

1. Investiga* sobre la subrutina de Fortran dgemv (parámetros que recibe y la salida).

Esta rutina se utiliza para realizar operaciones matriciales, como multiplicación escalar - matriz, matriz - vector.

Recibe los siguientes parámetros (mostrando también el tipo de dato en cada caso: character TRANS, integer M, integer N, double precision ALPHA, double precision, dimension(lda,*) A, integer LDA, double precision, dimension(*) X, integer INCX, double precision BETA, double precision, integer INCY).

- TRANS especifica una de las operaciones a realizar; existen 3 diferentes opciones y consisten en multiplicaciones de vectores, matrices y escalar.
- M: especifica el número de renglones de la matriz A.
- N: especifica el número de columnas de la matriz A.
- Alpha: especifica el valor del escalar.
- A: es un arreglo de doble precisión.
- LDA: es un parámetro que especifica la primera dimensión de A.
- X: es un arreglo de doble precisión.
- INCX: es un entero que especifica el incremento en los valor de X.
- Beta: indica el valor del escalar beta.
- Y: es un arreglo de doble precisión.
- INCY: indica el incremento en los valores de Y.

Como salida nos entrega el resultado de una de las operaciones matriciales que seleccionamos en TRANS:

```
TRANS = 'N' or 'n'    y := alpha*A*x + beta*y.
```

```
TRANS = 'T' or 't'    y := alpha*A**T*x + beta*y.
```

```
TRANS = 'C' or 'c'    y := alpha*A**T*x + beta*y.
```

2. En la carpeta analisis-numerico-computo-cientifico/C/BLAS/ejemplos/level2/ ejecuta el programa dgemv_mult_mat_vec.c y realiza pruebas con diferentes matrices y vectores definidos por ti.

Hice un par de pruebas adicionales aumentando el tamaño de la matriz A:

```

root@122751718bc8:/datos# ./programa.out 7 2
matriz 1:
matriz[0][0]=0.00000    matriz[0][1]=1.50000
matriz[1][0]=4.00000    matriz[1][1]=-5.00000
matriz[2][0]=-1.00000   matriz[2][1]=2.50000
matriz[3][0]=5.00000    matriz[3][1]=5.60000
matriz[4][0]=7.80000    matriz[4][1]=9.60000
matriz[5][0]=45.50000   matriz[5][1]=65.00000
matriz[6][0]=4.30000    matriz[6][1]=5.00000
-----
vector :
vector[0]=1.00000
vector[1]=0.00000
-----
vector resultado:
vector[0]=0.00000
vector[1]=4.00000
vector[2]=-1.00000
vector[3]=5.00000
vector[4]=7.80000
vector[5]=45.50000
vector[6]=4.30000

```

```

root@122751718bc8:/datos# ./programa.out 5 2
matriz 1:
matriz[0][0]=0.00000    matriz[0][1]=1.50000
matriz[1][0]=4.00000    matriz[1][1]=-5.00000
matriz[2][0]=-1.00000   matriz[2][1]=2.50000
matriz[3][0]=5.00000    matriz[3][1]=5.60000
matriz[4][0]=7.80000    matriz[4][1]=9.60000
-----
vector :
vector[0]=1.00000
vector[1]=0.00000
-----
vector resultado:
vector[0]=0.00000
vector[1]=4.00000
vector[2]=-1.00000
vector[3]=5.00000
vector[4]=7.80000

```

3. Después de haber estudiado y entendido los archivos de definiciones.h y funciones.c y realizado los puntos anteriores y la tarea 5 responde: ¿cómo fue que pudimos llamar a las rutinas de Fortran (que almacena en una forma column-major order los arreglos de dos dimensiones) para operaciones con arreglos 2-dimensionales sin haber instalado CBLAS, si en clase se dijo que almacenar arreglos de dos dimensiones en C es en un row-major order?
4. Investiga* sobre la subrutina de Fortran dgemm (parámetros que recibe y la salida).

Es una rutina que realiza una operación matriz - matriz.

Recibe los siguientes parámetros:

- Alpha: es un escalar.
- Beta: es un escalar.
- A es una matriz.
- B es una matriz.
- m y k son las dimensiones de la matriz A.
- k y n son las dimensiones de la matriz B.
- m y n son las dimensiones de la matriz C.

5. En la carpeta analisis-numerico-computo-cientifico/C/BLAS/ejemplos/level3/ ejecuta el programa dgemm_mult_mat.c y realiza pruebas con diferentes matrices definidas por ti.

