Sparse Matrices

09/14/2020

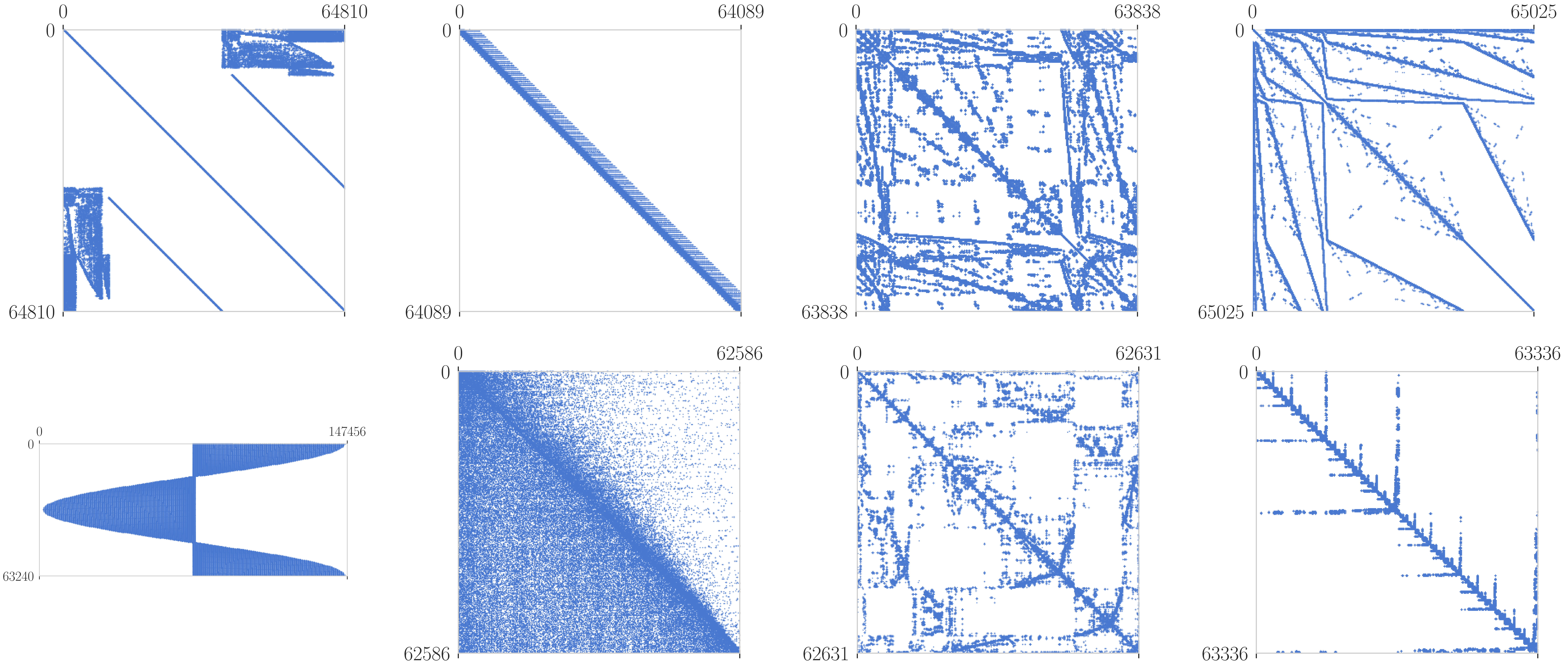
Matrices

- Looked at these pretty extensively

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 5 \\ 2 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 2 & 2 & 3 \\ 5 & 7 & 1 \\ 2 & 0 & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 18 & 16 & 14 \\ 21 & 13 & 25 \\ 11 & 11 & 10 \end{bmatrix}$$

- What if most elements are the number 0… we can ignore them

Some Sparse Matrices



Suitesparse Matrix Collection : https://sparse.tamu.edu/

Why Sparse Matrices?

- In real world applications, coming from all areas of STEM, each variable only depends on a subset of the other variables
- One big example : Partial Differential Equations - represented as sparse systems (sparse matrix, dense vectors)
- Ignoring the zeros improves
  - Memory requirements
  - Computation costs

Sparse Matrix Format

- Condensed Sparse Row (CSR) Format
- A sparse matrix is stored in three different variables:
  1. Row Pointer : points to position in 2. And 3. that start each row
  2. Column Indices : columns that hold a non-zero, ordered by row
  3. Data Values : Values associated with each column index in 2.

Example CSR Matrix

- How to step through a sparse matrix :

```
for row = 0 to num_rows:
    for idx = row_ptr[row] to row_ptr[row+1]
        col = col_indices[idx]
        val = data[idx]
```

- Exactly how many reads and writes to step through a matrix?
  - Number of non-zeros (**nnz**)

Matrix Multiplication

- Going back to dense matrices, what is the big O notation for multiplying two matrices? **O(n^3)**

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 5 \\ 2 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 2 & 2 & 3 \\ 5 & 7 & 1 \\ 2 & 0 & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 18 & 16 & 14 \\ 21 & 13 & 25 \\ 11 & 11 & 10 \end{bmatrix}$$

- What is the storage requirement? **3*n^2**

Sparse Matrix Multiplication

- What is the big O notation? **O(n^2)**

```
for row = 0 to num_rows:
    for idx = row_ptr[row] to row_ptr[row+1]
        col = col_indices[idx]
        val = data[idx]
```

- Storage requirements? **3*nnz (although nnz likely different for each)**

Harder to model performance:

- Easy to model / predict performance of structured codes, like dense matrix-matrix multiplication
- Performance is the same, regardless of the data in the matrix
- However, for sparse matrices, the performance and cost is dependent on the number of non-zeros, which is matrix dependent.

Multiplying Sparse Matrices

- We will step through an example of this is class
- Before Monday, think about how you might step through sparse matrix multiplication