



Simulación de redes eléctricas con procesamiento paralelo mediante Matlab

Electrical network simulation by using parallel processing techniques and Matlab

Dante López-Coronado¹, Iván Ceja-Cerrato²,
Verónica A. Galván-Sánchez³, José A. Gutiérrez-Robles⁴,
Carlos A. López-de-Alba⁵, Eduardo S. Bañuelos-Cabral⁶,
Jorge L. García-Sánchez⁷, Julián Sotelo-Castañón⁸,
José de Jesús Nuño-Ayón⁹

López-Coronado, D; Ceja-Cerrato, I; Galván-Sánchez, V.A; Gutiérrez-Robles, J.A; López-de-Alba, C.A; Bañuelos-Cabral, E.S; García-Sánchez, J.L; Sotelo-Castañón, J; Nuño-Ayón, J.J. Simulación de redes eléctricas con procesamiento paralelo mediante Matlab. *Tecnología en Marcha*. Vol. 34, especial. ALTAE. Diciembre 2021. Pág 142-157.

 <https://doi.org/10.18845/tm.v34i7.6021>

1 Universidad de Guadalajara. México.

2 Universidad de Guadalajara. México.

3 Universidad de Guadalajara. México.

Correo electrónico: veronica.galvan@academicos.udg.mx

4 Universidad de Guadalajara. México.

5 Universidad de Guadalajara. México.

6 Universidad de Guadalajara. México.

7 Universidad de Guadalajara. México.

8 Universidad de Guadalajara. México.

9 Universidad de Guadalajara. México.



Palabras Clave

Simulación de redes eléctricas; ecuaciones diferenciales; procesamiento paralelo; tiempo de viaje; EMTP.

Resumen

En este artículo se describe el desarrollo de una metodología para simular la dinámica de una red de energía eléctrica a través del procesamiento paralelo. Los sistemas eléctricos se modelan mediante ecuaciones diferenciales, por lo que es necesaria la implementación de métodos numéricos para encontrar la solución de dichas ecuaciones. La metodología consiste en la división de una red eléctrica en subredes que están desacopladas en tiempo entre sí, es decir, las subredes resultantes se encuentran conectadas entre sí mediante líneas de transmisión. El tiempo de viaje a través de las líneas produce el desacoplamiento en tiempo necesario para que la solución de las ecuaciones diferenciales pertenecientes a cada una de las subredes se pueda obtener en paralelo. El caso de estudio principal es una red eléctrica sencilla y para su simulación con Matlab se utiliza de la librería de procesamiento paralelo. Los resultados obtenidos con la técnica de procesamiento paralelo se comparan con la simulación secuencial también implementada en Matlab y con el ATP (del inglés Alternative Transient Program) para validar los resultados obtenidos.

Keywords

Simulation of electrical networks; differential equations; parallel processing; travel time; EMTP.

Summary

This article describes the development of a methodology to simulate the dynamics of an electric power network through parallel processing. Electrical systems are modeled by differential equations, thus it is necessary to implement numerical methods to find the solution of these equations. The proposed methodology consists of dividing an electrical network into subnets that are decoupled in time from each other, that is, the resulting subsystems are connected to each other by transmission lines. The travel time among the lines produces the necessary time decoupling for the solution of the differential equations belonging to each of the subsystems to be obtained at the same time in parallel. The main case study is a simple electrical network and it is simulated with Matlab and its parallel processing library. The results obtained with the parallel processing technique are compared with the sequential simulation also implemented in Matlab and with the ATP (Alternative Transient Program) to validate the obtained results.

Introducción

Los grandes sistemas eléctricos se modelan con ecuaciones diferenciales por lo que se implementan métodos numéricos para encontrar soluciones a dichas ecuaciones. Estos métodos implican el cálculo de múltiples soluciones simultáneas para las numerosas variables de los sistemas. Para obtener soluciones más eficientemente, la simulación paralela ha permitido el aprovechamiento de todos los núcleos de una computadora para repartir las operaciones de la metodología implementada y conseguir una reducción del tiempo de simulación.

Fue en la década de los 50 cuando se analizó y se planteó la programación en paralelo; principalmente fueron los investigadores de IBM (John Cocke y Daniel Slotnick) los que discutieron el uso del paralelismo en cálculos numéricos. Años después, en la década de los 60 se abrió un debate sobre su uso en la Conferencia de la Federación Americana de Sociedades

de Procesamiento de la Información donde se acuñó la Ley de Amdahl para visualizar los límites de aceleración que se pueden alcanzar debido al paralelismo. A finales de esta década la compañía estadounidense Honeywell introdujo el primer sistema con multiprocesador. La ley de Amdahl plantea que una vez teniendo cierta cantidad de procesadores la velocidad permanecerá constante, en dependencia de la parte no paralelizable del problema.

En la actualidad, la implementación de la computación paralela se ha vuelto muy común debido a la llegada de los procesadores de varios núcleos casi por defecto en la mayoría de los dispositivos computacionales. El software ha sido una parte activa en la evolución de la programación paralela. Los programas paralelos son más difíciles de escribir que los programas secuenciales, ya que se requiere que haya una comunicación y sincronización entre las tareas que se han paralelizado.

En los últimos años el esfuerzo se ha centrado en la ejecución en paralelo, así como en la simulación tiempo real; es en este marco donde J. R. Martí et. al. [1-3] han desarrollado nuevas estrategias donde proponen el uso de líneas de transmisión como enlaces entre subsistemas, equivalentes de red para compactar un módulo o bloque, bancos de filtros para acondicionar las señales de acoplamiento, etc.

Métodos numéricos para la simulación de transitorios eléctricos

Dos métodos de solución de ecuaciones diferenciales ordinarias que se utilizan comúnmente para resolver transitorios eléctricos son Euler hacia atrás y regla trapezoidal. Las fórmulas recursivas de actualización para estas dos reglas se muestran a continuación.

