Universidad Del Valle de Guatemala
Computación Paralela
Miguel Novella

Proyecto # 1

18/03/2022
Augusto Alonso #181085
Esteban Cabrera #17781
Mario Sarmientos #17055

Índice

Antecedentes:

Se presenta a continuación una breve revisión de datos de trabajos acerca del mismo tema o que tienen un enfoque parecido al expuesto en este proyecto:

1. A continuación se presenta un trabajo relacionado con la disipación de calor utilizando la técnica del punto material (MPM), su metodología fue trasladar el algoritmo a un lenguaje de programación, y una vez ahí jugar con los diferentes parámetros para observar y graficar el comportamiento.

   (John A. Nairn, n.d.) Modeling heat flow across material interfaces and cracks using the material point method.

2. Así mismo, hay precedente de trabajos anteriores que han utilizado tecnologías similares de la mano de CUDA para simular la disipación de calor en distintos materiales.
   (Vinaya Sivanandan, 2015) *Designing a parallel algorithm for Heat conduction using MPI, OpenMP and CUDA*.

3. El el trabajo “Parallel algorithm for solving the problems of heat and mass transfer in the open geothermal system” se explora la posibilidad de mejorar el algoritmo de disipación de calor en tres dimensiones, al igual que en el presente proyecto se propone la utilización de de openMP para la paralelización del algoritmo.

   (Vladimir E. Misilov, 2020) *Parallel algorithm for solving the problems of heat and mass transfer in the open geothermal system*

4. Se de han visto proyectos de alcance similar como “Heat transfer simulation of heat storage unit with nanoparticles and fins through a heat exchanger” En el que utilizan un algoritmo de dispersión de calor en superficies planas para crear un algoritmo mayor de transferencia de calor multicapa.

   (M.Sheikholeslami, 2019) *Heat transfer simulation of heat storage unit with nanoparticles and fins through a heat exchanger*.

5. Para problemas del manejo térmico en electrónicos se propuso un trabajo similar al presente en donde se aplica la computación paralela para resolver

ecuaciones multidimensionales y diferenciales en donde se pretende generar una mejor transferencia de calor.

(S. Sarkar, 2009) Direct Numerical Simulation of Heat Transfer in Spray Cooling Through 3D Multiphase Flow Modeling Using Parallel Computing

6. Un trabajo en donde se estudia la transferencia de calor y de humedad en suelo no saturado en donde se implementa la computación paralela para realizar un modelo que pueda simular en condiciones específicas.

   (T, Hywel, 1995). Modelling transient heat and moisture transfer in unsaturated soil using a parallel computing approach

7. El siguiente proyecto se desarrolló para ubicar zonas de calor a nivel geográfico dentro de áreas urbanas y se implementó la computación paralela para desempeñar su modelo propio.

   (L. Xiaojiang, 2021). GPU parallel computing for mapping urban outdoor heat exposure.

8.

Objetivos

- Implementa y diseña programas para la paralelización de procesos con memoria compartida usando OpenMP
- Aplicar el método PCAM y los conceptos de patrones de partición y programa para modificar un programa secuencial y volverlo paralelo

Anexo 1

Catálogo de funciones

double maxFromArray(double data[], int start, int end)
Esta función se deja como genérica para poder encontrar dentro de un array de doubles el máximo entre los índices start y end.
Define un max inicial de 0 para encontrar el verdadero máximo
Esta función principalmente se ejecuta en cada task para encontrar nuestro máx de cada thread y al finalizar el máximo del conjunto de n threads

Entradas
double data[] valor máximo de las diferencias de temperaturas.
int start es el inicio del tiempo de ejecución
int end es el final del tiempo de ejecución

Salida
Devuelve el máximo valor de las diferencias de temperaturas.

