03 and absolute error bound of 50 J/(kg-K). Use the 2-norm throughout this problem for consistency.

1.
For the case of a square system with $n_T = n_d + 1$ there will be values of **d** that fit the data. Express the system as **Ad = c**, where **d** contains the weights. Write out the forms of **A** and **c**.

<u>Matrix-vector form of the polynomial fit for $C_P(T)$</u>
Write the n_T equations for specific heat capacity assuming $n_T = n_d + 1$:

$$\begin{gathered}
d_0 + d_1 T_1 + d_2 T_1^2 + \cdots + d_{n_d} T_1^{n_d} = C_P(T_1) \\
d_0 + d_1 T_2 + d_2 T_2^2 + \cdots + d_{n_d} T_2^{n_d} = C_P(T_2) \\
d_0 + d_1 T_3 + d_2 T_3^2 + \cdots + d_{n_d} T_3^{n_d} = C_P(T_3) \\
\vdots \\
d_0 + d_1 T_{n_T} + d_2 T_{n_T}^2 + \cdots + d_{n_d} T_{n_T}^{n_d} = C_P(T_{n_T})
\end{gathered}$$

In vector matrix form:

$$\begin{bmatrix} 1 & T_1 & T_1^2 & \cdots & T_1^{n_d} \\ 1 & T_2 & T_2^2 & \cdots & T_2^{n_d} \\ 1 & T_3 & T_3^2 & \cdots & T_3^{n_d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & T_{n_T} & T_{n_T}^2 & \cdots & T_{n_T}^{n_d} \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{n_d} \end{bmatrix} = \begin{bmatrix} C_P(T_1) \\ C_P(T_2) \\ C_P(T_3) \\ \vdots \\ C_P(T_{n_T}) \end{bmatrix}$$

From this, the forms of **A** and **c** can easily be determined:

$$\boxed{\mathbf{A} = \begin{bmatrix} 1 & T_1 & T_1^2 & \cdots & T_1^{n_d} \\ 1 & T_2 & T_2^2 & \cdots & T_2^{n_d} \\ 1 & T_3 & T_3^2 & \cdots & T_3^{n_d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & T_{n_T} & T_{n_T}^2 & \cdots & T_{n_T}^{n_d} \end{bmatrix} \qquad \mathbf{c} = \begin{bmatrix} C_P(T_1) \\ C_P(T_2) \\ C_P(T_3) \\ \vdots \\ C_P(T_{n_T}) \end{bmatrix}}$$

Commented [U1]: You should title your homework problem.

Commented [U2]: Short introductions to the problem statement can be nice to define your nomenclature and set the stage for your answers. However, it is not necessary and if you do this limit your description to be as short as possible. MAKE IT LESS THAN a quarter of a page.

Commented [U3]: Adding lines/spacing can alert the grader to the start of a new part of the problem. Again, a nice feature but not necessary.

Commented [U4]: Boxing your final answers and giving short introductory sentences can be very helpful for graders. I suggest you do this.

2.
Let the T measurements be evenly spaced between 273 K and 373 K, using n_d = 2,3,…,8 (with n_T = 3,4,…,9) plot cond(**A**) and norm($\mathbf{A}^{-1}$) on the same axes as a function of n_T.

Discussion of the algorithm to produce A, cond(A), and norm(A^{-1})

The paulson_HW2_P2.m file submitted online accesses a user-written function (condA_norminvA_errorPredCp()), reference for more information regarding the specifics of the algorithm) that generates the matrix **A** for each specified size (3 through 9). Using **A**, the function returns both its condition number and the norm of its inverse using the built-in MATLAB® functions cond() and norm(), respectively.

The condA_norminvA_errorPredCp() function generates the matrix **A** using two nested for loops, the first of which spans the rows and the second of which spans the columns. The first row (i = 1) corresponds to the lower bound for temperature (T_low). The second loop creates entries in matrix **A** (1 to n_T) that correspond to the proper power of temperature (i.e. T^0 to T^{nT-1}). After each row is filled with the correct elements of T along its columns, the temperature value (T) is adjusted to an evenly spaced value between T_low and T_high (through the addition step of T_spacing). This step occurs before the row value, i, is updated, which is important because each row in **A** must correspond to a different temperature value. The algorithm repeats this process until increment i reaches the final row in **A** (i = n_T).

