



TELECOMUNICACIÓN

Campus Sur
POLITÉCNICA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y SISTEMAS DE TELECOMUNICACIÓN

PROYECTO FIN DE GRADO

TÍTULO: ANÁLISIS DE CIRCUITOS CON MATLAB

AUTOR: BELÉN LOBO SÁNCHEZ

TUTOR: DAVID OSÉS DEL CAMPO

**TITULACIÓN: GRADO EN INGENIERÍA ELECTRÓNICA DE
COMUNICACIONES**

DEPARTAMENTO: TSC

VºBº

Miembros del Tribunal Calificador:

PRESIDENTE: ALFREDO MÉNDEZ ALONSO

VOCAL: DAVID OSÉS DEL CAMPO

SECRETARIO: LUIS ARRIERO ENCINAS

Fecha de lectura:

Calificación:

El Secretario,

AGRADECIMIENTOS

En primer lugar, quería agradecer a mi tutor David Osés la posibilidad de realizar este proyecto que une el motivo por el que empecé este grado, la electrónica, y mi gran descubrimiento académico: programar. Gracias a él he podido unir estos dos conocimientos y me ha permitido hacerlo libremente, sin limitar mis ideas.

Especialmente dar las gracias a mi padre que es el culpable de que me guste este mundo, a mi madre por apoyarme siempre en todo y guiarme para que no me vaya del camino, a mi hermana, mis abuelos y mis tíos que siempre se han interesado por mis progresos y por lo que iba aprendiendo año tras año, y a mi primo Fer por ser más que un primo, por ser como mi hermano.

Agradecer a la universidad no sólo lo que he aprendido académicamente, sino también a la gente que he conocido y que estará conmigo toda la vida. Entre ellos está mi mejor amiga, María, que ha sido y es de lo más importante que tengo, una persona que siempre me ha complementado y me ha hecho ser mejor con el mundo. También están los niños, Andrés que le llevo a mi espalda desde 2004 pero no se estar sin él, clásico ejemplo de “ni contigo ni sin ti”; Alberto Carrasco, por sus ojitos claros y su léxico que ha sido la infraestructura principal en la vida de todos; Andrés LM por sus expresiones, su risa contagiosa y esas pestañas de chica que son la envidia de todas; Jairo con su manera peculiar de decir las cosas y con el pelo más suave que he tocado nunca; a Álvaro Asenjo, nuestro fideo, por ser él y por soportarnos mutuamente; a David Medina por tratarme siempre con cariño y preocuparse por mi; y a Dani Mora por aparecer en una clase de programación que marcó un antes y un después en mi vida. Finalmente, acordarme de una persona que ha llegado en la última etapa de la universidad y pisando fuerte, Coqui, gracias por ser tú, por estar ahí y por entenderme mejor que nadie.

Por otro lado, a mis amigas que han hecho que me despejara y me riera como nunca, Elena, Macarena y especialmente Maribel, por ser tan como yo. También han estado las que conservo del colegio, las de toda la vida: Marta, Jenny, Teresa, Abi... Que, aunque la vida haya cambiado y nuestros caminos no sean los mismos, siempre se hace lo posible porque coincidan de nuevo en algún momento.

En general, gracias a todos por estar ahí, apoyarme y quererme tal y como soy.

Os quiero.

Belén

A mi familia y amigos de verdad.

Hepburn, Audrey: "Haz en la medida de lo que esperas conseguir"

Disney, Walter Elias: "Pregúntate si lo que estás haciendo hoy te acerca al lugar en el que quieres estar mañana"

RESUMEN

En la actualidad y desde hace algunos años, muchas carreras y módulos de grado superior se basan en la electrónica. En algunos institutos hay asignaturas que hacen una breve introducción a este mundo como es el caso de Tecnología u optativas como Electrotecnia, pero no llegan a profundizar mucho, sólo dan una breve introducción de conceptos básicos.

Muchas veces resulta tedioso el resolver los circuitos y más cuando no se sabe si el resultado está bien o no ya que se plantean muchas ecuaciones con muchas incógnitas y es relativamente fácil equivocarse en un número. Esto hace que el análisis no haya servido para nada ya que el resultado final no es correcto, lo que puede llegar a ser frustrante y puede provocar el abandono de ciertos alumnos.

Hay algunos programas que pueden resultar de ayuda para este aprendizaje pero no son muy intuitivos y requieren un tiempo para conseguir manejarlos con cierta soltura y que, realmente, sirvan como soporte para alumnos que están empezando con esta materia. Además, la mayoría son de pago o tienen una versión de prueba limitada.

Con este proyecto se quiso hacer una herramienta sencilla que analizara circuitos básicos, es decir, resistivos de corriente continua de hasta cuatro nodos fundamentales para gente que estuviera empezando en este campo. Más concretamente, para estudiantes de primer año de ingeniería de telecomunicaciones que estuvieran cursando Análisis de Circuitos I.

Se propuso crear un método de aprendizaje que motivara al usuario y le reportase una ayuda en el proceso de aprendizaje de análisis de circuitos. Este método cuenta con una interfaz de usuario amigable, realizada en un proyecto paralelo, que se basa en importar una imagen, que puede ser hecha desde un teléfono móvil o una captura de pantalla, que reconoce los elementos del circuito y sus valores. Estos datos son procesados por una serie de algoritmos que dan como resultado la tensión en los nodos fundamentales, dato necesario para tener totalmente resuelto el circuito.

Este proyecto se centra en la realización de estos algoritmos, la mayoría matemáticos, basados en matrices y sus propiedades, y la detección de nodos para permitir simplificar el circuito lo máximo posible y poder adaptarse a cualquier circuito resistivo de hasta cuatro nodos.

Se pensó que una aplicación sencilla de usar ayudaría a los alumnos a mejorar y agilizar el aprendizaje de análisis de circuitos para establecer una base sólida que serviría para asentar mejor futuros conocimientos más avanzados.

ABSTRACT

Nowadays and for some years ago, many degrees and top grade modules are based on electronics. In some schools there are subjects that make a brief introduction to this world as Technology or optatives such as Electrical Engineering Technology, but do not reach too deeply, just give a brief introduction of basic concepts.

Often it is tedious solving circuits and more when you do not know if the result is good or not due to there are many equations with many unknown variables and is relatively easy to go wrong in a number. This makes that the analysis is no use at all because the final result is not correct. This can become frustrating for students and they could give up these studies.

There are some programs that can be helpful for this learning but are not very intuitive and require too much time to get handle with some looseness and they serve as a support for students who are beginning with this matter. In addition, the most of them are paid or have a limited trial version.

In this project, the goal is to make a simple tool to analyze basic circuits, resistive DC circuits with no more than four key nodes for people who were starting in this field, more specifically, for first-year engineering telecommunications students who were studying Circuit Analysis I.

The aim is to create a learning method that motivates and aids the user in the circuit analysis learning process. This method has a friendly interface, realized by some colleagues in a parallel project, which is based on import an image, which can be made from a mobile phone or a screenshot, which recognizes the circuit elements and their values. This data is processed by some algorithms that get the voltage on the fundamental nodes, data necessary to have completely solved the circuit.

This project focuses on the realization of these algorithms based on matrix and their properties, and detection of nodes to simplify the circuit. The algorithms are able to adapt to any resistive circuit up to four nodes.

It was thought that a simple application may help students to improve their learning circuit analysis. Students could establish a solid foundation that will help them in the future to settle more advanced knowledge.

ÍNDICE DE CONTENIDOS

AGRADECIMIENTOS	5
RESUMEN	3
ABSTRACT	4
ÍNDICE DE CONTENIDOS	7
ÍNDICE DE FIGURAS	9
ÍNDICE DE TABLAS	10
1. INTRODUCCIÓN Y OBJETIVOS	1
2. ESTADO DEL ARTE	3
3. DESCRIPCIÓN DE LA SOLUCIÓN	7
3.1. Elección del entorno de programación	7
3.2. Conceptos básicos de circuitos eléctricos	8
3.2.1. Carga eléctrica, tensión y corriente	8
3.2.2. Fuentes de tensión y de corriente	9
3.2.3. Ley de Ohm y resistencia eléctrica	10
3.2.4. Leyes de Kirchhoff	10
3.2.5. Asociación de resistencias	12
3.3. Descripción de problemas y soluciones propuestas	13
3.3.1. Método nodal	13
3.3.2. Método nodal modificado	15
3.3.3. Método ZBUS	18
3.3.4. Método elegido	19
4. DISEÑO DE LA APLICACIÓN	21
4.1. Fundamentos de Matlab	21
4.1.1. Formas de definir una matriz	21
4.1.2. Funciones de matrices	22
4.2. Análisis nodal	23
4.3. Funciones del análisis nodal	24
4.3.1. Un nodo fundamental	24
4.3.2. Dos nodos fundamentales	25
4.3.3. Tres nodos fundamentales	26
4.3.4. Cuatro nodos fundamentales	28
4.4. Funciones simplificadoras	29
4.4.1. Localización de resistencias en serie	29
4.4.2. Localización de fuentes independientes de tensión en serie con resistencias	31
4.4.3. Algoritmo ordenar	32
4.5. Funcionamiento global	33
5. RESULTADOS	35
6. MANUAL DE USUARIO	45
6.1. Inicio de la aplicación	45
6.2. Ejemplos	46
6.2.1. Ejemplo de circuito con un nodo fundamental	46
6.2.2. Ejemplo de circuito con dos nodos fundamentales	49
6.2.3. Ejemplo de circuito con tres nodos fundamentales	51
6.2.4. Ejemplo de circuito con cuatro nodos fundamentales	55

7. CONCLUSIONES Y LÍNEAS FUTURAS	59
7.1. Conclusiones	59
7.2. Líneas futuras	60
8. REFERENCIAS.....	61

ÍNDICE DE FIGURAS

<i>Figura 1: captura de la pantalla principal del programa Orcad PSpice</i>	<i>3</i>
<i>Figura 2: circuito eléctrico usando la herramienta Simulink.....</i>	<i>4</i>
<i>Figura 3: captura de pantalla del esquemático de un circuito hecho en CircuitLab</i>	<i>5</i>
<i>Figura 4: captura de pantalla del resultado del análisis de un circuito con CircuitLab.....</i>	<i>5</i>
<i>Figura 5: a)fuente independiente de tensión; b)fuente independiente de corriente.....</i>	<i>9</i>
<i>Figura 6: a)fuente de corriente dependiente de tensión; b)fuente de tensión dependiente de tensión; c)fuente de corriente dependiente de corriente; d)fuente de tensión dependiente de corriente.</i>	<i>9</i>
<i>Figura 7: ejemplo de circuito para aplicar la Ley de Kirchhoff de las corrientes.....</i>	<i>10</i>
<i>Figura 8: ejemplo de circuito para aplicar la Ley de Kirchhoff de las tensiones</i>	<i>11</i>
<i>Figura 9: a)circuito con resistencias en serie; b)circuito equivalente del anterior</i>	<i>12</i>
<i>Figura 10: a)circuito con resistencias en paralelo; b)circuito equivalente al anterior</i>	<i>12</i>
<i>Figura 11: fuente dependiente de tensión entre dos nodos.....</i>	<i>14</i>
<i>Figura 12: circuito con un transistor bipolar</i>	<i>14</i>
<i>Figura 13: modelo en pequeña señal de un transistor bipolar</i>	<i>14</i>
<i>Figura 14: modelo en pequeña señal del circuito de la figura 12.....</i>	<i>15</i>
<i>Figura 15: resistencia entre dos nodos</i>	<i>16</i>
<i>Figura 16: resistencia entre dos nodos con tensión entre ellos</i>	<i>16</i>
<i>Figura 17: circuito equivalente de pequeña señal con fuente de corriente dependiente de corriente.....</i>	<i>17</i>
<i>Figura 18: circuito con un supernodo.</i>	<i>36</i>
<i>Figura 19: circuito genérico de tres nodos fundamentales.....</i>	<i>37</i>
<i>Figura 20: circuito genérico de cuatro nodos fundamentales</i>	<i>37</i>
<i>Figura 21: captura de pantalla del script de pruebas, parte 1</i>	<i>38</i>
<i>Figura 22: captura de pantalla del script de pruebas, parte 2</i>	<i>38</i>
<i>Figura 23: captura de pantalla de mensaje de error en la ventana de comandos.....</i>	<i>39</i>
<i>Figura 24: captura de pantalla con el resultado del algoritmo de TresNodos en la ventana de comandos.....</i>	<i>39</i>
<i>Figura 25: circuito con resistencias en serie entre nodos fundamentales.....</i>	<i>40</i>
<i>Figura 26: circuito equivalente al de la Figura 25</i>	<i>41</i>
<i>Figura 27: circuito de cuatro nodos fundamentales con resistencias en serie y supernodo</i>	<i>41</i>
<i>Figura 28: circuito equivalente al de la Figura 27</i>	<i>42</i>
<i>Figura 29: captura de pantalla de la reducción de la matriz del circuito de la Figura 28... </i>	<i>43</i>
<i>Figura 30: circuito de la Figura 27 identificando todos los nodos.....</i>	<i>43</i>
<i>Figura 31: ficheros con la información del circuito de la Figura 30</i>	<i>44</i>
<i>Figura 32: ventana de comandos de Matlab tras ejecutar el programa con los ficheros de la Figura 31</i>	<i>44</i>
<i>Figura 33: fichero tensiones.txt, solución del circuito de la Figura 30</i>	<i>44</i>
<i>Figura 34: formato de los ficheros de las resistencias, fuentes de corriente y fuentes de tensión.....</i>	<i>45</i>
<i>Figura 35: formato del fichero tensiones.txt.....</i>	<i>46</i>
<i>Figura 36: ejemplo de circuito con un nodo fundamental.....</i>	<i>46</i>
<i>Figura 37: identificación de los nodos de la Figura 36.....</i>	<i>47</i>
<i>Figura 38: circuito equivalente al de la Figura 36</i>	<i>47</i>
<i>Figura 39: identificación del nodo fundamental del circuito de la Figura 38</i>	<i>47</i>
<i>Figura 40: ficheros con la información del circuito de la Figura 37</i>	<i>48</i>
<i>Figura 41: resolución del circuito de la Figura 40 (fichero tensiones.txt)</i>	<i>48</i>

<i>Figura 42: ejemplo de circuito con dos nodos fundamentales</i>	49
<i>Figura 43: identificación de los nodos de la Figura 42</i>	49
<i>Figura 44: ficheros con la información del circuito de la Figura 43</i>	50
<i>Figura 45: resolución del circuito de la Figura 43 (tensiones.txt)</i>	50
<i>Figura 46: ejemplo de circuito con tres nodos fundamentales</i>	51
<i>Figura 47: identificación de los nodos de la Figura 46</i>	51
<i>Figura 48: circuito equivalente al de la Figura 46</i>	52
<i>Figura 49: identificación de los nodos fundamentales del circuito de la Figura 48</i>	52
<i>Figura 50: ficheros con la información del circuito de la Figura 47</i>	54
<i>Figura 51: resolución del circuito de la Figura 47 (tensiones.txt)</i>	54
<i>Figura 52: ejemplo de circuito con cuatro nodos fundamentales</i>	55
<i>Figura 53: identificación de los nodos de la Figura 52</i>	55
<i>Figura 54: ficheros con la información del circuito de la Figura 53</i>	57
<i>Figura 55: resolución del circuito de la Figura 53</i>	58

ÍNDICE DE TABLAS

<i>Tabla 1: tabla comparativa de las alternativas a la aplicación desarrollada</i>	6
<i>Tabla 2: tabla comparativa de los posibles entornos de programación</i>	7
<i>Tabla 3: funciones útiles de matrices en Matlab</i>	22
<i>Tabla 4: términos importantes para el análisis nodal</i>	23

1. INTRODUCCIÓN Y OBJETIVOS

Es necesario resaltar que en la actualidad, en la mayoría de las ingenierías y en algunas formaciones profesionales relacionadas con el mundo de la tecnología, se estudian circuitos electrónicos ya que la electrónica está implantada en nuestras vidas desde el dispositivo más sencillo que se nos pueda ocurrir, como una bombilla conectada a una pila y un interruptor, hasta complejos sistemas autónomos, como un robot.