Comentarios
Se necesita pasar los valores del inicio y el final para que cada thread se encargue del bloque que le corresponde al correrlo de manera paralela

void resolveInitialValues()

Resuelve los valores iniciales de los deltas basados en la ecuación diferencial unidimensional de disipación de calor.

void sendTask(int start, int end, int num_threads, double temperatures2[], double temperatures1[], double diff[])

Esta función es la encargada de encontrar cada nueva temperatura entre temperatures2 y temperatures1 entre los índices start y end que se les pasa luego de que cada thread calcule su bloque. Luego de dicho cálculo se asigna la diferencia en diff para luego calcular su máximo de ese bloque.

Entradas
int startes el inicio del tiempo de ejecución
int end es el final del tiempo de ejecución
int num_threads cantidad de threads que se ejecutan
double temperatures2[] set de temperaturas actuales

double temperatures1[] set de temperaturas pasadas
double diff[] set de diferencias temperaturas

Se encarga de primero calcular las siguientes temperaturas y luego calcular sus diferencias para almacenar en el array diff. Al igual que maxFromArray se pasa el inicio y el final de la iteración para que cada thread se encargue del bloque que le corresponde

Catálogo de variables

Constantes

ERR 0.00001
Precisión o diferencia requerida

DIFUSIVE_VAL 10e-4
Es el número de intervalos discretos.

C_CONSTANT 10e-4
El valor de la constante del metal seleccionado

L
La longitud de la barra

N
E número de espacios en el que se dividirá la barral

Runtime

double DT;
Delta del tiempo calculada siempre conformado a nuestra constante C

double DX;
Delta X = L/N

double temperatures1[N];
Temperaturas iniciales.

double temperatures2[N];

Array donde se calcularon las temperaturas

double differences[N];
Array que contiene todas las diferencias entre nuestras temperaturas

double differences_pt[num_threads];
Aca se guarda la maxima diferencia de cada thread para ser comparado luego

double start_time
Variable donde se guarda timestamp de el inicio de la ejecución paralela

double elapsed_time_ms;
Tiempo transcurrido de la ejecución del programa.

double t0
La temperatura inicial en toda la barra excepto los extremos

double tl
La temperatura de la barra en la izquierda

double tr
La temperatura de la barra en la derecha

int num_threads
El número de threads que se estará corriendo

int m
variable para iterar los threads - 1 y mandar cada task con su respectivo bloque

int iterations
Número de iteraciones totales de los threads N-2 porque se excluyen los extremos

int block
El tamaño de cada bloque de cada thread

int end_l
Variable que contiene cuando más tiene que recorrer el ultimo thread asi no se gasta poder en cada corrida calculandolo

