G2S3: Quantum Linear Algebra **August 11, 2023**

Lecture 1: Classical Numerical Linear Algebra

Resources: Nick Higham's Blog (https://nhigham.com/blog/); *Matrix Computations*, Gene Golub, Charles Van Loan; *Numerical Linear Algebra*, Nick Trefethen; *Matrix Analysis*, Charles Johnson; *Matrix Algorithms: Volumes I & II*, G.W. Stewart

1.1 Review: Linear Algebra Basics

1.1.1 Vectors, Matrices, Norms

Let $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ be vectors in a vector space $\mathbb{V}$, and let c be a scalar.

1. $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
2. $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$
3. $c(\mathbf{u} + \mathbf{v}) = c\mathbf{u} + c\mathbf{v}$
4. $(c + d)\mathbf{u} = c\mathbf{u} + d\mathbf{u}$
5. $c(d\mathbf{u}) = (cd)\mathbf{u}$
6. $\mathbf{u} + \mathbf{0} = \mathbf{u}$
7. $1\mathbf{u} = \mathbf{u}$
8. $\mathbf{u} + (-\mathbf{u}) = \mathbf{0}$
9. $0\mathbf{u} = \mathbf{0}$
10. $(-1)\mathbf{u} = -\mathbf{u}$

Matrices

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Vector Norms Let $\|\mathbf{v}\|$ denote the norm of vector $\mathbf{v}$.

1. $\|\mathbf{v}\| \geq 0$
2. $\|\mathbf{v}\| = 0 \iff \mathbf{v} = \mathbf{0}$
3. $\|c\mathbf{v}\| = |c|\|\mathbf{v}\|$
4. $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$

Cauchy-Schwarz: For any vectors $\mathbf{u}$ and $\mathbf{v}$ in an inner product space, $|\langle \mathbf{u}, \mathbf{v} \rangle| \leq \|\mathbf{u}\| \cdot \|\mathbf{v}\|$ where $\langle \mathbf{u}, \mathbf{v} \rangle$ represents the inner product of vectors $\mathbf{u}$ and $\mathbf{v}$.

Matrix Norms An *induced matrix norm* is a norm defined for matrices based on a vector norm in a consistent way. Let A be a matrix and $\|\cdot\|$ be a vector norm. The induced matrix norm $\|A\|$ of matrix A is:

$$\|A\|_p = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|A\mathbf{x}\|_p}{\|\mathbf{x}\|_p}$$

Properties of Matrix Norms Let $\|\cdot\|_p$ be a matrix norm. Here are some key properties of matrix norms:

1. $\|A\|_p \geq 0$
2. $\|A\|_p = 0 \iff A \equiv 0$
3. $\|cA\|_p = |c|\|A\|_p$
4. $\|A + B\|_p \leq \|A\|_p + \|B\|_p$
5. $\|AB\|_p \leq \|A\|_p \cdot \|B\|_p$

1.1.1.1 Accuracy and Stability

Accuracy: $\frac{\|\tilde{\mathcal{F}}(x) - \mathcal{F}(x)\|}{\|\mathcal{F}(x)\|}$

Stability: $\frac{\|\tilde{\mathcal{F}}(x) - \mathcal{F}(\tilde{x})\|}{\|\mathcal{F}(\tilde{x})\|}$

1.1.1.2 Conditioning

The **condition number** of a matrix quantifies how sensitive the solution of a linear system is to small changes in the input data. Let A be a matrix and $\kappa(A)$ be its condition number.

$$\kappa(A) = \|A\|\|A^{-1}\|$$

1.2 Overview: Advanced Linear Algebra

1.2.1 Linear Systems of Equations

Direct Solvers

Nonsingular Matrix A matrix A is said to be *nonsingular* if A^{-1} exists. More carefully, A is nonsingular if all columns and rows of A are linearly independent. (Else, it is *singular*.)

Unitary Matrix A matrix U is said to be *unitary* if $UU^T = U^TU = I$ (or $UU^H = U^HU$). Unitary matrices preserve length, orthogonality, and eigenvalue modulus.

Normal Matrix A matrix A is said to be *normal* if $AA^T = A^TA$ (or $AA^H = A^HA$). Unitary matrices are a special case of normal matrices.

Symmetric Positive Definite Matrix A matrix A is said to be *symmetric positive definite* (SPD) if it is symmetric ($A = A^T$) and for any $\mathbf{x} \neq 0$, $\mathbf{x}^TA\mathbf{x} > 0$.

Cholesky Factorization The Cholesky factorization decomposes a symmetric positive definite matrix A into the product of a lower triangular matrix L and its transpose:

$$A = LL^T$$

where L is a lower triangular matrix with positive diagonal entries.

