On Computing with Diagonally Structured Matrices

Shahadat Hossain[1]

[1]Department of Mathematics and Computer Science
University of Lethbridge, Canada

The 2019 IEEE High Performance Extreme Computing Conference
Waltham, Massachusetts, USA, Sept 24–26, 2019
(Joint work with Mohammad Sakib Mahmud)

Outline

Numerical Linear Algebra and BLAS

GEMM (BLAS Level-3)

$$C \leftarrow \beta C + \alpha AB \tag{1}$$

and

$$C \leftarrow \beta C + \alpha A^{\top} B \tag{2}$$

$A \in \mathbb{R}^{m \times k}$; $B \in \mathbb{R}^{k \times n}$; $C \in \mathbb{R}^{m \times n}$; $\alpha, \beta \in \mathbb{R}$

Libraries

Intel MKL, ATLAS, GotoBLAS/OpenBLAS

Structured Problems

Problem (Long-range Transport of Air Pollutants (Zlatev at al.))

- *Model is described by 29 PDE's*
- *Advection, Diffusion, deposition, and Chemical Reactions*
- *Space discretization converts to 29N ODEs, N is the number of grid points in the space domain*
- *Numerical scheme leads to symmetric positive definite banded matrices.*

Observation

Narrow band (or small number of nonzero diagonals); organize computations by diagonals

Banded Matrix Storage Schemes

Notations

$A \in \mathbb{R}^{n \times n}$ is banded matrix with lower band-width k_l and upper bandwidth k_u if

$$j > k_u + i \implies a_{ij} = 0 \text{ and } i > k_l + j \implies a_{ij} = 0.$$

kth superdiagonal: $A_k = \{a_{ij} \mid j - i = k\}$; kth subdiagonal: $A_{-k} = \{a_{ij} \mid i - j = k\}$

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & 0 \\ a_{10} & a_{11} & a_{12} & a_{13} \\ 0 & a_{21} & a_{22} & a_{23} \\ 0 & 0 & a_{32} & a_{33} \end{pmatrix}$$

A banded matrix

$$\begin{pmatrix} * & * & a_{02} & a_{13} \\ * & a_{01} & a_{12} & a_{23} \\ a_{00} & a_{11} & a_{22} & a_{33} \\ a_{10} & a_{21} & a_{32} & * \end{pmatrix}$$

Compact representation (BLAS)

$$\begin{pmatrix} a_{02} & a_{13} & * & * \\ a_{01} & a_{12} & a_{23} & * \\ a_{00} & a_{11} & a_{22} & a_{33} \\ a_{10} & a_{21} & a_{32} & * \end{pmatrix}$$

Modified compact representation (Tsao and Turnbull, 1993)

Remark

$O({k_l}^2) + O({k_u}^2)$ *uninitialized memory*

Diagonal Storage Scheme: Our Proposal

Nonzero entries within the band are stored in an array of size

$$n(k_u + k_l + 1) - \frac{k_u(k_u+1)}{2} - \frac{k_l(k_l+1)}{2}$$

and are laid out by diagonals,

$$\left(A_0; A_1; \ldots; A_{k_u}; A_{-1}; A_{-2}; \ldots; A_{-k_l}\right).$$

Example,

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & 0 \\ a_{10} & a_{11} & a_{12} & a_{13} \\ 0 & a_{21} & a_{22} & a_{23} \\ 0 & 0 & a_{32} & a_{33} \end{pmatrix}$$

stored by diagonals:

$$\begin{pmatrix} a_{00} & a_{11} & a_{22} & a_{33}; & a_{01} & a_{12} & a_{23}; & a_{02} & a_{13}; & a_{10} & a_{21} & a_{32} \end{pmatrix}$$

Where is the kth diagonal?

x: address or index of the first element of A_0 The first element of the kth superdiagonal at

$$x + nk - \frac{k(k-1)}{2}.$$

For $k = k_u$,

$$x + nk_u - \frac{k_u(k_u-1)}{2}.$$

The first element of kth subdiagonal at

$$y + (k-1)(n-1) - \frac{k(k-1)}{2}$$

where,

$$y = x + nk_u - \frac{k_u(k_u-1)}{2} + (n-k_u).$$

Remark

- *Dense matrix:* $k_u = k_l = n-1$
- *Lower (Upper) triangular matrix:* $k_u(k_l) = 0(n-1), k_l(k_u) = n-1(0)$

Computing by Diagonal

$$\begin{pmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{pmatrix} \leftarrow \begin{pmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{pmatrix}$$

$$+ \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{pmatrix}$$

Compute c_{03}

- Scalar product:

$$c_{03} \leftarrow c_{03} + \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \cdot & \cdot & \cdot & b_{03} \\ \cdot & \cdot & \cdot & b_{13} \\ \cdot & \cdot & \cdot & b_{23} \\ \cdot & \cdot & \cdot & b_{33} \end{pmatrix}$$

