



Proyecto Programación Multihilo

Implementar el método iterativo de Jacobi utilizando programación multihilo.

El algoritmo en alto nivel que corresponde al método iterativo de Jacobi se recomienda que sea consultado en Burden, Faires Análisis Numérico Thomson ó en William H. Press Numerical Recipes in C: The Art of Scientific Computing Cambridge University Press. Los datos de entrada deben ser leídos desde un archivo de entrada(*.in). Los resultados deben ser escritos a un archivo de salida(*.out). El formato del archivo de entrada es el siguiente:

```
N
T
I
x1...xi...xN
a1,1 ...a1,i ...a1,N
...
aN,1...aN,i ...aN,N
b1... bi ...bN
```

N: dimensión de la matriz.

T: tolerancia.

I: número máximo de iteraciones.

x1...xi... xN: Aproximación Inicial

a1,1 ...a1,i ...a1,N: Elementos de la Matriz cuadrada A.

b1... bi ...bN: Elementos del vector b.

Si se obtuvo una solución, el formato del Archivo de Salida es:

La solución para una tolerancia de (=T) se obtuvo exitosamente en (=k) iteraciones. Los valores obtenidos son:

```
1 x1...xi...xN
...
k x1...xi...xN
```

Si se excedió el número máximo de iteraciones, el formato del Archivo de Salida es el siguiente:

Numero Maximo de Iteraciones excedido. Los valores obtenidos son:

1 x1...xi...xN

....

k x1...xi...xN

Para todos los casos mostrados el subíndice i es un número natural mayor o igual que 1 y menor o igual que N . El subíndice k utilizado en los archivos de salida, indica el numero de iteración para la cual se obtuvieron esos valores, y es un numero natural mayor o igual que 1 y que a lo sumo podrá ser igual que el número máximo de iteraciones.

Ejemplo de Archivo de Entrada

```
4
0.00001
50
0 0 0 0
10 -1 2 0
-1 11 -1 3
2 -1 10 -1
0 3 -1 8
6 25 -11 15
```

Ejemplo de Archivo de Salida

La solución para una tolerancia de 0.00001 se obtuvo exitosamente en 15 iteraciones. Los valores obtenidos son:

1	0.600000	2.272727	-1.100000	1.875000
2	1.047273	1.715909	-0.805227	0.885227
3	0.932636	2.053306	-1.049341	1.130881
4	1.015199	1.953696	-0.968109	0.973843
5	0.988991	2.011415	-1.010286	1.021351
6	1.003199	1.992241	-0.994522	0.994434
7	0.998128	2.002307	-1.001972	1.003594
8	1.000625	1.998670	-0.999036	0.998888
9	0.999674	2.000448	-1.000369	1.000619
10	1.000119	1.999768	-0.999828	0.999786
11	0.999942	2.000085	-1.000068	1.000109
12	1.000022	1.999959	-0.999969	0.999960
13	0.999990	2.000016	-1.000013	1.000019
14	1.000004	1.999993	-0.999994	0.999992
15	0.999998	2.000003	-1.000002	1.000003

Condiciones

1. El proyecto será desarrollado por grupos integrados por tres estudiantes.
2. La fecha tope de entrega del proyecto es el [23 de marzo de 2018](#) con los siguientes aspectos a saber:
 - Presentación de un informe técnico en el que se explique el problema y su abordaje en cuanto a: uso de las primitivas, identificación de los recursos críticos, hilos y estructuras de datos. Se deben incluir en la sección de **Análisis de los Resultados**, tablas y gráficas que muestren la relación número de hilos vs tiempo de procesamiento con respecto a cada caso de prueba. Entre los casos de prueba se deben incluir matrices de dimensión 10x10, 100x100,1.000x1.000,10.000x10.000. Con respecto al número de hilos: 1,2,4 y 8.

- Se deben elaborar casos de prueba que permitan evaluar las funcionalidades del programa y los casos extremos. Se deben incluir en el directorio del proyecto los casos de prueba utilizados.
- El código deberá ser desarrollado en Lenguaje ANSI C con la utilización de la biblioteca pthreads, intradocumentado y que sea capaz de reflejar con banderas los escenarios de ejecución y los distintos casos de prueba.
- Sólo será evaluado el proyecto si funciona correctamente y si resuelve el problema planteado en forma correcta y completa.
- La detección de copias llevará a la anulación inmediata de la entrega.

Éxito!