Las herramientas proporcionadas hasta la fecha son muy potentes pero enfocadas a un público que tenga un cierto conocimiento y destreza analizando circuitos como es el caso de Orcad PSpice.

Este proyecto surge de la necesidad de facilitar una herramienta muy sencilla de manejar para la resolución de circuitos dirigida para alumnos de nuevo ingreso en la Escuela matriculados de la asignatura Análisis de Circuitos I sin necesidad de que éstos apenas tengan conocimientos previos más allá de identificar los componentes.

Este proyecto tiene dos partes diferenciadas. Por un lado, la interfaz gráfica que identifica automáticamente los componentes del circuito y la conexión entre ellos a partir de una imagen importada, dando como resultado varios ficheros que recogen todos los datos necesarios para poder realizar el cálculo nodal en la segunda parte. Esta primera parte se realizará en un proyecto a parte. El cálculo nodal será una serie de algoritmos que interpretan los datos y, a partir de ellos, crea un fichero con las tensiones de los nodos significativos. Se consideran nodos significativos aquellos que tienen varios componentes conectados a ellos y no son simplificables.

El proyecto no tiene como objetivo que los alumnos lo usen para que les haga las tareas de clase, puesto que éstas se mandan para practicar los conocimientos adquiridos. Por ello, esta aplicación no muestra la metodología empleada para llegar a la solución, sino solamente el resultado para comprobar que los circuitos analizados estén correctamente resueltos.

Esta memoria se ha estructurado para poder entender el desarrollo del propio proyecto. Para ello, primero hay que conocer las otras alternativas al mismo problema planteado existentes en el mercado. Después hay que abordar una posible solución, eligiendo el entorno de programación más adecuado para esta finalidad, teniendo en cuenta que se trata de algoritmos matemáticos. Además de que se tiene que tener una noción básica de análisis de circuitos ya que sino es imposible usar la herramienta y plantear soluciones.

Análisis de circuitos mediante Matlab

Una vez estudiados estos conocimientos previos, se buscan alternativas de soluciones, estudiando sus pros y sus contras. Ya elegida la solución, se subdivide en objetivos que, unidos, consigan el funcionamiento global deseado:

1. Lectura y escritura de datos entre el entorno de programación elegido y los ficheros.
2. Algoritmos de simplificación de circuitos.
3. Métodos para clasificar los circuitos.
4. Funciones que obtengan la solución final.

2. ESTADO DEL ARTE

En este apartado se verán las herramientas usadas hoy en día que dan la misma solución que esta aplicación, es decir, herramientas que se usan para resolver análisis de circuitos por ordenador. Para poder hacer la comparativa habría que unir este proyecto y el desarrollado por los compañeros, ya que todas las aplicaciones tienen interfaz de usuario y un funcionamiento que procesa y analiza los circuitos.

En primer lugar, destacar la herramienta más utilizada en esta Universidad para el análisis de circuitos por ordenador, OrCad PSpice. Es un programa diseñado para ingenieros electrónicos que permite el desarrollo de un circuito eléctrico desde la creación del esquemático hasta la realización de una placa de circuito impreso generando el footprint de la misma. Además, ofrece una funcionalidad muy importante, la simulación de los circuitos que hace posible estudiar y ver la respuesta del diseño una vez realizado en una PCB.

Esta potente herramienta, la cuál sólo tiene sentido comparar con nuestra aplicación si nos centramos en la parte de simulación, fue creada en 1985 bajo la marca registrada OrCAD, por John Durbetaki, Ken Seymour, Seymour y Keith como "OrCAD Systems Corporation" en Hillsboro, Oregon. El primer producto de la compañía fue SDT y que solamente hacía esquemáticos. Al año siguiente se integró un simulador digital, VST, y algunas herramientas de diseño de PCBs pero no fue hasta 1997 cuando se fusionó con microSIM para adquirir PSpice, una simulación de señales mixtas analógicas y digitales basado en el diseño por ordenador de estas placas de circuito impreso.

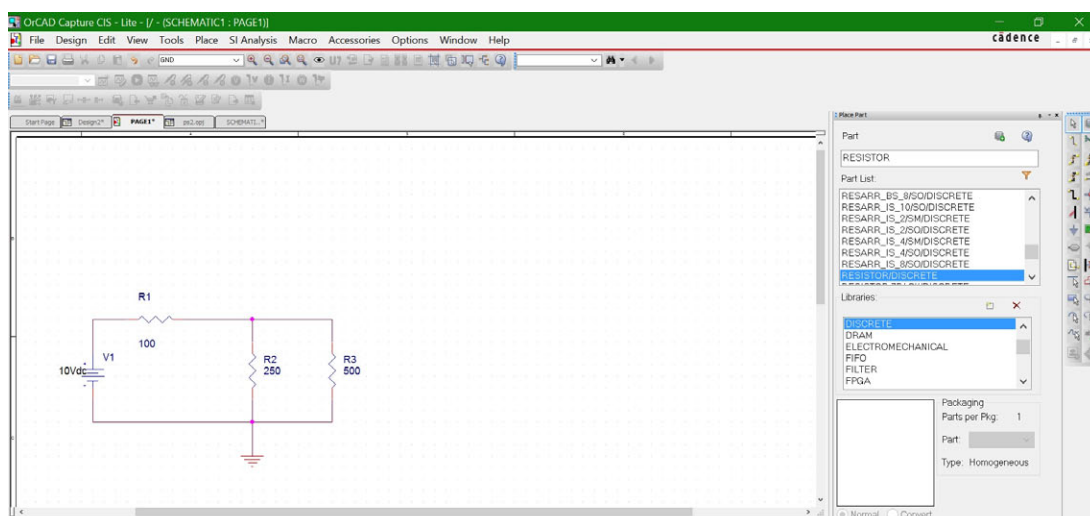


Figura 1: captura de la pantalla principal del programa Orcad PSpice

Análisis de circuitos mediante Matlab

Otra herramienta que se puede utilizar con el fin de analizar circuitos es un paquete adicional de Matlab, Simulink, que permite diseñar y simular un circuito entendido como un sistema, es decir, hay que entenderlo por la funcionalidad de cada uno de los bloques que lo forma.

Para poder usar Simulink como analizador de circuitos, hay que tener instalada la librería de "SimPower Systems", donde se encuentran todos los elementos que pueden formar parte de un circuito como resistencias, fuentes, masas... y cajas de medidas, es decir, bloques con la funcionalidad de voltímetro o amperímetro. Una vez montado el circuito, se ejecuta el archivo Simulink y se obtienen las medidas de corriente y tensión en cada punto donde se hayan puesto las cajas de medidas y mencionadas.

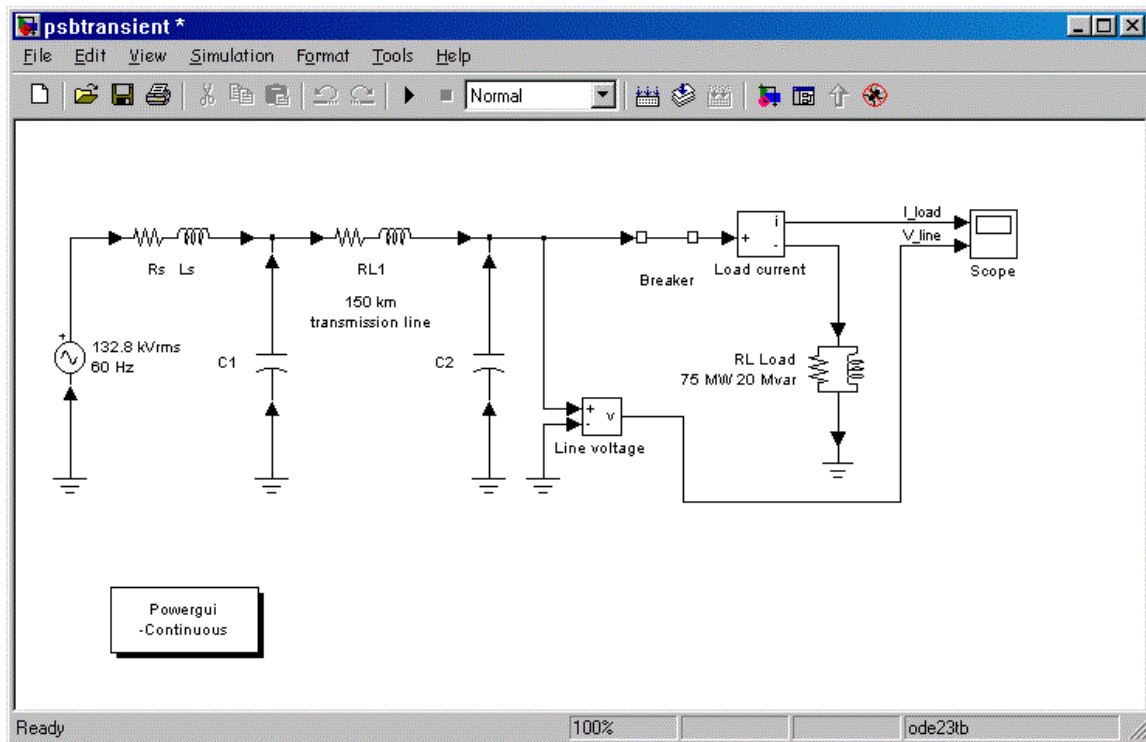


Figura 2: circuito eléctrico usando la herramienta Simulink

La aplicación CircuitLab permite capturar y simular esquemas electrónicos online. Tiene una interfaz de usuario muy sencilla y es, posiblemente, la que alternativa más parecida al programa diseñado. Además, permite compartir los circuitos simulados para que puedan verlos más usuarios y es compatible con varios sistemas operativos.

Hay disponible una versión limitada gratuita y otra de pago que permite realizar todos los circuitos que se quiera.

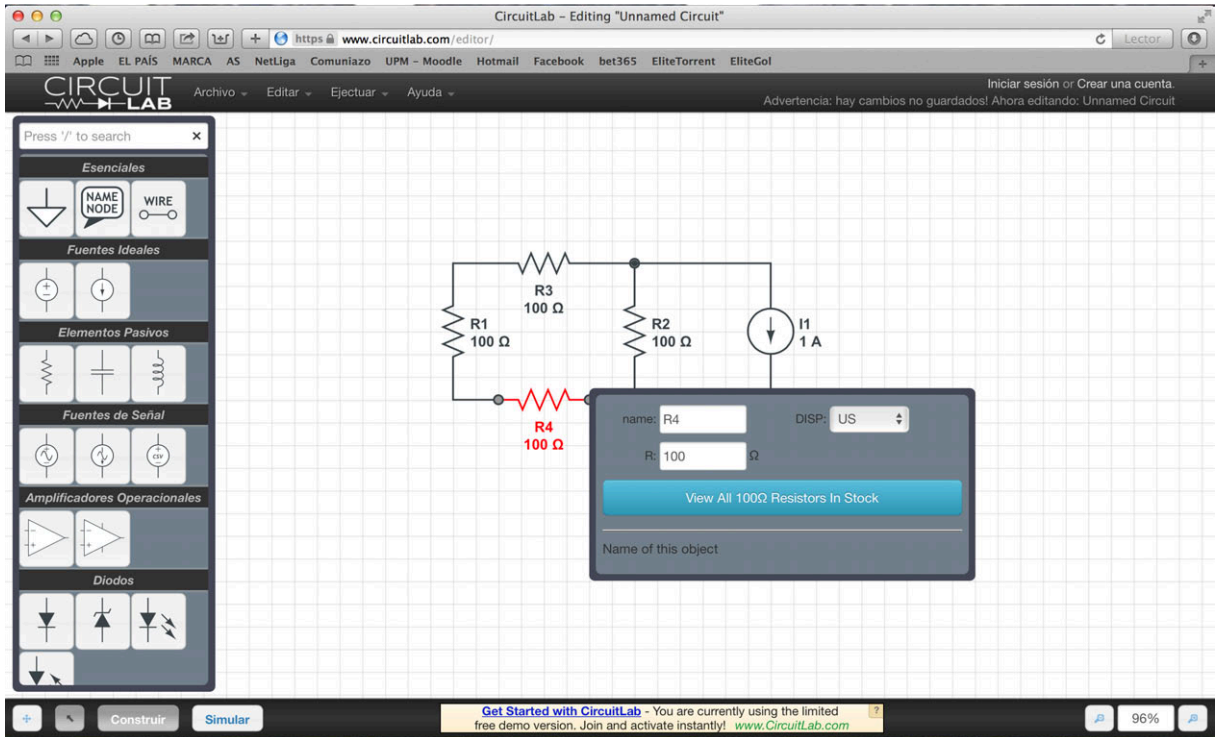


Figura 3: captura de pantalla del esquemático de un circuito hecho en CircuitLab

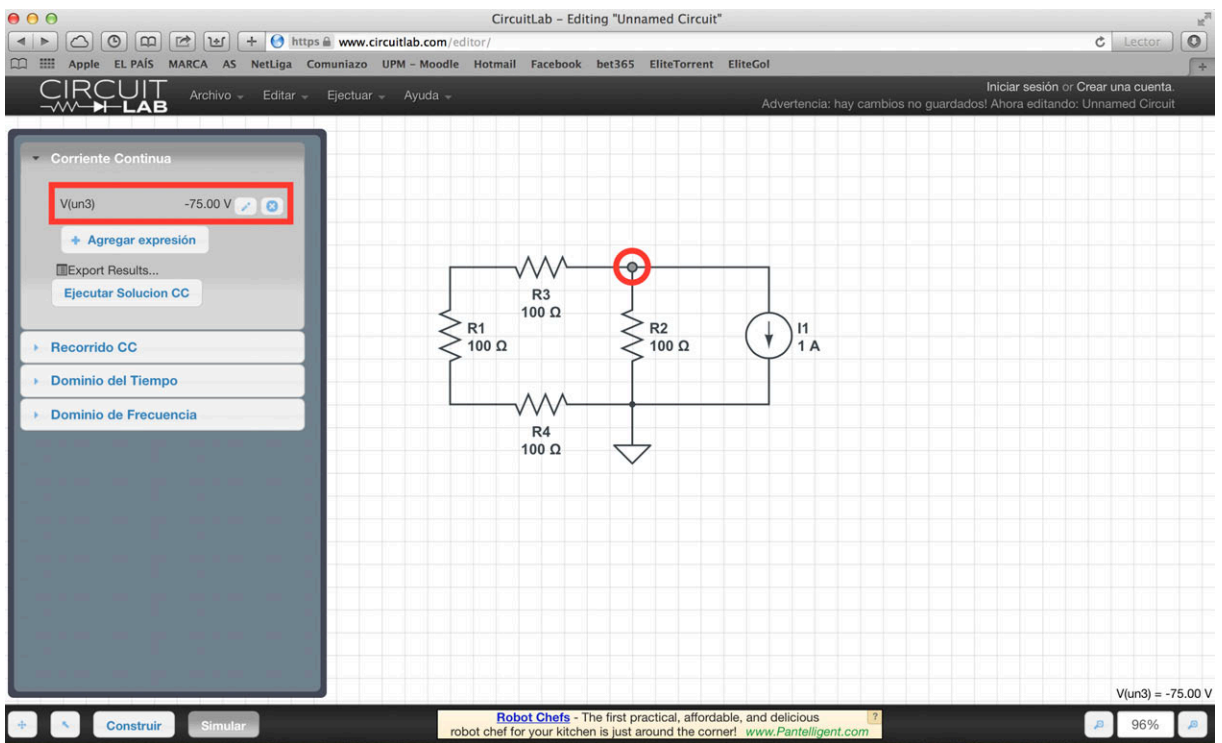


Figura 4: captura de pantalla del resultado del análisis de un circuito con CircuitLab