- Hadamard product :

$$C_3 \leftarrow C_3 + A_0 \times B_3 + A_1 \times B_2 + A_2 \times B_1 + A_3 \times B_0$$

Compute the *k*th Diagonal

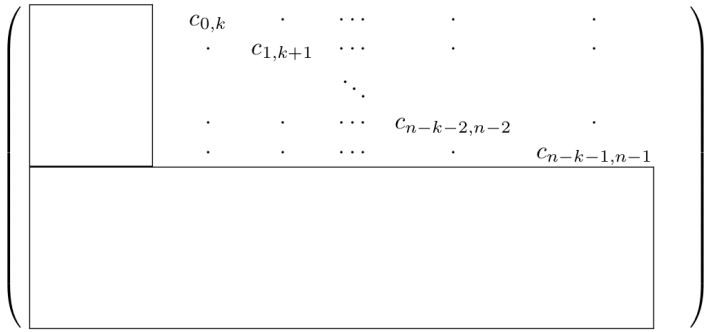

Fig. 4. kth super diagonal of the product matrix C

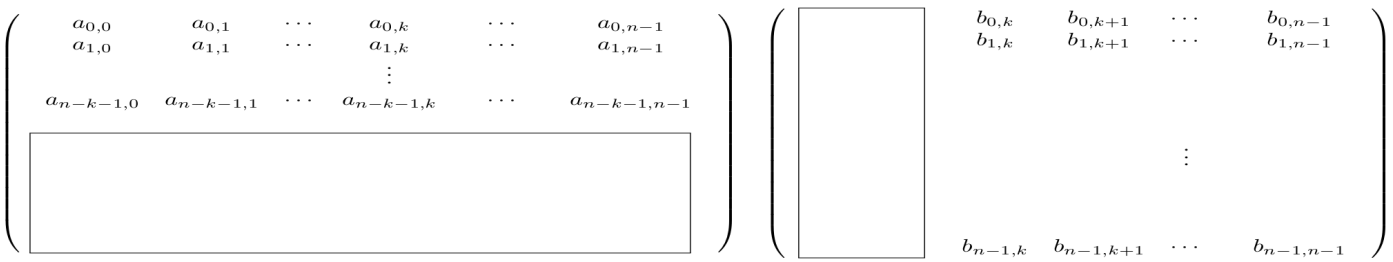

Fig. 3. Active sections of operands A and B for computing super diagonal C_k

Remark

Diagonal C_k is obtained from the sum of the Hadamard products of the diagonal pairs A_i and B_j where $k = i + j$

Compute the *k*th Diagonal

$$\begin{pmatrix} c_{00} & \cdot & \cdot & \cdot \\ \cdot & c_{11} & \cdot & \cdot \\ \cdot & \cdot & c_{22} & \cdot \\ \cdot & \cdot & \cdot & c_{33} \end{pmatrix} \leftarrow \begin{pmatrix} c_{00} & \cdot & \cdot & \cdot \\ \cdot & c_{11} & \cdot & \cdot \\ \cdot & \cdot & c_{22} & \cdot \\ \cdot & \cdot & \cdot & c_{33} \end{pmatrix} + \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{pmatrix}$$

$$C_0 \quad +A_0 \times B_0$$

$$+(A_1 \times B_{-1}) + (A_{-1} \times B_1)$$

$$+(A_{-2} \times B_2) + (A_2 \times B_{-2})$$

$$+(A_3 \times B_{-3}) + (A_{-3} \times B_3)$$

$$\begin{pmatrix} c_{00} \\ c_{11} \\ c_{22} \\ c_{33} \end{pmatrix} \leftarrow \begin{pmatrix} c_{00} \\ c_{11} \\ c_{22} \\ c_{33} \end{pmatrix} + \begin{pmatrix} a_{00} \\ a_{11} \\ a_{22} \\ a_{33} \end{pmatrix} \times \begin{pmatrix} b_{00} \\ b_{11} \\ b_{22} \\ b_{33} \end{pmatrix}$$

$$+ \begin{pmatrix} a_{01} \\ a_{12} \\ a_{23} \\ 0 \end{pmatrix} \times \begin{pmatrix} b_{10} \\ a_{21} \\ a_{32} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ a_{10} \\ a_{21} \\ a_{32} \end{pmatrix} \times \begin{pmatrix} 0 \\ b_{01} \\ a_{12} \\ a_{23} \end{pmatrix}$$

$$+ \begin{pmatrix} a_{02} \\ a_{13} \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} b_{20} \\ b_{31} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ a_{20} \\ a_{31} \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ b_{02} \\ b_{13} \end{pmatrix}$$

$$+ \begin{pmatrix} a_{03} \\ 0 \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} b_{30} \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ a_{30} \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 0 \\ b_{03} \end{pmatrix}$$