Para Euler hacia atrás:

$$x(t) = x(t - \Delta t) + \Delta t \cdot \dot{x}(t) \quad (1)$$

$$\dot{x}(t) = 2 \frac{x(t) - x(t - \Delta t)}{\Delta t} - \dot{x}(t) \quad (2)$$

Para la regla trapezoidal:

$$x(t) = x(t - \Delta t) + \frac{\Delta t}{2} [\dot{x}(t) + \dot{x}(t - \Delta t)] \quad (3)$$

$$\dot{x}(t) = 2 \frac{x(t) - x(t - \Delta t)}{\Delta t} - \dot{x}(t - \Delta t) \quad (4)$$

En (1) - (4), Δt es el paso de integración, \dot{x} es la derivada de x con respecto a t y $\dot{x} = f(x, t)$. Estas ecuaciones se expresan en términos del instante de tiempo actual, t , y del instante de tiempo pasado, $t - \Delta t$. Estos métodos de integración son A-estables, lo que significa que siempre arrojan resultados estables, independientemente de la precisión de la solución. Ésta depende del tamaño del paso de tiempo, las propiedades de la regla y de las características del circuito [4]. Para más información acerca de las reglas de integración se puede consultar [5].

Modelos en dominio del tiempo del resistor, del inductor y del capacitor

El circuito equivalente de cada elemento modelado mediante la Euler hacia atrás y regla trapezoidal se muestran en el cuadro 1.

Cuadro 1. Circuitos equivalentes de elementos R,L,C.

| | Elemento | | |
|-----------------------------------|---|---|--|
| | Resistor | Inductor | Capacitor |
| Circuito equivalente | | | |
| Fórmulas con Euler hacia atrás | $Y_R = \frac{1}{R}$ | $i_{km}(t) = Y_{L,EA} \cdot v_{km}(t) + hist_{km,EA}(t)$ $Y_{L,EA} = \frac{\Delta t}{L}$ $hist_{km,EA}(t) = i_{km}(t - \Delta t)$ | $i_{km}(t) = Y_{C,EA} \cdot v_{km}(t) + hist_{km,EA}(t)$ $Y_{C,EA} = \frac{C}{\Delta t}$ $hist_{km,EA}(t) = -\frac{C}{\Delta t} v_{km}(t - \Delta t)$ |
| Fórmulas con la Regla trapezoidal | $i_{km}(t) = \frac{1}{R} v_k(t) - \frac{1}{R} v_m(t)$ | $i_{km}(t) = Y_{L,RT} \cdot v_{km}(t) + hist_{km,RT}(t)$ $Y_{L,RT} = \frac{\Delta t}{2L}$ $hist_{km,RT}(t) = Y_L \cdot v_{km}(t - \Delta t) + i_{km}(t - \Delta t)$ | $i_{km}(t) = Y_{C,RT} \cdot v_{km}(t) + hist_{km,RT}(t)$ $Y_{C,RT} = \frac{2C}{\Delta t}$ $hist_{km,RT}(t) = -\frac{2C}{\Delta t} v_{km}(t - \Delta t) - i_{km}(t - \Delta t)$ |

Línea de transmisión ideal

Se utiliza el método de las características, también conocido como el método de Bergeron, para representar los parámetros distribuidos de la línea de transmisión ideal. A continuación, se presenta una breve descripción de este método, el circuito equivalente se muestra en la figura 1. Para más detalles se recomienda consultar [6].

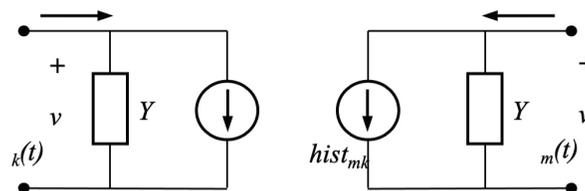


Figura 1. Circuito equivalente de una línea ideal de transmisión.

Así, las ecuaciones de dos terminales para las corrientes de la línea de transmisión son

$$\begin{aligned} i_{km}(t) &= Y_0 v_k(t) + hist_{km}(t) \text{ e} \\ i_{mk}(t) &= Y_0 v_m(t) + hist_{mk}(t), \end{aligned} \quad (5)$$

donde la admitancia característica es

$$Y_0 = \sqrt{\frac{C'}{L'}} \quad (6)$$

los términos $hist_{mk}$ e $hist_{km}$ son fuentes de corriente de historia que son conocidas en el instante t ya que dependen de valores calculados para el tiempo $t - \tau$,

$$\begin{aligned} hist_{km}(t) &= -Y_0 v_m(t - \tau) - i_{mk}(t - \tau) \\ hist_{mk}(t) &= -Y_0 v_k(t - \tau) - i_{km}(t - \tau) \end{aligned} \quad (7)$$

El tiempo de viaje de la línea es

$$\tau = \frac{d}{u} = d\sqrt{L'C'} \quad (8)$$

donde d es la longitud de la línea y u es a velocidad de propagación que se calcula como

$$u = \frac{1}{\sqrt{L'C'}} \quad (9)$$

Análisis nodal

Si en una red los elementos se reemplazan por sus circuitos equivalentes como en el cuadro 1 y en la figura 1, se puede aplicar el método nodal para obtener los voltajes en cada instante de tiempo, quedando la siguiente estructura algebraica,

$$[\mathbf{Y}][\mathbf{V}(t)] = [\mathbf{I}(t)] \quad (10)$$

donde \mathbf{Y} es la matriz de admitancias equivalentes, \mathbf{V} es la columna de voltajes nodales en el tiempo t e \mathbf{I} es la columna de corrientes que incluye corrientes de historia y las fuentes de corriente.

La metodología para encontrar los voltajes nodales de una red en un instante tiempo dado, t , es la siguiente:

1. Se selecciona un nodo de referencia y se asignan números a los nodos restantes.
2. Se crea la matriz de admitancias nodales \mathbf{Y} . Este paso es único si la conectividad de la red no cambia.
3. Se actualiza el vector de corrientes \mathbf{I} con las fuentes de corriente y las corrientes de historia.
4. Se resuelve el sistema de ecuaciones resultante para obtener los voltajes nodales mediante la ecuación $\mathbf{V} = \mathbf{Y}^{-1}\mathbf{I}$.

5. Se calculan las corrientes de historia de los elementos. Con los valores de voltajes actuales se obtienen los valores de las corrientes de historia que serán necesarios para calcular voltajes en instantes futuros.
6. Se repiten los pasos anteriores hasta alcanzar el tiempo de simulación especificado.