	Versión Gratuita	Precio versión completa	Limitaciones	Manejo
OrCAD PSpice	Sí	6600\$ (1000\$ fianza de permanencia para el primer año).	Nº de componentes del esquemático.	Complejo
SimPower Systems (Librería de Simulink).	No	Estudiantes: 69€ Matlab + 20€/ librería Standard: 2000€ Matlab + 1000€-3000€/librería	Necesaria comprar librería además de la licencia de Matlab.	Complejo
CircuitLab	Sí	1200€/6 meses	Tiempo limitado de estar haciendo circuitos	Sencillo

Tabla 1: tabla comparativa de las alternativas a la aplicación desarrollada

La aplicación desarrollada es de manejo sencillo y gratuita. El único problema que tiene es que los circuitos tienen que ser resistivos y reducibles a un máximo de cuatro nodos fundamentales. Para la finalidad que tiene es suficiente, ya que soluciona los circuitos de corriente continua estudiados en la asignatura *Análisis de Circuitos I*.

3. DESCRIPCIÓN DE LA SOLUCIÓN

3.1. Elección del entorno de programación

Para realizar la elección de la herramienta de desarrollo de la aplicación se han tenido en cuenta las siguientes razones:

- Compatibilidad con un lenguaje de programación que esté en uso y que se utilice en la Escuela en varias asignaturas.
- Que tenga un potente sistema de análisis matemático para que sea lo más rápido posible.
- Facilidad de análisis de imágenes.

Nombre	Lenguaje	Licencia	Uso	Aprendizaje	Estudiado durante la carrera	Eficiencia, orientado a la resolución de algoritmos matemáticos
Matlab	Matlab	Proporcionada por la UPM	En uso	Intermedio	Sí	Alta
Eclipse	Java	Libre	En uso	Intermedio	Sí	Normal
Crimson	C	Libre	En uso	Fácil	Sí	Baja
Otros editores de texto	C	Libre	En desuso	Fácil	No	Baja

Tabla 2: tabla comparativa de los posibles entornos de programación

Para la finalidad de la aplicación lo más idóneo es usar Matlab ya que se usa en varias asignaturas de la carrera, está disponible en todos los ordenadores de libre acceso de la Escuela (laboratorios y aula Reina Sofía) y hay una licencia para ordenadores personales al estar estudiando en la Universidad Politécnica de Madrid. Pero la principal diferencia frente a otros entornos y lenguajes es su gran capacidad para el cálculo matemático y procesado de datos e imágenes.

3.2. Conceptos básicos de circuitos eléctricos

En este apartado se verán definiciones de conceptos y palabras clave que resultarán de gran ayuda para entender los algoritmos desarrollados.

Para empezar se definen modelos de circuito, que son modelos matemáticos de uso común para los sistemas eléctricos los cuales están formados por una serie de elementos. Estos elementos se llaman componentes de circuito ideales. Un componente de circuito ideal no es otra cosa que un modelo matemático de un componente eléctrico ideal que representa el comportamiento real con buena precisión.

Al análisis del comportamiento del modelo de circuito y de sus componentes de circuito ideales se conoce como análisis de circuitos. Este análisis se hace formulando expresiones matemáticas con términos de magnitudes medibles, en este caso, en función de la corriente y la tensión.

3.2.1. Carga eléctrica, tensión y corriente

Para poder describir estos dos fenómenos eléctricos hay que partir del concepto de carga. Una carga eléctrica es una magnitud física característica de los fenómenos eléctricos y es una propiedad de los cuerpos ya que cualquier cuerpo eléctrico puede adquirir carga eléctrica. La unidad con la que se mide la carga es el Coulomb (C), cuya unidad corresponde a $6,25 \cdot 10^{18}$ electrones, por lo que la carga de un electrón es de $1,6022 \cdot 10^{-19}$ C.

Aplicando este concepto a circuitos, la tensión es la fuerza eléctrica que se crea cuando se separan cargas mientras que la corriente es el flujo eléctrico creado por el movimiento de una carga.

La tensión se define como la energía por unidad de carga creada por la separación de cargas positivas y negativas y sus unidades del Sistema Internacional son los voltios (V).

$v = \frac{dw}{dq}$ donde v es la tensión en voltios, w=energía en Julios y q la carga en Coulombios.

La corriente eléctrica es la tasa de flujo de carga por unidad de tiempo que recorre un material, producida por el movimiento de cargas en el interior de un material. Se produce dicho movimiento cuando hay una diferencia de potencial entre dos puntos, es decir, dos puntos con carga de diferente signo. La unidad con la que se mide la corriente es el Amperio (A) y equivale a C/s del Sistema Internacional. Se define como:

$i = \frac{dq}{dt}$ donde i es la corriente en Amperios, q la carga en Coulombios y t el tiempo en segundos.

3.2.2. Fuentes de tensión y de corriente

Una fuente eléctrica es un dispositivo capaz de convertir energía no eléctrica en eléctrica y viceversa. Estas fuentes pueden generar o absorber energía eléctrica.

Una fuente ideal de tensión es un elemento de circuito que mantiene una tensión fija entre sus dos terminales, independientemente de la corriente que fluya a través de dichos terminales. De la misma manera, una fuente ideal de corriente es un elemento de circuito que mantiene una misma corriente entre sus dos terminales, independientemente de la tensión entre los bornes de los mismos.

Las fuentes ideales de tensión y de corriente pueden subdividirse en fuentes independientes y dependientes.

Se entiende por fuente independiente aquella que establece una corriente o una tensión que no depende de otra corriente u otra tensión existente en otra parte del circuito. Por lo que, como su propio nombre indica, en una fuente dependiente su valor sí que depende de otra intensidad o tensión de un punto diferente del circuito.

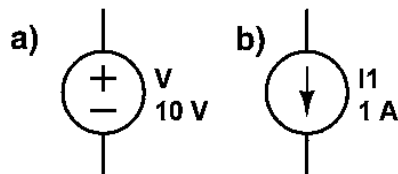


Figura 5: a) fuente independiente de tensión; b) fuente independiente de corriente

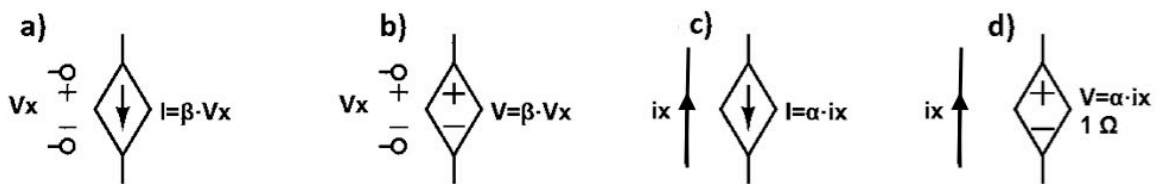


Figura 6: a) fuente de corriente dependiente de tensión; b) fuente de tensión dependiente de tensión; c) fuente de corriente dependiente de corriente; d) fuente de tensión dependiente de corriente.

3.2.3. Ley de Ohm y resistencia eléctrica

Hay que diferenciar dos tipos de elementos en un circuito: los activos y los pasivos. Los elementos activos son aquellos que modelan un dispositivo capaz de generar energía eléctrica, como es el caso de las fuentes. Los elementos pasivos, por el contrario, no generan electricidad, como por ejemplo las resistencias, las bobinas o los condensadores.

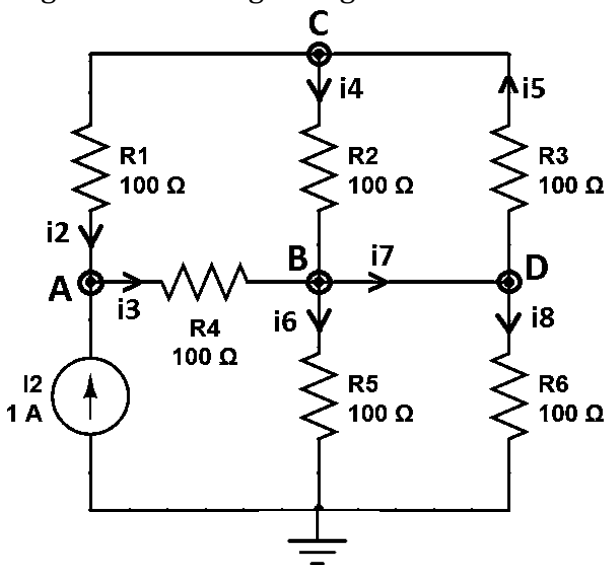
La Ley de Ohm, nombre en honor al físico alemán Georg Simon Ohm que estableció esta ley a principios del siglo XIX, es la relación algebraica existente entre la tensión entre los terminales de una resistencia y la corriente que la pasa, tal que $V=I \cdot R$. La resistencia eléctrica es la oposición que muestra un conductor a los electrones cuando intentan moverse a través de él. Sus unidades en el Sistema Internacional son los Ohmios y se representan con la letra griega Omega (Ω).

3.2.4. Leyes de Kirchhoff

Antes de enunciar las leyes de Kirchhoff es conveniente conocer algunos conceptos previos, como un nodo o un camino cerrado. Un nodo es el punto en el que se juntan dos o más elementos de circuito. Un camino cerrado o lazo es el camino que recorre los nodos de un circuito que empieza y acaba en el mismo nodo sin pasar a través de un nodo intermedio más de una vez.

Hay dos leyes, una asociada a las corrientes y otra a las tensiones, las cuales se entienden mejor con un ejemplo real.

La Ley de Kirchhoff de las corrientes es “la suma algebraica de todas las corrientes existentes en un nodo de un circuito es igual a cero”. Para hacer un uso práctico de esta ley hay que asignar a cada nodo un signo algebraico que indique la dirección de referencia. Asignar un signo positivo a una corriente que salga de un nodo implica que asignaremos un signo negativo a todas las corrientes que entren al mismo.



Nodo A: $i1 + i2 - i3 = 0$

Nodo B: $i3 - i6 - i7 = 0$

Nodo C: $i5 - i4 - i2 = 0$

Nodo D: $i7 - i5 - i8 = 0$

Figura 7: ejemplo de circuito para aplicar la Ley de Kirchhoff de las corrientes

La Ley de Kirchhoff de las tensiones dice “la suma algebraica de todas las tensiones alrededor de cualquier camino cerrado en un circuito es igual a cero”. Para utilizar esta Ley hay que fijar una tensión de referencia a cada tensión del camino cerrado.

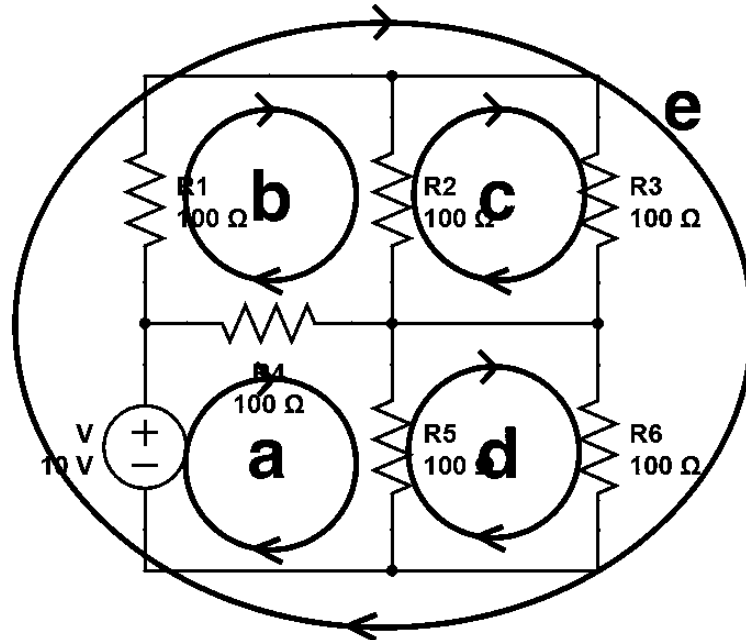


Figura 8: ejemplo de circuito para aplicar la Ley de Kirchhoff de las tensiones

Lazo A: $V_1 + V_2 - V_6$

Lazo B: $V_3 - V_4 - V_2 = 0$

Lazo C: $V_4 - V_5 = 0$

Lazo D: $V_6 - V_7 = 0$

Lazo E: $V_1 + V_3 - V_5 - V_7 = 0$

3.2.5. Asociación de resistencias

Los circuitos eléctricos, en muchas ocasiones, tienen una disposición de componentes que permiten simplificarlos haciendo modelos equivalentes. En este apartado se verán algunas posibilidades de asociación de resistencias, que son las que se utilizan en los algoritmos.