Result

Theorem

In the matrix-matrix multiplication operation $C \leftarrow C + AB$ for $A, B, C \in \mathbb{R}^{n \times n}$ performed by diagonals, the kth diagonal for $k \geq 0$ is given by

$$C_k = \sum_{j=0}^{n-1} \mathrm{A_j} \times \mathrm{B}_{k-j} + \sum_{j=k+1}^{n-1} \mathrm{A}_{k-j} \times \mathrm{B}_j \tag{3}$$

and for $k \leq 0$

$$C_k = \sum_{j=0}^{n-1} \mathrm{A}_{k+j} \times \mathrm{B}_{-j} + \sum_{j=-k+1}^{n-1} \mathrm{A}_{-j} \times \mathrm{B}_{k+j} \tag{4}$$

Observation

The kth superdiagonal of square matrix A is the kth subdiagonal of $A^\top$ and vice-versa.

Banded Matrix-Matrix Multiplication

For A, B banded
Facts:

$$l_C = \min(l_A + l_B, n-1) \text{ and } u_C = \min(u_A + u_B, n-1)$$

where, l_A, l_B, l_C and u_A, u_B, u_C denote, respectively, the lower bandwidth and the upper bandwidth for matrices A, B, and C.

(compute C_k for $0 \leq k \leq \min(u_A + u_B, n-1)$)

$$C_k = \sum_{j=0}^{n-1} \mathrm{A_j} \times \mathrm{B}_{k-j} + \sum_{j=k+1}^{n-1} \mathrm{A}_{k-j} \times \mathrm{B_j}$$

The constraints on the diagonal indices j:
For the first subexpression $j \leq u_A$ and

$$\begin{cases} k-j & \leq & u_B, & k \geq j \\ j-k & \leq & l_B, & k < j \end{cases}$$

For the second subexpression,

$$\begin{cases} j & \leq & u_B, \\ j-k & \leq & l_A, & k < j \end{cases}$$

Matrix-vector Multiplication

For $A \in \mathbb{R}^{m \times n}$ compute,

$$y \leftarrow y + Ax \tag{5}$$

Compute result vector y as the main diagonal (Y_0) of appropriate matrices X and Y:

$$Y \leftarrow Y + Ax\mathbf{1}^\top$$

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_0 & x_0 & x_0 & x_0 \\ x_1 & x_1 & x_1 & x_1 \\ x_2 & x_2 & x_2 & x_2 \\ x_3 & x_3 & x_3 & x_3 \end{pmatrix}$$

Testing Environment (1)

Table: System Information

Operating System:	CentOS
GCC version	4.4.7
Linux version	2.6.32
OpenMP Version	3.0

Table: Test Environment Specifications

Processor Model:	AMD Opteron(tm) Processor 4284
CPU(s)	16
Thread(s) per core	2
Core(s) per socket	4
Socket(s)	2
CPU MHz	1400.00
L1d cache	16K
L1i cache	64K
L2 cache	2048K
L3 cache	6144K

Parallel Banded Matrix-Matrix Multiplication (1)

Data type: Float; no optimization

Table: Speedup for matrix dimension 100000 with 16 threads

Speedup				
$k_b = 101$	$k_b = 201$	$k_b = 801$	$k_b = 1601$	$k_b = 3201$
3.30	4.50	10.00	11.74	11.56

Table: Efficiency for matrix dimension 100000 with 16 threads

Efficiency				
$k_b = 101$	$k_b = 201$	$k_b = 801$	$k_b = 1601$	$k_b = 3201$
0.21	0.28	0.63	0.73	0.72

Parallel Banded Matrix-Matrix Multiplication (2)

Data type: Float; no optimization

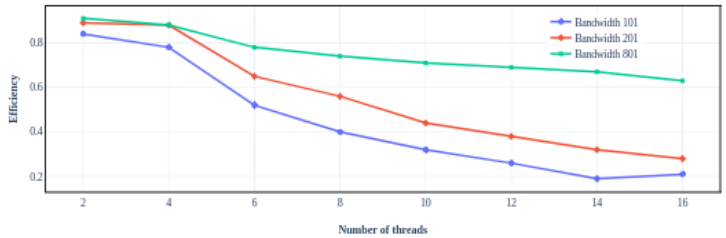

Figure: Efficiency for chunk size 15 with matrix dimension 100000

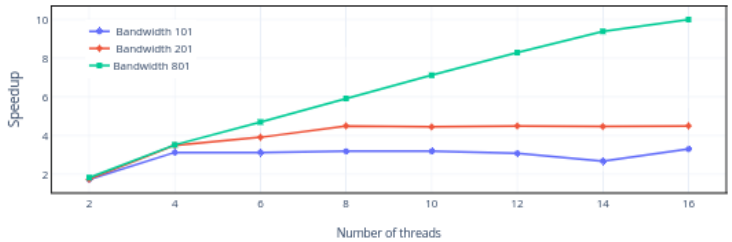

Figure: Speedup for chunk size 15 with matrix dimension 100000

Testing Environment (2)