Procesamiento paralelo para la simulación de redes eléctricas

En esta sección se exponen los conceptos básicos del procesamiento paralelo de datos y aplicarlos al sistema de cómputo “MATrix LABORatory” (MATLAB) para el análisis de transitorios eléctricos circuitos eléctricos. De esta aplicación de conceptos se plantea un algoritmo de simulación en paralelo, el cual también se explica y se utiliza en la simulación de transitorios eléctricos de un circuito eléctrico específico.

El procesamiento paralelo es una técnica de programación en la que muchas instrucciones se ejecutan simultáneamente. Se basa en el principio de que los problemas grandes se pueden dividir en partes más pequeñas que pueden resolverse simultáneamente.

MATLAB es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio. Este sistema cuenta con un conjunto de herramientas, denominadas “Parallel Computing Toolbox” (PCT), o bien caja de herramientas para el procesamiento paralelo, permiten el uso total del poder de procesamiento de computadoras de escritorio multinúcleo para resolver problemas computacionales y de uso intensivo de datos.

Procesamiento paralelo

El fundamento principal detrás del procesamiento paralelo es que una vez que se alcanzan los límites de velocidad impuestos por cierta tecnología, la única manera de tener procesos más veloces es la realización de operaciones en forma simultánea como se ilustra en la figura 2. Esto es posible sólo en sistemas que se pueden paralelar, como es el caso del sistema de ecuaciones que modela una red eléctrica de potencia [7].

El procesamiento paralelo tiene tres características inherentes que restringen hasta cierto punto la implementación en paralelo [8]:

- **Programación de las tareas.** La eficiencia con la cual el trabajo está disponible se divide entre el número de procesadores. Esto se conoce como balance de carga.
- **Sincronización.** La sincronización de los procesadores se refiere a que las operaciones se lleven a cabo en el orden correcto.
- **Comunicación.** Los diferentes procesadores casi siempre necesitan enviarse datos entre sí en alguna etapa del algoritmo; se requiere tiempo para realizar esta tarea, el cual podría utilizarse para realizar cálculos.

Parallel Computing Toolbox

La PCT permite resolver problemas con un uso intensivo de cálculos y datos mediante procesadores multinúcleo, Graphics Processing Units (GPUs), que significa Unidades de Procesamiento de Gráficos y clústeres de ordenadores [7]. Las herramientas que PCT proporciona permiten hacer uso de toda la potencia de procesamiento de los equipos multinúcleo gracias a la ejecución de aplicaciones en workers (motores de cálculo de MATLAB), que se ejecutan localmente, como se muestra en la figura 3.

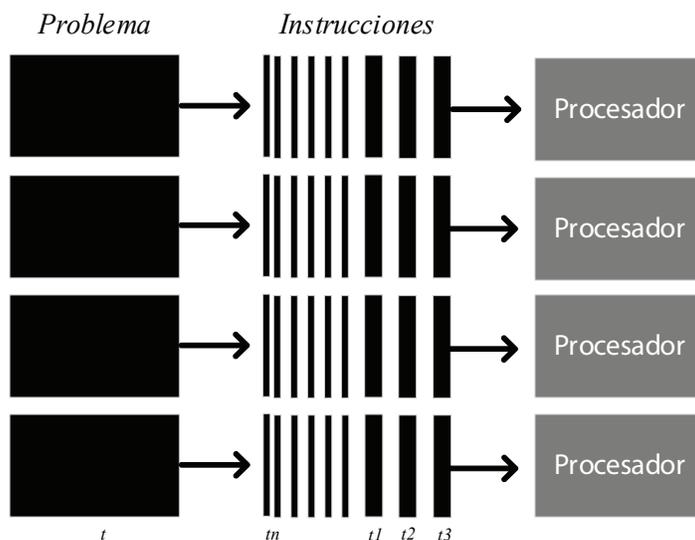


Figura 2. Representación gráfica de procesamiento en paralelo.

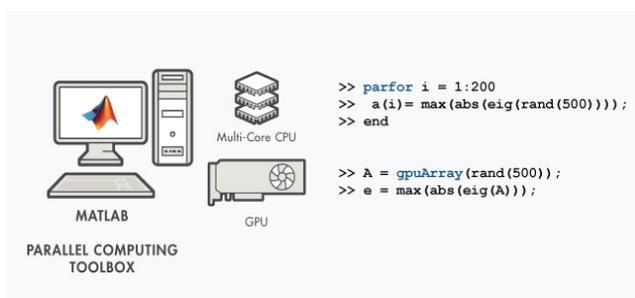


Figura 3. PCT local [9].

Ejemplo de encabezado terciario

También se puede utilizar la toolbox con MATLAB Parallel Server (Servidor Paralelo de MATLAB) para ejecutar cálculos matriciales que no caben en la memoria de un único equipo; esto se presenta en la figura 4.

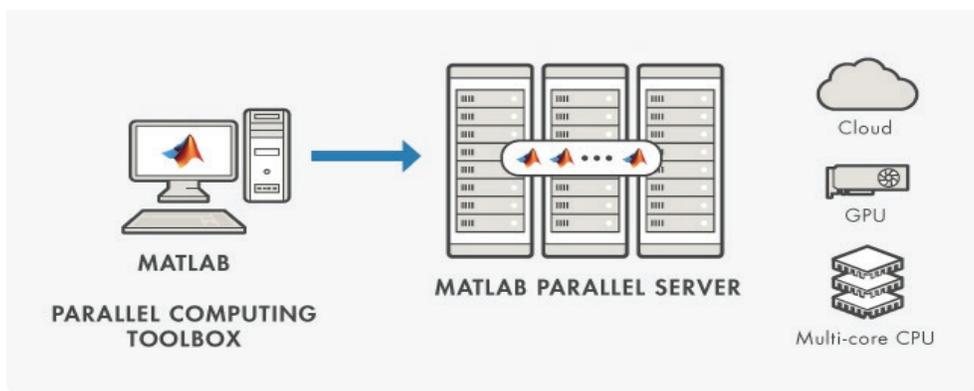


Figura 4. PCT Servidor paralelo [9].

Caja de Herramientas

Las herramientas de la PCT que se seleccionaron para las simulaciones paralelas son:

- parpool
- spmd
- labSendReceive
- labBarrier

En el cuadro 2 se presentan las descripciones principales del funcionamiento de estos comandos.