Se dice que dos elementos están en serie cuando sólo estos dos están conectados a un mismo nodo. La característica más importante que tienen estos elementos es que transportan la misma corriente. Por ello, se puede calcular una resistencia equivalente cuyo valor numérico sea la suma de las resistencias individuales. Esto se debe a la Ley de Kirchoff de las tensiones.

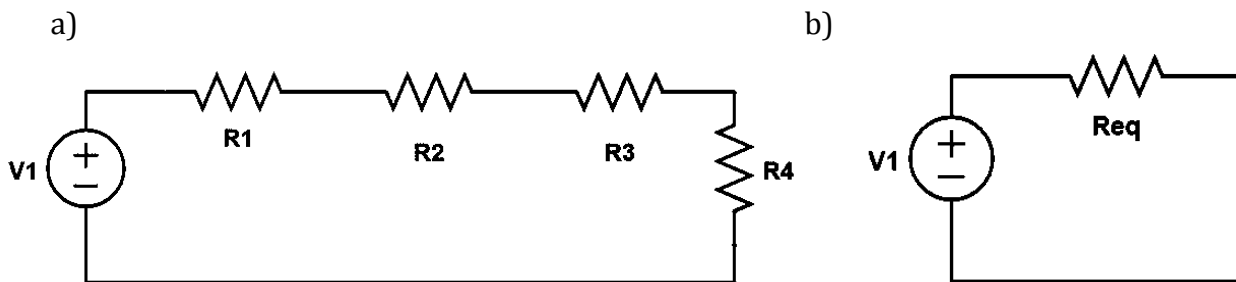


Figura 9: a) circuito con resistencias en serie; b) circuito equivalente del anterior

$$V = i \cdot R_1 + i \cdot R_2 + i \cdot R_3 + i \cdot R_4 = i \cdot (R_1 + R_2 + R_3 + R_4)$$

$$V = I \cdot R_{EQ} \text{ tal que } R_{EQ} = R_1 + R_2 + R_3 + R_4$$

Cuando dos elementos están en paralelo es cuando ambos están conectados directamente a una misma pareja de nodos. La característica que define esta conexión es que tienen la misma tensión en bornes de sus terminales. Si se aplica la Ley de Kirchoff de las corrientes y la Ley de Ohm, se puede simplificar el circuito obteniendo una resistencia equivalente entre estos dos terminales.

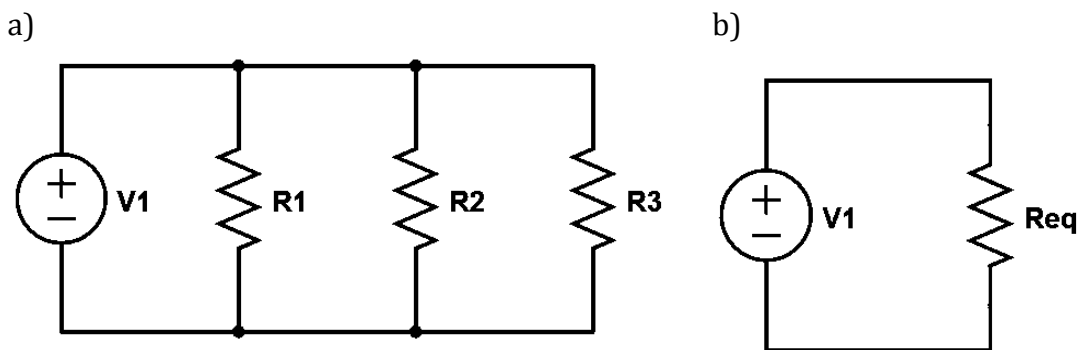


Figura 10: a) circuito con resistencias en paralelo; b) circuito equivalente al anterior

$$i = i_1 + i_2 + i_3 + i_4 \quad \text{Aplicando la Ley de Ohm } V = i_1 \cdot R_1 = i_2 \cdot R_2 = i_3 \cdot R_3$$

Sustituyendo y agrupando:

$$i = V \cdot \left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} \right) \rightarrow \frac{i}{V} = \frac{1}{R_{EQ}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$$

3.3. Descripción de problemas y soluciones propuestas

Una vez elegido el entorno de programación se tuvo que elegir el método de desarrollo. Se optó por hacer algoritmos basados en matrices usando sus propiedades. Hay varias vías implementadas de analizar circuitos, las cuales se combinaron dando lugar a las funciones que hacen posible el funcionamiento de la aplicación.

A continuación se expondrán dichas vías.

3.3.1 Método nodal

Es un método basado en análisis de circuitos por nudos. Estos circuitos están compuestos por resistencias, bobinas o condensadores, fuentes independientes y dependientes de corriente.

Este método es de aplicación inmediata en el análisis de circuitos RLC con fuentes independientes de corriente J ya que las ecuaciones tienen una estructura adecuada para el análisis de circuitos por ordenador. Estas corrientes se recogerán en un vector cuyos términos son igual a la suma de los valores de las fuentes independientes de corriente entrantes en cada uno de los nodos, excepto en el de referencia. Por otro lado, los valores RLC se recogerán en una matriz de admitancias nodales Y donde los términos de la diagonal principal son la suma de las admitancias conectadas al nodo y los términos restantes es la suma, cambiada de signo, de las admitancias conectadas entre dos nodos distintos al de referencia. Esta matriz siempre será simétrica. Finalmente se definirá el vector de las tensiones nodales V . Se cumplirá $Y \cdot V = J$, por lo que para obtener la matriz de tensiones V habrá que despejarla, es decir, V será el producto de la inversa de la matriz de admitancia nodal Y y el vector de corrientes J , según la expresión: $V = Y^{-1} \cdot J$.

En el caso de tratarse de un circuito de tres nodos fundamentales la estructura de la matriz y los vectores sería la siguiente:

$$\begin{bmatrix} Y_{11} & Y_{12} & Y_{13} \\ Y_{21} & Y_{22} & Y_{23} \\ Y_{31} & Y_{32} & Y_{33} \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} J_1 \\ J_2 \\ J_3 \end{bmatrix}$$

Este método se puede adaptar de tal manera que se pueda incorporar fuentes de corriente controladas por tensión añadiendo al vector J estas intensidades. La consecuencia de esta modificación es que se pierde la simetría de la matriz de admitancia nodal.

La forma genérica de expresar la fuente de corriente dependiente de tensión entre dos nodos es la siguiente:

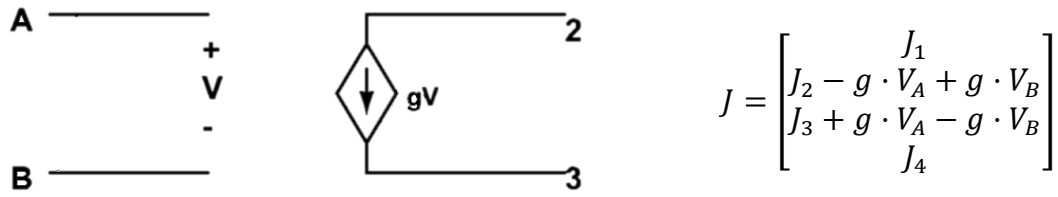


Figura 11: fuente dependiente de tensión entre dos nodos

Este método también es aplicable a fuentes de corriente dependientes de corriente teniendo la precaución de transformarlas previamente en fuentes de corriente dependientes controladas por tensión, es decir, expresando la propia dependencia en función de las tensiones nodales.

Un ejemplo podría ser analizar un circuito que tuviera, entre otros componentes, un transistor bipolar.

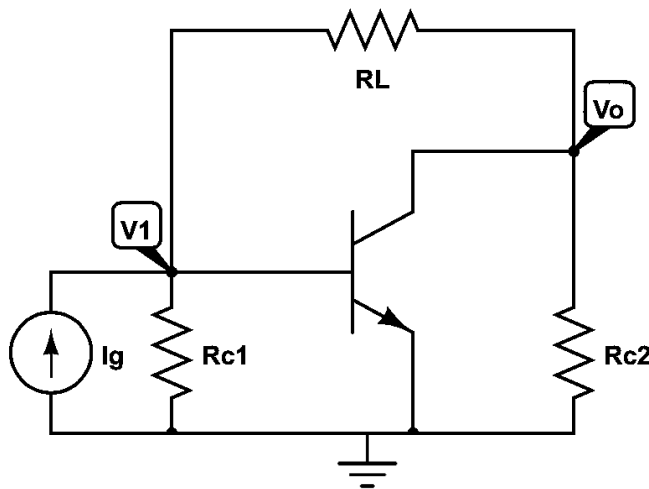


Figura 12: circuito con un transistor bipolar

Teniendo en cuenta cómo es el modelo en pequeña señal de dicho transistor:

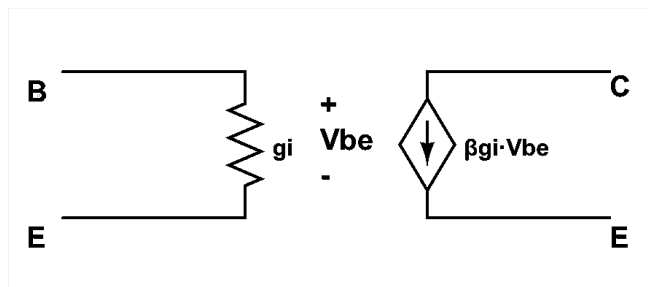


Figura 13: modelo en pequeña señal de un transistor bipolar

Transformando y reordenando el circuito, se obtiene el siguiente cuya matriz de admitancias y vectores de corriente y tensión son los siguientes:

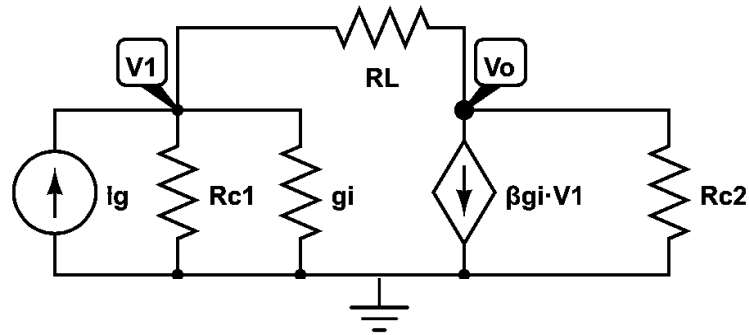


Figura 14: modelo en pequeña señal del circuito de la figura 12.

$$\begin{bmatrix} \frac{1}{R_{c1}} + \frac{1}{g_i} + \frac{1}{R_L} & -\frac{1}{R_L} \\ \beta \cdot g_i - \frac{1}{R_L} & \frac{1}{R_{c2}} + \frac{1}{R_L} \end{bmatrix} \begin{bmatrix} V_1 \\ V_0 \end{bmatrix} = \begin{bmatrix} I_g \\ 0 \end{bmatrix}$$

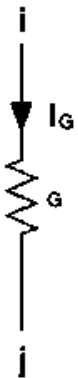
3.3.2. Método nodal modificado

Este método es una generalización del método nodal y permite el análisis de casi cualquier circuito electrónico, lo único que hay que hacer es añadir ecuaciones a las matrices y vectores vistos en el anterior apartado.

No se verá el desarrollo de todas las posibilidades que se podrían analizar pero sí que se citarán la mayoría y se explicarán los circuitos que tienen trascendencia para el desarrollo de la aplicación.

Al igual que en el Método Nodal, las resistencias no se definen en la matriz por su valor óhmico aunque se podría hacer pero implicaría añadir una ecuación adicional que aumentara las dimensiones de la matriz. Por lo tanto, generalmente, se definen por sus conductancias, es decir, por la facilidad que ofrece el material al paso de la corriente eléctrica. En otras palabras, la conductancia es la propiedad inversa de la resistencia, su símbolo es G y su unidad es el Siemens.

La forma de definir la submatriz de admitancias de una única resistencia, como el circuito de la figura 15 es la siguiente:



$$T = \begin{pmatrix} V_i \text{ respecto al nodo } i & V_j \text{ respecto al nodo } i \\ V_i \text{ respecto al nodo } j & V_j \text{ respecto al nodo } j \end{pmatrix}$$

$$T = \begin{matrix} & V_i & V_j \\ \begin{matrix} i \\ j \end{matrix} & \begin{pmatrix} G & -G \\ -G & G \end{pmatrix} \end{matrix}$$

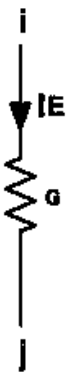
Figura 15: resistencia entre dos nodos

Si se quiere añadir la corriente I_G habría que añadir una fila y una columna correspondiente a una ecuación adicional, en este caso $G \cdot V_i - G \cdot V_j - I_G = 0$ por lo que la matriz T quedaría:

$$T = \begin{matrix} & V_i & V_j & I_G \\ \begin{matrix} i \\ j \\ EA \end{matrix} & \begin{pmatrix} G & -G & 1 \\ -G & G & -1 \\ G & -G & -1 \end{pmatrix} \end{matrix}$$

Esta construcción de matriz es aplicable a condensadores y bobinas, teniendo en cuenta su transformada de Laplace, donde la transformada de la conductancia del transformador es Cs y de la bobina $(Ls)^{-1}$. Aunque, en el caso del inductor, se suele añadir en serie una fuente de tensión dependiente de corriente, fuente de transconductancia, para evitar términos de s^{-1} en la matriz T. Esto implica añadir una ecuación adicional.

Un elemento muy importante que añade el Método Nodal Modificado es el de las fuentes independientes de tensión. Para este caso no sólo hay que añadir una ecuación a la matriz T, sino que también hay que hacerlo al vector de corrientes W. Puesto que se trabaja con conductancias e intensidades, la fuente no se tratará como la tensión entre sus terminales sino como la corriente que pasa por los mismos, como se puede ver en el circuito de la figura 16.



$$T = \begin{matrix} & V_i & V_j & I_E \\ \begin{matrix} i \\ j \\ EA \end{matrix} & \begin{pmatrix} & & 1 \\ & & -1 \\ 1 & -1 & \end{pmatrix} \end{matrix} \quad W = \begin{bmatrix} \dots \\ \dots \\ E \end{bmatrix}$$

Figura 16: resistencia entre dos nodos con tensión entre ellos

De la misma forma, añadiendo estas filas y columnas en la matriz T y en el vector W, se pueden hacer cualquier fuente de corriente o tensión dependiente de las mismas.

Este método también permite analizar circuitos con amplificadores operacionales, bipuertos, circuitos con acoplamiento magnético...

Ya vistas la formación de estas matrices en algunos casos particulares, es conveniente ver su composición de forma genérica:

$$T = \begin{pmatrix} Y_R & \vdots & B \\ \cdots & \cdots & \cdots \\ C & \vdots & D \end{pmatrix} \quad X = \begin{bmatrix} V_n \\ \cdots \\ I \end{bmatrix} \quad W = \begin{bmatrix} J \\ \cdots \\ E \end{bmatrix} \quad \text{tal que} \quad TX = W$$

Donde T se compone por cuatro submatrices:

Y_R es la matriz reducida de la admitancia total de orden n para circuitos de n+1 nodos. Incluye valores de conductancias G, tansconductancias g y admitacias de los condensadores.