Table: Test Hardware Specifications

Processor	Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz
CPU Cores(s)	4
L1 d and L1 i cache	32KB
L2 cache	256KB
L3 cache	8192KB
CPU MHz	3400.132

Table: System Information

Operating System:	CentOS Linux release 7.3.1611 (Core)
GCC version	4.4.7

Serial Dense Matrix-Matrix Multiplication (1)

Table 6: **Loop order k-i-j Vs. DIAS using -O0 for float**

Dimension	k-i-j(Seconds)	DIAS (Seconds)	Ratio(DIAS/k-i-j)
1000	9.06	10.09	1.11
2000	73.34	80.46	1.08
2500	143.25	156.87	1.08
3000	249.19	271.87	1.08
3500	392.59	428.93	1.08
4000	589	646.75	1.09
8000	5098.84	4707.69	1.08

Serial Dense Matrix-Matrix Multiplication (2)

Table 7: **Loop order k-i-j Vs. DIAS using -O2 for float**

Dimension	k-i-j(Seconds)	DIAS (Seconds)	Ratio(DIAS/k-i-j)
1000	0.56	1	1.78
2000	5	8.93	1.78
2500	9.81	17	1.73
3000	17	29.31	1.72
3500	27	45.63	1.69
4000	39.63	67.06	1.69
4500	55.88	95.06	1.67
8000	314.562	525.125	1.66

Table 8: **Loop order k-i-j Vs. DIAS using -O3 for Float Type**

Dimension	k-i-j (Seconds)	DIAS (Seconds)	Ratio(DIAS/k-i-j)
1000	0.38	0.56	1.47
2000	3.93	5.18	1.31
2500	8	9.68	1.21
3000	14.18	16.87	1.18
3500	22.75	26.37	1.15
4000	32.94	39.12	1.15
4500	47.75	54.13	1.13

Banded matrix multiplied with dense matrix

Table: Execution Time of Intel MKL (`cblas_sgbmv()`) Vs. DIAS where A is banded and B is dense for matrix size 10000×10000

Bandwidth	MKL (Seconds)	DIAS (Seconds)
100	3.50	9.25
200	9.5	18.5
300	15.75	29.75
400	20	39.5
500	23.5	49.75
600	27.75	60.25
700	32.00	68.75
800	36.25	79.5
900	41	87.5
1000	45.75	97.25

Concluding Remarks

1. *New storage scheme for diagonally structured computation*
   1. *Avoids allocation of extraneous memory*
   2. *Unlike Ellpack-Itpack format, storage for nonzero elements only*
   3. *Elements in specific diagonal are accessed in the order they are stored*
2. *Pointwise multiplication over long vectors (diagonals)*
3. *Diagonally-structured calculation extends to dense and matrices with regular structure*

Future Research

1. *Employ blocking/tiling (cache and register) and loop-unrolling*
2. *Block LU and block Gram-Schmidt rich in level-3 BLAS (structured Matrix-Matrix operation)*
3. *GPU acceleration*

Acknowledgements

Thank you!!

This research was partially supported by the Natural Sciences and Engineering Research Council (NSERC) under Discovery Grants Program.

Bibliography


Niel K. Madsen, Garry H. Rodrigue, and Jack I. Karush. “Matrix multiplication by diagonals on a vector/parallel processor”. Information Processing Letters vol. 5, no. 2 (1976): 41–45.


Anna Tsao, and Thomas Turnbull. “A comparison of algorithms for banded matrix multiplication”. Supercomputing Research Center, 1993.


Zahari Zlatev, Phuong Vu, Jerzy Wasniewski, and Kjeld Schaumburg. “Computations with symmetric, positive definite and band matrices on a parallel vector processor”. Parallel Computing, vol 8, Issues 1–3, 1988, pp. 301 – 312.


Gene H. Golub and Charles F. Van Loan. “Matrix computations (4th ed.)”. Johns Hopkins University Press, Baltimore, MD, USA, 2013.


E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, Jack J. Dongarra, J. Du Croz, S. Hammarling, A. Greenbaum, A. McKenney, and D. Sorensen. 1999. “LAPACK Users’ Guide (Third Ed.)”. SIAM Philadelphia, PA, 1999.


J. J Dongarra and A. J. van der Steen. "High-performance computing systems: Status and outlook". Acta Numerica 21 (2012): 379-474.