# MATRIX WORKSHOP LEARNING SHEET

**Goal:** Understand and practice matrix operations, inverses, transposes, multiplications, change of basis, and numerical derivatives.

## I. Concept Refresher — Without Computing

**1. Matrix Inverse & Identity**
If A is invertible:
$A * A^{-1} = A^{-1} * A = I$ (identity matrix).

**2. Matrix Multiplication Basics**
If A is n x m and B is m x k, then A * B is n x k.
Matrix multiplication is not commutative ($A * B \neq B * A$).

**3. Simplifying Matrix Expressions**
Use $A * A^{-1} = I$ and $B * B^{-1} = I$ repeatedly to simplify long products.

## II. Working with NumPy

Useful functions:
- Transpose: np.transpose(X) or X.T
- Multiply: np.dot(X, Y) or X.dot(Y)
- Inverse: np.linalg.inv(X)
- Addition/Subtraction: +, -

## III. Change of Basis Verification

Formula: $T_{new} = P^{-1} * T_{std} * P$
Check that $T_{new} * P^{-1} * v_{std} = P^{-1} * T_{std} * v_{std}$.
If np.allclose(left, right) returns True, the formula is verified.

General rotation matrix:
T_std(theta) = [[cos(theta), -sin(theta)], [sin(theta), cos(theta)]]

## IV. Numerical Derivative

Approximation formula:
$f'(x_i) \approx (f(x_{i+1}) - f(x_{i-1})) / (x_{i+1} - x_{i-1})$
Use central difference for better accuracy.

## V. Self-Learning: Properties of Square Matrices

**Determinant:**
• For 2x2: ad - bc
• det(AB) = det(A) * det(B)
• det(A^T) = det(A)
• det(A^-1) = 1 / det(A)
• det(A) = 0 → matrix not invertible.

**Eigenvalues & Eigenvectors:**
A * v = lambda * v; solve |A - lambda * I| = 0 for lambda.
Each lambda has one or more eigenvectors.

**Diagonalization:**
A = P * D * P^-1, where D is diagonal (eigenvalues) and columns of P are eigenvectors.
Possible only if A has enough independent eigenvectors.

**Summary Checklist:**
• Simplify A * A^-1 = I
• Predict output matrix sizes
• Compute transpose, inverse, and product
• Verify change-of-basis formula
• Implement central-difference derivative
• Understand determinant, eigenvalues, diagonalization