B es la matriz de los coeficientes de las variables añadidas por las ecuaciones adicionales. Su valor siempre es de ± 1 ó 0 excepto en el caso de las fuentes de corriente dependientes de corriente.

C es la matriz que incluye los coeficientes de las EA de las tensiones de nodo.

D es la matriz que incluye los coeficientes de las EA de las variables adicionales.

En cuanto al vector X está compuesto por el vector V_n que está compuesto por las tensiones nodales y por el vector I que es el vector de las intensidades adicionales.

Finalmente, el vector W, que es el resultado del producto de la matriz T por el vector X, está compuesto por el vector de las fuentes de corriente, J, y por el vector E que está compuesto por las tensiones adicionales.

Se tomará como ejemplo el mismo circuito del apartado anterior, figura 14, del modelo en pequeña señal de un transistor bipolar, con la diferencia de que la fuente de corriente dependiente de tensión ahora es dependiente de corriente.

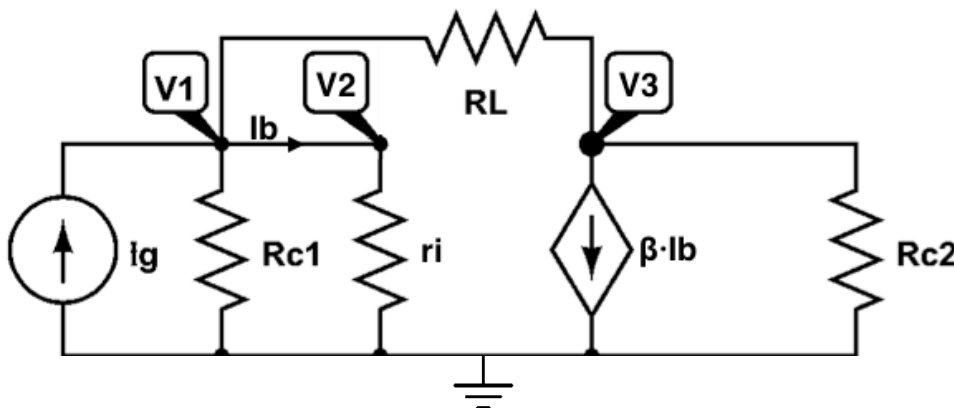


Figura 17: circuito equivalente de pequeña señal con fuente de corriente dependiente de corriente.

En primer lugar se define la matriz reducida de la admitancia total Y_R y el vector de tensiones nodales, sin tener en cuenta el de referencia.

$$Y_R = \begin{bmatrix} \frac{1}{Rc1} + \frac{1}{RL} & 0 & -\frac{1}{RL} \\ 0 & \frac{1}{ri} & 0 \\ -\frac{1}{RL} & 0 & \frac{1}{Rc2} \end{bmatrix} \quad V_n = \begin{bmatrix} V1 \\ V2 \\ V3 \end{bmatrix}$$

En este caso, la EA que hay que añadir tiene que ver con la corriente I_b , la cual es entrante al nodo 2 y saliente del nodo 1.

$$T = \begin{bmatrix} \frac{1}{Rc1} + \frac{1}{RL} & 0 & -\frac{1}{RL} & 1 \\ 0 & \frac{1}{ri} & 0 & -1 \\ -\frac{1}{RL} & 0 & \frac{1}{Rc2} & \beta \\ 1 & -1 & 0 & 0 \end{bmatrix} \quad X = \begin{bmatrix} V1 \\ V2 \\ V3 \\ I_b \end{bmatrix} \quad W = \begin{bmatrix} I \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Poniendo valores a los componentes del circuito y operando con las matrices se puede obtener fácilmente los valores de las tensiones en los nodos fundamentales.

3.3.3. Método ZBUS

Es un método muy parecido al nodal, se basa en definir una matriz de impedancias que tendrá que ser nxn siendo n el número de nodos a analizar, sin contar con el nodo de referencia.

El formato de la matriz de impedancias de un circuito de 3 nodos será:

$$Y_{BUS} = \begin{bmatrix} Y_{NODO1} & -Y_{1-2} & -Y_{1-3} \\ -Y_{2-1} & Y_{NODO2} & -Y_{2-3} \\ -Y_{3-1} & -Y_{3-2} & Y_{NODO3} \end{bmatrix}$$

Y_{NODO1} es la suma de todas las admitancias que llegan al nodo 1.

Y_{NODO2} es la suma de todas las admitancias que llegan al nodo 2.

Y_{NODO3} es la suma de todas las admitancias que llegan al nodo 3.

Y_{1-2} es la suma de todas las admitancias que se encuentran en las ramas que comparten los nodos 1 y 2. Por lo que $Y_{1-2} = Y_{2-1}$.

Y_{1-3} es la suma de todas las admitancias que se encuentran en las ramas que comparten los nodos 1 y 3. Por lo que $Y_{1-3} = Y_{3-1}$.

Y_{2-3} es la suma de todas las admitancias que se encuentran en las ramas que comparten los nodos 2 y 3. Por lo que $Y_{2-3} = Y_{3-2}$.

Una vez definida la matriz Y_{BUS} se hace la matriz inversa para conseguir la matriz Z_{BUS} .

Por otro lado, se define la matriz I_{NODO} que será de $1 \times n$, es decir, tendrá tantas columnas como nodos tenga el circuito:

$$I_{NODO} = \begin{bmatrix} I_{NODO1} \\ I_{NODO2} \\ I_{NODO3} \end{bmatrix}$$

I_{NODO1} es la suma de las corrientes entrantes al nodo 1 y resta de las salientes al mismo.

I_{NODO2} es la suma de las corrientes entrantes al nodo 2 y resta de las salientes al mismo.

I_{NODO3} es la suma de las corrientes entrantes al nodo 3 y resta de las salientes al mismo.

Finalmente, para obtener las tensiones de los nodos de referencia se multiplicarán las matrices I_{NODO} y Z_{BUS} obteniendo como resultado una matriz con dichas tensiones. El formato de la citada matriz será el que se muestra a continuación siendo cada elemento la tensión en cada nodo de referencia:

$$V = \begin{bmatrix} V_{NODO1} \\ V_{NODO2} \\ V_{NODO3} \end{bmatrix}$$

3.3.4. Método elegido

Primero se fijó la finalidad de la aplicación con sus objetivos. Ya que se parte de cero, se decidió analizar cualquier circuito con fuentes de tensión y de corriente independientes de corriente continua por lo que está orientado para el uso de resistencias, pero este aspecto ya se verá mejor en el apartado 7.2. *Líneas futuras*.

Por lo tanto, después de estudiar y analizar los posibles métodos para el análisis de circuitos por ordenador se optó por usar los aspectos provechosos de cada uno de tal manera que se llegó a un método basado en el ZBUS y en el Método Nodal. Para los aspectos que no plantean estos métodos, como las fuentes de tensión independientes, se investigó un algoritmo matemático que atendiera a todos los casos ya que el Método Nodal Modificado, por ejemplo, es bastante complejo de diseñar para cualquier composición de circuito que se resuelva automáticamente ya que hay que ver la ecuación adicional y cómo añadirla a la matriz T y al vector W. Se llegó a la conclusión de que éste método es cómodo para un uso personal cuando se conoce en qué se basa y cómo se crean las matrices, no para que una aplicación se basara en él para hacer los cálculos automáticamente.

Los aspectos tomados de los métodos explicados son el uso de matrices, de signos en función del sentido de las fuentes... Pero se verá más detalladamente en el siguiente apartado.

4. DISEÑO DE LA APLICACIÓN

En este apartado se pretende determinar las características funcionales de la aplicación.

En primer lugar se describirán los conocimientos de Matlab necesarios para desarrollar la aplicación. Una vez vista esta parte se detallarán las funciones encargadas del análisis nodal del circuito propiamente dicho, su dependencia de los elementos que le integran y de la cantidad de los mismos. Estas funciones se verán de forma incremental según su complejidad, empezando con la de un nodo fundamental hasta llegar a cuatro nodos fundamentales.

Después se verán los algoritmos encargados de simplificar el circuito, es decir, simplificación de fuentes y detección de resistencias en serie para hacer una única.

Finalmente se explicará el funcionamiento global de la aplicación, con todas las partes ya explicadas integradas, con la dependencia entre ellas.

4.1. Fundamentos de Matlab

Matlab es una herramienta muy potente y extensa que permite multitud de aplicaciones pero, en este caso, la funcionalidad que buscamos está ligada a las matrices y sus propiedades. A continuación se verá formas de definir una matriz y ciertas funciones que sirven para trabajar con las mismas.

4.1.1. Formas de definir una matriz

Para definir una matriz A en Matlab, tal que $A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{bmatrix}$ habrá que definir A de la siguiente forma: $A=[1 \ 2 \ 3; 2 \ 3 \ 4; 3 \ 4 \ 5];$

En Matlab la separación entre elementos de la misma fila se hace mediante espacios o comas y la separación entre columnas con punto y coma como se ha visto anteriormente.

Análisis de circuitos mediante Matlab

Si se quiere definir una matriz como parte de otra, es decir, una submatriz, se hace de la siguiente manera:

$B=A(:,n)$; Siendo B la matriz formada por los elementos de la columna n.

$C=A(m,:)$; Siendo C la matriz formada por los elementos de la fila m.

Otra opción es poner los elementos de una matriz en una única columna:

$D=A(:)$; Siendo D dicha matriz.

Ejemplo práctico a raíz de la matriz A definida en este apartado:

$$B=A(:,3); \rightarrow B = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \quad C=A(1,:); \rightarrow C = [1 \quad 2 \quad 3] \quad D=A(:); \rightarrow D = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

4.1.2. Funciones de matrices

Algunas funciones de interés para el desarrollo de las funciones se ven en la siguiente tabla.

Función	Descripción
size(A)	Devuelve el tamaño de la matriz A
B=A'	Trasponer una matriz A en una matriz B
B=inv(A)	Hacer la matriz inversa de la matriz A en B

Tabla 3: funciones útiles de matrices en Matlab

4.2. Análisis nodal

En primer lugar, se verán las definiciones de los términos necesarios para poder desarrollar la explicación del método nodal que está recogidos en la siguiente tabla aunque algunos ya se han visto en el apartado 3.2. *Conceptos básicos de circuitos eléctricos*.

Nombre	Definición
Nodo	Punto en el que se unen más de un elemento del circuito.
Nodo fundamental	Nodo en el que se unen tres o más elementos del circuito.
Nodo de referencia	Nodo fundamental del circuito con el que se comparan los demás.
Camino	Sucesión de elementos básicos adyacentes.
Rama	Camino que conecta dos nodos.

Tabla 4: términos importantes para el análisis nodal

Una vez aclaradas las palabras que se van a utilizar y su significado hay que elegir el nodo de referencia. Normalmente se opta por escoger el que mayor número de ramas tenga que, en los libros de ejercicios de electrónica, suele ser el inferior. Después se definen las tensiones de nodo que son la diferencia de tensión entre el nodo de referencia y los otros nodos del circuito.

El siguiente paso es plantear las ecuaciones de tensión de nodo. Se escribe la corriente que sale de cada rama conectada a un nodo distinto al de referencia. Estas corrientes se escriben en función de las tensiones aplicando la ley de Ohm ($V=I \cdot R$) y luego se suman igualándolas a cero.

Finalmente, hay que resolver el sistema de ecuaciones para obtener las tensiones de los nodos fundamentales.

4.3. Funciones del análisis nodal

Estas funciones están desarrolladas para un único elemento por cada rama, es decir, en el caso de que el circuito original tenga varias resistencias en serie en una misma rama, el valor introducido como parámetro de entrada será el equivalente de las mismas. 0, en el caso de fuente de tensión independiente en serie con una resistencia, se hará el equivalente de Thévenin-Norton por lo que los siguientes algoritmos verán una resistencia en paralelo con una fuente de corriente independiente.

4.3.1. Un nodo fundamental

El primer paso es empezar con el algoritmo encargado de obtener la tensión de un único nodo. Esta función tiene como cabecera $(V) = \text{UnNodo}(R, I_i, \text{ind})$ donde los parámetros de entrada son R que es la matriz de resistencias entre el nodo a calcular y el de referencia, I_i es la matriz o vector de intensidades, ind es un parámetro con el valor de tensión de una fuente independiente en una de las ramas en el caso que la hubiera, sino su valor es cero y V es el parámetro de salida cuyo valor es el de la tensión en el nodo fundamental que se quiere calcular. Como se puede deducir, si hay una fuente de tensión independiente entre el nodo y referencia, ind , el parámetro de salida V será igual a éste.

El código de este algoritmo es muy sencillo y de poca longitud, a continuación se presenta el pseudocódigo:

Función $V = \text{UnNodo}(R, I_i, \text{ind})$

```
Y ← 0
I ← 0
Si matriz R no está vacía entonces
    Para i ← 1 hasta longitud de R con pasos de 1 hacer
        Y ← Y + (1/Valor de R en la posición i)
    FinPara
FinSi
Si matriz Ii no está vacía entonces
    Para i ← 1 hasta longitud de Ii con pasos de 1 hacer
        I ← I + Ii en la posición i
    FinPara
FinSi
Si ind es igual a 0 entonces
    V ← 1/Y · I
Sino
    V ← ind
FinSi
FinFunción
```

4.3.2. Dos nodos fundamentales

Este algoritmo es similar al anterior pero aumentando los parámetros de entrada ya que el valor devuelto por la función es de mayor magnitud. Por lo tanto, este algoritmo calcula las dos tensiones referentes a los nodos fundamentales distintos al de referencia. En este caso hay que tener en cuenta las ramas entre cada uno de los nodos y el de referencia y las que pueda haber entre los mismos.

Por lo tanto, la cabecera de la función se define como $V = \text{DosNodos}(R1, R2, R12, I, \text{ind1}, \text{ind2}, S12)$ donde el parámetro de salida ahora es un vector y no un escalar ya que contiene los valores de las tensiones de los dos nodos. En este caso, R1 es la matriz de resistencias entre el nodo1 y el de referencia, R2 entre el nodo2 y el de referencia, R12 la matriz de resistencias entre los nodos 1 y 2, ind1 contiene el valor de la fuente independiente de tensión entre el nodo1 y masa y el ind2 hace la misma función pero entre el nodo2 y masa, en el caso de que hubiera alguna de estas fuentes. En este algoritmo aparece un nuevo parámetro de entrada, S12, que recoge el valor de una fuente independiente de tensión, en el caso de que existiera, entre los nodos 1 y 2 considerándose positivo si el terminal positivo está en el nodo2 y el negativo en el nodo1.

