What Is a Blocked Algorithm?

Nicholas J. Higham[*]

October 28, 2021

In numerical linear algebra a blocked algorithm organizes a computation so that it works on contiguous chunks of data. A blocked algorithm and the corresponding unblocked algorithm (with blocks of size 1) are mathematically equivalent, but the blocked algorithm is generally more efficient on modern computers.

A simple example of blocking is in computing an inner product $s = x^T y$ of vectors $x, y \in \mathbb{R}^n$. The unblocked algorithm

```
1. $s = x_1 y_1$
2. for $i = 2\colon n$
3.     $s = s + x_k y_k$
4. end
```

can be expressed in blocked form, with block size b, as

```
1. for $i = 1\colon n/b$
2.     $s_i = \sum_{j=(i-1)b+1}^{ib} x_j y_j$ % Compute by the unblocked algorithm.
3. end
4. $s = \sum_{i=1}^{n/b} s_i$
```

The sums of b terms in line 2 of the blocked algorithm are independent and could be evaluated in parallel, whereas the unblocked form is inherently sequential.

To see the full benefits of blocking we need to consider an algorithm operating on matrices, of which matrix multiplication is the most important example. Suppose we wish to compute the product $C = AB$ of $n \times n$ matrices A and B. The natural computation is, from the definition of matrix multiplication, the "point algorithm"

```
1. $C = 0$
2. for $i = 1\colon n$
3.    for $j = 1\colon n$
4.       for $k = 1\colon n$
5.          $c_{ij} = c_{ij} + a_{ik} b_{kj}$
6.       end
7.    end
8. end
```

Let $A = (A_{ij})$, $B = (B_{ij})$, and $C = (C_{ij})$ be partitioned into blocks of size b, where $r = n/b$ is assumed to be an integer. The blocked computation is

```
1. $C = 0$
2. for $i = 1\colon r$
3.    for $j = 1\colon r$
```

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

```
4.        for $k = 1\colon r$
5.            $C_{ij} = C_{ij} + A_{ik}B_{kj}$
6.        end
7.    end
8. end
```

On a computer with a hierarchical memory the blocked form can be much more efficient than the point form if the blocks fit into the high speed memory, as much less data transfer is required. Indeed line 5 of the blocked algorithm performs $O(b^3)$ flops on about $O(n^2)$ data, whereas the point algorithm performs $O(1)$ flops on $O(1)$ data on line 5, or $O(n)$ flops on $O(n)$ data if we combine lines 4–6 into a vector inner product. It is the $O(b)$ flops-to-data ratio that gives the blocked algorithm its advantage, because it masks the memory access costs.

The LAPACK (first released in 1992) was the first program library to systematically use blocked algorithms for a wide range of linear algebra computations.

An extra advantage that blocked algorithms have over unblocked algorithms is a reduction in the error constant in a rounding error bound by a factor b or more and, typically, a reduction in the actual error (Higham, 2021).

The adjective "block" is sometimes used in place of "blocked", but we prefer to reserve "block" for the meaning described in the next section.

Block Algorithms

A block algorithm is a generalization of a scalar algorithm in which the basic scalar operations become matrix operations ($\alpha + \beta$, $\alpha\beta$, and α/β become $A + B$, AB, and AB^{-1}). It usually computes a block factorization, in which a matrix property based on the nonzero structure becomes the corresponding property blockwise; in particular, the scalars 0 and 1 become the zero matrix and the identity matrix, respectively.

An example of a block factorization is block LU factorization. For a 4×4 matrix A an LU factorization can be written in the form

$$A = \left[\begin{array}{cc|cc} 1 & & & \\ \mathrm{x} & 1 & & \\ \hline \mathrm{x} & \mathrm{x} & 1 & \\ \mathrm{x} & \mathrm{x} & \mathrm{x} & 1 \end{array}\right] \left[\begin{array}{cc|cc} \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \hline & & \mathrm{x} & \mathrm{x} \\ & & & \mathrm{x} \end{array}\right].$$

A block LU factorization has the form

$$A = \left[\begin{array}{cc|cc} 1 & 0 & & \\ 0 & 1 & & \\ \hline \mathrm{x} & \mathrm{x} & 1 & 0 \\ \mathrm{x} & \mathrm{x} & 0 & 1 \end{array}\right] \left[\begin{array}{cc|cc} \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \hline & & \mathrm{x} & \mathrm{x} \\ & & \mathrm{x} & \mathrm{x} \end{array}\right].$$

Clearly, these are different factorizations. In general, a block LU factorization has the form $A = LU$ with L block lower triangular, with identity matrices on the diagonal, and U block upper triangular. A blocked algorithm for computing the LU factorization and a block algorithm for computing the block LU factorization are readily derived.

The adjective "block" also applies to a variety of matrix properties, for which there are often special-purpose block algorithms. For example, the matrix

$$\begin{bmatrix} A_{11} & A_{12} & & & \\ A_{21} & A_{22} & A_{23} & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & A_{n-1,1} \\ & & & A_{n,n-1} & A_{n,n} \end{bmatrix}$$

is a block tridiagonal matrix, and a block Toeplitz matrix has constant block diagonals:

$$\begin{bmatrix} A_1 & A_2 & \dots & \dots & A_n \\ A_{-1} & A_1 & A_2 & & \vdots \\ \vdots & A_{-1} & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & A_2 \\ A_{1-n} & \dots & \dots & A_{-1} & A_1 \end{bmatrix}.$$

One can define block diagonal dominance as a generalization of diagonal dominance. A block Householder matrix generalizes a Householder matrix: it is a perturbation of the identity matrix by a matrix of rank greater than or equal to 1.

References

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

- Nicholas J. Higham, Accuracy and Stability of Numerical Algorithms, second edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002. (Chapter 13, "Block LU Factorization".)
- Nicholas J. Higham, Numerical Stability of Algorithms at Extreme Scale and Low Precisions, MIMS EPrint 2021.14, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, September 2021.

Related Blog Posts

- Can We Solve Linear Algebra Problems at Extreme Scale and Low Precisions? (2021)
- What Is a Block Matrix? (2020)
- What Is an LU Factorization? (2021)

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`.