School of Mathematics and Statistics
MAST30028 Numerical Methods & Scientific Computing
Week 8

Drag and drop the folder `Week8` from `L: \MAST30028` to `C:\...\MATLAB` and include it in the path. Now MATLAB knows how to find the files in `Week8`. The folder includes a pdf file of Chapter 5 in Moler's textbook (`leastsquares.pdf`).

1 Condition numbers

This relates to material in Lecture 14.

Exercise Set 1

a. Compute by hand $\kappa_1(\mathbf{A}), \kappa_\infty(\mathbf{A})$ for the matrices :

a) $\begin{bmatrix} 1001 & 1000 \\ 1 & 1 \end{bmatrix}$ b) $\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$

In each case, find a nearby singular matrix and check that the bound

$$\frac{1}{\kappa_p(A)} \leq \frac{\|\mathbf{\Delta A}\|_p}{\|\mathbf{A}\|_p}$$

where $\mathbf{A} + \mathbf{\Delta A}$ is singular, is satisfied.

b. Run the demo M-file `CondEgs.m` to see the effect of ill-conditioning on solution values.

For another example, run `ErrChol.m`. We can compute the true answer because the inverse of a Hilbert matrix is known exactly. Hilbert matrices are symmetric, positive definite, hence the use of Cholesky factorization here.

c. Adapt `CondEgs` or `ErrChol.m` so it also shows the relative residual

$$\eta = \frac{||\mathbf{r}||}{||\mathbf{A}||||\mathbf{x}||}$$

You should see that the residuals stay pretty small but the solution is inaccurate because of the large condition number.

2 Data fitting

There are 2 common ways to solve Linear Least Squares problems — the normal equations

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$$

or preferably by using the QR factorization of $\mathbf{A}$. From $\mathbf{A} = \mathbf{QR}$, we solve the triangular system

$$\mathbf{R}\mathbf{x} = \mathbf{Q}^T\mathbf{b}$$

This is what the MATLAB \ command does in the case that $\mathbf{A}$ is rectangular with $m > n$ (overdetermined).

Exercise Set 2

Linear models

Which of the following models are linear? nonlinear? separable [have parameters appearing both linearly and nonlinearly]? which of the nonlinear models can be transformed to linear ones?

a) $y \approx c_1 + c_2 x^3$ b) $y \approx c_1 e^{c_2 x} + c_2 e^{c_4 x}$ c) $y \approx c_1 e^{c_2 x}$
d) $y \approx c_1 x^{c_2}$ e) $y \approx \sin(c_1 x) e^{c_2 x}$ f) $y \approx c_1 \sin(\omega x) + c_2 \sin(2\omega x) + c_3 \sin(3\omega x)$ where ω is known.

Curve fitting

a. Type `censusgui` and explore the various methods of fitting the U.S. census data from 1900-2010.

   - Which methods appear to be interpolating the data [making the curve go though every data point]? Which methods are fitting the data?
   - How sensitive is the extrapolated value at 2020 to the kind of model?

b. Now change the $t = 1950$ data point $y = 150.697$ to $y = 50.697$ to create an *outlier*. How sensitive are the various fits to this outlier?

The methods `pchip` and `spline` involve C^1 piecewise cubic functions i.e. the derivatives match where the pieces join, and C^2 piecewise cubic functions, respectively. We don't cover them in this subject.

Do it yourself

Generate the synthetic noisy data $\{t_i, y_i\}$ using the code

```
t = 1:0.1:3;
y = 1+2*sin(3*t)+0.5*randn(size(t));
```

Use Linear Least Squares to fit this data to the model

$$y \sim c_1 + c_2 \sin(3t)$$

How close are the coefficients to the underlying values?

Curve fitting App [Toolbox]

The Curve Fitting App (was Toolbox) fits both linear and nonlinear models, using a variety of methods, including least squares fitting.

To get an idea how to compare fits from different models, work through the example from
`Documentation Home->Curve Fitting Toolbox->Linear and Nonlinear Regression->Compare Fits in Curve`

Notice how high degree polynomial models must be centred and scaled to avoid conditioning problems. Also see how fits can be compared by examining residual plots, the SSE (residual to the least squares fit) and confidence intervals for coefficients.

Clicking on Fit Options shows that the polynomial models use Linear least Squares and the exponential model uses nonlinear least squares, which we meet next week.

You can save your fitting session for later re-use, and also generate the MATLAB commands equivalent to the fits you did by using the Curve Fitting App.

When the normal equations fail

This relates to material in Lecture 15.

Run the script `normalFails.m` to see an example where the Cholesky factorization of the normal equations fails, but a QR factorization successfully gives the same information.

Explain each line of output.