La metodología que sigue es muy similar a la del algoritmo de un nodo, como se puede ver observando la cabecera. A continuación se muestra el pseudocódigo de la función:

```

Función  $V = \text{DosNodos}(R1, R2, R12, I, \text{ind1}, \text{ind2}, S12)$ 
     $Y1 \leftarrow 0$ 
     $Y2 \leftarrow 0$ 
     $Y12 \leftarrow 0$ 
    Si S12 es cero entonces
        Si matrizR1 no está vacía entonces
            Para  $i \leftarrow 1$  hasta longitud de R1 con pasos de 1 hacer
                 $Y1 \leftarrow Y1 + (1/\text{Valor de R1 en la posición } i)$ 
            FinPara
        FinSi
        /*Se repite el bucle anterior para R2 y para R12
         $Y1 \leftarrow Y1 + Y12$ 
         $Y2 \leftarrow Y2 + Y12$ 
         $Y \leftarrow [Y1 \ -Y12; \ -Y12 \ Y2]$ 
         $V \leftarrow \text{inversa}(Y) \cdot I$ 
    Sino
         $V \leftarrow$  valor devuelto por la función SuperNodo
    FinSi
    Si ind1 no es 0 entonces
        Y en la posición (1,1)  $\leftarrow 1$ 
        Y en la posición (1,2)  $\leftarrow 0$ 
        I en la posición 1  $\leftarrow \text{ind1}$ 
         $V \leftarrow \text{inversa}(Y) \cdot I$ 

```

```
FinSi
/*Mismo bucle para el caso de ind2
FinFunción
```

4.3.3. Tres nodos fundamentales

La complejidad del algoritmo y su longitud va aumentando exponencialmente según aumenta el número de nodos tal y como era de prever. Este algoritmo calcula las tensiones correspondientes a los tres nodos fundamentales del circuito teniendo en cuenta que en cada rama que los interconecta puede haber cualquier elemento.

Siguiendo con la línea de explicación se procede a analizar la cabecera y sus elementos (V)= TresNodos(R1, R2, R3, R12, R13, R23, I, ind1, ind2, ind3, S12, S13, S23). Las resistencias que forman cada rama van en las matrices R1, para las que están entre el nodo1 y masa, R2, lo mismo pero de la rama del nodo2 a referencia, R3, ídem pero con las del nodo3, R12, resistencias en ramas entre el nodo1 y el nodo2, R13, entre el nodo1 y el nodo3 y R23, entre el nodo2 y el nodo3. La matriz de intensidades I sigue siendo igual que en los otros algoritmos, es decir, en la primera posición almacena las corrientes que afectan al nodo1, siendo positivas las entrantes y negativas las salientes, en la posición dos las corrientes del nodo2 y en la tercera posición las del nodo3. Los indicadores ind1, ind2, ind3 almacenan el valor de la fuente independiente de tensión entre el nodo correspondiente y referencia siendo ind1 el que guarda el voltaje entre el nodo1 y referencia, ind2 entre el nodo2 y referencia e ind3 entre el nodo3 y referencia. Finalmente, la presencia de fuentes independientes de tensión en ramas que están entre dos nodos distinto al de referencia se almacena en las variables S12, S13 y S23 siendo las fuentes entre los nodos 1 y 2, 1 y 3, y 2 y 3 respectivamente.

En esta parte del código se comprueba si hay incongruencias tales como que haya varias fuentes de tensión que marquen en un mismo punto distintas tensiones, es decir, si, por ejemplo, hay una fuente entre el nodo1 y masa de 10V, hay otra entre el nodo3 y masa de 5V y hay otra entre los nodos 1 y 3 de 3V con el terminal positivo de la misma sobre el nodo3 el nodo3 valdría, por un lado, 13V y, por el otro, 5V. En estos casos aparece en la pantalla un mensaje de error informando del mismo y el valor devuelto por la función es un cero.

Además hay que destacar que la diversidad de circuitos que se pueden dar con tres nodos fundamentales ha hecho que se tengan en cuenta muchos factores y, eso mismo, ha desencadenado en que se usen más las propiedades de las matrices y el análisis de circuitos a mano que el descrito en los métodos del apartado 3 de este documento.

A continuación se verá un resumen del código de la función en formato de pseudocódigo:

Función V=TresNodos (R1, R2, R3, R12, R13, R23, ind1, ind2, ind3, S12, S13, S23)

Y1←0

Y2←0

Y3←0

Y12←0

Y13←0

Y23←0

Si matrizR1 no está vacía entonces

Para i← 1 hasta longitud de R1 con pasos de 1 hacer

Y1←Y1+(1/Valor de R1 en la posición i)

FinPara

FinSi

/*Se repite el bucle anterior para R2, R3, R12, R13 y R23

Y1←Y1+Y12+Y13

Y2←Y2+Y12+Y23

Y3←Y3+Y13+Y23

Y←[Y1 -Y12 -Y13; -Y12 Y2 -Y23; -Y13 -Y23 Y3]

V←inversa(Y)·I

/*Se analizan los posibles casos de combinaciones de fuentes

Si S12 no es 0 e ind1 es 0 e ind2 es 0 entonces

Si S13 no es 0 e ind3 es 0 entonces

Si S23 no es 0 entonces

*/Comprobamos incongruencias

Si S13 es S12+S23 entonces

*/Construimos una matriz M y la reducimos

*/guardando el resultado en Vaux

V en la posición 1←Vaux en la posición (1,2)

V en la posición 2←V en la posición 1+S12

V en la posición 3←V en la posición 2+S23

Sino

Escribir que no es posible distinta tensión en un mismo punto

FinSi

Sino

*/Se construye otra matriz M similar y se asigna la tensión

FinSi

Sino

*/Se comprueban más posibles casos de fuentes

FinSi

Sino

/*Se comprueban otras combinaciones de fuentes

FinSi

4.3.4. Cuatro nodos fundamentales

El algoritmo correspondiente a cuatro nodos fundamentales es igual que el de tres con la diferencia de que tiene más parámetros de entrada y más casos de combinaciones entre fuentes y resistencias, por lo que no se detallará un pseudocódigo. El valor devuelto, en este caso, será la misma matriz V y tendrá cuatro columnas, una por cada nodo fundamental del circuito.

La cabecera de la función es $(V)= \text{CuatroNodos}(R1, R2, R3, R4, R12, R13, R14, R23, R24, R34, I, \text{ind1}, \text{ind2}, \text{ind3}, \text{ind4}, S12, S13, S14, S23, S24, S34)$. Comparando con el caso de tres nodos, se añade $R4$ que es la matriz formada por las resistencias que forman cada rama conectadas entre el cuarto nodo y el nodo de referencia, $R14, R24$ y $R34$ las resistencias entre el primer y cuarto nodo, segundo y cuarto nodo y tercer y cuarto nodo respectivamente. En el vector de intensidades I se añade una cuarta columna con la suma de las corrientes entrantes menos las salientes al nodo4. Se añade el indicador de presencia de fuente independiente de tensión entre el nodo4 y referencia, ind4 , así como la presencia de las mismas entre los nodos uno y cuatro, dos y cuatro y tres y cuatro almacenadas en las matrices $S14, S24$ y $S34$.

También, como en el apartado anterior, se comprueba, en cada caso, incongruencias entre fuentes de tensión mostrando un mensaje informativo en la ventana de comandos de Matlab y devolviendo un cero en la matriz de tensiones, el valor devuelto de la función.

4.4. Funciones simplificadoras

Los algoritmos que se explicarán en este apartado corresponden con encargados de procesar la información proporcionada por el usuario mediante ficheros, para depurarla y ordenarla. Básicamente hay dos que, a su vez, usan otras para conseguir el resultado esperado que es reducir el circuito en nodos fundamentales ya que la información leída de los ficheros está dada simplemente en nodos. Éstas se llaman desde el programa principal que se explicará en el apartado 4.5 de este documento.

4.4.1. Localización de resistencias en serie

El primer filtro por el que pasa el programa es el de detectar dos o más resistencias en serie para reducir el número de nodos, dejando el circuito bastante más simplificado. Todo ello se hace ordenando los elementos de la matriz de resistencias en función de los nodos a los que están conectadas las mismas.

La cabecera del algoritmo es (resistencias ,fuentesCorriente ,fuentesTension) = serie(resistenciasLeido ,fuentesCorrienteLeido ,fuentesTensionLeido). Como se puede deducir, las matrices de salida son iguales que las de entrada pero procesadas. Para entender mejor el funcionamiento se analizará el pseudocódigo del algoritmo:

Función (resistencias,fuentesCorriente,fuentesTension) =
serie(resistenciasLeido,fuentesCorrienteLeido,fuentesTensionLeido)

```
/*Se ordenan las matrices resistenciasLeido, fuentesCorrienteLeido,
/*fuentesTensionLeido y se almacenan en las matrices resistencias,
/*fuentesCorriente y fuentes de tensión, respectivamente.
```

```
/*Se recorre la matriz de resistencias hasta encontrar nodos comunes distintos al
/*de referencia.
```

```
Para j← 1 hasta longitud de resistencias con pasos de 1 hacer
```

```
/*Se inicializa este parámetro auxiliar que vale uno cuando no ha
/*encontrado resistencias en serie y cero en caso contrario.
```

```
noSerie← 1
```

```
Para i← 1 hasta longitud de resistencias con pasos de 1 hacer
```

```
Si la resistencia del primer bucle para y la del segundo bucle tienen
un sólo no en común entonces
```

```
noSerie← comprobarSerie(resistencias comparadas)
```

```
/*La función comprobarSerie se verá al terminar el
/*pseudocódigo.
```

```
nodoComun← nodoComun localizado
```

```
Si noSerie=0 entonces
```

```
auxR← resistencia equivalente
```

```
borrar las resistencias que forman la equivalente
```

```
FinSi
```

```

    FinSi
    Si noSerie=0 entonces
    /*Se recorren todas las matrices y se decrementan los nodos
    /*mayores o iguales al común
        Para y←1 hasta longitud de resistencias con pasos de 1
        hacer
            Si la resistencia tiene un nodo >= a 1 entonces
                Si el nodo1 de la resistencia >= al nodo común
                hacer
                    Nodo1(resistencia) ←Nodo1(resistencia)-1
                FinSi
                Si el nodo2 de la resistencia >= al nodo común
                hacer
                    Nodo2(resistencia) ←Nodo2(resistencia)-1
                FinSi
            Sino si nodo1(resistencia)=0 o 1 y nodo2>= al nodo
            común hacer
                Nodo2(resistencia) ←Nodo2(resistencia)-1
            Sino si nodo2(resistencia)=0 o 1 y nodo1>= al nodo
            común hacer
                Nodo1(resistencia) ←Nodo1(resistencia)-1
            FinSi
        FinSi
    FinPara
    resistencias←[resistencias;auxR ]
    resistencias←ordenar(resistencias)
    FinSi
    FinPara
    FinPara

    /*Se hace lo mismo con las matrices de fuentes independientes de corrientes y de
    /*tensión.
FinFunción

```

Como se ha visto en el pseudocódigo, se usan dos funciones dentro de él. La función para ordenar se detallará en el apartado 4.4.3 ya que se usa en varias ocasiones. Pero el algoritmo comprobarSerie es una función auxiliar de serie.m que se ha externalizado para evitar repeticiones innecesarias de código.

La cabecera de comprobarSerie es (noSerie) = comprobarSerie (i, resComp1, resComp2, resistencias, fuentesCorriente, fuentesTension). El valor devuelto sería un booleano, hablando en otros entornos de programación, valiendo únicamente uno o cero. En cuanto a los parámetros de entrada, está i, que es el punto o nodo en común que tienen las dos resistencias a comparar, resComp1 y resComp2 que son las propias resistencias que se comparan, y tienen los tres parámetros que las caracterizan: su valor y los dos nodos a los que están conectadas, y las matrices de resistencias, fuentesCorriente, fuentesTension usadas en el algoritmo que invoca esta función.

El algoritmo recorre las tres matrices, la de resistencias, la de fuentes independientes de tensión y la de fuentes independientes de corriente, para detectar si hay algún otro elemento en el nodo que tienen en común las dos resistencias a comparar. En el caso de que sí que lo haya, se descartará que están en serie y, en caso contrario, se confirmará que lo están. Por lo tanto, la variable `noSerie` valdrá cero sólo en el caso de que sí que estén en serie las dos resistencias, en cualquiera de los demás, será un uno.

4.4.2. Localización de fuentes independientes de tensión en serie con resistencias

Una vez reducido el circuito a equivalentes de resistencias en serie, el siguiente paso es hacer más cómodo y sencillo el análisis haciendo transformación de fuentes, aunque se haya llamado al algoritmo equivalente de Thévenin-Norton. Este nombre viene porque ambos procesos se han quedado en la mente de la mayoría de la gente que lo ha estudiado alguna vez en su vida. Ambos métodos sirven para simplificar un circuito y, para hacer el paso entre ellos, hay que usar la transformación de fuentes. A continuación se explicará brevemente en qué consisten estos equivalentes y la transformación de fuentes.

La técnica de la transformación de fuentes permite sustituir una fuente de tensión en serie con una resistencia por una fuente de corriente en paralelo con la misma resistencia, o viceversa. La equivalencia entre fuentes se basa en aplicar la ley de Ohm, de tal manera que, si se tiene una fuente de tensión en serie con una resistencia, el valor de la fuente de corriente será la tensión de la fuente de tensión dividida entre el valor de la resistencia.

Por otro lado, el equivalente de Thévenin es un circuito formado por una fuente independiente de tensión en serie con una resistencia, con el que se sustituye otro circuito más complejo formado por una interconexión de fuentes y resistencias. El equivalente de Norton está compuesto por una fuente de corriente independiente en paralelo con la resistencia equivalente del circuito. El paso de un equivalente a otro se puede hacer mediante transformación de fuentes.

La cabecera del algoritmo descrito en este apartado es `(resistencias, fuentesCorriente, fuentesTension) = equivalenteTheveninNorton(RIn, IIn, VIn)` siendo los parámetros de entrada las matrices de resistencias, fuentes de corriente y fuentes de tensión sin resistencias en serie y, los parámetros de salida las mismas matrices con las transformaciones de fuentes correspondientes.

El funcionamiento del algoritmo sigue una línea bastante intuitiva, en primer lugar se recorre, a la vez, la matriz de resistencias y la matriz de fuentes independientes de tensión para analizar las conexiones de los elementos de cada matriz. Es decir, se contrastan los nodos de una resistencia con los de todas las fuentes de tensión hasta haber probado todas las posibles combinaciones de elementos. El fin de esta comparación es encontrar los casos, si los hubiera, de fuentes de tensión en serie con una resistencia.

Para corroborar que están realmente en serie, o sea, que no hay ningún otro elemento en el nodo que tienen en común, se usa la función `buscarNodo` que busca el número de veces que se repite un nodo, de tal manera que si se repite más de dos veces, una por cada elemento, quiere decir que no están en serie.