Cuadro 2. Tabla de descripciones de los comandos utilizados.

| Herramienta | Descripción |
|----------------|--|
| parpool | Inicia un conjunto de motores computacionales usando el perfil de clúster predeterminado. |
| Spmd | Ordena la ejecución del cuerpo de spmd en múltiples workers simultáneamente. |
| labSendReceive | Envía y recibe información simultáneamente entre los workers de un conjunto paralelo. |
| labBarrier | Bloquea la ejecución de un algoritmo paralelo hasta que todos los workers hayan alcanzado el llamado a labBarrier. |

El tamaño del conjunto paralelo iniciado por parpool se especifica por las preferencias de la simulación paralela y el perfil del clúster como en el ejemplo que presenta en la figura 5. En caso de no haber especificación, MATLAB utiliza todos los workers del conjunto que estén a disposición del equipo. Es esta herramienta la que permite la completa funcionalidad de las características del lenguaje paralelo en MATLAB las cuales incluyen a parfor y spmd. De manera que para poder ejecutar las declaraciones de spmd, primero se debe crear el conjunto de workers mediante parpool.

La herramienta spmd proviene de Single Program Multiple Data, que se traduce como Programas Únicos en Múltiples grupos de Información. La parte de “Programas Únicos” significa que el mismo programa es ejecutado en múltiples trabajadores. Y “Múltiples grupos de Información” indica que, aunque las declaraciones de spmd ejecuten códigos idénticos en todos los workers, cada uno de ellos puede tener información diferente, única para ese código. De manera que múltiples grupos de información pueden ser acomodados por múltiples workers. Cada worker tiene un único valor entero positivo de labindex, mientras que numlabs denota el número total de workers ejecutando el bloque en paralelo.

Los valores obtenidos del cuerpo de una declaración de spmd son convertidos a objetos compuestos (composite) en el usuario de MATLAB. Un objeto compuesto contiene referencias de los valores almacenados en los workers remotos, y esos valores pueden ser recuperados usando arreglos de celdas. La información real en los workers permanece disponible en ellos para ejecuciones subsecuentes del spmd, en tanto el objeto compuesto exista en el usuario y el conjunto paralelo permanezca abierto.

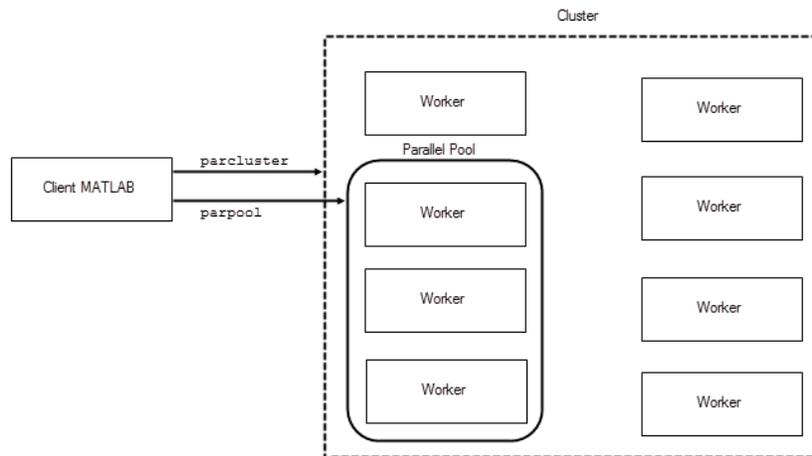


Figura 5. Proceso de inicialización de workers dentro del clúster [9].

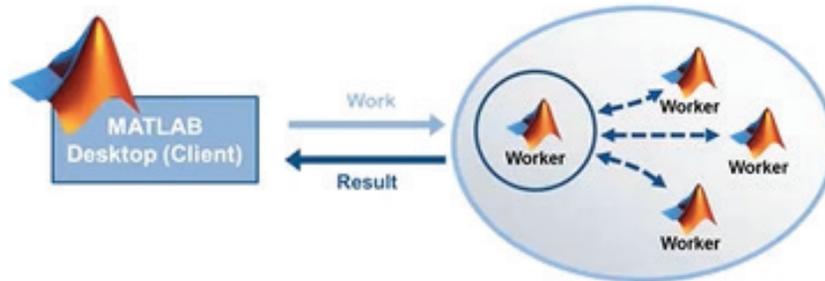


Figura 6. Comunicación entre trabajadores.

Tanto `spmd` como `parfor` sirven para ejecutar códigos en paralelo usando workers de `parpool`. Se decide utilizar `spmd` porque `parfor` distribuye y asigna todas las tareas de una simulación entre todos los trabajadores del conjunto de la manera más equitativa posible sin recurrir al intercambio de información entre los workers, es decir, son independientes. Mientras que `spmd` permite distribuir y asignar las tareas de manera personalizada y además si permite comunicación de los workers como se ilustra en la figura 6. Esta comunicación se da gracias a funciones como `labSend`, `labReceive` y `labSendReceive`.

`labSend` y `labReceive` permiten el envío de datos y la recepción de estos respectivamente por separado, por lo que `labSendReceive` agiliza este proceso e incluso puede llegar a eliminar puntos muertos en el código que se producen por llamar equivocadamente a `labSend`. La importancia de `labBarrier` radica en que previene errores en la simulación causados por los puntos muertos en el programa.

Simulación en paralelo de redes eléctricas con el método nodal

La representación matemática de un sistema físico se realiza mediante un sistema de ecuaciones. Cuando este sistema se puede paralelar, entonces se pueden usar varios procesadores para resolver el sistema de ecuaciones. Esta condición conlleva el poder separar el sistema en bloques de tal forma que su ejecución individual dependa sólo de las variables internas con la particularidad de que los datos que se transfieren entre bloques se utilizan en tiempo futuro [7].

La transferencia de datos entre bloques, o bien, subsistemas se realiza como se muestra en la figura 7. La información que se intercambia son los voltajes del nodo donde está conectada la línea de enlace y las corrientes de dicha línea.