Since the problem statement asks for parameters at various n_T values, I decided to add a third nested loop above the two that generate **A** which runs from a lower to an upper bound on n_T. After the two loops (discussed above) generate **A**, the condition number, norm of **A** inverse, and absolute error of C_P (see part 4) are stored in vectors based on the corresponding n_T value. The outer loop repeats this process from a user-input lower bound of n_T (nT_low) to a user-input upper bound of n_T (nT_high) After the three loops are completely iterated, these parameter vectors (that have stored every specified parameter for all n_T's input to the function) are designated as the function output. These vectors (cond_A, norm_invA, abs_error_Cp_pred) are then plotted versus n_T in the main function.

Commented [U5]: Here is an example of going TOO FAR. I describe the submitted code in detail. This is not necessary especially for this problem where we want you to worry more about the equations than any algorithm (the algorithm is very simple, just make an A matrix and compute the values). It is completely fine to do this, but a waste of time. I would recommend given a much shorter description (e.g., "My submitted MATLAB code generates matrix A from part 1 and computes the condition number of norm of A inverse using built-in functions").

Condition number of A and norm of A inverse

The condition number of **A** and the norm of **A** inverse are plotted versus n_T below on Figure P2.1. The algorithm used to generate this plot is described above (reference paulson_HW2_P2.m for more information regarding any part of problem 2).

Figure P2.1 shows that the condition number of **A** grows in order of magnitudes with small changes in n_T. Additionally, their linear trends on a semilog plot imply that these parameters have an exponential correlation with respect to n_T.

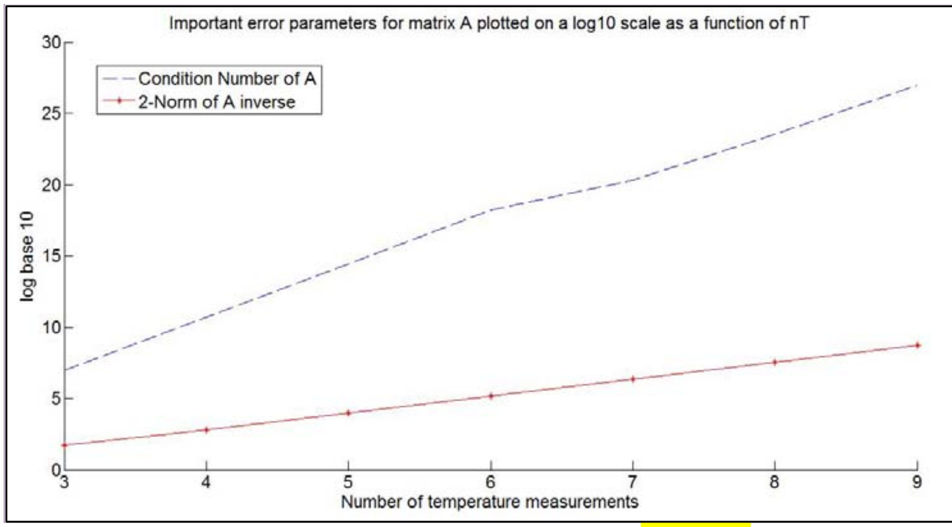

Figure P2.1. Condition Number and norm of inverse for **A** on a log10 scale as a function of n_T

Commented [U6]: The plot is commented well and nicely labeled. You should do this in your reports. Notice that the quality of the plot is not very good, you can achieve much better quality in a variety of ways (please ask Joel for more information on this).

3.
Again for the square system, compute relative and absolute error bounds on **d** for several values of n_d. Let the T measurements be evenly spaced between 273 and 373 K for the same n_T and n_d values in part 2. Plot the relative and absolute error bounds as a function of n_T. Note: Express your error bounds in terms of the relative and absolute error bounds in heat capacity.

Error matrix rearrangement
The matrix vector form of the equation for heat capacity is:

Commented [U7]: Derivations such as this can be nice and helpful for you. I would recommend that you include it when it is not obvious as you can reference it during exams (you should be allowed to have all your notes and graded HWs). Here, this is fairly obvious so I could have jumped to the boxed equation directly.

