# MATLAB Code
# AY23/24, Y1S1
# Notes

**Sim Ray En Ryan**

December 3, 2025

# Contents

# 1 General MATLAB Commands

## 1.1 Basic Operations and Initialization

- **Clear Command Window**: clc
- **Initialize Matrix**: A = [1 2 3; 4 5 6; 7 8 9]
- **Concatenate Horizontally**: C = [A, B]
- **Concatenate Vertically**: C = [A; B]
- **Zero Matrix**: zeros(m, n)
- **One Matrix**: ones(m, n)
- **Identity Matrix**: eye(n)
- **Most Recent Answer**: ans

## 1.2 Display and Symbolic Operations

- **Format Long Decimal**: format long
- **Format Rational (Fraction)**: format rat
- **Declare Symbolic Variables**: syms a b c
- **Simplify Expression**: simplify(A)

## 1.3 Matrix Manipulation

- **Transpose**: A' or transpose(A)
- **Normalize Columns**: normc(A)
- **Normalize Rows**: normr(A)

# 2 Linear Algebra Core Commands

- **Rank of Matrix**: rank(A)
- **Trace (Sum of Diagonal Elements)**: trace(A)
- **Norm of A**: norm(A)
- **Null Space Basis**: null(A)
- **Orthonormal Basis**: Orth(A)

# 3 Linear Equations and Matrix Algebra

## 3.1 Elementary Row Operations (EROs)

- **R1 ← R1 + 2R2**: A(1,:) = A(1,:) + 2*A(2,:)
- **R1 ↔ R2**: A([1 2],:) = A([2 1],:)
- **R1 ← 2R1**: A(1,:) = 2*A(1,:)
- **Reduced Row Echelon Form (RREF)**: rref(A)

## 3.2 Matrix Inversion and Decomposition

- **Inverse of A**: inv(A)
- **LU Decomposition ($A = LU$)**: [L U] = lu(sym(A))
- **Forward Substitution ($Ly = b$)**: y = inv(L) * b
- **Backward Substitution ($Ux = y$)**: x = inv(U) * y

## 3.3 Determinants

- **Determinant of Square Matrix A**: det(A)
- **Adjugate Matrix**: adjoint(A)
- **Cramer's Rule (For $3\times3$ matrix $A\mathbf{x} = \mathbf{b}$)**: A1=A;A1(:,1)=b;A2=A;A2(:,2)=b;A3=A;A3(:,3)=b; x=(1/det(A))*[det(A1);det(A2);det(A3)]

# 4 Orthogonal Projection and Least Square Approximation

## 4.1 Orthogonal Basis and Projection

- **Gram-Schmidt (QR Decomposition)**: [Q R] = qr(sym(A), 0)
- **Simplify Q**: simplify(Q)
- **Projection Formula**: vproj=(v'*w1)/(w1'*w1)*w1 + ...
- **Manual Gram-Schmidt Step**: v2=u2−(u2'*v1)/(v1'*v1)*v1

## 4.2 Least Square Solution (LSS)

- **Solve $A^TA\mathbf{x} = A^T\mathbf{b}$**: rref([A'*A A'*b])

- **Solve LSS when $A = QR$**: rref ([R Q'*b])

### 4.3 Least Square Approximation (Best Fit Curve)

- **Vandermonde Matrix**: fliplr (vander(v))
- **Set up Data Vectors**: x = [x values ]'; y = [y values ]'
- **Generate Matrix N**: N = fliplr (vander(x))
- **Select Columns for Degree $p$**: N=N(:,1:p) (where $p$ is the degree + 1)
- **Solve for Coefficients c (RREF)**: rref ([N'*N N'*y])
- **Solve for Coefficients c (Direct)**: inv(N'*N)*N'*y
- The resulting vector's elements are coefficients for $x^0, x^1, x^2, \ldots$.

## 5 Diagonalization and Eigenspaces

### 5.1 Eigen-decomposition

- **Characteristic Polynomial**: poly(A)
- **Solve Eigenvalues**: solve (det(x*eye(3) − A))
- **Diagonalization ($A = PDP^{-1}$)**: [P D] = eig(sym(A))

### 5.2 Singular Value Decomposition (SVD)

- **SVD ($A = USV^T$)**: [U S V] = svd(A)
- **Display Symbolic SVD**: sym(U), sym(S), sym(V)
- **Manual SVD Steps**:
  - Compute $B = A^T A$.
  - **Singular Values ($S$)**: Compute eigenvalues of $B$. $S$ has $\sqrt{\lambda_i}$ in descending order.
  - **Right Singular Vectors ($V$)**: Find eigenspace of $B$: sym(null(eigenvalue ∗ eye(n) − B)).
  - **Left Singular Vectors ($U$)**: $\mathbf{u}_i = \frac{1}{\sqrt{\lambda_i}} A\mathbf{v}_i$: V1 = 1/sqrt(eigenvalue) ∗ A ∗ associated eigenspac

## 6 Vectors and Vector Spaces

- Used setdiff to select sub-matrices by excluding columns: V(:, setdiff (a,i)).

# 7 Linear Transformations

- Listed as a topic, no specific commands provided.