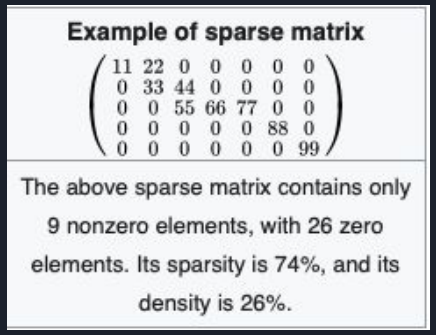
Sparse Matrix-Vector Multiplication

Computación de altas prestaciones y aplicaciones:
Erik Manuel Velásquez Rodríguez

Tabla de contenido:

- Matriz Dispersa (Sparse Matrix)
- Formatos de Matrices
- Sparse Matrix-Vector Multiplication
- Parallel Sparse Matrix-Vector Multiplication
- Performance

Matriz Dispersa (Sparse Matrix)



La matriz dispersa es una matriz (real, compleja) donde la mayoría de los elementos son ceros: $A \in R^{N,N}$ entonces el número de elementos distintos de cero (NNZ) es O (N).

Utilizadas generalmente:

- Método de los elementos finitos (MEF en castellano o finite element method (FEM) en inglés)
- Ecuación en derivadas parciales (a veces abreviada como EDP o partial differential equation (PDE) en inglés)
- Esquemas de discretización numérica: finite volume (FV), finite element (FE), y finite difference (FD) methods.
- Sistemas de ecuaciones no lineales dispersos
- Ecuaciones lineales

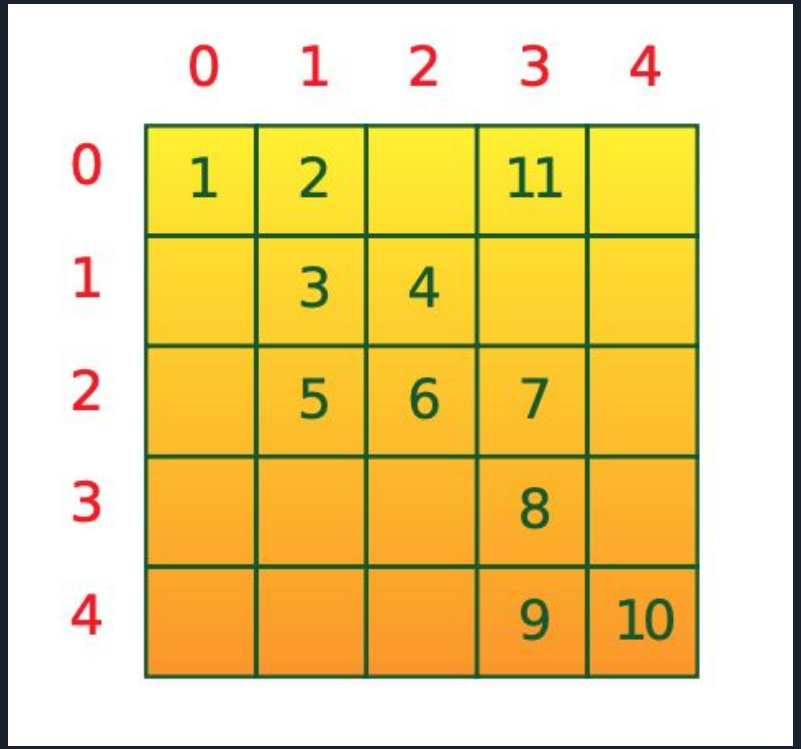
Formato de Matrices:

Los formatos los podemos dividir en dos grupos:

- Aquellos que apoyan la modificación eficiente, como son:
  a. DOK (Diccionario de claves)
  b. LIL (Lista de listas)
  c. COO (Lista coordinada)
- Aquellos que apoyan operaciones eficientes de acceso y matriz, como son:
  a. CSR (Compressed Sparse Row)
  b. CSC (Compressed Sparse Column)

