Title: **Benchmarking the Numerical Stability of Direct and Iterative Solvers in Image Decryption: A Comparative Study of Dense and Sparse Matrix Systems**

Algorithm	Method Type	Key Strength to Test
Gaussian Elimination	Direct	How well it handles pivoting in sparse systems.
LU Decomposition	Factorization	Speed efficiency when reusing the same sparse key.
Gauss-Seidel	Iterative	Its famous ability to converge faster on sparse matrices.

The Two Matrix Types:

1. **Dense Matrix:** Every entry in the 8×8 key is a non-zero number.
2. **Sparse Matrix:** Only about 20% of the key entries are non-zero.

Methodology

Step 1: Image Discretization

You convert your image into a numerical matrix and divide it into 8×8 pixel blocks.

Step 2: Key Generation

You generate two different keys:

- Key_Dense: A standard random matrix.
- Key_Sparse: A matrix where most elements are zeroed out.

Step 3: Noise Injection (The Stress Test)

Before decryption, you add **Gaussian White Noise** to the encrypted blocks. This is your independent variable—how much noise can each algorithm handle before the image is destroyed?

Possible Result

Eye Test					
Condition	**Original**	**Encrypted + Noise**	**Gaussian Result**	**LU Result**	**Gauss-Seidel Result**
Dense Key	[Image]	[Noise]	Clear	Clear	Grainy
Sparse Key	[Image]	[Noise]	Clear	Clear	**Cleanest**

Numerical Accuracy				
Matrix Type	**Metric**	**Gaussian**	**LU Decomposition**	**Gauss-Seidel**
Dense	PSNR (dB)	34.5	34.3	28.1
	MSE	0.021	0.022	0.085
Sparse	PSNR (dB)	33.9	33.7	**36.4**

	MSE	0.025	0.026	**0.015**

Section	Content Strategy for your Paper
Introduction	**The Context:** Define the importance of matrix-based encryption (Hill Cipher) in digital image security. **The Problem:** Discuss how high-dimensional matrices are vulnerable to rounding errors and data corruption (stochastic noise). **The Objective:** To evaluate the numerical robustness, speed, and accuracy of three solvers across **Sparse(why we also choose this too)** and **Dense** key matrices.
Methods	**Algorithms:** Provide the pseudocode for **Gaussian Elimination (with Partial Pivoting)**, **LU Decomposition**, and the **Gauss-Seidel Iterative Method**. **The Experiment:** Explain the process of partitioning an image into blocks, generating two types of key matrices (Dense vs. Sparse), and injecting Gaussian noise ($C' = C + \epsilon$) to simulate transmission errors.
Results	**Visual Data:** Present the "Wall of Decryption"—side-by-side images of the original, the noisy encrypted version, and the recovered results for all three algorithms. **Quantitative Data:** Provide comparison tables for **PSNR (Peak Signal-to-Noise Ratio)**, **MSE (Mean Squared Error)**, and **Execution Time** (in milliseconds) for both matrix types.

Discussion	**The Comparison:** Contrast the performance of **Direct methods (Gaussian/LU)** vs. **Iterative methods (Gauss-Seidel)**. **The Discovery:** Explain why **Gauss-Seidel** converged faster on sparse matrices and how it potentially filtered out stochastic noise better than the exact solvers. **Stability:** Analyze the impact of the **Condition Number** on the final image quality.