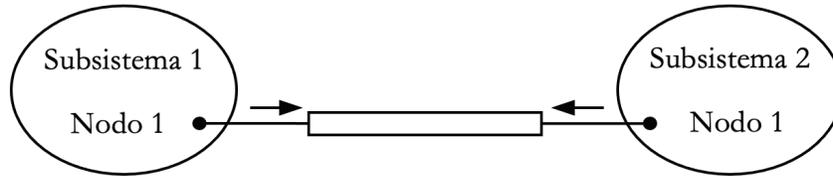


Figura 7. Esquema de transferencia de datos.

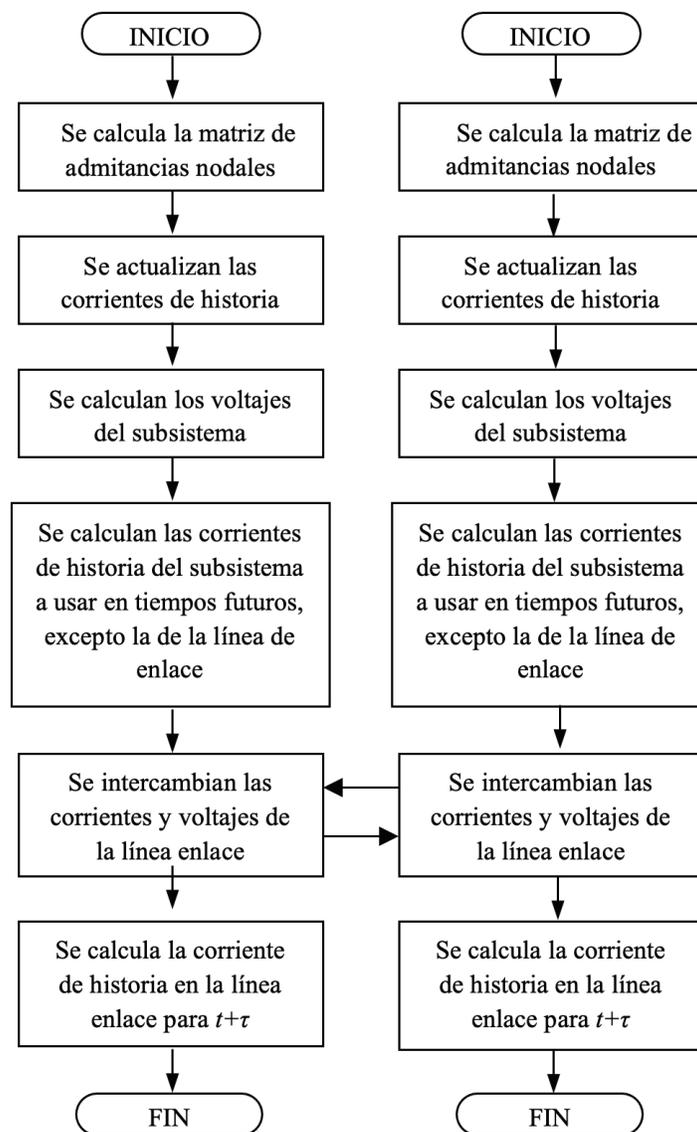


Figura 8. Esquema de simulación paralela para un instante de tiempo t .

Una vez establecidos los subsistemas en el diagrama se procede a determinar las matrices de admitancias nodales $Y1$ y $Y2$ que son particulares de cada subred. Después se determinan sus respectivas matrices inversas para obtener las matrices de impedancias $Z1$ y $Z2$. Estos vectores son constantes para todo instante de tiempo, y dependen del tamaño del paso de tiempo que se especifique para el análisis. El ciclo comienza con la formulación de las matrices de fuentes de corriente $I1$ e $I2$, correspondientes a cada subsistema en ese tiempo t . Se utilizan las respectivas Z e I de cada subsistema para calcular los voltajes nodales $V1$ y $V2$ correspondientes. Con los voltajes nodales de la iteración actual que se han determinado, se calculan las corrientes que circulan por las líneas de transmisión y por los otros elementos que conforman cada subsistema. En cada subred las corrientes y voltajes actuales de estos elementos son los que se convierten en los valores de historia del instante de solución siguiente, es decir, con los datos actuales obtenidos se determinan los valores de historia correspondientes a $t + \Delta t$ y $t + \tau$ según sea el elemento en cuestión. Es necesario aclarar que las corrientes de historia futuras del nodo que comunica los subsistemas no se pueden calcular inmediatamente después de tener los voltajes debido a que cada subsistema requiere de valores de corriente y voltaje del otro subsistema. Es en esta parte del proceso que se realiza un intercambio de estos datos entre las subredes para después realizar los cálculos de historia del nodo de enlace, con lo que finaliza el primer ciclo. Todo el proceso del ciclo iterativo se repetirá hasta alcanzar el tiempo de simulación, con lo que se alcanza el fin del procedimiento paralelo de la simulación. Este procedimiento se ilustra en el diagrama de flujo de la figura 8.

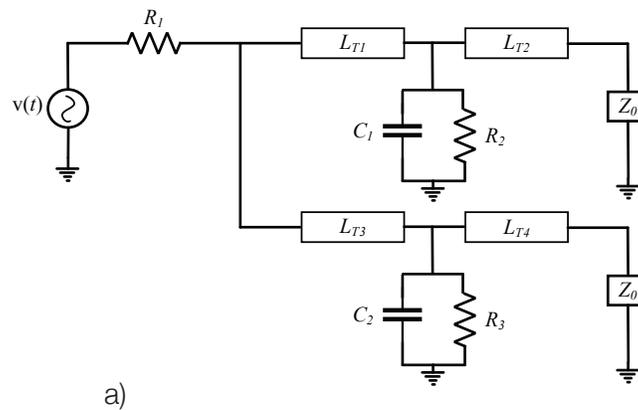
Simulación en paralelo de una red eléctrica

En esta sección se aplica el esquema de simulación en paralelo ya presentado la una red eléctrica con líneas de transmisión que se muestra en la figura 9a. La simulación se hace en código de programación de MATLAB. Este estudio se elabora aplicando tanto el método de Euler hacia atrás como la regla trapezoidal. Con el objeto de realizar una comparación de resultados de la paralelización, los resultados se comparan con los resultados obtenidos del programa ATP. En la figura 9b se muestra el circuito simulado en el Alternative Transient Program (ATP).