Si se da el caso buscado, se hace la transformación de fuentes y se reordenan y actualizan todos los nodos de todos los componentes del circuito que se pudieran ver afectados por este cambio.

La función comentada anteriormente, `buscarNodo`, consiste en un algoritmo formado por tres bucles `for`, uno para cada una de las matrices de los diferentes elementos del sistema, que comparan los nodos de los mismos con el nodo que tienen en común la fuente independiente de tensión y la resistencia. La cabecera de la función es `(veces) = buscarNodo(nodoComun, tensiones, corrientes, resistencias, longV, longI, longR)` donde el valor devuelto, `veces`, es el número de casos que se repite el nodo a comparar, introducido como parámetro de entrada. Otros parámetros de entrada son las tres matrices de componentes y las longitudes de las mismas.

4.4.3. Algoritmo ordenar

Este algoritmo está basado en un método sencillo de ordenación conocido como algoritmo de la burbuja. Consiste en comparar pares de elementos adyacentes en un array y, si están desordenados, intercambiarlos hasta que estén todos ordenados. El pseudocódigo correspondiente al algoritmo de la burbuja para ordenar los elementos de un array de menor a mayor sería así:

```
Para i←0 hasta la longitud del array -1 en pasos de 1 hacer
    Para j←0 hasta la longitud del array-1-i en pasos de 1 hacer
        Si el array en la posición (j+1)< el array en la posición j entonces
            Variable auxiliar←el array en la posición (j+1)
            El array en la posición (j+1) ←el array en la posición j
            El array en la posición j←variable auxiliar
        FinSi
    FinPara
FinPara
```

Adaptando este método de ordenación a la funcionalidad que se quiere dar en esta aplicación, se hace la función `ordenar` que tiene como cabecera `(elementoOrdenado)=ordenar (elementoAOrdenar, res)` donde los parámetros de entrada son el elemento a ordenar, que es la matriz que corresponda en cada caso, y la variable `res` que vale uno en el caso de que dicho elemento sea la matriz de resistencias y cero en el caso contrario. El valor devuelto, `elementoOrdenado`, corresponde, como su propio nombre indica, a la matriz pasada como parámetro de entrada ordenada. El caso de las resistencias se estudia ya que éstas no tienen polaridad por lo que se ordenan también los propios nodos considerándose el positivo el mayor de los dos

4.5. Funcionamiento global

El funcionamiento global del sistema parte del desarrollo de un script, traductor.m, que usa los algoritmos y funciones detallados anteriormente para conseguir la funcionalidad esperada. Consiste en coger los datos del circuito a analizar proporcionados por el usuario, ordenarlos, clasificarlos y realizar el cálculo de las tensiones de los nodos de referencia. Este algoritmo se puede dividir en siete pasos:

1. Se leen los ficheros con la información, uno para las resistencias que forman el circuito, resistencias.txt, otro para las fuentes independientes de tensión, fuentesTension.txt, y otro para las fuentes independientes de corriente, fuentesCorriente.txt. Estos datos se guardan en tres matrices.
2. Se simplifica el circuito localizando las resistencias en serie mediante los algoritmos detallados en apartados anteriores.
3. Se hace otra simplificación buscando fuentes independientes de tensión en serie con resistencias usando el algoritmo equivalenteTheveninNorton.m.
4. Se crean e inicializan las matrices y vectores que recogerán las resistencias y fuentes de corriente y tensión que hay en cada nodo fundamental.
5. Se agrupan las resistencias y fuentes en función de los nodos a los que están conectados y se guardan en los parámetros inicializados en el paso anterior.
6. Se comprueba el número de nodos fundamentales del circuito a analizar y, en función de cuántos sean, se usa el algoritmo que corresponda de los cuatro posibles de las cuatro funciones de análisis nodal.
7. Finalmente, se escribe el resultado de las tensiones de los nodos fundamentales en el fichero tensiones.txt.

5. RESULTADOS

En este capítulo se hará un repaso de las pruebas a las que ha sido sometida la aplicación según se ha ido desarrollando la misma. No procede poner las pruebas realizadas ya que se basan en resolver un circuito a mano y estudiar cómo parametrizarlo para poder solucionarlo de una forma sistemática, se han estudiado cerca de doscientos ejemplos.

En primer lugar, se empezó estudiando circuitos con un único nodo, buscando qué particularidades podrían tener unos sobre otros como por ejemplo varias resistencias en paralelo o en serie, fuentes de tensión entre el nodo de referencia y el fundamental o fuentes de tensión en serie con resistencias, fuentes de corriente independientes en cada sentido... Para poder pasar estos resultados a ordenador fue necesario partir de circuitos que contaran únicamente con fuentes independientes de corriente y resistencias o algunos en los que poder hacer transformación de fuentes y que quedaran circuitos con sólo estos dos tipos de componentes.

Por lo tanto, haciendo el análisis de circuitos con nudos se vio que lo primero que había que solucionar era cuando había varias resistencias en paralelo, cómo tratarlas y dónde hacerlo. Se decidió que irían todas ellas en una matriz y que al principio del algoritmo se haría su paralelo para obtener la equivalente. En cuanto a las resistencias en serie, se decidió hacer la suma de ellas a mano antes de introducir sus valores en la aplicación, era un asunto que se resolvió más adelante.

Una vez conseguido para circuitos de un nodo, se escaló para circuitos de dos, tres y cuatro nodos añadiendo únicamente más parámetros de entrada en la ecuación siendo éstos resistencias entre los nodos fundamentales y el de referencia, resistencias entre nodos fundamentales y fuentes independientes de corriente entre cualquier punto.

El siguiente paso fue incorporar la transformación de fuentes, es decir, la detección de resistencia en serie con fuente de tensión. Al principio se hizo un algoritmo que aplicaba únicamente la Ley de Ohm dentro de la propia función que calcula la tensión de los nodos. Luego se hizo uno más complejo que, además de localizar estos casos, hacía la transformación y recategorizaba los nodos para posteriormente pasarlo a la función que correspondiera de las ya citadas. Para entenderlo bien lo mejor es verlo con un ejemplo. Si había un circuito con una fuente de tensión entre el nodo 1 y referencia y una resistencia entre el nodo 1 y el nodo 2, de tal manera que estuvieran en serie, se hacía la transformación estando tanto la fuente de corriente independiente como la resistencia paralela a la misma entre los nodos 1 y el de referencia, por lo que todos los nodos mayores de uno, que en este caso son todos, había que reducirlos en uno.

Esta función devuelve a la aplicación principal las tres matrices una vez hecha la transformación, si procede hacerla, para posteriormente analizar si se trata de un circuito de uno, dos, tres o cuatro nodos y usar el algoritmo correspondiente para calcular las tensiones de los nodos fundamentales.

El ampliar la variedad de circuitos añadiendo fuentes independientes de tensión entre nodo fundamental y el de referencia a priori no tendría por qué haber supuesto un desafío ya que este tipo de fuentes lo que hacen es fijar la tensión en el nodo por lo que en cálculos teóricos y manuales es una ventaja ya que reducen mucho la cantidad de ecuaciones. Pero en el caso de diseñar un algoritmo matemático y automático para cualquier posible caso no fue tan sencillo, se tuvo que cambiar la manera de tratar las matrices, es decir, se tuvo que añadir parámetros nuevos de entrada a las funciones que calculan las tensiones de los nodos para que indicaran la presencia de estos elementos. Se creó un parámetro por cada nodo de referencia del algoritmo tratándose de cuatro elementos en el caso de cuatro nodos, tres en el caso de tres nodos y así sucesivamente. Se fijó un criterio de signos, de tal manera que si el parámetro era positivo quería decir que el terminal positivo estaba sobre el nodo fundamental y el negativo sobre el de referencia. Las matrices se tratarían igual al principio de la función pero al final, antes de crear el valor devuelto, es decir, las tensiones en los nodos fundamentales, se comprobaban estos parámetros y se recalculaban las tensiones resultado.

A partir de este momento, se estudió cómo incorporar a los algoritmos la posibilidad de que hubiera fuentes independientes de tensión entre dos nodos fundamentales. En el análisis de circuitos, lo que se hace es considerar un supernodo, es decir, fijar la diferencia de tensión entre dos nodos y dejar uno en función de otro y aquí se hizo un algoritmo que hiciera prácticamente lo mismo pero automáticamente.

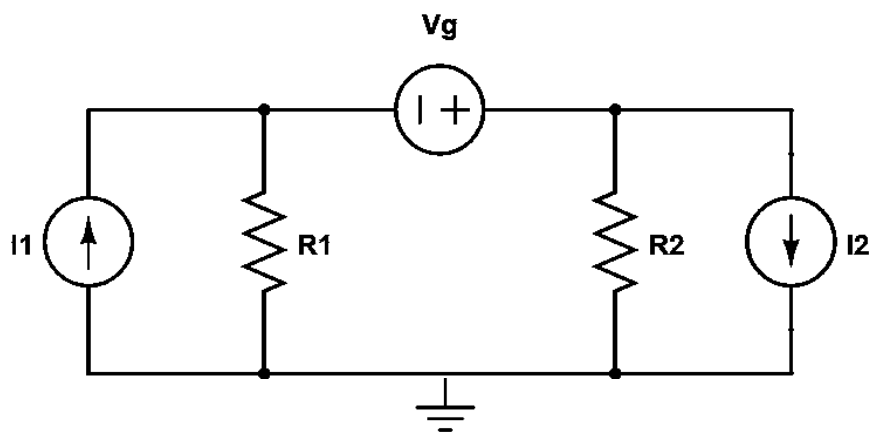


Figura 18: circuito con un supernodo.

No se podrían utilizar ecuaciones de nudos ya que no hay ninguna resistencia entre los nodos 1 y 2, habría que hacerlo por corrientes o usando el concepto de supernodo como se ve a continuación:

$$\frac{V_1}{R_1} + \frac{V_2}{R_2} = I_1 - I_2, \quad V_2 - V_1 = V_g \rightarrow V_2 = V_1 + V_g$$

$$\frac{V_1}{R_1} + \frac{(V_1 + V_g)}{R_2} = I_1 - I_2 \rightarrow V_1 \cdot \left(\frac{1}{R_1} + \frac{1}{R_2} \right) = I_1 - I_2 - \frac{V_g}{R_2}$$

Dando valores a los componentes y despejando se obtendría V_1 y, sustituyendo este valor en la relación de V_1 , V_2 y V_g el de V_2 . Por lo que un circuito con dos nodos fundamentales con un supernodo hace que tenga una única ecuación en lugar de dos, facilitando los cálculos del circuito.

Para el caso de uno o dos nodos no fue muy complejo, puesto que las posibilidades son bastante limitadas, pero en el caso de tres y, sobretodo, de cuatro nodos se tuvo que buscar un circuito de ejemplo para ver qué dependencias tenían y así poder formar adecuadamente las matrices. Ya que todos estos análisis se realizaban a mano, se buscaron circuitos de prueba para no realizar todos los cálculos cada vez que se cambiaba un elemento y se hizo un fichero de Matlab donde estaban todas las pruebas con los resultados que tenían que dar.

