10.34 Numerical Methods Applied to Chemical Engineering Fall 2015

Homework #0: MATLAB®and Linear Algebra

Please reference the homework guidelines for how to format and submit this assignment.

Problem 1. Cylinder with heat source

Crystalline silicon is annealed to remove defects, before it is sliced to make the substrate for computer chips and photovoltaic devices. One way to do this is to run an electric current down the long cylindrical crystal rods. In this problem you will compute the steady-state temperature distribution inside the rod during this process.

Consider a cylinder of radius R with uniformly distributed heat sources with heat generation $\dot{q}$ and temperature-dependent thermal conductivity $k(T)$. If the cylinder is sufficiently long so that the temperature can be considered a function of radius only, the differential equation for temperature can be written as:

$$\frac{d^2T}{dr^2} + \frac{1}{r}\frac{dT}{dr} + \frac{\dot{q}}{k} = 0$$

This second-order differential equation can be converted into a system of two coupled first-order differential equations by defining new variables u_1 and u_2 as follows:

$$\begin{aligned} u_1 &= T \\ u_2 &= \frac{dT}{dr} \end{aligned}$$

The two differential equations thus obtained are:

$$\begin{aligned} \frac{du_1}{dr} &= u_2 \\ \frac{du_2}{dr} &= -\frac{u_2}{r} - \frac{\dot{q}}{k} \end{aligned}$$

The surface temperature of the rod $T(R) = u_1(r = R)$ is measured to be 393 Kelvin. If the system is at steady state, and the rod is so long that axial heat flow can be neglected, the heat flow to the surface must equal the total heat generated inside the rod, i.e. at the surface,

$$\begin{aligned} -2\pi R L k(393\,\text{Kelvin})\frac{dT}{dr} &= \pi R^2 L \dot{q} \\ \text{i.e., } u_2(r = R) &= -R\frac{\dot{q}}{2k(393\,\text{Kelvin})} \end{aligned}$$

Standard ODE solvers expect to integrate starting from zero, not from R. To accomplish this, we can replace r with a new variable y defined this way:

$$\begin{aligned} y &= R - r \\ u_1(y=0) &= 393\,\text{Kelvin} \\ u_2(y=0) &= -R\frac{\dot{q}}{2k(393\,\text{Kelvin})} \\ \frac{du_1}{dy} &= -u_2 \\ \frac{du_2}{dy} &= \frac{u_2}{R-y} + \frac{\dot{q}}{k(u_1)} \end{aligned}$$

A function you can modify to make your solution `username_HW0_P1.m` is provided to you, WITH AN IMPORTANT CATCH. The input and outputs for the function ode23 have been removed. Your task is to fill in the blanks.

Assume $R = 0.20$ m and $\dot{q} = 20$ kW/m^3. The thermal conductivity of silicon is given by

$$k(T) = \frac{148\,\text{W}}{\text{m}\cdot\text{K}}\left(\frac{T}{300\,\text{K}}\right)^{-1.3}.$$

1. In `username_HW0_P1.m`, the `ode23` function calls an imbedded (pre-written) function `dudy` with the entry `udash(2)` blank. Fill in `udash(2)` with the expression provided above for $\frac{du_2}{dy}$. Type `help ode23` in the MATLAB command window to better understand the usage and syntax of the `ode23` function. This will be useful in the remaining parts of the problem.

2. Write an additional function in `username_HW0_P1.m` that plots T and dT/dr versus r using the `subplot` function. Include comments at the start of your plotting function by describing its inputs and outputs briefly. Attach the plot generated from the function.

3. How would you alter the function `username_HW0_P1.m` if the surface temperature was $T =$ 290 K, and $\dot{q}$ was 34 kW/m^3? Copy-paste the original and modified lines of code into your report. Attach plots of T and dT/dr versus r for the new parameters.

4. How would you have the `ode23` function report the temperature in the cylinder at radius = 0.156 meters? Copy-paste the original and modified lines of code into your report.

5. How would you change the absolute or relative error tolerance used by `ode23`? Copy-paste the original and modified lines of code into your report.

6. If the function `dudy` was not imbedded in the function `username_HW0_P1.m` (i.e., `dudy` was stored in a separate m-file, `dudy.m`), would you have to change anything in the code `username_HW0_P1.m`? If so, what?