LDL^T Factorization The LDL^T factorization decomposes an SPD matrix A into the product of a lower triangular matrix L with unit diagonal entries, and a diagonal matrix D, and the transpose of L:

$$A = LDL^T$$

where L is lower triangular, D is diagonal with positive entries.

Sparse Matrices A matrix is said to be *sparse* if it contains a significant number of zero entries compared to its total number of entries.

Least Squares Approximation

Pseudoinverse: $A^+ = (A^T A)^{-1} A^T$

Given a linear system $Ax = b$ where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, the **least squares solution** $x_{LS} \in \mathbb{R}^n$ minimizes the residual vector $\|Ax - b\|$:

$$x_{LS} = \arg\min_x \|Ax - b\|_2^2$$

The solution can be expressed using the pseudoinverse and is given by:

$$x_{LS} = A^+ b$$

The **QR Factorization** decomposes a matrix A into the product of an orthogonal matrix Q and an upper triangular matrix R:

$$A = QR.$$

Singular Value Decomposition

The Singular Value Decomposition (SVD) is a fundamental matrix factorization that represents any $m \times n$ matrix as $A = U\Sigma V^T$.

- U is an $m \times m$ orthogonal matrix with columns as the left singular vectors of A:

$$U = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_m \\ | & | & & | \end{bmatrix}$$

- Σ is an $m \times n$ diagonal matrix with non-negative real numbers $\sigma_1, \sigma_2, \ldots, \sigma_{\min(m,n)}$ on its diagonal, called the singular values of A:

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{\min(m,n)} \end{bmatrix}$$

- V is an $n \times n$ orthogonal matrix with columns as the right singular vectors of A:

$$V = \begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \\ | & | & & | \end{bmatrix}$$

1. **Orthogonality**: The matrices U and V in the SVD are orthogonal matrices, meaning their columns form orthonormal bases.

2. **Rank Approximation**: The SVD allows us to approximate a matrix by keeping only a subset of its singular values and corresponding singular vectors.

3. **Solving Linear Systems**: The SVD can be used to solve linear systems by expressing the solution in terms of the pseudoinverse.

4. **Principal Component Analysis (PCA)**: SVD is a fundamental step in PCA, a technique used for dimensionality reduction and feature extraction in data analysis.
5. **Image Compression**: The SVD can be employed for image compression, where a large image matrix is approximated using a small number of singular values and vectors.
6. **Machine Learning**: SVD is utilized in various machine learning algorithms, such as matrix factorization and collaborative filtering.
7. **Signal Processing**: SVD plays a role in signal processing tasks like noise reduction, signal enhancement, and channel equalization.
8. **Numerical Stability**: SVD is numerically stable, making it suitable for solving ill-conditioned or singular linear systems.

1.3 Iterative Methods for Linear Systems

1.3.1 Fixed Point Iterations

Matrix splitting: $A = M - N$

The **Jacobi Method** is a matrix splitting technique where the matrix A is decomposed into a diagonal matrix D, and the remaining entries are placed in matrix R: $A = D - R$

The Jacobi iteration equation is then given by:

$$Dx^{(k+1)} = Rx^{(k)} + b$$

Properties: Jacobi Method

- **Sufficient Condition for Convergence**:
- **Rate of Convergence**:

The **Gauss-Seidel Method** method is an improvement over the Jacobi method, where the entries of the lower triangular part of matrix A are included in matrix L, while the strictly upper triangular part is included in matrix U: $A = L + D + U$

The Gauss-Seidel iteration equation is:

$$(D + L)x^{(k+1)} = Ux^{(k)} + b$$

Properties: Gauss-Seidel Method

- **Sufficient Condition for Convergence**:

- **Rate of Convergence**:

1.3.2 Conjugate Gradient

Algorithm 1 Conjugate Gradient Method

Require: Symmetric positive definite matrix A, vector b, initial guess $\mathbf{x}_0$, and tolerance tol.

Initialize:
  $\mathbf{r}_0 \leftarrow \mathbf{b} - A\mathbf{x}_0$ { Compute initial residual }
  $\mathbf{p}_0 \leftarrow \mathbf{r}_0$ { Set initial search direction }
  $k \leftarrow 0$ { Initialize iteration counter }
while not converged **do**
  Compute $\mathbf{Ap}_k = A\mathbf{p}_k$
  $\alpha_k \leftarrow \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{p}_k^T \mathbf{A}\mathbf{p}_k}$ { Compute step size }
  $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{p}_k$ { Update solution }
  $\mathbf{r}_{k+1} \leftarrow \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k$ { Update residual }
  $\beta_{k+1} \leftarrow \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k}$ { Compute conjugate direction update }
  $\mathbf{p}_{k+1} \leftarrow \mathbf{r}_{k+1} + \beta_{k+1} \mathbf{p}_k$ { Update search direction }
  $k \leftarrow k + 1$ { Increment iteration counter }
  if $\|\mathbf{r}_{k+1}\| <$ tol **then**
    break