$$\mathbf{Ad} = \mathbf{c}$$

Add a perturbation (**Δd**) in **d** that creates a perturbation (**Δc**) in **c**

$$\mathbf{A}(\mathbf{d} + \Delta\mathbf{d}) = \mathbf{c} + \Delta\mathbf{c}$$

Distribute **A** on the LHS

$$\mathbf{Ad} + \mathbf{A}\Delta\mathbf{d} = \mathbf{c} + \Delta\mathbf{c}$$

Notice the original equation **Ad** = **c**, substitute this into the equation above

$$\mathbf{c} + \mathbf{A}\Delta\mathbf{d} = \mathbf{c} + \Delta\mathbf{c}$$

$$\boxed{\mathbf{A}\Delta\mathbf{d} = \Delta\mathbf{c}}$$

Induced Norm Inequality Proof

The induced norm of matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ by the vector $\mathbf{x} \in \mathbb{R}^N$ is:

$$\|\mathbf{A}\| = \max\left(\frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|}\right) \longleftarrow$$

NOTE: THROUGHOUT THIS PROBLEM
$\|\mathbf{A}\|$ implies 2-norm of matrix $\mathbf{A}$

By the definition of the maximum

$$\frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|} \le \|\mathbf{A}\|$$

$$\boxed{\|\mathbf{Ax}\| \le \|\mathbf{A}\|\|\mathbf{x}\|, \ \forall \Delta\mathbf{x} \in \mathbb{R}^N : \mathbf{x} \neq \mathbf{0}}$$

*Induced norm inequality proof (referenced throughout the following sections)

Commented [U8]: This was an example of something given in the notes that you could have referenced. NOT REQUIRED AND A WASTE OF TIME. You would still need to give the final boxed equation but you would not have to go through any details.

Absolute Error of d

In order to calculate the absolute error of $\mathbf{d}\left(\|\Delta\mathbf{d}\|\right)$, left multiply the above relationship by the inverse of $\mathbf{A}$

$$\Delta\mathbf{d} = \mathbf{A}^{-1}\Delta\mathbf{c}$$

Take the 2-norm of the above relationship

$$\|\Delta\mathbf{d}\| = \left\|\mathbf{A}^{-1}\Delta\mathbf{c}\right\|$$

From the induced norm inequality (proved in part 3)

$$\boxed{\|\Delta\mathbf{d}\| \le \left\|\mathbf{A}^{-1}\right\|\|\Delta\mathbf{c}\|}$$

*Absolute error inequality expression

Relative Error of d

In order to calculate the relative error of **d**, take the 2-norm of the original matrix equation:

$$\|\mathbf{Ad}\| = \|\mathbf{c}\|$$

Divide the relative error inequality (above) by this expression

$$\frac{\|\Delta\mathbf{d}\|}{\|\mathbf{Ad}\|} \le \frac{\left\|\mathbf{A}^{-1}\right\|\|\Delta\mathbf{c}\|}{\|\mathbf{c}\|}$$

Multiply the norm of **Ad** on both sides of the equation

$$\|\mathbf{\Delta d}\| \le \frac{\|\mathbf{A}^{-1}\|\|\mathbf{\Delta c}\|}{\|\mathbf{c}\|}\|\mathbf{Ad}\|$$

From the induced norm inequality (proved in part 3)

$$\|\mathbf{\Delta d}\| \le \frac{\|\mathbf{A}^{-1}\|\|\mathbf{\Delta c}\|}{\|\mathbf{c}\|}\|\mathbf{A}\|\|\mathbf{d}\|$$

Divide both sides by the norm of **d**

$$\boxed{\frac{\|\mathbf{\Delta d}\|}{\|\mathbf{d}\|} \le \|\mathbf{A}^{-1}\|\|\mathbf{A}\|\frac{\|\mathbf{\Delta c}\|}{\|\mathbf{c}\|}}$$

*Relative error inequality expression

2-Norm of Δc

$$\|\mathbf{\Delta c}\| = \sqrt{\sum_{i=1}^{n_T} \Delta C p_i^{\,2}}$$

Expand the summation term

$$\|\mathbf{\Delta c}\| = \sqrt{\Delta C p_1^{\,2} + \Delta C p_2^{\,2} + \cdots + \Delta C p_{n_T}^{\,2}}$$

Since each of the measurement error terms, ΔC_{Pi} are equivalent, they can be group together n_T times. ΔC_P is the absolute measurement error in heat capacity, $\Delta C_P = 50$ J/kg-K (Given).

$$\|\mathbf{\Delta c}\| = \sqrt{n_T \cdot \Delta C p^2}$$

$$\|\mathbf{\Delta c}\| = \sqrt{n_T} \cdot |\Delta C p|$$

$$\boxed{\|\mathbf{\Delta c}\| = \sqrt{n_T} \cdot 50 \frac{J}{kgK}}$$

*2-Norm of the heat capacity error vector

Error bounds for d

The given absolute and relative error bounds for the measured heat capacities (**c**) are:

Absolute Error: $\|\mathbf{\Delta c}\| = \sqrt{n_T} \cdot 50 \frac{J}{kgK}$ Relative Error: $\frac{\|\mathbf{\Delta c}\|}{\|\mathbf{c}\|} = 0.03$

The absolute and relative errors for weights **d** can be calculated using the error inequalities derived above in part 3:

$$\|\Delta\mathbf{d}\| \le \|\mathbf{A}^{-1}\|\|\Delta\mathbf{c}\| \qquad \frac{\|\Delta\mathbf{d}\|}{\|\mathbf{d}\|} \le \|\mathbf{A}^{-1}\|\|\mathbf{A}\|\frac{\|\Delta\mathbf{c}\|}{\|\mathbf{c}\|}$$

$$\|\Delta\mathbf{d}\| \le \sqrt{n_T}\cdot 50\frac{J}{kgK}\|\mathbf{A}^{-1}\| \qquad \frac{\|\Delta\mathbf{d}\|}{\|\mathbf{d}\|} \le 0.03\|\mathbf{A}^{-1}\|\|\mathbf{A}\|$$

Define the norm of **A** times the norm of its inverse to be the condition number (cond(**A**)).

$$\boxed{\|\Delta\mathbf{d}\| \le \sqrt{n_T}\cdot 50\frac{J}{kgK}\|\mathbf{A}^{-1}\| \qquad \frac{\|\Delta\mathbf{d}\|}{\|\mathbf{d}\|} \le 0.03\cdot cond(\mathbf{A})}$$

*Used to calculate absolute and relative error bounds of **d** in terms of the error bounds of **c**

<u>Plot of the error bounds of d as a function of n_T</u>

The relative and absolute errors of C_P weights (**d**) are plotted as a function of n_T below on Figure P2.2. The methodology used to generate these error values for **d** is described above. (Figure P2.2 also shows cond(**A**) and the norm($\mathbf{A}^{-1}$) from part 2 for comparison)

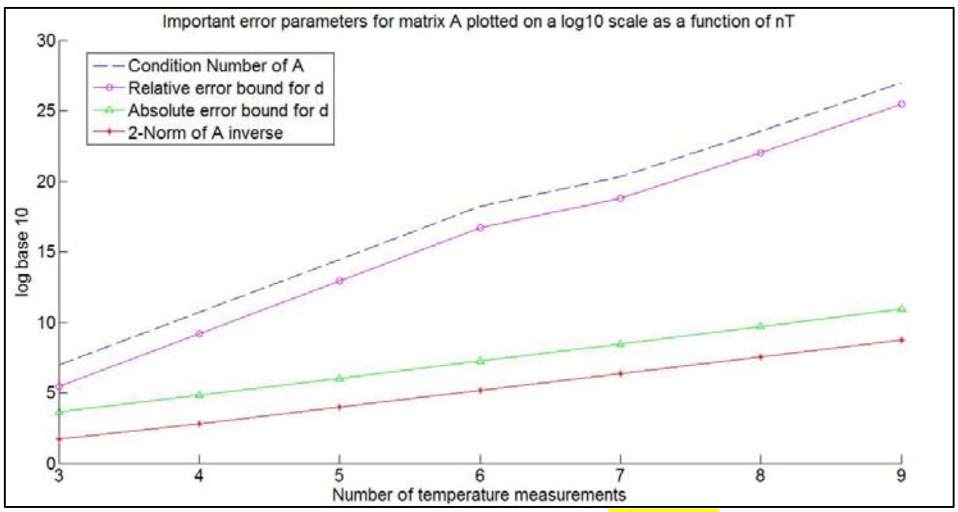

Figure P2.2. Relative and absolute error of **d** plotted on a log10 scale as a function of n_T

Figure P2.2 shows that the relative error bound of **d** follows the same trend as the condition number of **A** and the absolute error bound of **d** follows the same trend as the 2-norm of **A** inverse. Due to these trends the error bounds of **d** grow exponentially fast making this polynomial fit unsuitable for C_P as a function of T.