Task

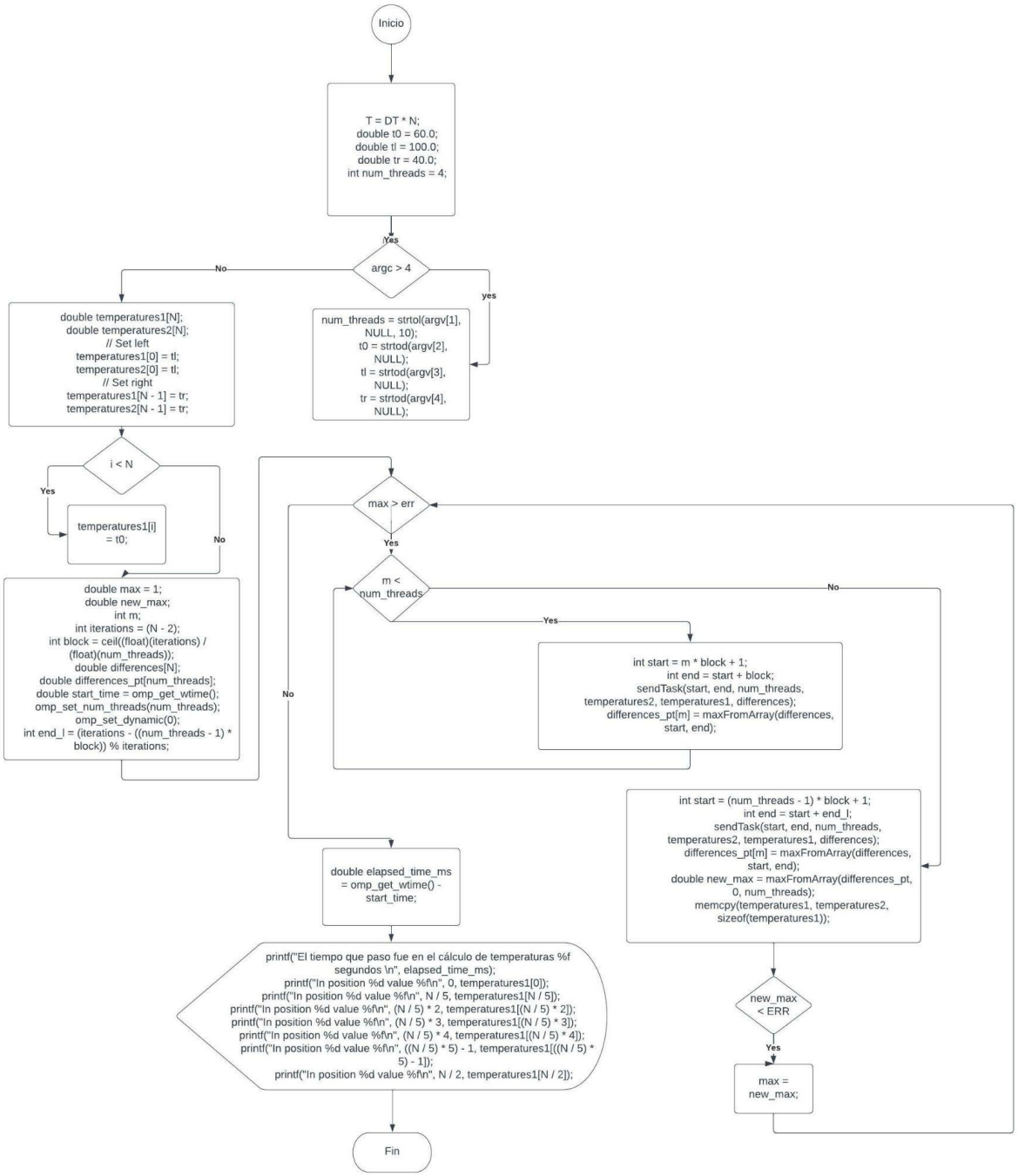
A nivel de los tasks tenemos ciertas variables que vamos a explicar

int start
Donde inicia a recorrer el array nuestro thread

int end
Donde finaliza

Anexo 2

Diagrama de flujo



Paralelización

Resultados

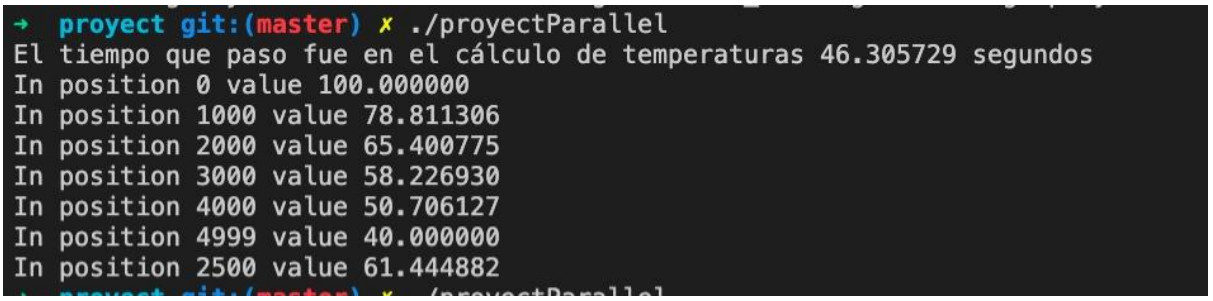


Imagen 1. Ejecución 4 threads, N=5000 en paralelo.