Hice un par de pruebas adicionales.

```
root@122751718bc8:/datos# ./programa.out 5 2 2 5
matriz 1:
matriz[0][0]=0.00000    matriz[0][1]=1.50000
matriz[1][0]=4.00000    matriz[1][1]=-5.00000
matriz[2][0]=-1.00000   matriz[2][1]=2.50000
matriz[3][0]=-12.00000  matriz[3][1]=5.00000
matriz[4][0]=4.50000    matriz[4][1]=6.70000
-----
matriz 2:
matriz[0][0]=1.00000    matriz[0][1]=0.00000    matriz[0][2]=0.00000    matriz[0][3]=3.00000    matriz[0][4]=-1.00000
matriz[1][0]=0.00000    matriz[1][1]=-1.00000   matriz[1][2]=1.00000    matriz[1][3]=2.00000    matriz[1][4]=-2.00000
-----
matriz resultado:
matriz[0][0]=0.00000    matriz[0][1]=-1.50000   matriz[0][2]=1.50000   matriz[0][3]=3.00000   matriz[0][4]=-3.00000
matriz[1][0]=4.00000    matriz[1][1]=5.00000   matriz[1][2]=-5.00000  matriz[1][3]=2.00000   matriz[1][4]=6.00000
matriz[2][0]=-1.00000   matriz[2][1]=-2.50000  matriz[2][2]=2.50000   matriz[2][3]=2.00000   matriz[2][4]=-4.00000
matriz[3][0]=-12.00000  matriz[3][1]=-5.00000  matriz[3][2]=5.00000   matriz[3][3]=-26.00000 matriz[3][4]=2.00000
matriz[4][0]=4.50000    matriz[4][1]=-6.70000  matriz[4][2]=6.70000   matriz[4][3]=26.90000  matriz[4][4]=-17.90000
```

```
root@122751718bc8:/datos# ./programa.out 3 2 2 3
matriz 1:
matriz[0][0]=0.00000    matriz[0][1]=1.50000
matriz[1][0]=4.00000    matriz[1][1]=-5.00000
matriz[2][0]=-1.00000   matriz[2][1]=2.50000
-----
matriz 2:
matriz[0][0]=1.00000    matriz[0][1]=0.00000    matriz[0][2]=0.00000
matriz[1][0]=0.00000    matriz[1][1]=-1.00000   matriz[1][2]=1.00000
-----
matriz resultado:
matriz[0][0]=0.00000    matriz[0][1]=-1.50000   matriz[0][2]=1.50000
matriz[1][0]=4.00000    matriz[1][1]=5.00000   matriz[1][2]=-5.00000
matriz[2][0]=-1.00000   matriz[2][1]=-2.50000  matriz[2][2]=2.50000
```

6. En la carpeta del punto anterior encuentras la sección Multiplicación matriz-matriz con trick. Ejecuta el programa de esta sección con diferentes matrices definidas por ti y resuelve la pregunta ¿por qué funciona este trick?.

Hice un par de pruebas adicionales aumentando el tamaño de la matriz A:

```

root@122751718bc8:/datos# ./programa.out 8 2 2 3
matriz 1:
matriz[0][0]=0.00000    matriz[0][1]=1.50000
matriz[1][0]=4.00000    matriz[1][1]=-5.00000
matriz[2][0]=-1.00000   matriz[2][1]=2.50000
matriz[3][0]=4.00000    matriz[3][1]=56.00000
matriz[4][0]=54.00000   matriz[4][1]=5.00000
matriz[5][0]=2.00000    matriz[5][1]=1.00000
matriz[6][0]=2.00000    matriz[6][1]=3.00000
matriz[7][0]=5.00000    matriz[7][1]=6.00000
-----
matriz 2:
matriz[0][0]=1.00000    matriz[0][1]=0.00000    matriz[0][2]=0.00000
matriz[1][0]=0.00000    matriz[1][1]=-1.00000   matriz[1][2]=1.00000
matriz 3:
matriz[0][0]=0.00000    matriz[0][1]=-1.50000   matriz[0][2]=1.50000
matriz[1][0]=4.00000    matriz[1][1]=5.00000    matriz[1][2]=-5.00000
matriz[2][0]=-1.00000   matriz[2][1]=-2.50000   matriz[2][2]=2.50000
matriz[3][0]=4.00000    matriz[3][1]=-56.00000  matriz[3][2]=56.00000
matriz[4][0]=54.00000   matriz[4][1]=-5.00000   matriz[4][2]=5.00000
matriz[5][0]=2.00000    matriz[5][1]=-1.00000   matriz[5][2]=1.00000
matriz[6][0]=2.00000    matriz[6][1]=-3.00000   matriz[6][2]=3.00000
matriz[7][0]=5.00000    matriz[7][1]=-6.00000   matriz[7][2]=6.00000

root@122751718bc8:/datos# ./programa.out 3 2 2 3
matriz 1:
matriz[0][0]=0.00000    matriz[0][1]=1.50000
matriz[1][0]=4.00000    matriz[1][1]=-5.00000
matriz[2][0]=-1.00000   matriz[2][1]=2.50000
-----
matriz 2:
matriz[0][0]=1.00000    matriz[0][1]=0.00000    matriz[0][2]=0.00000
matriz[1][0]=0.00000    matriz[1][1]=-1.00000   matriz[1][2]=1.00000
matriz 3:
matriz[0][0]=0.00000    matriz[0][1]=-1.50000   matriz[0][2]=1.50000
matriz[1][0]=4.00000    matriz[1][1]=5.00000    matriz[1][2]=-5.00000
matriz[2][0]=-1.00000   matriz[2][1]=-2.50000   matriz[2][2]=2.50000

```

7. Haz un programa que utilice la subrutina dsymm de Fortran.