4.
Again for the square system, derive an expression bounding the absolute uncertainty of the specific heat capacity prediction from the fitted d for some arbitrary $\hat{T}$ between 273 K and 373 K, for arbitrary n_d, and justify the bound. Plot the numerical values for the bounds for the values of n_d and n_T above, for $\hat{T}$= 300 K.

Assume:

$$\begin{bmatrix} 1 & \hat{T} & \hat{T}^2 & \cdots & \hat{T}^{n_d} \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{n_d} \end{bmatrix} = C_P^{\mathrm{Pred}}(\hat{T}) = C_P^{True}(\hat{T}) + \Delta C_P(\hat{T})$$

And that there exists some $\hat{\mathbf{d}}$ within the error bound on $\mathbf{d}$ computed above such that:

$$\begin{bmatrix} 1 & \hat{T} & \hat{T}^2 & \cdots & \hat{T}^{n_d} \end{bmatrix} \begin{bmatrix} \hat{d}_0 \\ \hat{d}_1 \\ \hat{d}_2 \\ \vdots \\ \hat{d}_{n_d} \end{bmatrix} = C_P^{True}(\hat{T})$$

Subtracting these two equations yields:

$$\begin{bmatrix} 1 & \hat{T} & \hat{T}^2 & \cdots & \hat{T}^{n_d} \end{bmatrix} \left(\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{n_d} \end{bmatrix} - \begin{bmatrix} \hat{d}_0 \\ \hat{d}_1 \\ \hat{d}_2 \\ \vdots \\ \hat{d}_{n_d} \end{bmatrix} \right) = \Delta C_P(\hat{T})$$

Define $\mathbf{\Delta d}$ to be the difference between $\mathbf{d}$ and $\hat{\mathbf{d}}$ and, $\mathbf{T}$ to be the polynomial expansion of $\hat{T}$. **S**ubstitute these relations into the above equation

$$\mathbf{T}\Delta\mathbf{d} = \Delta C_P(\hat{T})$$

Take the 2-norm of both sides of the equation

$$\|\mathbf{T}\Delta\mathbf{d}\| = \| \Delta C_P(\hat{T}) \|$$

Using the matrix induced norm inequality: $\|\mathbf{T}\Delta\mathbf{d}\| \le \|\mathbf{T}\|\|\Delta\mathbf{d}\|$

$$\| \Delta C_P(\hat{T}) \| \le \|\mathbf{T}\| \|\Delta \mathbf{d}\|$$

Absolute error inequality from part 3: $\|\Delta \mathbf{d}\| \le \|\mathbf{A}^{-1}\| \|\Delta \mathbf{c}\|$

$$\| \Delta C_P(\hat{T}) \| \le \|\mathbf{T}\| \|\mathbf{A}^{-1}\| \|\Delta \mathbf{c}\|$$

Since ΔC_P is a scalar its 2-norm is equivalent to the absolute value of the quantity itself

$$| \Delta C_P(\hat{T}) | \le \|\mathbf{T}\| \|\mathbf{A}^{-1}\| \|\Delta \mathbf{c}\|$$

Where ΔC_P is the difference in the predicted and true heat capacity values
T is the row vector of $\hat{T}$ values substituted into the polynomial fit for C_P
A is the temperature matrix representing the C_P polynomial fit (part 1)
Δ**c** is the absolute measurement error bound for heat capacity = 50 J/(kg-K)
$\|\Delta \mathbf{c}\| = \sqrt{n_T} \cdot \Delta \mathbf{c}$ (reference part 3 for derivation)

*Used to calculate absolute uncertainty in predicted C_P values

Commented [U9]: Here is a nice example of a completely summarized equation. I am telling the graders everything they need to know for the final answer (including defining my terms) while referencing above for derivations. If you define things previously, it is ok to NOT define them again, but remember graders are human and make mistakes.

Predicted C_P uncertainty at $\hat{T}$ = 300 K

The absolute uncertainty in the predicted C_P value (defined as $|\Delta C_P|$) plotted versus n_T = 3,..,9 is shown below in Figure P2.3. The values of this plot were obtained using the condA_norminvA_errorPredCp() function in the submitted paulson_HW2_P2.m file. The algorithm is discussed in detail in part 2.