Formato de Matrices: Ejemplo

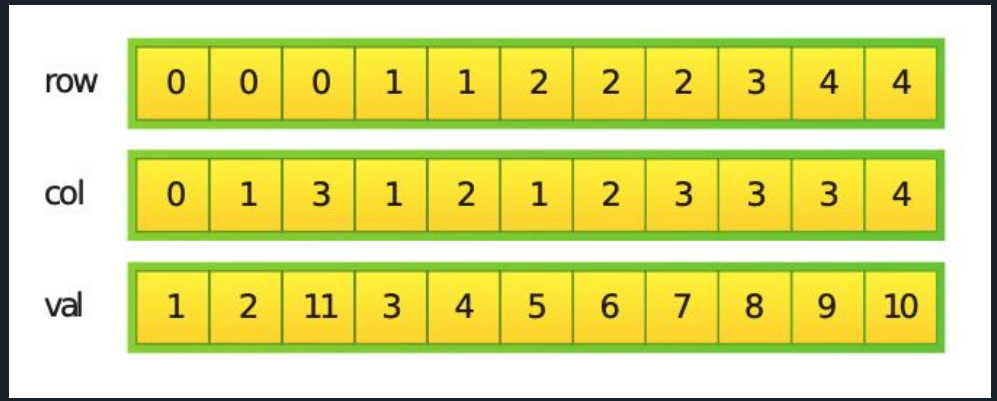
$A \in R^{5,5}$, NNZ=11, sin patrón



Formato de Matrices: COO

Lista Coordinada:

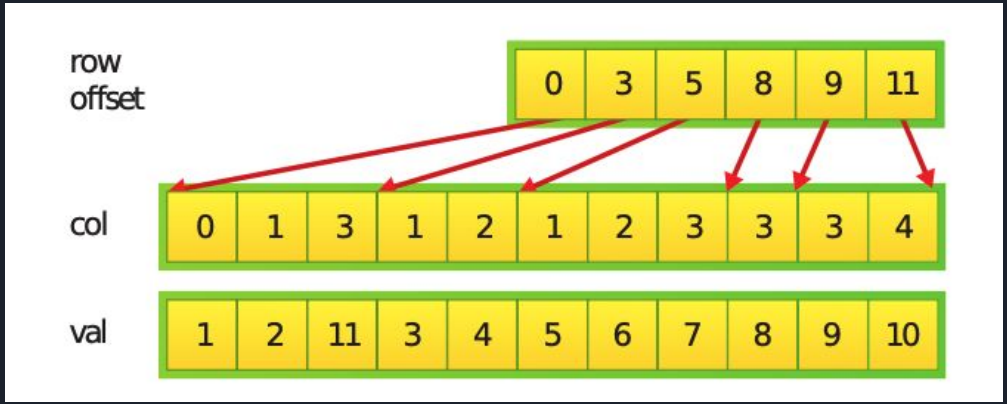
- Índice de Filas - (int) (NNZ)
- Índice de columnas - (int) (NNZ)
- Valores - (data type) (NNZ)



