Systems of Linear Equations

Exercise 7

Systems of Linear Equations Report

Professor
Annie Cuyt

Author
Cesare De Cal

Assistant Professor
Ferre Knaepkens

October 28, 2019

1 Introduction

In this exercise I am going to use solve a linear system $Ax = y$ using LU decomposition. The first element of the unknowns vector x will contain an approximation of e - 2.

2 Tools

The following programming language and libraries have been used in this exercise:

- C
- GSL (GNU Scientific Library)

The following GSL data types have been used in the exercise:

- `gsl_vector`
- `gsl_matrix`
- `gsl_permutation`

The following GSL methods have been used in the exercise:

- `gsl_matrix_alloc(size1, size2)`
- `gsl_matrix_set_zero(matrix)`
- `gsl_matrix_set(matrix, row, column, value)`
- `gsl_matrix_get(matrix, row, column)`
- `gsl_vector_alloc(size)`
- `gsl_vector_set_zero(vector)`
- `gsl_vector_set(size)`
- `gsl_vector_get(vector, index)`
- `gsl_permutation_alloc(size)`

In order to factorize a matrix into the LU decomposition, and then solve the square system $Ax = y$ using the decomposition of A, I've used the following methods:

- `gsl_linalg_LU_decomp(A, permutation, signum)`
- `gsl_linalg_LU_solve(LU, permutation, b, x)`

3 Solving the system

In order to solve the system $Ax = y$, I first need to build the matrix A by understanding how it's build. The requirements are to build a tridiagonal matrix with the values -1 on the adjacent upper diagonal, the entries $+1$ on the adjacent lower diagonal, and on the main diagonal the values b_i, with $i = 1, \dots, n$ given by

$$b_i = \frac{2(i+1)}{3}, \quad i+1 = 3, 6, 9, \dots$$
$$b_i = 1, \quad i+1 = 2, 4, 5, 7, 8, \dots$$

By looking closely at the first rule, we see that the $i+1$ are all multiples of 3 ($i+1 = 3*k$, for some k). Hence the i are of the form $i = 3*k-1$, for some k. For $n = 10$, for example, this is what the matrix approximately looks like:

$$\begin{bmatrix}
1.0000 & -1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
1.0000 & 2.0000 & -1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 1.0000 & 1.0000 & -1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 1.0000 & 1.0000 & -1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 1.0000 & 4.0000 & -1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 1.0000 & -1.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 1.0000 & -1.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 6.0000 & -1.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 1.0000 & -1.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 1.0000
\end{bmatrix}$$

The coefficients matrix A is first alloced by using the `gsl_matrix_alloc` method, then I set all the elements to zero with `gsl_matrix_set_zero` and finally nested `for` loops fill the diagonal values. The coefficients reported above on the diagonal have 5 significant digits for improve the readability of this report.

After that, I created the matrix b
I then found the LU decomposition matrix:

4 Observations