What is a Sparse Matrix?

Nicholas J. Higham*

September 8, 2020

A sparse matrix is one with a large number of zero entries. A more practical definition is that a matrix is sparse if the number or distribution of the zero entries makes it worthwhile to avoid storing or operating on the zero entries.

Sparsity is not to be confused with data sparsity, which refers to the situation where, because of redundancy, the data can be efficiently compressed while controlling the loss of information. Data sparsity typically manifests itself in low rank structure, whereas sparsity is solely a property of the pattern of nonzeros.

Important sources of sparse matrices include discretization of partial differential equations, image processing, optimization problems, and networks and graphs. In designing algorithms for sparse matrices we have several aims.

- Store the nonzeros only, in some suitable data structure.
- Avoid operations involving only zeros.
- Preserve sparsity, that is, minimize *fill-in* (a zero element becoming nonzero).

We wish to achieve these aims without sacrificing speed, stability, or reliability.

An important class of sparse matrices is *banded matrices*. A matrix A has bandwidth p if the elements outside the main diagonal and the first p superdiagonals and subdiagonals are zero, that is, if $a_{ij} = 0$ for $j > i + p$ and $i > j + p$.

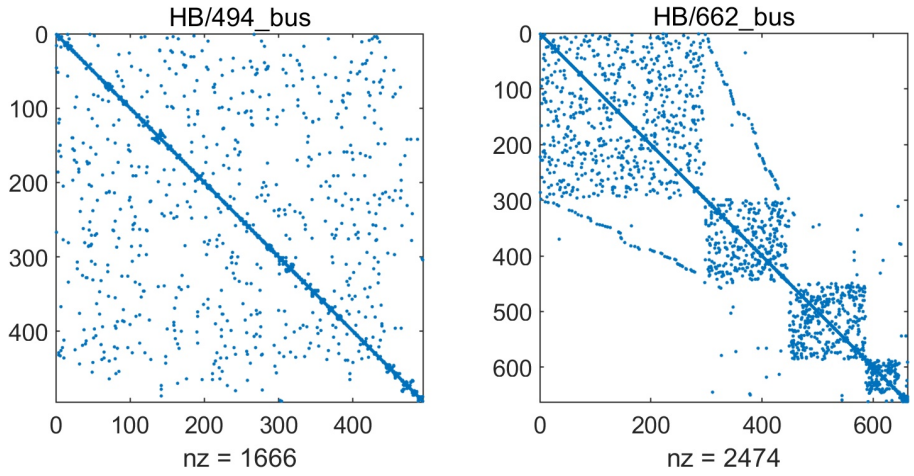
The most common type of banded matrix is a tridiagonal matrix ($p = 1$), of which an archetypal example is the second-difference matrix, illustrated for $n = 5$ by

$$A_5 = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}.$$

This matrix (or more precisely its negative) corresponds to a centered finite difference approximation to a second derivative: $f''(x) \approx (f(x+h) - 2f(x) + f(x-h))/h^2$.

The following plots show the sparsity patterns for two symmetric positive definite matrices. Here, the nonzero elements are indicated by dots.

*Department of Mathematics, University of Manchester, Manchester, M13 9PL, UK (nick.higham@manchester.ac.uk).



The matrices are both from power network problems and they are taken from the SuiteSparse Matrix Collection (`https://sparse.tamu.edu/`). The matrix names are shown in the titles and the `nz` values below the x-axes are the numbers of nonzeros. The plots were produced using MATLAB code of the form

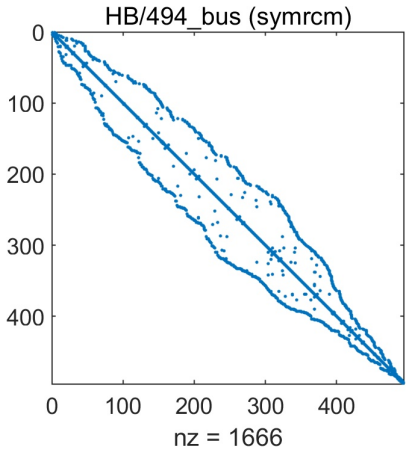
```
W = ssget('HB/494_bus'); A = W.A; spy(A)
```

where the `ssget` function is provided with the collection. The matrix on the left shows no particular pattern for the nonzero entries, while that on the right has a structure comprising four diagonal blocks with a relatively small number of elements connecting the blocks.