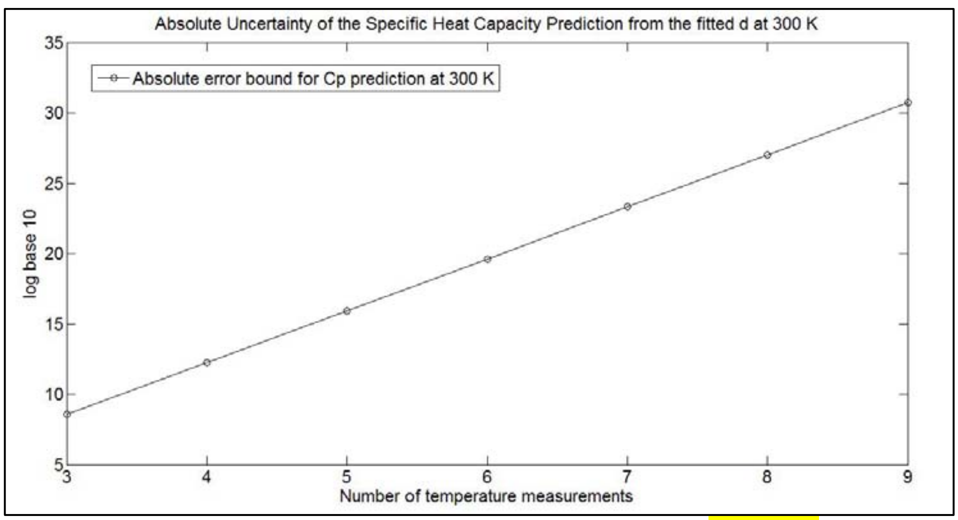

Figure P2.3. Absolute uncertainty in predicted C_P at 300 K (log10 scale)

Figure P2.3 shows that at $\widehat{T}$ = 300 K, the predicted specific heat capacity has an absolute error bound of order 30 which corresponds to a polynomial fit with nine temperature measurements (n_T = 9). These large magnitudes in the predicted C_P values are a direct result of the large-normed **T** vector and the ill-conditioned **A** matrix. Together, these parameters significantly amplify the small absolute error bound in the measured heat capacities.

Commented [U10]: I would recommend giving little descriptions of the plots such as this one so that the plot is not free-floating in space. You can keep it short and sweet.

5.
For the more general case $n_T > n_d + 1$, derive expressions for absolute error bounds for the least-squares estimates $\mathbf{d}^{LS}$ using the relation seen in Homework 0:

$$\mathbf{A}^T\mathbf{A}\mathbf{d}^{LS} = \mathbf{A}^T\mathbf{c}$$

Derivation
Starting with the general least-squares estimate equation:

$$\mathbf{A}^T\mathbf{A}\mathbf{d}^{LS} = \mathbf{A}^T\mathbf{c}$$

Add a perturbation ($\Delta\mathbf{d}^{LS}$) in $\mathbf{d}^{LS}$ that creates a perturbation ($\Delta\mathbf{c}$) in $\mathbf{c}$

$$\mathbf{A}^T\mathbf{A}(\mathbf{d}^{LS} + \Delta\mathbf{d}^{LS}) = \mathbf{A}^T(\mathbf{c} + \Delta\mathbf{c})$$

Distribute $\mathbf{A}^T\mathbf{A}$ on the LHS and $\mathbf{A}^T$ on the RHS

$$\mathbf{A}^T\mathbf{A}\mathbf{d}^{LS} + \mathbf{A}^T\mathbf{A}\Delta\mathbf{d}^{LS} = \mathbf{A}^T\mathbf{c} + \mathbf{A}^T\Delta\mathbf{c}$$

Notice the original equation $\mathbf{A}^T\mathbf{A}\mathbf{d}^{LS} = \mathbf{A}^T\mathbf{c}$, cancel on both sides of the equation

$$\boxed{\mathbf{A}^T\mathbf{A}\Delta\mathbf{d}^{LS} = \mathbf{A}^T\Delta\mathbf{c}}$$

*Least squares (LS) error equation

Multiply both sides of the LS error equation by the inverse of $\mathbf{A}^T\mathbf{A}$:

$$\Delta\mathbf{d}^{LS} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\Delta\mathbf{c}$$

Take the 2-norm of both sides of this equation

$$\left\|\Delta\mathbf{d}^{LS}\right\| = \left\|(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\Delta\mathbf{c}\right\|$$

Using similar ideas to the induced norm inequality (proved in part 4), the RHS norm can be shown to be less than the product of the individual vectors/matrices norms. This results in:

$$\boxed{\left\|\Delta\mathbf{d}^{LS}\right\| = \left\|(\mathbf{A}^T\mathbf{A})^{-1}\right\|\left\|\mathbf{A}^T\right\|\left\|\Delta\mathbf{c}\right\|}$$

