Large-Scale Linear Algebra and Applications

Daniel Masotti
March 4, 2019

1 Introduction

Many applications of science and engineering reduce to solving large linear systems of equations. Accordingly, the solution to the equation $A\mathbf{x} = \mathbf{b}$, where $A \in \mathbb{C}^{n \times m}$ and $\mathbf{b} \in \mathbb{C}^m$, is important to our understanding of physical systems. One of the simplest algorithms for solving these problems is Gaussian elimination, which is usually taught in an introductory linear algebra course. By the 20th century however, very large systems motivated iterative methods, which could potentially compute solutions more efficiently than could traditional direct methods [7]. Furthermore, advancements in computing revealed the effectiveness of these relatively new algorithms.

Iterative methods work by computing increasingly accurate approximations for the solution at each step of the algorithm. The earliest iterative solvers included Gauss-Seidel (1823), Jacobi (1846) [2], and the Successive Over-relaxation, or SOR, (1950) methods. These solvers are called relaxation methods and, in general, are computed by splitting the matrix into its lower triangular, diagonal, and upper triangular components and modifying and replacing one entry at a time. These methods were the state of the art until the 1980s, at which point, iterative methods using Krylov subspaces began to dominate [5]. Theoretically, these methods approximate the solution by iteratively building the Krylov subspace, $K_m = \text{span}\{\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, \ldots, A^{m-1}\mathbf{b}\}$, in $m$ iterations and extracting an approximate solution from $K_m$ at each step. Examples of Krylov subspace methods include the conjugate gradient (CG) method and generalized minimum residual (GMRES) method. The CG method was introduced in 1952 as a direct solver [3]. Today, it is used as an iterative solver for symmetric positive-definite systems of linear equations. GMRES was introduced in 1986 for solving asymmetric systems [8]. Combined with preconditioning, a technique which helps the approximations converge, GMRES is a powerful algorithm with some elegant mathematical properties [6].

This semester, we will implement some of these Krylov subspace methods, including the Arnoldi Iteration, the Full Orthogonalization method (FOM), CG, and GMRES as well as investigate their theoretical properties. We will also study preconditioning techniques and some applications to physical systems.

2 Background

Krylov subspace methods are projection methods onto the Krylov space, $K_m$. In general, projection methods are iterative methods that extract an approximation $x_m \in x_0 + K_m$,
where \( x_0 \) is an initial guess, such that the residual, \( r_m = b - A x_m \), is orthogonal to the the space \( L_m \). In theory, these methods work as shown below [6].

<table>
<thead>
<tr>
<th>Algorithm 1: A typical projection method for solving a linear system.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> A matrix ( A \in \mathbb{R}^{n \times n} ) and vector ( b \in \mathbb{R}^n ) corresponding to the linear system ( A x = b ).</td>
</tr>
<tr>
<td><strong>Output:</strong> An approximate solution ( x ) to the system ( A x = b ).</td>
</tr>
<tr>
<td>1 while the solutions haven’t converged do</td>
</tr>
<tr>
<td>2 build ( V = \begin{bmatrix} v_1 &amp; \ldots &amp; v_m \end{bmatrix} ), a basis for ( K_m )</td>
</tr>
<tr>
<td>3 build ( W = \begin{bmatrix} w_1 &amp; \ldots &amp; w_m \end{bmatrix} ), a basis for ( L_m )</td>
</tr>
<tr>
<td>4 ( r = b - A x )</td>
</tr>
<tr>
<td>5 ( y = (W^T A V)^{-1} W^T r )</td>
</tr>
<tr>
<td>6 ( x = x + V y )</td>
</tr>
<tr>
<td>7 end</td>
</tr>
</tbody>
</table>

In the Full Orthogonalization Method, \( L_m = K_m \), and in GMRES, \( L_m = A K_m \). Accordingly, computing a basis for \( K_m \) is central to Krylov subspace methods. In practice, however, computing the basis for Krylov subspace poses numerical difficulties as the basis vectors may become nearly linearly dependent. As such, many of these methods are similar mathematically, but differ in the way that they build the basis for \( K_m \). For example, a first pass of the Arnoldi Iteration, which was first used to solve eigenvalue problems, builds an orthonormal basis for \( K_m \) using the Gram-Schmidt procedure [1]. Some numerical improvements to the Arnoldi iteration include using the Modified Gram-Schmidt or Householder implementations for building the basis. The extension of the Arnoldi Iteration to solve linear systems is the FOM method. GMRES is an improvement from FOM because it minimizes the residual, which is now orthogonal to the space \( A K_m \), in the least squares sense at each step.

In many applications, we adjust the execution of these algorithms for practical purposes. **Restarting**, for example, is the process in which we compute \( x_m \) from subspace \( K_m \) using the chosen method (1), discard the previous basis, and restart the algorithm but use \( x_m \) as our initial guess. We perform this restart procedure when storing the orthonormal basis matrix in memory is impractical. For instance, if the original system has millions of entries, we are limited in the number of basis vectors we could compute for our approximation by virtue of the problem having space-complexity \( \mathcal{O}(mn) \), where \( m \) is the size of \( K_m \) and \( n \) is the size of our matrix. While these restarted algorithms do not perform as well, they are much more space efficient than their counterparts.

Another important technique we will cover is preconditioning. We can solve \( A x = b \) by solving \( M^{-1} A x = M^{-1} b \) instead. We do this when the condition number, the ratio of the dominant eigenvalue to the smallest eigenvalue, of the matrix \( A \) is very large, which creates numerical instability and makes convergence towards a solution difficult. We hope that by applying the transformation \( A \rightarrow M^{-1} A, b \rightarrow M^{-1} b \), we can see an improvement in the convergence of the Krylov subspace methods. In this way, we can make our iterative solvers more effective.


3 Proposed Methodology

We will implement FOM, GMRES, CG, and MINRES using MATLAB and use test matrices from the Matrix Market website. We use the pseudocode from Saad’s “Iterative Methods for Sparse Linear Systems” for reference [6]. First, we will implement the Arnoldi Iteration. Next, we will implement the FOM method, which is an extension of the Arnoldi Iteration for solving linear systems. We will investigate the behavior of the residual at each step. Then, we will do the same for GMRES and the restarted versions of both these algorithms. This corresponds roughly to the material covered in chapter 6 of [6].

After spending some time implementing these iterative methods on some test matrices, we will investigate preconditioners and how they can improve our results. Some preconditioners we will study are the Jacobi, SOR, and incomplete LU (ILU) preconditioners. For reference, we will mostly follow chapters 9 and 10 of [6].

Finally, we will explore some applications from soft-body dynamics and pattern formation. In particular, we will investigate the Gray-Scott equations (1), which is a reaction-diffusion model for chemical systems [4]. The Gray-Scott equations are described by

\[
\begin{align*}
\frac{\partial u}{\partial t} &= r_u \nabla^2 u + uv^2 + f(1 - u) \\
\frac{\partial v}{\partial t} &= r_v \nabla^2 v + uv^2 + (f + k)v
\end{align*}
\]

(1)

(2)

corresponding to the pair of chemical reactions

\[ U + 2V \rightarrow 3V \]
\[ V \rightarrow P. \]

Here, \( u, v \) are the concentrations of the chemicals \( U, V \) respectively, \( r_u, r_v \) are the diffusion rates, \( k \) is the rate at which \( V \) is converted to \( P \), and \( f \) is the rate at which the process feeds \( U \) and drains \( U, V, P \). By discretizing this system of nonlinear partial differential equations, we obtain a system of linear equations on which we can apply our solvers.

References