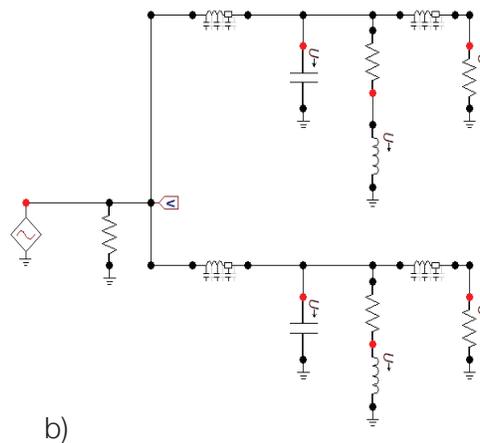
La separación de la red eléctrica en subredes es posible por los desacoples naturales de la misma en ambas extensiones del nodo que las comunica. Como ya se explicó estos desacoples se producen debido al tiempo de viaje de las ondas de voltaje y corriente a través de las líneas de transmisión. El circuito por analizar es el presentado en la figura 9a. Para su análisis se realizó la conversión de la fuente de voltaje en serie con R_1 del circuito por una fuente de corriente con R_1 en paralelo siendo la equivalen $i(t)=v(t)/R_1$. El cuadro 3 muestra los parámetros de los elementos de la red mostrada en la figura 9.

Al incorporar los circuitos equivalentes del cuadro 1 y de la figura 1 el circuito a analizar queda como se muestra en la figura 10. La red de la figura 10 tiene de manera natural (como resultado de la aplicación de los modelos) una separación en 5 islas, es decir, se puede simular como 5 regiones paralelas; lo anterior se muestra en la figura 10 marcando con diferentes colores la separación de cada sector. Para el caso de aplicación e implementación en MATLAB se tomaron los dos sectores de la parte superior y el de la parte izquierda como un subsistema y los dos sectores de la parte inferior como otro subsistema; es decir, los sectores azul, naranja y verde forman el subsistema 1, y los sectores amarillo y gris forman el subsistema 2. Se realizaron 2 simulaciones paralelas de la red, estas corresponden a la utilización de Euler hacia atrás y de la regla trapezoidal como métodos de solución. Esto con el objeto de hacer una comparación de resultados entre métodos.

Las figuras 11 a 17 muestran los resultados de las simulaciones utilizando procesamiento paralelo tanto con regla trapezoidal como con Euler hacia atrás; los resultados que se obtienen se comparan con los resultados que arroja el ATP. Como se puede notar en todos los nodos, las soluciones del ATP y de la regla trapezoidal se superponen, esto es debido a que el ATP utiliza también regla trapezoidal como método de solución y por esta razón coincide perfectamente con los resultados de la implementación en paralelo utilizando esa misma regla.



a)



b)

Figura 9. (a) Diagrama de la red de prueba (b) Red de prueba simulada en el ATP.

Cuadro 3. Parámetros del esquema de simulación.

| Parámetro | Valor (unidad) | | Parámetro | Valor (unidad) | |
|-------------|----------------|--------------|-----------|----------------|--------------|
| Dt | 50 | (μs) | R_1 | 1 | (Ω) |
| f | 60 | (Hz) | R_2 | 20 | (Ω) |
| L_{T1} | 45 | (km) | R_3 | 20 | (Ω) |
| L_{T2} | 105 | (km) | L_1 | 50 | (mH) |
| L_{T3} | 105 | (km) | L_2 | 50 | (mH) |
| L_{T4} | 195 | (km) | C_1 | 20 | (μF) |
| $Z_0=1/Y_0$ | 399.89 | (Ω) | C_2 | 20 | (μF) |

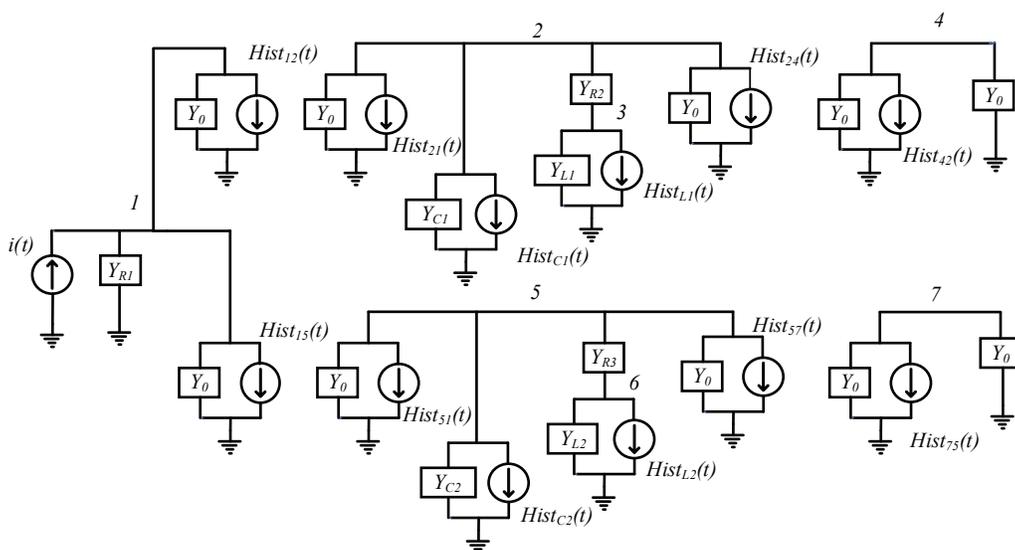
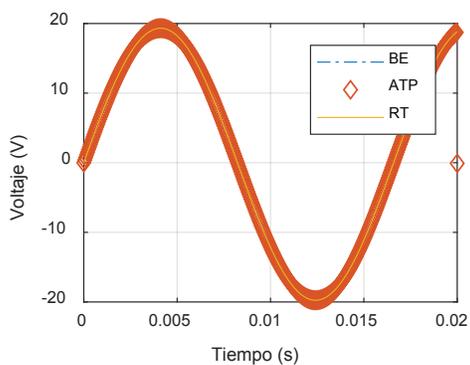
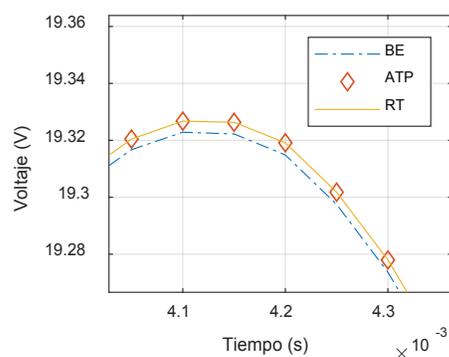


Figura 10. Circuito discreto de la red de prueba.