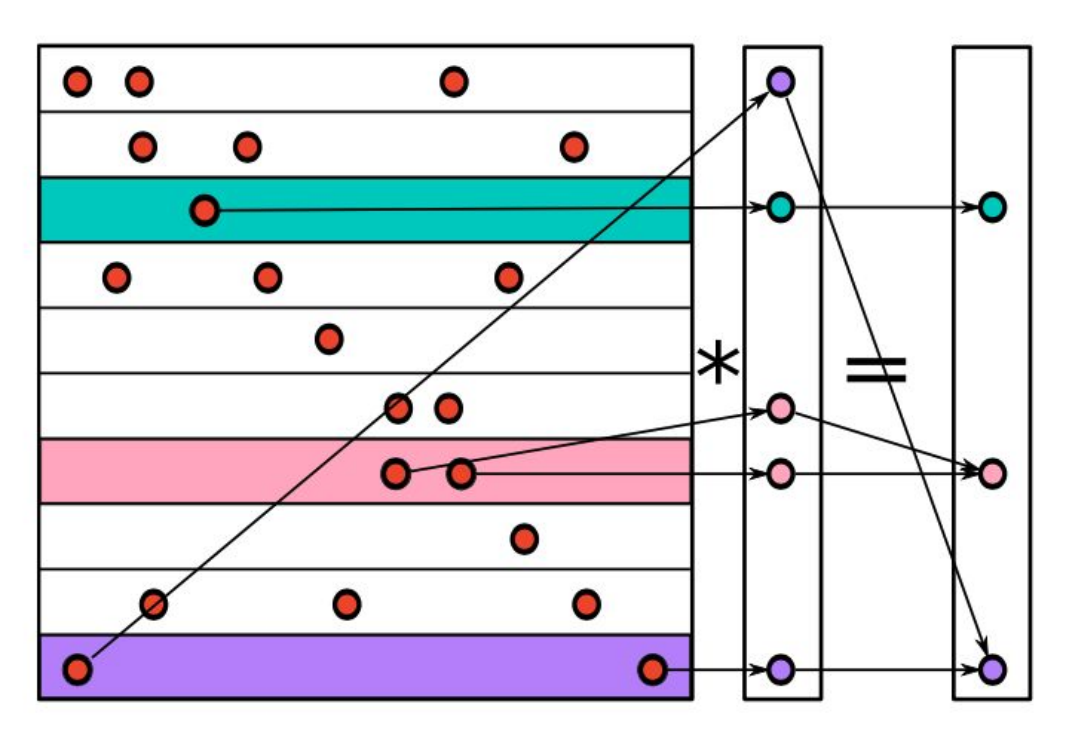
Formato de Matrices: CSR

Compressed Sparse Row:

- Desplazamientos de filas - (int) (N+1)
- Índice de columnas - (int) (NNZ)
- Valores - (data type) (NNZ)

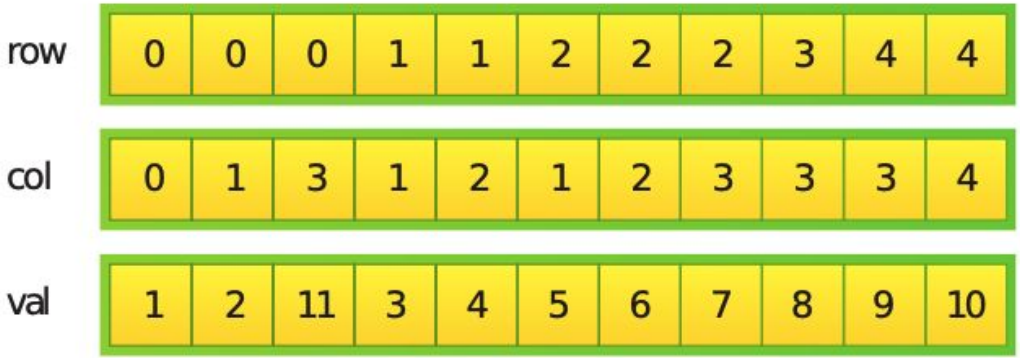


Sparse Matrix-Vector Multiplication q=A.p



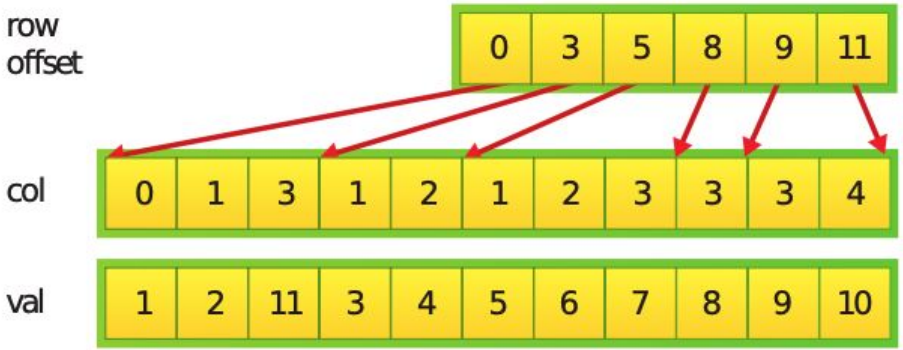
SpMV - COO

```
for (int i=0; i<nnz; ++i)
 y[row[i]] += val[i]*x[col[i]];
```



SpMV - CSR

```
for (int i=0; i<n; ++i) {
 y[i] = 0.0;
 for (int j=row_off[i]; j<row_off[i+1]; ++j)
  y[i] += val[j]*x[col[j]];
}
```



Parallel SpMV

CSR

```
#pragma omp parallel for
for (int i=0; i<n; ++i) {
 y[i] = 0.0;
 for (int j=row_off[i]; j<row_off[i+1]; ++j)
  y[i] += val[j]*x[col[j]];
}
```

COO

????????????

