Numerical Mathematics I, 2016/2017, Lab session 3

Keywords: linear algebra, direct methods, LU, pivoting, conditioning, least-squares

Remarks

- Make a new folder called `NM1_LAB_3` for this lab session, save all your functions in this folder.
- Whenever a new `MATLAB` function is introduced, try figuring out yourself what this function does by typing `help <function>` in the command window.
- Make sure that you have done the preparation before starting the lab session. The answers should be worked out either by pen and paper (readable) or with any text processing software (LaTeX, Word, etc.).

1 Preparation

1.1 LU factorisation and partial pivoting

1. Study (Textbook, Section 1.4), (Textbook, Section 5.1 - 5.5) (except 5.4.1) and (Textbook, Section 5.7). Recall from Linear Algebra (use your Linear Algebra 1 book if needed): overdetermined, underdetermined, inverse, nonsingular, range, nullspace, eigenvalue, norm, orthogonality.

2. Consider the linear system of equations $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is given by

$$\mathbf{A} = \begin{pmatrix} \epsilon & 1 \\ 1 & 1 \end{pmatrix}.$$

For the following three situations

(i) $\epsilon \neq 1$ and $\mathbf{b} = (1, 2)^T$

(ii) $\epsilon = 1$ and $\mathbf{b} = (1, 2)^T$

(iii) $\epsilon = 1$ and $\mathbf{b} = (1, 1)^T$,

answer the following questions

(a) Is $\mathbf{A}$ nonsingular?

(b) What is the nullspace of $\mathbf{A}$? And the range of $\mathbf{A}$? Is $\mathbf{b} \in \text{range}(\mathbf{A})$?

(c) How many solutions are there?

(d) Give the general form of the solution(s) if there exist any.

3. A matrixnorm $\|.\|_*$ on $\mathbb{R}^{n\times n}$ is said to be sub-multiplicative if

$$\|\mathbf{AB}\|_* \leq \|\mathbf{A}\|_* \|\mathbf{B}\|_*,$$

for all matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$. Show that in this case the corresponding condition number

$$K_*(\mathbf{A}) = \|\mathbf{A}\|_* \|\mathbf{A}^{-1}\|_*$$

satisfies

$$K_*(\mathbf{AB}) \leq K_*(\mathbf{A})K_*(\mathbf{B}),$$

for all nonsingular matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$.

4. For a linear system of equations $\mathbf{Ma} = \mathbf{f}$ with square nonsingular matrix $\mathbf{M}$, give a general upper bound for the relative error in $\mathbf{a}$ if we perturb both $\mathbf{f}$ and $\mathbf{M}$ (Textbook, Section 5.5). How does this upper bound simplify if we only perturb $\mathbf{f}$?

5. Consider again solving $\mathbf{Ma} = \mathbf{f}$ with square nonsingular matrix $\mathbf{M}$. Due to round-off errors we obtain an approximate solution $\hat{\mathbf{a}}$. Let the residual $\mathbf{r}$ be given by

$$\mathbf{r} = \mathbf{f} - \mathbf{M}\hat{\mathbf{a}}.$$

The *scaled residual norm* is defined as the 2-norm of the residual $\mathbf{r}$ divided by the 2-norm of $\mathbf{f}$. Is the scaled residual norm always a good measure of the relative error in $\hat{\mathbf{a}}$? How does this depend on $K_2(\mathbf{M})$? Hint: use the result of the previous question.

1.2 Conditioning of the least squares problem

1. Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{y} \in \mathbb{R}^m$ be given with $m \geq n$. Show that the minimization of the 2-norm of

$$\mathbf{Ac} - \mathbf{y}$$

(with respect to $\mathbf{c}$) leads to the *normal equations*

$$\mathbf{A}^T\mathbf{Ac} = \mathbf{A}^T\mathbf{y}. \tag{1}$$

2. Subject to what condition on $\mathbf{A} \in \mathbb{R}^{m\times n}$ does there always (for any $\mathbf{y}$) exist a unique solution to the normal equations?

3. Under the condition derived in the point above, show that the unique solution $\mathbf{c}$ to the normal equations (1) can be found by solving the upper triangular system

$$\mathbf{Rc} = \mathbf{Q}^T\mathbf{y}, \tag{2}$$

where $\mathbf{A} = \mathbf{QR}$ is the reduced ('economy size') QR factorisation of $\mathbf{A}$ with $\mathbf{Q} \in \mathbb{R}^{m\times n}$ and $\mathbf{R} \in \mathbb{R}^{n\times n}$.