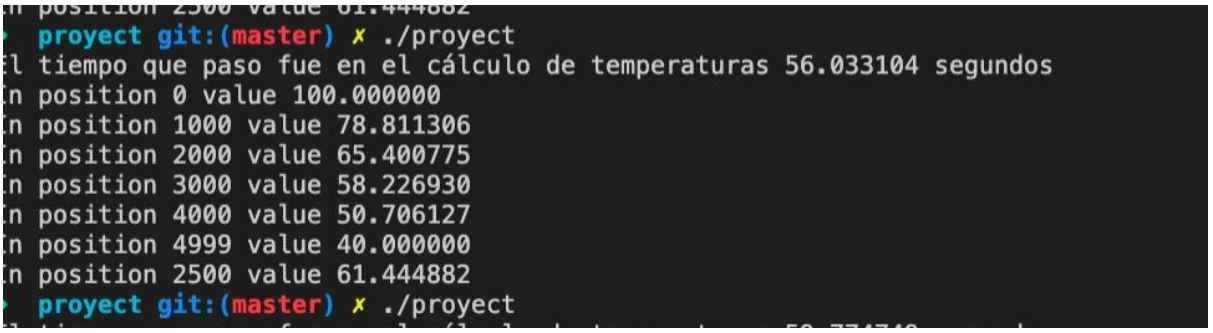


Imagen 2. Ejecución en secuencial

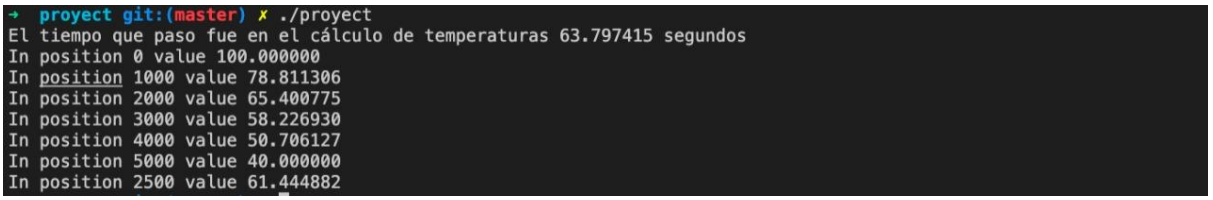


Imagen 3. Ejecución en paralelo con el tiempo más tardado obtenido.

Paralelo

Ejecuciones con :

T0 = 58 , Tl = 40 Tr = 60

Medición 1

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./hello 8 58 40 60
El tiempo que paso fue en el cálculo de temperaturas 11.762450 segundos
In position 0 value 40.000000
In position 1000 value 52.888535
In position 2000 value 57.424412
In position 3000 value 58.040883
In position 4000 value 58.568581
In position 4999 value 60.000000
In position 2500 value 57.881758
```

Medición 2

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./hello 8 58 40 60
El tiempo que paso fue en el cálculo de temperaturas 11.532036 segundos
In position 0 value 40.000000
In position 1000 value 52.888535
In position 2000 value 57.424412
In position 3000 value 58.040883
In position 4000 value 58.568581
In position 4999 value 60.000000
In position 2500 value 57.881758
```

Medición 3

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./hello 8 58 40 60
El tiempo que paso fue en el cálculo de temperaturas 11.695699 segundos
In position 0 value 40.000000
In position 1000 value 52.888535
In position 2000 value 57.424412
In position 3000 value 58.040883
In position 4000 value 58.568581
In position 4999 value 60.000000
In position 2500 value 57.881758
```

T0 = 58 , TI = 40 Tr = 20

medicion 1

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./hello 8 58 40 20
El tiempo que paso fue en el cálculo de temperaturas 24.291838 segundos
In position 0 value 40.000000
In position 1000 value 49.583530
In position 2000 value 54.449330
In position 3000 value 52.177548
In position 4000 value 40.412803
In position 4999 value 20.000000
In position 2500 value 54.341875
```

