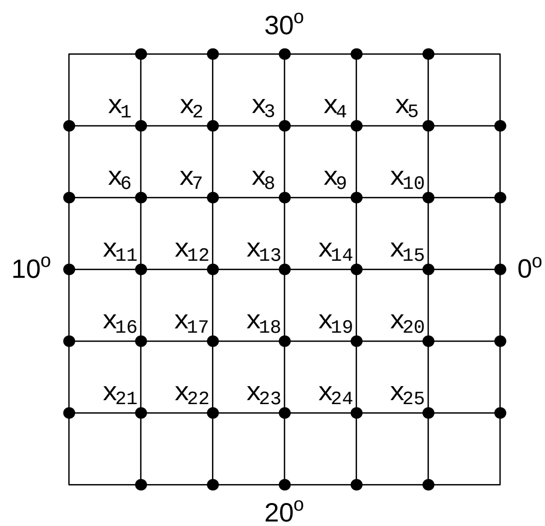
2.4 Partitioned Matrices

In real world problems, systems can have huge numbers of equations and unknowns. Thousands of equations and hundreds of thousands of variables are not uncommon. Standard computation techniques are inefficient in such cases, so we need to develop techniques which exploit the internal structure of the matrices. In most cases, the matrices of interest have lots of zeros: they are **sparse**.

For example, consider a fluid dynamics problem such as computing equilibrum tmperatures. It is solved by placing a fine grid over a region of interest and analyzing the situation at each grid point. The data at each point only effects the data at neighboring points, and we know the data at the edges of the grid. In the grid



there is an equation for each of the 25 internal grid points; thus 25 equations in 25 unknowns, but each equation has at most 5 variables. A sparse coefficient matrix results:

$$\left[\begin{array}{ccccccccccccccccccccccccc}
\bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \bullet & \bullet & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\bullet & 0 & 0 & 0 & 0 & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & 0 & 0 & 0 & 0 & \bullet \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & \bullet & \bullet & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet
\end{array}\right]$$

Partitioning a matrix: Splitting it up into smaller matrices.

Consider

$$A = \begin{bmatrix} 2 & 1 & -3 & 5 & 7 \\ 0 & 1 & 4 & -1 & 2 \\ 3 & -2 & 1 & 1 & 5 \\ 0 & 4 & 0 & 1 & -3 \end{bmatrix}$$

Example 1: One way to partition A:

$$\left[\begin{array}{cc|ccc} 2 & 1 & -3 & 5 & 7 \\ 0 & 1 & 4 & -1 & 2 \\ \hline 3 & -2 & 1 & 1 & 5 \\ 0 & 4 & 0 & 1 & -3 \end{array}\right]$$

which is recorded

$$A_{11} = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}, A_{12} = \begin{bmatrix} -3 & 5 & 7 \\ 4 & -1 & 2 \end{bmatrix}$$

$$A_{21} = \begin{bmatrix} 3 & -2 \\ 0 & 4 \end{bmatrix}, A_{22} = \begin{bmatrix} 1 & 1 & 5 \\ 0 & 1 & -3 \end{bmatrix}$$

Example 2: Another way to partition A:

$$\left[\begin{array}{rrr|rr} 2 & 1 & -3 & 5 & 7 \\ 0 & 1 & 4 & -1 & 2 \\ 3 & -2 & 1 & 1 & 5 \\ \hline 0 & 4 & 0 & 1 & -3 \end{array}\right]$$

Example 3: Another way to partition A:

$$\left[\begin{array}{r|r|r|r|r} 2 & 1 & -3 & 5 & 7 \\ 0 & 1 & 4 & -1 & 2 \\ 3 & -2 & 1 & 1 & 5 \\ 0 & 4 & 0 & 1 & -3 \end{array}\right]$$

Example 4: Another way to partition A:

$$\left[\begin{array}{rrrrr} 2 & 1 & -3 & 5 & 7 \\ \hline 0 & 1 & 4 & -1 & 2 \\ \hline 3 & -2 & 1 & 1 & 5 \\ \hline 0 & 4 & 0 & 1 & -3 \end{array}\right]$$

Operations with Partitioned Matrices

Addition: if 2 $m \times n$ matrices are partitioned in the same way, add corresponding blocks.

Scalar Multiplication: multiply each block by the scalar.

Matrix Multiplication: to multiply two partitioned matrices A and B, the column partition of A must match the row partition of B (the partition is **conformable**.) Use the usual row-column rule considering each block as a single entry.

Example 5 – Partition 1:

$$A = \left[\begin{array}{rrr|rr} 2 & 1 & -3 & 5 & 7 \\ 0 & 1 & 4 & -1 & 2 \\ 3 & -2 & 1 & 1 & 5 \\ \hline 0 & 4 & 0 & 1 & -3 \end{array}\right] = \left[\begin{array}{cc} A_{11} & A_{12} \\ A_{21} & A_{22} \end{array}\right]$$

$$B = \left[\begin{array}{rrr} 1 & -2 & 1 \\ -1 & 2 & 0 \\ 5 & 0 & -1 \\ \hline 6 & 3 & -4 \\ 0 & 3 & -1 \end{array}\right] = \left[\begin{array}{c} B_1 \\ B_2 \end{array}\right]$$

$$AB = \left[\begin{array}{c} A_{11}B_1 + A_{12}B_2 \\ A_{21}B_1 + A_{22}B_2 \end{array}\right] = \left[\begin{array}{rrr} 16 & -38 & -8 \\ 13 & -1 & 2 \\ 16 & -28 & 3 \\ \hline 2 & 14 & -7 \end{array}\right]$$