2 Lab experiments

Introduction

In this lab session you will consider the following problem: for a given set of n data points

$$(x_i, y_i), \quad i = 1, \ldots, n,$$

find the coefficients $\mathbf{c}$ of the polynomial f of degree $\leq r$,

$$f(x) = \sum_{j=1}^{r+1} c_j x^{r-j+1},$$

such that the error

$$E := \sum_{i=1}^{n} (f(x_i) - y_i)^2$$

is minimized. Here we assume that $r \leq n-1$. If we define the *Vandermonde* matrix $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{n\times(r+1)}$ by

$$a_{ij} = x_i^{r-j+1},$$

then we can write the error E as

$$E = \|\mathbf{Ac} - \mathbf{y}\|_2^2,$$

showing that the problem consists of finding the least squares solution of the overdetermined linear system

$$\mathbf{Ac} = \mathbf{y}. \tag{3}$$

In the preparation you have shown that, under certain conditions, such a problem can be uniquely solved by solving the corresponding normal equations

$$\mathbf{A}^T\mathbf{Ac} = \mathbf{A}^T\mathbf{y}$$

for the coefficients $\mathbf{c}$. In this lab session you will investigate several solution techniques for this problem.

Write a function `makeVandermondeMatrix` which, given the abscissae vector $\mathbf{x}$ and degree r, generates the corresponding Vandermonde matrix $\mathbf{A}$. The header of this function should be

```
% INPUT
% x          nodal points x_i
% r          polynomial degree
function A = makeVandermondeMatrix(x, r)
```

2.1 LU factorisation and partial pivoting

Introduction

In this section we consider the case $r = n - 1 = 1$. We are looking for the best linear fit through the two data points corresponding to the vectors

$$\mathbf{x} = \begin{pmatrix} \epsilon \\ 1 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

Since $r = n - 1$, the corresponding linear system (3) is actually not overdetermined but square, so instead of solving the normal equations we will directly solve the square system

$$\mathbf{A}_\epsilon \mathbf{c}_\epsilon = \mathbf{y}, \tag{4}$$

where $\mathbf{A}_\epsilon \in \mathbb{R}^{2\times 2}$ is the Vandermonde matrix corresponding to the given data.

Experiment without pivoting

We want to solve this system of equations using an LU factorisation: write a function called `luNaive` which solves an *arbitrary* square linear system by using the LU factorisation technique. To test that it works (for arbitrary square linear systems), you can try it out as follows: generate a random 10 × 10 matrix $\mathbf{A}$ and solution vector $\mathbf{x}$, compute $\mathbf{b} = \mathbf{Ax}$, and then solve for $\mathbf{x}$ using your `luNaive` function. You should write the code to compute the triangular factors $\mathbf{L}$ and $\mathbf{U}$ yourself, but please note that example pseudocode is given in (Textbook, Equation 5.13). Solving the upper and lower triangular subproblems may be done by using the backslash command \ (which recognizes the triangular structure of $\mathbf{L}$ and $\mathbf{U}$). The header of this function should be

```
% INPUT
% A          square matrix
% b          right hand side
% OUTPUT
% x          solution such that A*x=b
% L          lower triangular matrix such that A = L*U
% U          upper triangular matrix such that A = L*U
function [x, L, U] = luNaive(A, b)
```

Hence the triangular factors $\mathbf{L}$ and $\mathbf{U}$ should be returned as well. Now consider solving (4) for the following values of ϵ,

$$\epsilon = 10^{-i}, \quad i = 1, \dots 16.$$

The resulting solutions will be denoted by $\hat{\mathbf{c}}_\epsilon$. For each ϵ compute the following quantities:

- The relative error
$$\frac{\|\hat{\mathbf{c}}_\epsilon - \mathbf{c}_\epsilon\|_2}{\|\mathbf{c}_\epsilon\|_2}$$
in the obtained solution $\hat{\mathbf{c}}_\epsilon$ (compute the exact solution $\mathbf{c}_\epsilon$ by hand).
- The K_2 condition numbers of $\mathbf{A}_\epsilon$, $\mathbf{L}_\epsilon$ and $\mathbf{U}_\epsilon$
- The factorisation error
$$\|\mathbf{A}_\epsilon - \mathbf{L}_\epsilon \mathbf{U}_\epsilon\|_F,$$
where $\|\mathbf{M}\|_F$ denotes the Frobenius norm which is defined as
$$\|\mathbf{M}\|_F := \sqrt{\sum_{i,j=1}^{m,n} m_{ij}^2},$$
which in MATLAB can be computed by `norm(M, 'fro')`.