Problem 2. Bessel Approximation with Finite Sums

Bessel functions $J_\nu(r)$ arise in many problems with cylindrical or spherical symmetry (see for example Deen's text, section 4.7).

A series expansion is known for these Bessel functions:

$$J_\nu(r) = \sum_{k=0}^{\infty} \frac{(-1)^k (r/2)^{\nu+2k}}{k!\Gamma(\nu+k+1)}$$

where Γ is the Gamma function. For integer $\nu < 0$, the first term of this series is not defined, and those Bessel functions are computed using a different formula. However, this function works for both positive and negative non-integer values of ν.

1. Write a MATLAB function `my_bessel` which takes in `nu`, `r`, and a tolerance `tol` as inputs and returns an estimate of $J_\nu(r)$ based on a finite truncated series expansion (i.e., run the sum for k up to $k = k_{\max}$ rather than $k = \infty$). The estimate should converge to the true value of the Bessel function as tol gets very small. If the input `nu` is a negative integer, your function should return an error message. Your code should be sufficiently flexible and vectorized to evaluate for one r value or a vector of r values, returning either a single value or the corresponding vector of evaluations. For testing your function: it is known that $J_0(0) = 1$, $J_{1/2}(r) = \mathrm{sqrt}(2/\pi \mathrm{r}) \sin \mathrm{r}$, $J_{-1/2}(r) = \mathrm{sqrt}(2/\pi \mathrm{r}) \cos \mathrm{r}$; and $J_2(11.6198) \approx 0$. Paste your error checking and Bessel calculation loop in your solution.

2. Write a function `plot_bessel` which reads in `nu`, `r`, and a vector `tolvec` and generates a nicely labeled plot with several traces overlapped:

   (a) First trace is a plot of $J_\nu(r)$ from $r = 0$ to $r =$ `r_max`, for `tol=tolvec(1)`

   (b) Second trace is a plot of $J_\nu(r)$ from $r = 0$ to $r =$ `r_max`, for `tol=tolvec(2)`

   (c) Etc.

   The last trace is the corresponding plot of $J_\nu(r)$ generated by the built-in MATLAB function `besselj.m`. Please limit the number of traces to 10. Use this function to generate a pretty plot which shows how your Bessel function routine converges as you tighten the tolerance using `nu` $= 0$ and `r_max` $= 30$.

Problem 3. The paper "A colloidal quantum dot spectrometer" by Bao and Bawendi (Nature 523:67, 2015) describes a 2-D array of thin films built from different colloidal quantum dots and used as a filter for a CCD camera such as the one in your cell phone. The device is a micro-spectrometer that utilizes the the tunable photonic band gap of quantum dots to filter the light transmitted to pixels of the CCD camera. If each element of the 2-D array of filter films is designated by a number, $i = 1, 2, ..., n_f$, where n_f is the number of films in the array, then the intensity of light transmitted to the CCD under filter i can be approximated as

$$J_i = \sum_{j=1}^{n_\lambda} = T_i(\lambda_j)\Phi(\lambda_j), \quad \text{for each } i = 1, 2, \ldots, n_f. \tag{1}$$

where $\Phi(\lambda)$ is the spectrum of light incident on the filter, $T_i(\lambda)$ is the transmission spectrum of filter i, and λ_j with $j = 1, 2, ..., n_\lambda$ is one of a discrete set of wavelengths of light over the range of interest, say 400-600 nm. n_λ is the number of wavelengths at which the incident light will be sampled. This device aims to use the intensities measured by the CCD under different filter elements to infer the spectrum of the incident light $\Phi(\lambda)$ in a process called spectral reconstruction.

You will simulate the transmission spectrum of each filter using a simple sigmoidal function that mimics the properties of a colloidal quantum dot film. For a device having n_f filters, filter i has the simulated transmission spectrum:

$$T_i(\lambda) = \frac{1}{1 + \exp^{-(\lambda - s_i)/w}}$$

with λ measured in nanometers, $w = 50$ nm and $s_i = 400 + 200(i-1)/(n_f - 1)$ nm. You may use any built-in MATLAB functions needed during this problem.

