Solving Systems of Equations, Errors and Explorations

David Tran and Spencer Kelly

March 9, 2024

Abstract

1 Introduction

2 The PA = LU factorization method for linear systems

2.1 Why is PA = LU needed for solving linear systems approximately?

2.2 How to identify systems Ax = b for which PA = LU is not suited

2.3 Larger applications of PA = LU factorization

3 Iterative solution of systems of linear equations

Jacobi FPI is a method for solving systems of linear equations of the form $Ax = b$. It is a multi-dimensional analogue of the one-dimensional fixed-point iteration we explored in Lab 2. We require that A is diagonally dominant, which means that the absolute value of the diagonal element of A is greater than the sum of the absolute values of the other elements in the row. This is a sufficient condition for the convergence of the Jacobi FPI. The Jacobi FPI is given by the following formula:

$$x^{(k+1)} = D^{-1}(L + U)x^{(k)} + D^{-1}b \tag{1}$$

where D is the diagonal of A, L is the lower triangular part of A, and U is the upper triangular part of A. The Jacobi FPI is a simple and easy-to-implement method for solving systems of linear equations, but it is not the most efficient method. We will explore the convergence of the Jacobi FPI and compare it to the PA = LU factorization method.

3.1 Solving an equation for n = 100,000

We solve the following system of linear equations

$$\begin{bmatrix} 3 & -1 & 0 & 0 & 0 & \frac{1}{2} \\ -1 & 3 & -1 & 0 & \frac{1}{2} & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 3 & -1 \\ \frac{1}{2} & 0 & 0 & 0 & -1 & 3 \end{bmatrix} x = b \tag{2}$$

where $b = (5/2, 3/2, \ldots, 3/2, 1, 1, 3/2, \ldots, 5/2)^T$, where there are $n - 4$ 3/2's in the middle of the vector. Using the code in Listing 1, we solve the system of linear equations for $n = 100,000$, with the following:

```
>> n = 100000;
>> [A, b] = sparsesetup(n);
>> x = jacobi(A, b, 50);
```

which converges to $(1, \ldots, 1)^T$ after just 50 iterations.

```
% Inputs: full or sparse matrix a, r.h.s. b,
% number of Jacobi iterations, k
% Output: solution x
function x = jacobi(a,b,k)
    n=length(b); % find n
    d=diag(a); % extract diagonal of a
    r=a-diag(d); % r is the remainder
    x=zeros(n,1); % initialize vector x
    for j=1:k % loop for Jacobi iteration
        x = (b-r*x)./d;
    end % End of Jacobi iteration loop
```

Listing 1: The code for the Jacobi FPI

3.2 Comparison of PA = LU and Jacobi Iteration

Theoretically, Jacobi is guaranteed to converge for all n if A is diagonally dominant, which A is in this case. If we try $PA = LU$ decomposition for this problem, using

```
% Solve a system of equations with PA = LU
function x = solve_lu(A, b)
    [L, U, P] = lu(A);
    x = U \ (L \ (P * b));
end
```

Listing 2: The code for the LU decomposition

the function takes much longer due to its $O(n^3)$ time complexity. Even for $n = 10000$, the function takes almost a minute on an Apple M1 Pro chip. Usually for $PA = LU$, if the matrix is rank-deficient (multiple rows/columns that are linearly dependent), the LU decomposition may not be unique and thus may fail to find the correct solution. However in this case, A is not rank-deficient, so the LU decomposition should work. But, the Jacobi method is much faster and more reliable for this problem.

3.3 Why is solving such large systems important in applications?

Solving large systems of linear equations is important in applications because it is a fundamental problem in many areas of science and engineering. For example, in statstics, a linear model may depend on millions or even billions of parameters which translates to a system of equations of the same size. As a concrete example, a technique such as principal component analysis requires solving a system with as many equations as there are desired dimensions. Or, for example, modelling fluid dynamics similarily requires very large systems of equations due to the complexity of the problem. Data is naturally highly multi-dimensional, which makes solving large systems important in practice.