Summarise your results in a single double-logarithmic plot (use the `loglog` function) where the five quantities are plotted as a function of ϵ.

Experiment using partial pivoting

Repeat the previous experiment, but now use the LU factorisation with partial pivoting such that

$$\mathbf{PA} = \mathbf{LU},$$

where $\mathbf{P}$ stands for a permutation matrix. For this you should write a function called `luPivot` which solves an *arbitrary* square linear system using the LU factorisation technique with partial pivoting. This function should also return the permutation matrix $\mathbf{P}$. Just like you did for the previous function `luNaive`, first test the new function on a random 10 × 10 linear system. Next solve system (4) for the same values of ϵ as before and summarise your results in a similar double-logarithmic plot.

2.2 Conditioning of the least squares problem

Introduction

Consider the set of 21 data points given by

$$x_i = (i-1)/20, \quad y_i = x_i^8, \quad i = 1, \dots, 21,$$

which correspond to the eighth degree monomial $f(x) = x^8$. If we set the maximum polynomial degree equal to $r = 8$, the least squares solution $\mathbf{c}$ of the overdetermined system (3) is easily seen to be given by $\mathbf{c} = \mathbf{e}_1$.

Contrary to the previous problem, system (3) is overdetermined, and therefore we must solve the corresponding normal equations. As you have shown in the preparation, this can be done in at least two ways: either we solve (1) where we consider $\mathbf{A}^T\mathbf{A}$ as the system matrix, or we use a QR factorisation of $\mathbf{A}$ and solve (2) with $\mathbf{R}$ as the system matrix. Note that in `MATLAB` one should compute the QR factorisation as `[Q, R] = qr(A, 0)`.

Experiment

To illustrate the difference between the two solution methods, we perturb the data points (both x and y) by adding random perturbations from the interval $(-\epsilon, \epsilon)$, thereby simulating measurement errors of $\mathcal{O}(\epsilon)$, and quantify the effect these perturbations have on the resulting solution. We consider the following values of ϵ,

$$\epsilon = 10^{-i}, \quad i = 1, \dots 16,$$

and denote the resulting solution by $\hat{\mathbf{c}}_\epsilon$. For each value of ϵ, and for each of the solution methods (hence in total you should run 32 experiments), compute the following quantities:

- The relative error of the solution. We now consider the exact solution $\mathbf{c} = \mathbf{e}_1$ which is, contrary to the previous experiment, independent of ϵ since we now consider ϵ as a measurement error.
- The upper bound (which you found in the preparation) for the relative error of the solution of the resulting perturbed linear system. Note that the linear system solved is different for each of the solution methods, and so is the right-hand side.

Summarise your results in a single double-logarithmic plot, where each of the quantities is plotted as a function of ϵ.

3 Discussion

3.1 LU factorisation and partial pivoting

1. What can go wrong when computing the "normal" LU factorisation of a matrix and how does partial pivoting circumvent this problem?
2. What is the computational cost of making an LU factorisation (for a full matrix)? What if the matrix is banded with bandwidth b?
3. When making an LU factorisation you basically split the problem of finding $\mathbf{x}$ such that $\mathbf{Ax} = \mathbf{b}$ into two subproblems. Which subproblems? In your numerical experiments,

how do the condition numbers (K_2) of these two subproblems compare to that of the original problem? Make here a distinction between the experiments with and without partial pivoting. Would it ever be possible that the two subproblems are well-conditioned, and the original problem is not? Hint: the matrix 2-norm $\|.\|_2$ is sub-multiplicative. Is it possible that the original problem is well-conditioned, but one of the two subproblems is not? Do your experiments confirm this?

4. What happens to the relative error as $\epsilon \to 0$ when not using pivoting? Can you predict this behaviour?

3.2 Conditioning of the least squares problem

1. What is the meaning of the "`0`" in `[Q, R] = qr(A, 0)`?

2. Are the 32 experiments that you did in Section 2.2 in agreement with the theoretical upper bounds for the relative error that you provided in Question 4 of Preparation section 1.1?

3. How does the conditioning of the triangular system (2) compare to the conditioning of the normal equations (1)?