medicion 2

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./hello 8 58 40 20
El tiempo que paso fue en el cálculo de temperaturas 24.644169 segundos
In position 0 value 40.000000
In position 1000 value 49.583530
In position 2000 value 54.449330
In position 3000 value 52.177548
In position 4000 value 40.412803
In position 4999 value 20.000000
In position 2500 value 54.341875
```

medicion 3

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./hello 8 58 40 20
El tiempo que paso fue en el cálculo de temperaturas 25.450515 segundos
In position 0 value 40.000000
In position 1000 value 49.583530
In position 2000 value 54.449330
In position 3000 value 52.177548
In position 4000 value 40.412803
In position 4999 value 20.000000
In position 2500 value 54.341875
```

T0 = 58 , TI = 15 Tr = 25

medicion 1

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./hello 8 58 15 25
El tiempo que paso fue en el cálculo de temperaturas 29.020286 segundos
In position 0 value 15.000000
In position 1000 value 36.825851
In position 2000 value 49.637128
In position 3000 value 50.912827
In position 4000 value 41.638381
In position 4999 value 25.000000
In position 2500 value 51.681259
```

medicion 2

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./hello 8 58 15 25
El tiempo que paso fue en el cálculo de temperaturas 28.838976 segundos
In position 0 value 15.000000
In position 1000 value 36.825851
In position 2000 value 49.637128
In position 3000 value 50.912827
In position 4000 value 41.638381
In position 4999 value 25.000000
In position 2500 value 51.681259
```

medicion 3

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./hello 8 58 15 25
El tiempo que paso fue en el cálculo de temperaturas 27.599972 segundos
In position 0 value 15.000000
In position 1000 value 36.825851
In position 2000 value 49.637128
In position 3000 value 50.912827
In position 4000 value 41.638381
In position 4999 value 25.000000
In position 2500 value 51.681259
```

T0 = 58 , TI = 15 Tr = 25

medicion 1

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./hello 8 58 45 50
El tiempo que paso fue en el cálculo de temperaturas 8.317738 segundos
In position 0 value 45.000000
In position 1000 value 55.303845
In position 2000 value 57.846845
In position 3000 value 57.904165
In position 4000 value 56.337189
In position 4999 value 50.000000
In position 2500 value 57.965879
```

medicion 2

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./hello 8 58 45 50
El tiempo que paso fue en el cálculo de temperaturas 8.280759 segundos
In position 0 value 45.000000
In position 1000 value 55.303845
In position 2000 value 57.846845
In position 3000 value 57.904165
In position 4000 value 56.337189
In position 4999 value 50.000000
In position 2500 value 57.965879
```

medicion 3

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./hello 8 58 45 50
El tiempo que paso fue en el cálculo de temperaturas 8.223544 segundos
In position 0 value 45.000000
In position 1000 value 55.303845
In position 2000 value 57.846845
In position 3000 value 57.904165
In position 4000 value 56.337189
In position 4999 value 50.000000
In position 2500 value 57.965879
```

Secuenciales:

T0 = 58 , TI = 40 Tr = 60

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./secu 58 40 60
El tiempo que paso fue en el cálculo de temperaturas 21.253257 segundos
In position 0 value 40.000000
In position 1000 value 52.888535
In position 2000 value 57.424412
In position 3000 value 58.040883
In position 4000 value 58.568581
In position 4999 value 60.000000
In position 2500 value 57.881758
```

medicion 2

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./secu 58 40 60
El tiempo que paso fue en el cálculo de temperaturas 21.360747 segundos
In position 0 value 40.000000
In position 1000 value 52.888535
In position 2000 value 57.424412
In position 3000 value 58.040883
In position 4000 value 58.568581
In position 4999 value 60.000000
In position 2500 value 57.881758
```