Properties: Conjugate Gradient Method

- **Sufficient Condition for Convergence**:

- **Rate of Convergence**:

- **Optimal Solution in A-Norm**:

1.3.3 Krylov Methods

The Krylov subspace $\mathcal{K}_m(A, v)$ is a vector space spanned by powers of the matrix A applied to a vector v. It is defined as:

$$\mathcal{K}_m(A, v) = \text{span}\{v, Av, A^2v, \ldots, A^{m-1}v\}$$

Arnoldi Iteration

Algorithm 2 Arnoldi Algorithm

Require: Matrix $A \in \mathbb{R}^{n\times n}$, vector $v \in \mathbb{R}^n$, and the desired subspace size m

Initialize: $\mathbf{V} = [\mathbf{v}]$, $\mathbf{H} = \mathbf{0} \in \mathbb{R}^{(m+1)\times m}$
for $j = 1$ to m **do**
  $\mathbf{w} = A\mathbf{v}_j$
  for $i = 1$ to j **do**
    $h_{ij} = \mathbf{w}^T\mathbf{v}_i$
    $\mathbf{w} = \mathbf{w} - h_{ij}\mathbf{v}_i$
  $h_{j+1,j} = \|\mathbf{w}\|$
  if $h_{j+1,j} = 0$ **then**
    break
  $\mathbf{v}_{j+1} = \mathbf{w}/h_{j+1,j}$
  Update $\mathbf{H}$ and $\mathbf{V}$ with new column $\mathbf{v}_{j+1}$

GMRES (Generalized Minimal Residual)

Algorithm 3 GMRES Algorithm

Require: Matrix $A \in \mathbb{R}^{n\times n}$, vector $b \in \mathbb{R}^n$, initial guess $x_0 \in \mathbb{R}^n$, and the desired subspace size m

Initialize: Compute initial residual $r_0 = b - Ax_0$
Apply Arnoldi process to generate $\mathbf{V}_m$ and $\mathbf{H}_m$ for $\mathcal{K}_m(A, r_0)$
Solve the least squares problem $\min_y \|\mathbf{H}_m y - \|r_0\|e_1\|_2$ for y
Update solution: $x_{\text{new}} = x_0 + \mathbf{V}_m y$

1.3.4 Preconditioning

Left Preconditioning

Right Preconditioning

Symmetric Preconditioning

1.4 Eigenvalue Decomposition

Properties:

- Every matrix with n linearly independent eigenvectors has an eigenvalue decomposition.
- If A has n linearly independent eigenvectors, it can be diagonalized as $A = V\Lambda V^{-1}$, where columns of V are eigenvectors of A and Λ contains eigenvalues.
- Eigenvalues of a real matrix can be complex, even if the matrix is real.
- Symmetric matrices have real eigenvalues and orthogonal eigenvectors.
- Positive definite matrices have positive real eigenvalues and orthogonal eigenvectors.
- Algebraic multiplicity counts how many times an eigenvalue appears in the characteristic polynomial.
- Geometric multiplicity counts how many linearly independent eigenvectors correspond to an eigenvalue.
- Small perturbations in a matrix can cause small or large changes in its eigenvalues and eigenvectors.
- The eigenvalues of a matrix lie within the numerical range, or field of values.
- Matrix powers can be computed using the eigenvalue decomposition: $A^k = V\Lambda^k V^{-1}$.
- The matrix exponential can be computed using the eigenvalue decomposition: $e^A = Ve^{\Lambda}V^{-1}$.

Iterative Methods for Computing Eigenpairs:

1. Power Iteration: Computes the dominant eigenvalue and corresponding eigenvector of a matrix A, by computing repeated matrix-vector multiplications and normalization.
2. Inverse Power Iteration: Used to find the eigenvalue closest to a specified value and requires solving linear systems in each iteration.
3. Rayleigh Quotient Iteration: Improves the convergence of power iteration by using the Rayleigh quotient as an estimate of the desired eigenvalue.
4. QR Algorithm: Uses QR factorizations to compute all eigenvalues of a matrix A and can handle both symmetric and non-symmetric matrices.
5. Arnoldi Iteration: Used for finding a few eigenvalues and corresponding eigenvectors of a large matrix by constructing an orthogonal basis for the Krylov subspace.
6. Subspace Iteration: A generalization of power iteration that computes a basis for a subspace containing the desired eigenvectors.
7. Shifted Inverse Iteration: Combines inverse iteration with shifting to find eigenvalues near a given value.
8. Implicitly Restarted Arnoldi Method: An enhancement of Arnoldi iteration that improves convergence and stability by using deflation and restarting techniques.
9. Krylov-Schur: Avoids the numerical stability of IRAM and does not restrict the decomposition (IRAM: Arnoldi decomp, KS: Schur decomp).