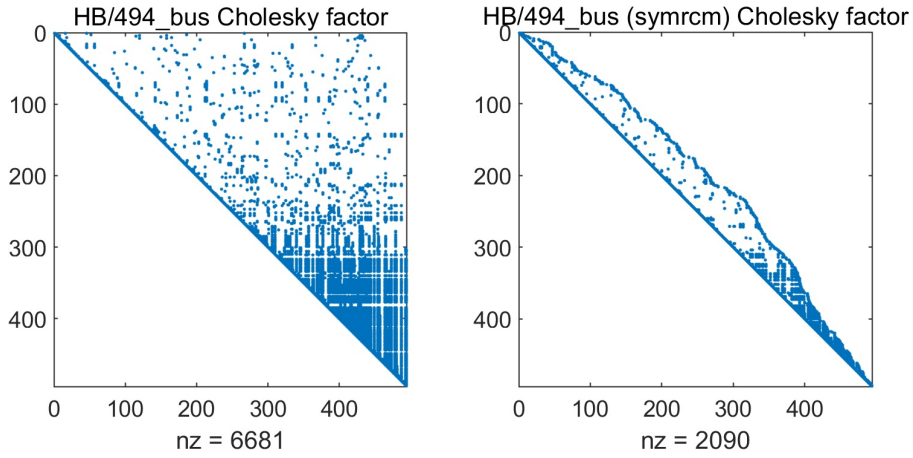
It is important to realize that while the sparsity pattern often reflects the structure of the underlying problem, it is arbitrary in that it will change under row and column reorderings. If we are interested in solving $Ax = b$, for example, then for any permutation matrices P and Q we can form the transformed system $PAQ(Q^*x) = Pb$, which has a coefficient matrix PAQ having permuted rows and columns, a permuted right-hand side Pb, and a permuted solution. We usually wish to choose the permutations to minimize the fill-in or (almost equivalently) the number of nonzeros in L and U. Various methods have been derived for this task; they are necessarily heuristic because finding the minimum is in general an NP-complete problem. When A is symmetric we take $Q = P^T$ in order to preserve symmetry.

For the `HB/494_bus` matrix the symmetric reverse Cuthill-McKee permutation gives a reordered matrix with the following sparsity pattern, plotted with the MATLAB commands

```
r = symrcm(A); spy(A(r,r))
```



The reordered matrix with a variable band structure that is characteristic of the symmetric reverse Cuthill-McKee permutation. The number of nonzeros is, of course, unchanged by reordering, so what has been gained? The next plots show the Cholesky factors of the `HB/494_bus` matrix and the reordered matrix. The Cholesky factor for the reordered matrix has a much narrower bandwidth than that for the original matrix and has fewer nonzeros by a factor 3. Reordering has greatly reduced the amount of fill-in that occurs; it leads to a Cholesky factor that is cheaper to compute and requires less storage.



Because Cholesky factorization is numerically stable, the matrix can be permuted without affecting the numerical stability of the computation. For a nonsymmetric problem the choice of row and column interchanges also needs to take into account the need for numerical stability, which complicates matters.

The world of sparse matrix computations is very different from that for dense matrices. In the first place, sparse matrices are not stored as $n \times n$ arrays, but rather just the nonzeros are stored, in some suitable data structure. Programming sparse matrix computations is, consequently, more difficult than for dense matrix computations. A second difference from the dense case is that certain operations are, for practical purposes, forbidden, Most notably, we never invert sparse matrices because of the possibly severe fill-in. Indeed the inverse of a sparse matrix is usually dense. For example, the inverse

of the tridiagonal matrix given at the start of this article is

$$A_5^{-1} = \frac{1}{6}\begin{bmatrix} 5 & 4 & 3 & 2 & 1 \\ 4 & 8 & 6 & 4 & 2 \\ 3 & 6 & 9 & 6 & 3 \\ 2 & 4 & 6 & 8 & 4 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}.$$

While it is always true that one should not solve $Ax = b$ by forming $x = A^{-1} \times b$, for reasons of cost and numerical stability (unless A is orthogonal!), it is even more true when A is sparse.

Finally, we mention an interesting property of A_5^{-1}. Its upper triangle agrees with the upper triangle of the rank-1 matrix

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} \begin{bmatrix} 5 & 4 & 3 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 4 & 3 & 2 & 1 \\ 10 & 8 & 6 & 4 & 2 \\ 15 & 12 & 9 & 6 & 3 \\ 20 & 16 & 12 & 8 & 4 \\ 25 & 20 & 15 & 10 & 5 \end{bmatrix}.$$

This property generalizes to other tridiagonal matrices. So while a tridiagonal matrix is sparse, its inverse is data sparse—as it has to be because in general A depends on $2n-1$ parameters and hence so does A^{-1}. One implication of this property is that it is possible to compute the condition number $\kappa_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty$ of a tridiagonal matrix in $O(n)$ flops.

References

This is a minimal set of references, which contain further useful references within.

- Timothy A. Davis, Direct Methods for Sparse Linear Systems, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006.
- Timothy A. Davis, Sivasankaran Rajamanickam, and Wissam M. Sid-Lakhdar, A Survey of Direct Methods for Sparse Linear Systems, Acta Numerica 25, 383–566, 2016.
- Timothy A. Davis and Yifan Hu, The University of Florida Sparse Matrix Collection, ACM Trans. Math. Software 38 (1), 1:1–1:25, 2011. *Note*: this collection is now called the SuiteSparse Matrix Collection.
- Gareth I. Hargreaves, Computing the Condition Number of Tridiagonal and Diagonal-Plus-Semiseparable Matrices in Linear Time, SIAM J. Matrix Anal. Appl. 27, 801–820, 2006.
- Gérard Meurant, A Review on the Inverse of Symmetric Tridiagonal and Block Tridiagonal Matrices, SIAM J. Matrix Anal. Appl. 13, 707–728, 1992.
- Yousef Saad, Iterative Methods for Sparse Linear Systems, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

This article is part of the "What Is" series, available from `https://nhigham.com/category/what-is` and in PDF form from the GitHub repository `https://github.com/higham/what-is`.