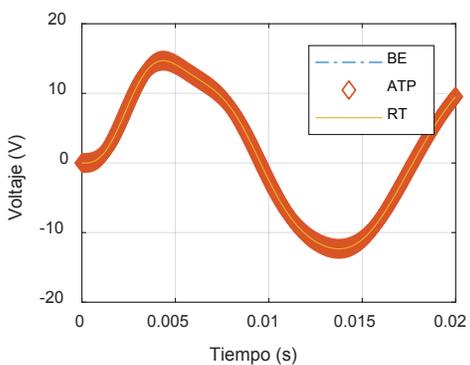


(a)

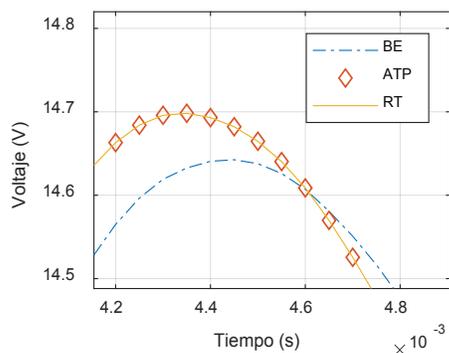


(b)

Figura 11. Respuesta de simulación, voltaje de nodo 1. (a) Vista General. (b) Acercamiento.



(a)



(b)

Figura 12. Respuesta de simulación, voltaje de nodo 2. (a) Vista General. (b) Acercamiento.

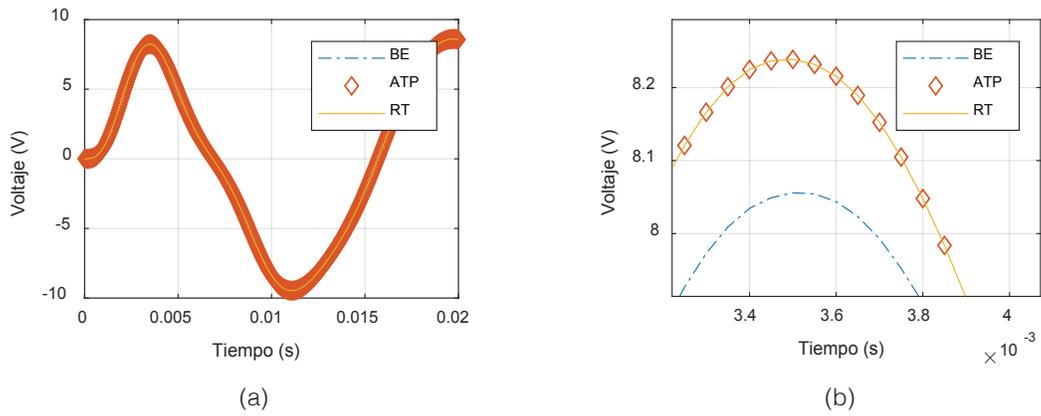


Figura 13. Respuesta de simulación, voltaje de nodo 3. (a) Vista General. (b) Acercamiento.

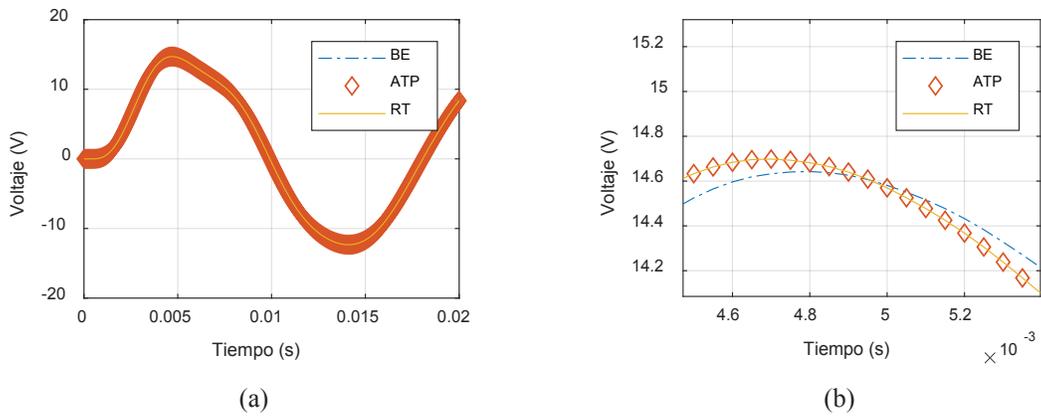


Figura 14. Respuesta de simulación, voltaje de nodo 4. (a) Vista General. (b) Acercamiento.

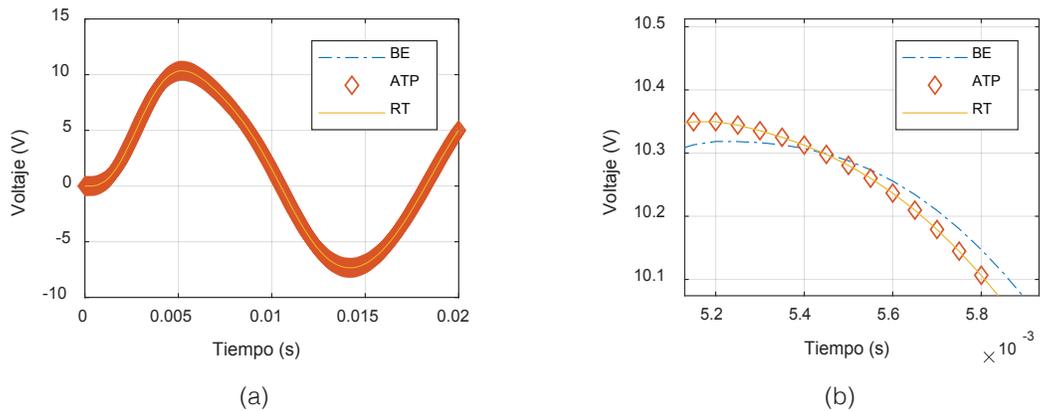


Figura 15. Respuesta de simulación, voltaje de nodo 5. (a) Vista General. (b) Acercamiento.