*Notes: (1) $||\Delta\mathbf{d}^{LS}||$ is the relative error bound for the least square estimates ($\mathbf{d}^{LS}$)
(2) **A** and **c** are equivalent to those defined in part 1

6.
Explain the intuitive meaning of the ill-conditioning of this system

For a linear equation, **Ax** = **b**, the condition number of matrix **A** is said defined as the operator of two norms, cond(**A**) = ||**A**|| ||$\mathbf{A}^{-1}$||. The condition number is important when solving linear equations because it is an indicator as to how much a perturbation in **b** will alter the solution vector **x**. If the condition number is high, small changes in **b** can create large errors in **x**. Such a system is said to be ill-conditioned. This usually implies **A** has small-valued eigenvalues making it close to singular (zero-valued determinant) and/or the system is badly scaled.

In this problem, the A matrix equals (part 1):

$$\mathbf{A} = \begin{bmatrix} 1 & T_1 & T_1^2 & \cdots & T_1^{n_d} \\ 1 & T_2 & T_2^2 & \cdots & T_2^{n_d} \\ 1 & T_3 & T_3^2 & \cdots & T_3^{n_d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & T_{n_T} & T_{n_T}^{\ 2} & \cdots & T_{n_T}^{\ n_d} \end{bmatrix}$$

This matrix has a large magnitude discrepancy within each of its columns due to the "Taylor-series type" polynomial fit for heat capacity (C_P) in terms of temperature (T) which has increasing orders of T from 0 to n_d. Since the values of temperature are usually order 3, these polynomial factors result in extremely large magnitude differences from column to column. This intuitively means that either the chosen scale for the problem parameters are wrong or the fit for C_P as a function of temperature was a poor choice. For this particular problem, I believe the polynomial expansion fit was a poor choice since high order functions usually result in large magnitude terms (Terms >= 10^{10}). Additionally, these values span so many orders that it is difficult to estimate each term accurately. Moreover, the large-magnitude terms result in an enormous condition number for **A** which, in turn, takes the relatively small errors in **c** and amplifies them within the solution vector **d**.

I would recommend choosing a better function to fit C_P with respect to T instead of forcing data into random polynomial expansions and preforming least squares analysis blindly. This more in-depth analysis would require investigating how these variables normally interact with each other in various relationships. Hopefully, this newer fit would eliminate the large magnitude discrepancy between columns; however, rescaling the variables to some small finite range could also be a viable option (i.e. try using relationships that make the temperature and heat capacity dimensionless, maybe through the addition of other important parameters).

7.
Use Chebyshev polynomials in place of the T_is to construct matrix **A**. You can use the MATLAB function ChebyshevPoly. For better conditioning, scale the temperature so that the Chebyshev polynomial is evaluated on [-1, 1] by evaluating at (2T - T_{low} - T_{high}) / (T_{high} - T_{low}) rather than at T. Comment on the difference between the error bounds for T_is vs. Chebyshev polynomials on the rescaled temperature range.

Discussion of the Chebyshev polynomial algorithm

The paulson_HW2_P2.m file submitted online accesses a user-written function (chebyshev_A()), reference for more information regarding the specifics of the algorithm) that generates a modified matrix **A** for each specified size (3 through 9). Using the ChebyshevPoly() MATLAB function a polynomial is generated for every power of T in the assumed fit for C_P. This polynomial is then evaluated at a modified T value on the scale of [-1, 1].

The main idea behind this type of modification is to alter **A** so that its condition number is significantly lower effectively reducing the error bounds for the solution vector. The user-written chebyshev_A() function is just a slightly modified version of the cond_A_norminvA_errorPredCp() function discussed in detail in part 2 (reference for more information regarding the framework of this algorithm). The only change to this is inside the third nested for loop, which no longer stores some power of T to the matrix **A**, but instead performs this substitution/evaluation for the chebyshev polynomial discussed in the paragraph above. From this, it can be seen that a multiple polynomials (evaluated along the altered T scale) are stored in **A** which effectively adds more entries to the spanning set of the matrix **A**. Additionally, the rescaled temperature reduces the extremely large magnitude values dealt with in part 2 (i.e. $\sim 273^9$).

Plot of the error bounds of d as a function of n_T evaluated using Chebyshev polynomials

The relative and absolute errors of C_P weights (**d**) calculated using Chebyshev polynomials are plotted as a function of n_T below on Figure P2.4.