Example 5 – Partition 2:

$$A = \left[\begin{array}{rrrrr} 2 & 1 & -3 & 5 & 7 \\ 0 & 1 & 4 & -1 & 2 \\ 3 & -2 & 1 & 1 & 5 \\ 0 & 4 & 0 & 1 & -3 \end{array}\right]$$

$$B = \left[\begin{array}{r|r|r} 1 & -2 & 1 \\ -1 & 2 & 0 \\ 5 & 0 & -1 \\ 6 & 3 & -4 \\ 0 & 3 & -1 \end{array}\right] = \left[\begin{array}{ccc} B_1 & B_2 & B_3 \end{array}\right]$$

$$AB = \left[\begin{array}{ccc} AB_1 & AB_2 & AB_3 \end{array}\right]$$

This is the definition of matrix multiplication: the columns of AB are A multiplied by the columns of B.

Example 5 – Partition 3:

$$A = \left[\begin{array}{rrrrr} 2 & 1 & -3 & 5 & 7 \\ \hline 0 & 1 & 4 & -1 & 2 \\ \hline 3 & -2 & 1 & 1 & 5 \\ \hline 0 & 4 & 0 & 1 & -3 \end{array}\right] = \left[\begin{array}{c} A_1 \\ A_2 \\ A_3 \\ A_4 \end{array}\right]$$

$$B = \begin{bmatrix} 1 & -2 & 1 \\ -1 & 2 & 0 \\ 5 & 0 & -1 \\ 6 & 3 & -4 \\ 0 & 3 & -1 \end{bmatrix}$$

$$AB = \begin{bmatrix} A_1B \\ A_2B \\ A_3B \\ A_4B \end{bmatrix}$$

Another understanding of matrix multiplication: the rows of AB are the rows of A multiplied by B.

Example 5 – Partition 4:

$$A = \left[\begin{array}{c|c|c|c|c} 2 & 1 & -3 & 5 & 7 \\ 0 & 1 & 4 & -1 & 2 \\ 3 & -2 & 1 & 1 & 5 \\ 0 & 4 & 0 & 1 & -3 \end{array}\right] = \begin{bmatrix} A_1 & A_2 & A_3 & A_4 & A_5 \end{bmatrix}$$

$$B = \left[\begin{array}{ccc} 1 & -2 & 1 \\ \hline -1 & 2 & 0 \\ \hline 5 & 0 & -1 \\ \hline 6 & 3 & -4 \\ \hline 0 & 3 & -1 \end{array}\right] = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{bmatrix}$$

$$AB = A_1B_1 + A_2B_2 + A_3B_3 + A_4B_4 + A_5B_5$$

This is called expressing AB as a sum of **outer products**, and it turns out to be a nifty way of decomposing the product AB.

Why partition matrices?

Example 6: Consider multiplying

$$\begin{bmatrix} A & O \\ I & B \end{bmatrix} \begin{bmatrix} C & I \\ O & D \end{bmatrix}$$

where A, B, C, and D are $n \times n$, O is the $n \times n$ zero matrix, and I is the $n \times n$ identity matrix.

Direct multiplication: to compute each term uses $2n$ multiplications and $2n$ additions, or $4n$ flops (floating point operations). Since there are $4n^2$ terms in the resulting matrix, direct multiplication uses __________ flops. So if $n = 1000$, the number of flops would be __________.

Partitioned multiplication:

$$\begin{bmatrix} A & O \\ I & B \end{bmatrix} \begin{bmatrix} C & I \\ O & D \end{bmatrix} = \begin{bmatrix} AC + OO & AI + OD \\ IC + BO & I + BD \end{bmatrix} = \begin{bmatrix} AC & A \\ C & I + BD \end{bmatrix}$$

To compute AC takes $2n \cdot n^2$ flops, to compute BD takes $2n \cdot n^2$ flops, to add $I + BD$ takes n flops, so partitioned multiplication uses __________ flops. So if $n = 1000$, the number of flops would be __________.

Example 7: Consider the original example. Partitioning can exploit internal structure:

$$\left[\begin{array}{ccccc|ccccc|ccccc|ccccc|ccccc}
\bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \bullet & \bullet & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\bullet & 0 & 0 & 0 & 0 & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & 0 & 0 & 0 & 0 & \bullet \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & 0 & \bullet & \bullet & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet & \bullet \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bullet & 0 & 0 & 0 & \bullet & \bullet
\end{array}\right]$$

Solving for blocks:

Example 8: Consider the equation

$$\begin{bmatrix} A & B \\ O & D \end{bmatrix} \begin{bmatrix} Z & O \\ X & Y \end{bmatrix} = \begin{bmatrix} O & O \\ I & O \end{bmatrix}$$

where all blocks are $n \times n$ and A is invertible. Find formulas for X, Y and Z.

Multiplying and setting the results equal gives:

$$AZ + BX = O, AO + BY = O, OZ + DX = I, OO + DY = O$$

or

$$AZ + BX = O, BY = O, DX = I, DY = O$$

Since X is square, the IMT gives $X = D^{-1}$. Since D is also invertible by the IMT, $DY = O$ implies $Y = D^{-1}O = O$. Finally, $AZ + BX = O$ is equivalent to $AZ+BD^{-1} = O$. Thus $AZ = -BD^{-1}$, and since A is invertible, $Z = A^{-1}BD^{-1}$.