

## Actividad 1.6

### A01740197

Erick Jair Garcia Barradas

Desarrollar 2 programas que resuelvan las siguientes ecuaciones diferenciales de forma numérica usando el método de Runge-Kutta de orden 4:

$$y' = te^{3t} - 2y$$

$$y' = 1 + (t - y)^2$$

$$y' = 1 + y/t$$

$$y' = \cos(2ty) + \sin(3ty)$$

en el intervalo  $[0, \pi]$  (con 50, mil puntos en el intervalo de evaluación). Para todas las ecuaciones la condición inicial es  $y(0) = \pi/4$ . Se debe de entregar un archivo por cada una de las soluciones determinadas en el programa.

Uno de los programas a desarrollar deberá hacerlo de forma secuencial y el otro en forma paralela usando task level parallelism (TLP). Realizar la documentación en Github y mostrar los tiempos de ejecución de cada uno de los programas.

#### Programa 1 Secuencial

```
void main(int argc, char const *argv[])
{
    const double start = omp_get_wtime();
    int N = 50000;
    runge1(N, 1, "runge_ns_1.txt");
    runge2(N, 2, "runge_ns_2.txt");
    runge3(N, 3, "runge_ns_3.txt");
    runge4(N, 4, "runge_ns_4.txt");
    const double end = omp_get_wtime();
    printf("tomo (%lf) segundos\n", (end - start));
}
```

#### Programa 2 Paralelo

```

void main(int argc, char const *argv[])
{
    const double start = omp_get_wtime();
    int N = 50000;

    omp_set_num_threads(4);
    #pragma omp parallel
    {
        #pragma omp sections
        {
            #pragma omp section
                runge1(N, 1, "runge_np_1.txt");
            #pragma omp section
                runge2(N, 2, "runge_np_2.txt");
            #pragma omp section
                runge3(N, 3, "runge_np_3.txt");
            #pragma omp section
                runge4(N, 4, "runge_np_4.txt");
        }
    }
    const double end = omp_get_wtime();
    printf("tomo (%lf) segundos\n", (end - start));
}

```

Al ejecutarlos se consigue la siguiente diferencia.

```

[erickg@erick-1 1.6]$ ./secu
Numero de pasos:50000 Atendido por thread:1
Numero de pasos:50000 Atendido por thread:2
Numero de pasos:50000 Atendido por thread:3
Numero de pasos:50000 Atendido por thread:4
tomo (0.498356) segundos
[erickg@erick-1 1.6]$ gcc -fopenmp -o para para.c -lm
[erickg@erick-1 1.6]$ ./para
Numero de pasos:50000 Atendido por thread:1
Numero de pasos:50000 Atendido por thread:2
Numero de pasos:50000 Atendido por thread:3
Numero de pasos:50000 Atendido por thread:4
tomo (0.402502) segundos
[erickg@erick-1 1.6]$ █

```

Donde el programa secuencial es 0.498 segundos de operación y el programa paralelo es 0.402 segundos de operación. Con una diferencia de 0.096 segundos aproximadamente.