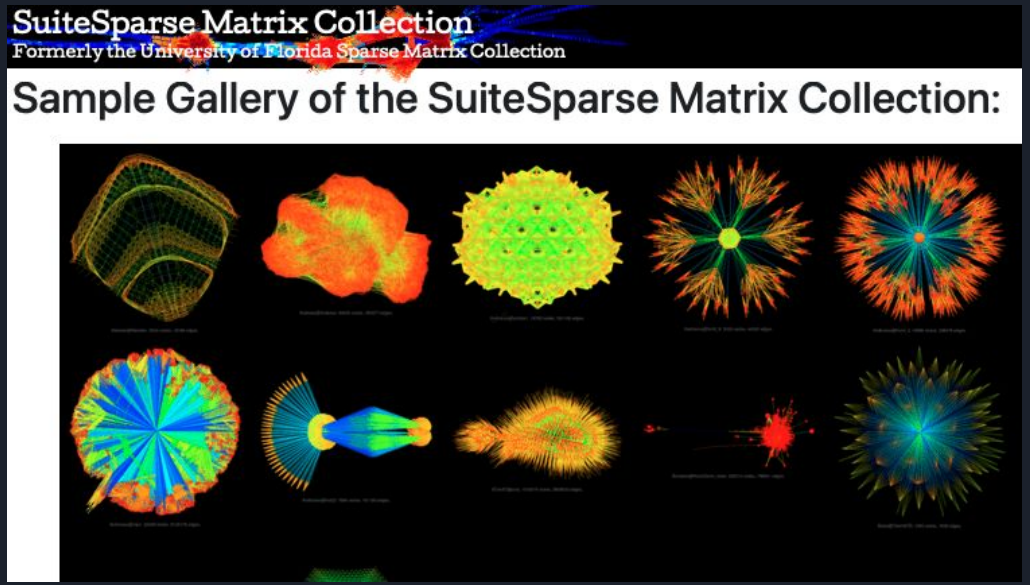
```
for (int i=0; i<nnz; ++i)
 y[row[i]] += val[i]*x[col[i]];
```

row	0	0	0	1	1	2	2	2	3	4	4
col	0	1	3	1	2	1	2	3	3	3	4
val	1	2	11	3	4	5	6	7	8	9	10

Performance

Datos:

The SuiteSparse Matrix Collection (formerly the University of Florida Sparse Matrix Collection) (https://sparse.tamu.edu)



Otras Librerías:

Intel® oneAPI Math Kernel Library (MKL) (https://software.intel.com/content/www/us/en/develop/tools/oneapi/components/onemkl.html)