1. Plot the transmission spectra of the filters assuming $n_f = 30$ and $n_\lambda = 100$ with the sampled wavelengths evenly spaced between 400 and 600 nm.

2. For the special case of a square system with $n_\lambda = n_f = n$, there will be values of $\mathbf{\Phi} := [\Phi(\lambda_1), \Phi(\lambda_2), \cdots, \Phi(\lambda_n)]^{\mathrm{T}}$ that fit the data exactly. Write a system of equations in the form $\mathbf{A\Phi} = \mathbf{J}$ where $\mathbf{\Phi}$ contains the unknown incident spectrum at different discrete wavelengths: $\{\Phi_j\}_{j=1}^{n}$ and $\mathbf{J} := [J_1, J_2, \cdots, J_n]^{\mathrm{T}}$ contains the integrated intensity measured by the CCD under different filters. Explicitly write out the form of $\mathbf{A}$.

3. Let the λ samples be evenly spaced between 400 and 600 nm. Using $n_f = n_\lambda = n = 2, 3, \cdots, 20$, plot cond($\mathbf{A}$), $\|\mathbf{A}\|$, and $\|\mathbf{A}^{-1}\|$ on the same axes as a function of n. How does cond($\mathbf{A}$) compare to $\|\mathbf{A}\|\|\mathbf{A}^{-1}\|$ for each value of n. Explain any differences.

4. Assume that each of the measured intensities has a relative error no bigger than 0.1%. Derive an expression giving an upper bound for the relative error in $\mathbf{\Phi}$ in terms of the relative error in the measured intensities and any of cond($\mathbf{A}$), $\|\mathbf{A}\|$, and $\|\mathbf{A}^{-1}\|$. Let the λ samples be evenly spaced between 400 and 600 nm. Using $n_f = n_\lambda = n = 2, 3, \cdots, 20$, compute this error bound for several values of n. Plot these relative error bounds for $\mathbf{\Phi}$ as a function of n.

5. Assume that each of the transmission spectra for the filters was measured with a relative error no bigger than 0.1% at each sampled wavelength. Derive an expression giving an upper bound for the relative error in $\mathbf{\Phi}$ in terms of the relative error in the transmission spectra and any of cond($\mathbf{A}$), $\|\mathbf{A}\|$, and $\|\mathbf{A}^{-1}\|$. Be sure to note the conditions under which this upper bound is valid. Let the λ samples be evenly spaced between 400 and 600 nm. Plot the values

of the bound in $n_f = n_\lambda = n = 2, 3, \cdots, 30$. For which values of n is the relative error in $\mathbf{\Phi}$ bounded from above? Explain what it means for the relative error in $\mathbf{\Phi}$ to have no upper bound.

6. Consider an alternative case in which $n_f > n_\lambda$ for which there does not exist an exact solution for $\mathbf{\Phi}$. A linear least squares solution can be constructed from the system of equations: $\mathbf{A}^T\mathbf{A}\mathbf{\Phi} = \mathbf{A}^T\mathbf{J}$, however. We wish to estimate the sensitivity of the least squares solution to perturbations in the measured intensities, $\mathbf{J}$. Assume that the relative error in $\mathbf{J}$ is no bigger than 0.1%. Derive an upper bound on the relative error in the least squares solution. Is this bound independent of the measured values $\mathbf{J}$? You may use the fact that the matrix $\mathbf{A}$ has full column rank when $n_f > n_\lambda$ and the $T_i(\lambda)$ are linearly independent so that $\mathbf{A}^T\mathbf{A}$ has an inverse. Explain what factors will enhance the sensitivity of the least squares solution.

7. Explain the intuitive meaning of the ill-conditioning for these systems of equations. Examine extended data figure 1 of the cited Nature article and compare the simulated transmission spectra to those of the experimental colloidal quantum dot thin films. Suggest properties for the basis functions (transmission spectra) needed for accurate calculation of the incident light spectrum in the range of wavelengths: 400-600 nm.

MIT OpenCourseWare
https://ocw.mit.edu

10.34 Numerical Methods Applied to Chemical Engineering
Fall 2015

For information about citing these materials or our Terms of Use, visit: https://ocw.mit.edu/terms.