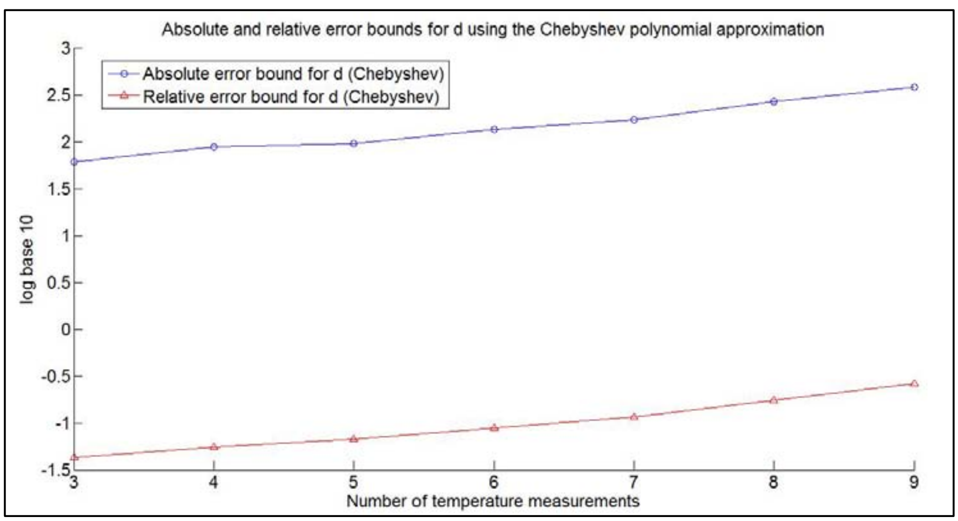

Figure P2.4. Absolute and relative error bounds for **d** calculated using Chebyshev polynomials

Error bound comparison for d

For easier comparison, the relative and absolute errors of **d** calculated using the full polynomial expansion (part 2) and Chebyshev polynomials (part 7) are plotted together on Figure P2.5 below. The full polynomial fit produces an absolute error range for **d** of ~10^4 to ~10^{10} (over n_T = 3 to 9) whereas the Chebyshev polynomials produce a range of ~$10^{1.75}$ to ~$10^{2.5}$. Furthermore, the full polynomial fit produces a relative error range for **d** of ~10^5 to ~10^{26} (over n_T = 3 to 9) whereas the Chebyshev polynomials produce a range of ~$10^{-1.5}$ to ~$10^{-0.5}$. This shows that the both the absolute and relative error bounds for **d** drop significantly when Chebyshev polynomials are used in place of the full polynomial expansion to calculate **A**.

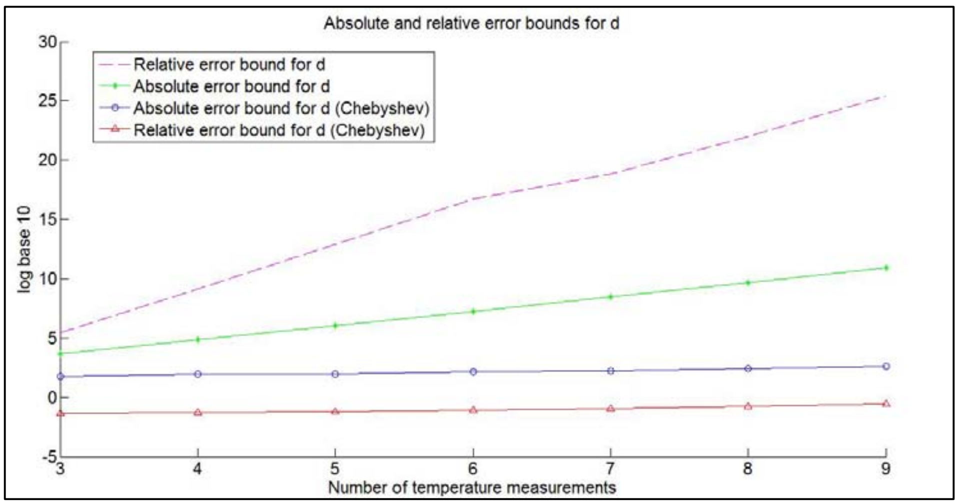

Figure 2.5. Error bounds for **d** using the full polynomial expansion and Chebyshev polynomials