medicion 3

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./secu 58 40 60
El tiempo que paso fue en el cálculo de temperaturas 21.404373 segundos
In position 0 value 40.000000
In position 1000 value 52.888535
In position 2000 value 57.424412
In position 3000 value 58.040883
In position 4000 value 58.568581
In position 4999 value 60.000000
In position 2500 value 57.881758
```

T0 = 58 , TI = 40 Tr = 20

medicion 1

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./secu 58 40 20
El tiempo que paso fue en el cálculo de temperaturas 45.668318 segundos
In position 0 value 40.000000
In position 1000 value 49.583530
In position 2000 value 54.449330
In position 3000 value 52.177548
In position 4000 value 40.412803
In position 4999 value 20.000000
In position 2500 value 54.341875
```

medicion 2

```
mario@DESKTOP-BL3KA26:/mnt/c/wsl/p1$ ./secu 58 40 20
El tiempo que paso fue en el cálculo de temperaturas 45.498171 segundos
In position 0 value 40.000000
In position 1000 value 49.583530
In position 2000 value 54.449330
In position 3000 value 52.177548
In position 4000 value 40.412803
In position 4999 value 20.000000
In position 2500 value 54.341875
```

Discusión

Como podemos observar desde las primeras tandas de corridas en las secuenciales tuvimos 21 segundos versus 11 segundos en promedio de las paralelas. Se puede observar en este caso casi un 50% de speed up. En diferentes también se probó con menor cantidad de threads que bajó un poquito pero realmente no varió tanto pudo ser utilizado de la misma manera. En la compu ejecutada 8 threads una por cada core fue la mejor opción

Conclusiones

- las mediciónes paralelas si tuvieron mayor speed up
- El speedup fue de 1.931, confirmando que el cálculo paralelo se hizo en menos de la mitad del tiempo del cálculo secuencial.
- No se pueden añadir más unidades de ejecución paralela debido a que el límite de ejecución teórico no lo permite.

Recomendaciones

Se tienen como recomendaciones las siguientes:

1. Se debe tomar en cuenta que el último thread del bloque gastará tiempo de ejecución del cálculo debido a que este se debe ejecutar independientemente de las iteraciones que se hagan. Es por esto que las iteraciones tienen que tomar en cuenta la cantidad de threads menos uno.

2. La paralelización funciona perfectamente para disminuir el tiempo de ejecución de las implementaciones de ecuaciones diferenciales como las que involucran el método de euler.

3. Para ver si realmente se tiene un speed up calcular con un N > 4000 que no se verá reflejado antes probablemente también se recomiendo optimizar los cálculos de cada bloque desde un inicio.

Bibliografía

John A. Nairn. (n.d.). Modeling heat flow across material interfaces and cracks using the material point method. Retrieved March 19, 2022, from http://www.cof.orst.edu/cof/wse/faculty/Nairn/papers/InterfaceHeat.pdf

M.Sheikholeslami. (2019, 02 03). *Heat transfer simulation of heat storage unit with nanoparticles and fins through a heat exchanger*. ScienceDirect. https://www.researchgate.net/publication/297613897_Designing_a_parallel_algorithm_for_Heat_conduction_using_MPI_OpenMP_and_CUDA

Vinaya Sivanandan. (2015, 02 01). *Designing a parallel algorithm for Heat conduction using MPI, OpenMP and CUDA*. Research Gate. https://www.researchgate.net/publication/297613897_Designing_a_parallel_algorithm_for_Heat_conduction_using_MPI_OpenMP_and_CUDA

Vladimir E. Misilov. (2020, 04 02). *Parallel algorithm for solving the problems of heat and mass transfer in the open geothermal system*. Parallel algorithm for solving the problems of heat and mass transfer in the open geothermal system. https://aip.scitation.org/doi/10.1063/5.0035531

Apéndice

Formula1: Eficiencia

$$E(n) = \frac{S(n)}{n} = \frac{T(1)}{n * T(n)}$$

Formula 2: Ley de Ahmdal

$$T_m = T_a \cdot \left((1 - F_m) + \frac{F_m}{A_m} \right)$$

formula 3: Disipación De calor una barra

$$\frac{\partial T(x,t)}{\partial t} = c\frac{\partial^2 T(x,t)}{\partial x^2}$$