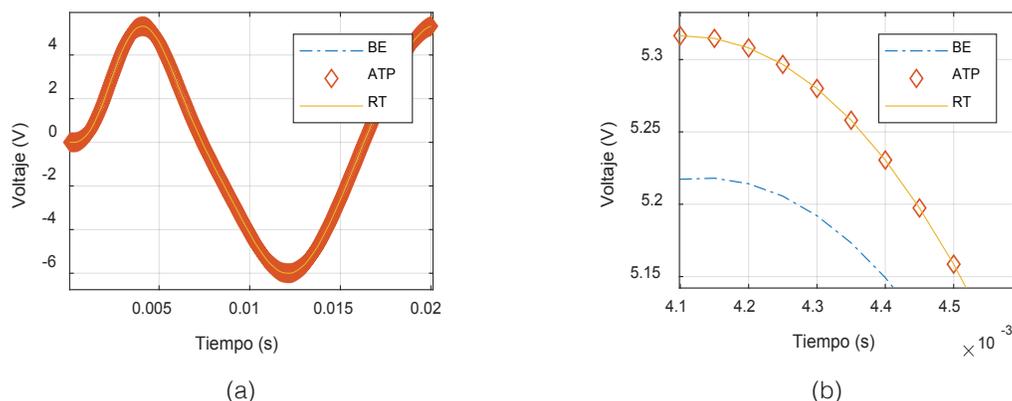


Figura 16. Respuesta de simulación, voltaje de nodo 6. (a) Vista General. (b) Acercamiento.

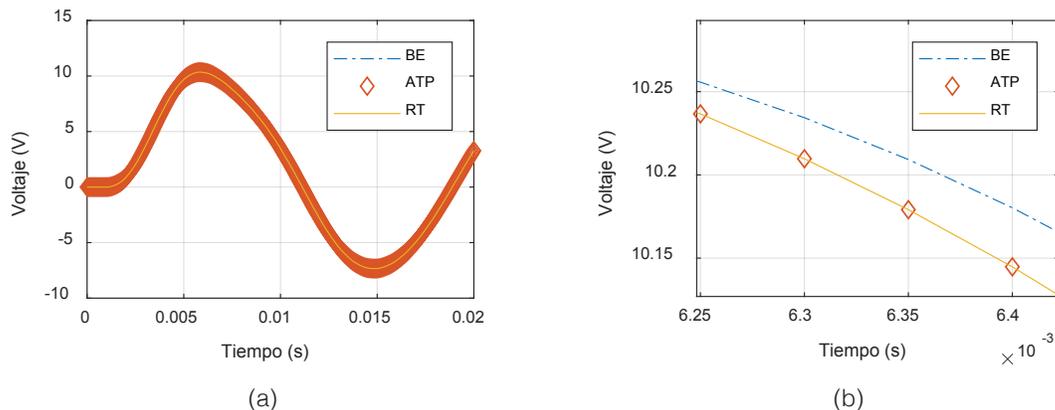


Figura 17. Respuesta de simulación, voltaje de nodo 7. (a) Vista General. (b) Acercamiento.

Conclusiones

Se ha mostrado una metodología para simular redes eléctricas con procesamiento paralelo. La recomendación para aplicar este tipo de procedimiento es que las redes sean grandes y que sean parecidas en número de nodos. Si las redes son pequeñas como el ejemplo que se presentó, el tiempo total de simulación en paralelo es mayor que la simulación lineal; esto se debe al tiempo de comunicación entre los subsistemas. Este tiempo de simulación debe ser mucho menor que el tiempo de simulación de los subsistemas para que se pueda observar el ahorro total de tiempo. El tamaño de los subsistemas debe ser parecido para que los tiempos de espera sean lo más cortos posible. Los tiempos de ejecución de cada subsistema son diferentes, es así que el subsistema más rápido debe esperar a que el más lento termine su rutina para esperar el intercambio de información.

Referencias

- [1] Tomim, Marcelo A.; Martí, José R.; De Rybel, Tom; Wang, Lei; Yao, Michael. "MATE network tearing techniques for multiprocessor solution of large power system networks". Power and Energy Society General Meeting, 2010 IEEE. IEEE, 2010, pp. 1-6.
- [2] Tomim, Marcelo A.; De Rybel, Tom; Martí, José R. "Extending the Multi-Area Thevenin Equivalents method for parallel solutions of bulk power systems". *International Journal of Electrical Power & Energy Systems*, 2013, vol. 44, no 1, pp. 192-201.

- [3] Tomim, Marcelo A.; Martí, José R.; Wang, Lei. "Parallel solution of large power system networks using the Multi-Area Thévenin Equivalents (MATE) algorithm." *International Journal of Electrical Power & Energy Systems* 31, no. 9, 2009, p. 497-503.
- [4] Ken Kundert, *The Designer's Guide to Spice and Spectre®*, Springer Science & Business Media, 2006.
- [5] Chapra, S. C., Canale, R. P., Ruiz, R. S. G., Mercado, V. H. I., Díaz, E. M., & Benites, G. E. (2011). *Métodos numéricos para ingenieros* (Vol. 5). McGraw-Hill.
- [6] H. W. Dommel, "Digital Computer Solution of Electromagnetic Transients in Single and Multiphase Networks", *IEEE Transactions on Power Apparatus and Systems*, PAS-88, #4, pp. 388-399, abril de 1969.
- [7] Galván Sánchez, V. A., Gutiérrez Robles, J. A., Naredo, J. L., Bañuelos Cabral, E. S., García Sánchez, J. L., Sotelo Castañón, J., (2013). "Simulación Rápida y Precisa de la Dinámica de un Sistema de Potencia Mediante el Uso de Técnicas de Multirresolución y de Procesamiento en Paralelo". Congreso internacional de alta tensión y aislamiento eléctrico ALTAE 2013.
- [8] Davidson, David B.: "A parallel processing tutorial". *Antennas and Propagation Magazine*, IEEE, 1990, vol. 32, no 2, pp. 6-19.
- [9] MATLAB 2020a, The MathWorks, Inc., Natick, Massachusetts, United States.