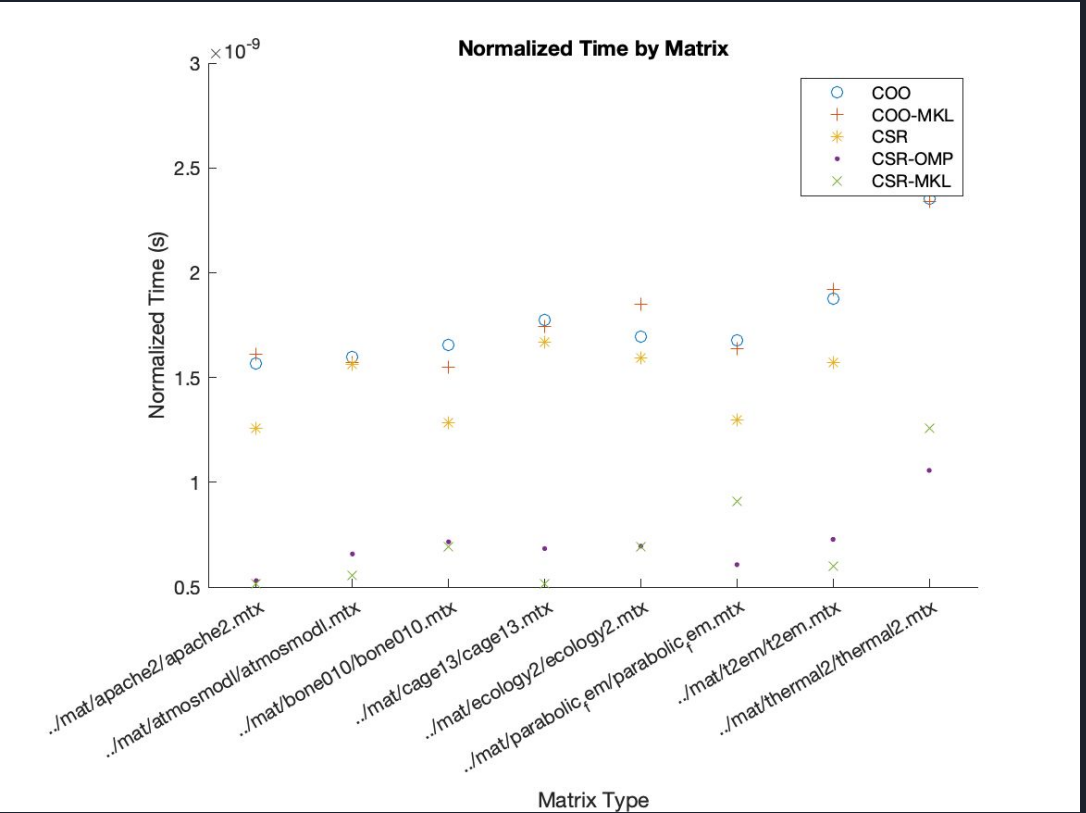
Performance

Intel(R) Core(TM) i5-8257U CPU @ 1.40GHz

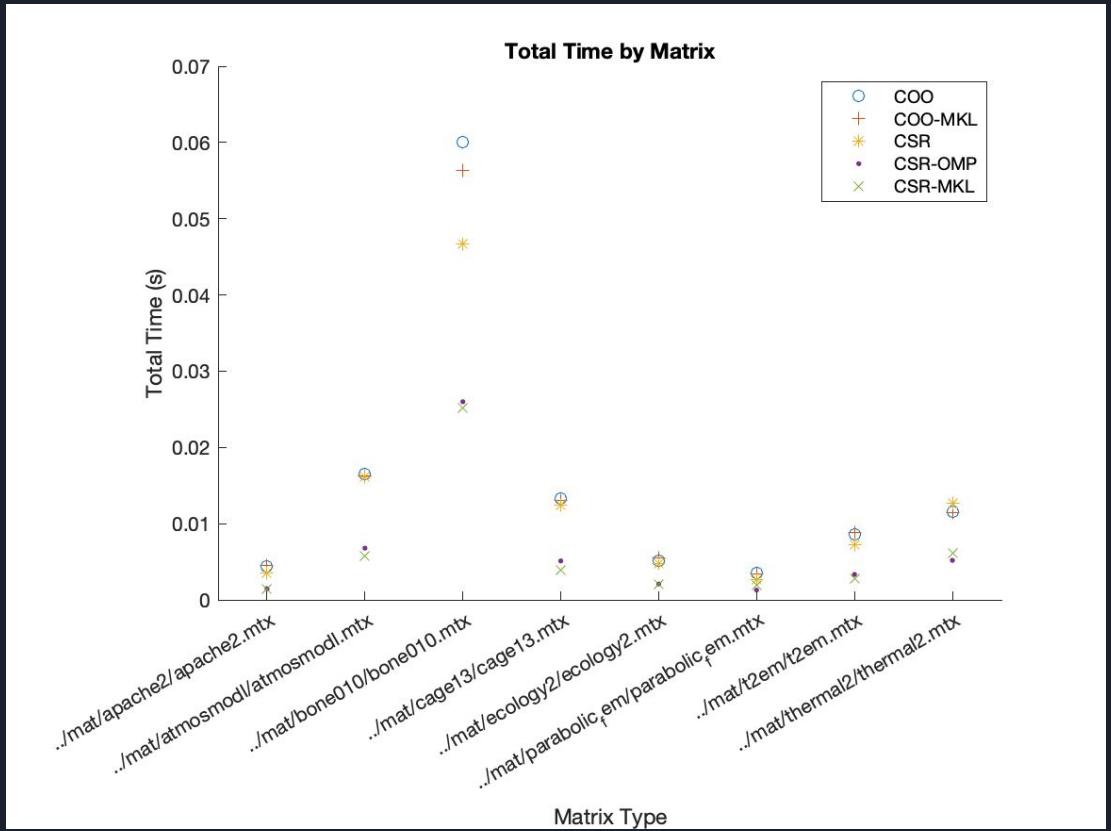
Cores = 4

Threads = 8

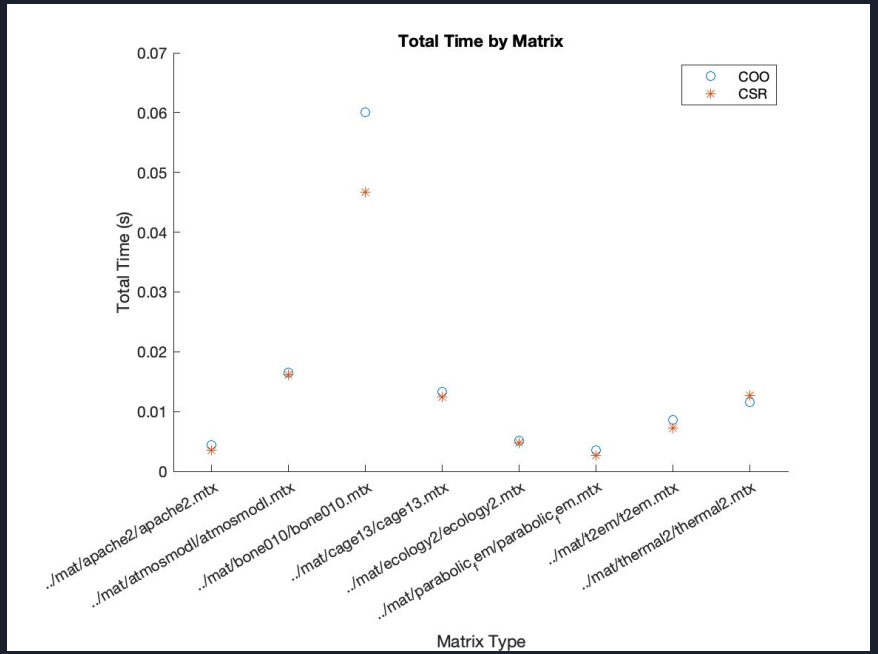
Cache = 6MB

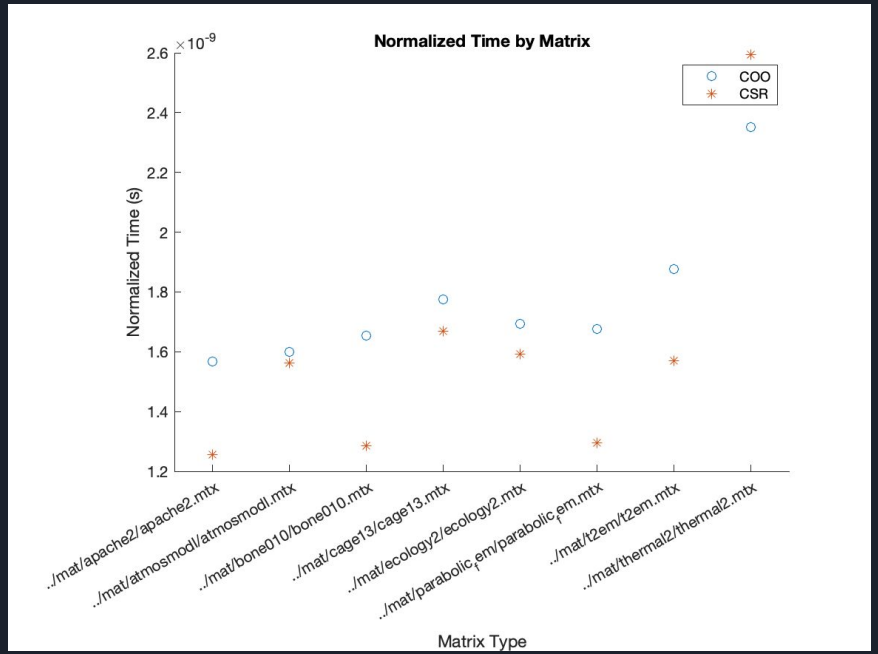


Performance



Performance





Repositorio

https://github.com/velasquezerik/master-capa-spmv

Bibliografía

- https://en.wikipedia.org/wiki/Sparse_matrix#Compressed_sparse_column_(CSC_or_CCS)
- Timothy A. Davis and Yifan Hu. 2011. The University of Florida Sparse Matrix Collection. ACM Transactions on Mathematical Software 38, 1, Article 1 (December 2011), 25 pages. DOI: https://doi.org/10.1145/2049662.2049663
- https://software.intel.com/content/www/us/en/develop/tools/oneapi/components/onemkl.html
- Dimitar Lukarski, Uppsala University, Parallel Scientific Computing, April 11, 2013

Gracias...