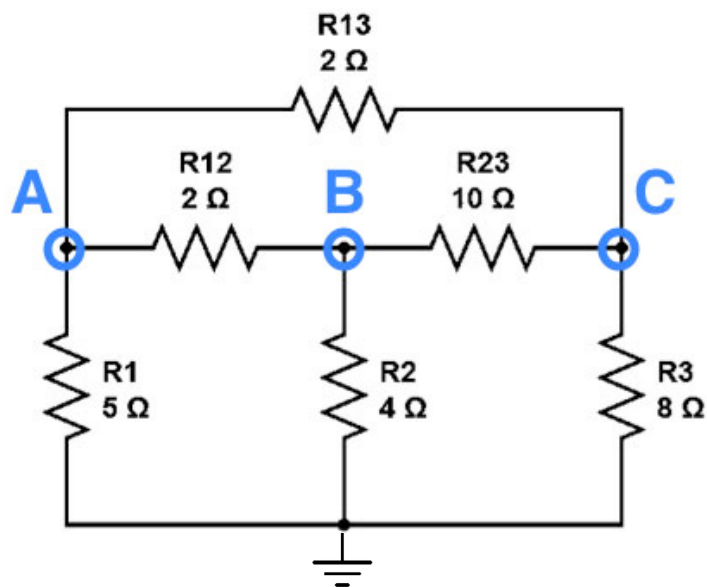


Figura 19: circuito genérico de tres nodos fundamentales

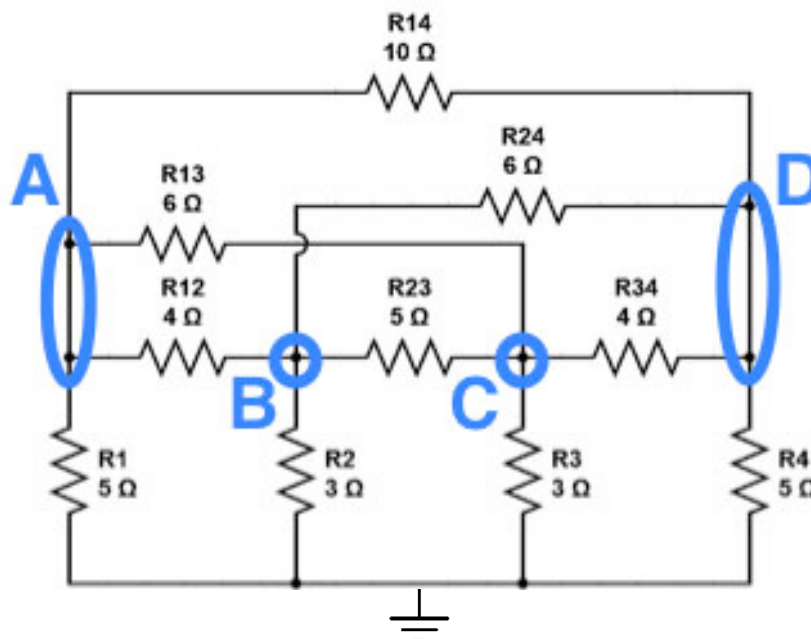


Figura 20: circuito genérico de cuatro nodos fundamentales

Análisis de circuitos mediante Matlab

Esta fue la parte que conllevó más trabajo ya que se hicieron siete circuitos para el caso de tres nodos en fuentes entre nodos fundamentales, además de la genérica, es decir, sin fuentes, y las tres de fuentes entre nodo fundamental y de referencia con posibilidad de que hubiera más de una entre ellos. Para tres nodos un total de quince circuitos pero para el caso de cuatro nodos este número aumentó exponencialmente. Se realizaron quince circuitos para los casos de generadores de tensión entre nodo fundamental y de referencia, cincuenta y seis para casos entre nodos fundamentales y algunos casos genéricos, es decir, se hizo un total de setenta y tres circuitos.

```
1  %% Prueba de 1 nodo:
2  %Cabecera: function [ V ] = UnNodo(R, Ii, ind)
3  %1. Sin fuentes de tensión (V=280)
4  UnNodo ([80;(20+50)], [7.5], 0)
5  %2. Con transformación de fuentes: (V=16)
6  UnNodo( [5,20,4], [CambioFuente(25,5);3], 0)
7  %3. Con fuentes de tensión (V=7)
8  UnNodo( [10,4], [4;6;9], 7)
9
10 %% Prueba de 2 nodos:
11 %Cabecera: function [V]= DosNodos(R1, R2, R12, I, ind1, ind2,S)
12 %1. Sin fuentes de tensión (Valores: V1=60 V2=10)
13 DosNodos( [60,15], [2], [5], [15;-5], 0, 0, 0)
14 %2. Fuente de tensión entre nodo y referencia
15 %2.1. Nodo 1 y referencia (Valores: V1=50 V2=60.8696)
16 DosNodos([40], [50,100], [5], [30;4], 50, 0, 0)
17 %2.2. Nodo 2 y referencia (Valores: V1=60.8696 V2=50)
18 DosNodos([50,100], [40], [5], [4;0], 0, 50, 0)
19 %3. Dos fuentes de tensión entre nodo y referencia (Valores: V1=50 V2=30)
20 DosNodos([40], [20], [5], [450;30000], 50, 30, 0)
21 %4. Fuente de tensión entre nodos (Valores: V1=1.9565 V2=11.9565)
22 DosNodos([2,3], [2,5], [0], [8;2], 0, 0, 10)
23 %5. Fuente de tensión entre nodos y referencia (Valores: V1= V2=)
24 %% Prueba de 3 nodos:
25 %Cabecera: function [V]= TresNodos(R1, R2, R3, R12, R13, R23, I, ind1, ind2, ind3, S12, S13, S23)
26 %1. Sin fuentes de tensión (Valores: V1=13.5205 V2=9.3991 V3=11.0075)
27 TresNodos(2,6, [7;10], 4, 12, 3, [0;0;3], 0, 0, 0, 0, 0)
28 %2. Fuente de tensión entre nodo y referencia
29 %2.1. Nodo 1 y referencia (Valores: V1=5 V2=5.11898 V3=7.7677)
30 TresNodos(2,6, [7,10], 4, 12, 3, [0;0;3], 5, 0, 0, 0, 0, 0)
31 %2.2. Nodo 2 y referencia (Valores: V1=2.2358 V2=5 V3=7.3583)
32 TresNodos(2,6, [7,10], 4, 12, 3, [0;0;3], 0, 5, 0, 0, 0, 0)
33 %2.3. Nodo 3 y referencia (Valores: V1=5.2963 V2=3.9877 V3=5)
34 TresNodos(2,6, [7,10], 4, 12, 3, [3;0;0], 0, 0, 5, 0, 0, 0)
Ln 1 Col 1
```

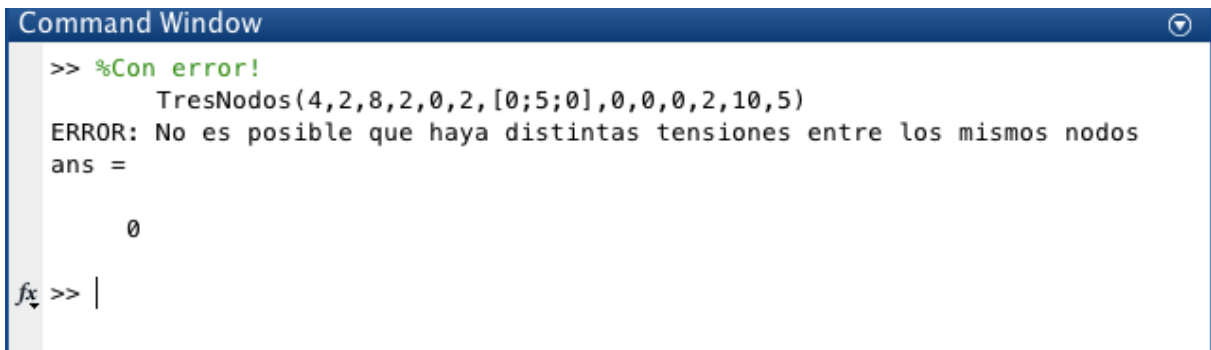
Figura 21: captura de pantalla del script de pruebas, parte 1

```
76 %% Prueba de 4 nodos
77 %Cabecera: function [V]= CuatroNodos(R1, R2, R3, R4, R12, R13, R14, R23, R24, R34, I, ind1, ind2, ind3, S12, S13, S23, S34)
78 %1. Sin fuentes de tensión (Valores: V1=9.0694 V2=6.5853 V3=13.9525 V4=3.7821)
79 CuatroNodos(4,6,8, [2,3,5,6,10,6,4], [3;0;6;-2], 0, 0, 0, 0, 0, 0, 0, 0, 0)
80 %2. Fuente de tensión entre nodo y referencia
81 %2.1. Fuente entre nodo1 y ref.
82 %(Valores: V1=4 V2=2.3238 V3=2.0664 V4=1.4498)
83 CuatroNodos(0,6,8,2,3,5,6,10,6,4, [0;0;0;0], 4, 0, 0, 0, 0, 0, 0, 0, 0)
84 %2.2. Fuente entre nodo2 y ref.
85 %(Valores: V1=2.9736 V2=6 V3=2.4945 V4=1.9562)
86 CuatroNodos(4,0,8,2,3,5,6,10,6,4, [0;0;0;0], 0, 6, 0, 0, 0, 0, 0, 0, 0)
87 %2.3. Fuente entre nodo3 y ref.
88 %(Valores: V1=3.2571 V2=3.0726 V3=8 V4=2.8199)
89 CuatroNodos(4,6,0,2,3,5,6,10,6,4, [0;0;0;0], 0, 0, 8, 0, 0, 0, 0, 0, 0)
90 %2.4. Fuente entre nodo4 y ref.
91 %(Valores: V1=0.9478 V2=0.9994 V3=1.1696 V4=2)
92 CuatroNodos(4,6,8,0,3,5,6,10,6,4, [0;0;0;0], 0, 0, 0, 2, 0, 0, 0, 0, 0)
93 %2.5. Fuente entre nodo1 y ref. y nodo2 y ref.
94 %(Valores: V1=4 V2=6 V3=2.8910 V4=2.2056)
95 CuatroNodos(0,0,8,2,3,5,6,10,6,4, [0;0;0;0], 4, 6, 0, 0, 0, 0, 0, 0, 0)
96 %2.6. Fuente entre nodo1 y ref. y nodo3 y ref.
97 %(Valores: V1=4 V2=3.4325 V3=8 V4=2.9896)
98 CuatroNodos(0,6,0,2,3,5,6,10,6,4, [0;0;0;0], 4, 0, 8, 0, 0, 0, 0, 0, 0)
99 %2.7. Fuente entre nodo1 y ref. y nodo4 y ref.
100 %(Valores: V1=4 V2=2.4729 V3=2.2923 V4=2)
101 CuatroNodos(0,6,8,0,3,5,6,10,6,4, [0;0;0;0], 4, 0, 0, 2, 0, 0, 0, 0, 0)
102 %2.8. Fuente entre nodo2 y ref. y nodo3 y ref.
103 %(Valores: V1=4.3939 V2=6 V3=8 V4=3.4452)
104 CuatroNodos(4,0,0,2,3,5,6,10,6,4, [0;0;0;0], 0, 6, 8, 0, 0, 0, 0, 0, 0)
105 %2.9. Fuente entre nodo2 y ref. y nodo4 y ref.
106 %(Valores: V1=2.9854 V2=6 V3=2.5142 V4=2)
```

Figura 22: captura de pantalla del script de pruebas, parte 2

Al igual que para añadir fuentes entre un nodo fundamental y el de referencia, se añadieron parámetros que indicaran la presencia de generadores de tensión entre nodos fundamentales, uno por cada posibilidad. Aumentando en gran cantidad las cabeceras de las funciones.

Además, se tuvo que comprobar casos de posibles incongruencias como que hubiera una fuente en una rama de un nodo y en otra rama del nodo otra fuente con una tensión distinta. Para estos casos el valor devuelto es un cero y sale un mensaje de error en la consola de comandos de Matlab. La siguiente imagen es un ejemplo, no es posible que entre los nodos 1 y 2 haya una fuente de 2V, entre 1 y 3 de 10V y entre 2 y 3 de 5V.



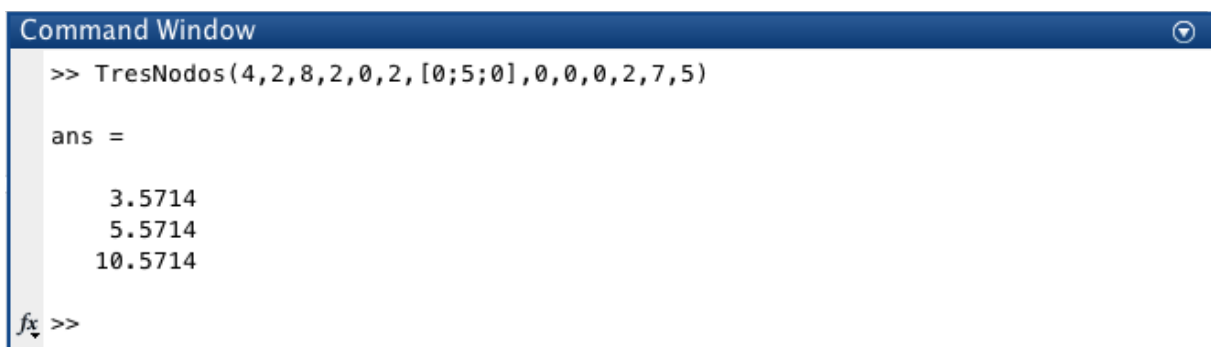
```

Command Window
>> %Con error!
      TresNodos(4,2,8,2,0,2,[0;5;0],0,0,0,2,10,5)
ERROR: No es posible que haya distintas tensiones entre los mismos nodos
ans =
     0
fx >> |

```

Figura 23: captura de pantalla de mensaje de error en la ventana de comandos

Para que fuera correcto el ejemplo anterior, el valor de la tensión entre el nodo 1 y el 3 debería ser de 7V, se comprueba con la siguiente imagen:



```

Command Window
>> TresNodos(4,2,8,2,0,2,[0;5;0],0,0,0,2,7,5)

ans =

     3.5714
     5.5714
    10.5714
fx >>

```

Figura 24: captura de pantalla con el resultado del algoritmo de TresNodos en la ventana de comandos

El siguiente paso fue implementar la lectura de ficheros, lo cual se hizo directamente desde el programa principal, traductor.m. Después de cargar estos datos se procesaron para clasificarlos y ponerlos en el formato de las cabeceras de los algoritmos que calculan las tensiones y se descubrió que no tienen por qué estar ordenados. Se hizo una función auxiliar que coloca las filas de las matrices en orden ascendente según sus nodos, la explicación del algoritmo está más detallada en el apartado 4.4.3. *Algoritmo ordenar* de este documento.

Una vez hechas las pruebas bien, estas matrices cargadas de ficheros se pasaron primero por la función de transformación de fuentes y, posteriormente, por las de cálculo de tensión de los nodos fundamentales.

En último lugar, se buscó una solución para localizar y reducir circuitos con resistencias en serie. Para ello se partió de las características que tienen que tener: no puede haber nada más conectado al nodo que hay entre ellas, esto implica buscar que no haya ninguna otra resistencia, fuente de corriente o tensión. Si esto ocurre se hace una resistencia cuyo valor es la suma de las otras dos y que esté entre el nodo más pequeño de los tres y el intermedio. Además también hay que disminuir en uno el valor de todos los nodos mayores iguales que el intermedio sin contar con el de la resistencia que se acaba de reducir.

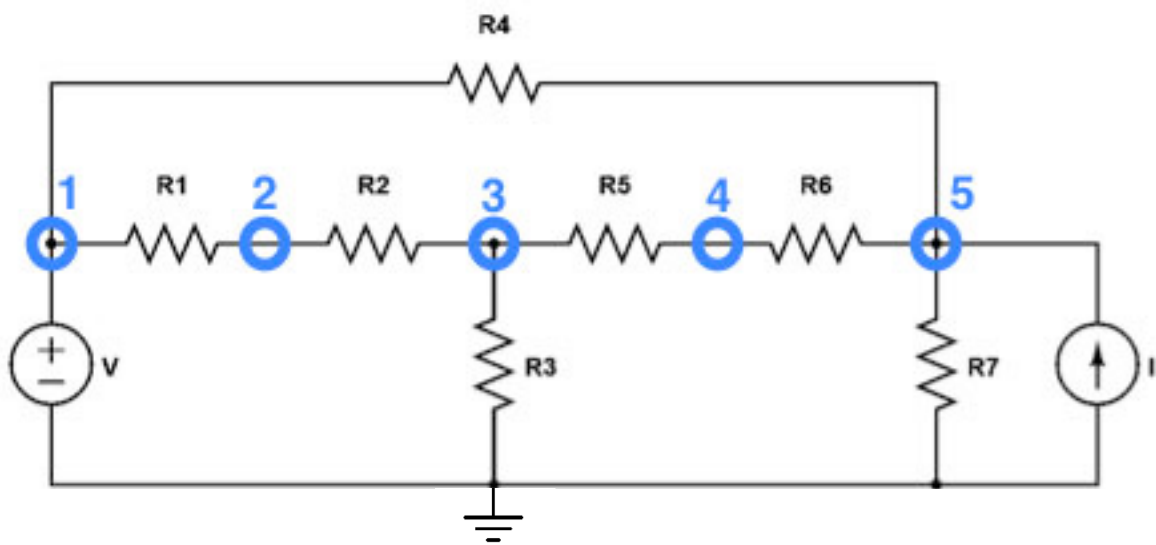


Figura 25: circuito con resistencias en serie entre nodos fundamentales

El circuito de la Figura 25 tiene cinco nodos, pero no son fundamentales ya que entre los nodos 1 y 3 hay dos resistencias en serie y ocurre lo mismo entre los nodos 3 y 5. El algoritmo desarrollado procesa el circuito obteniendo el de la Figura 26, es decir, el circuito equivalente con únicamente tres nodos fundamentales, cuyas tensiones serán el resultado proporcionado por el programa.

En el circuito de la Figura 26, el valor de la resistencia $Req1$ es igual a la suma de las resistencias $R1$ y $R2$, y está situada entre los nodos 1 y 2 por lo que hay que reducir todos los nodos, habiendo un total de cuatro nodos. Ocurre lo mismo con la resistencia $Req2$, cuyo valor es la suma de $R5$ y $R6$, y ahora está situada entre los nodos 2 y 3.

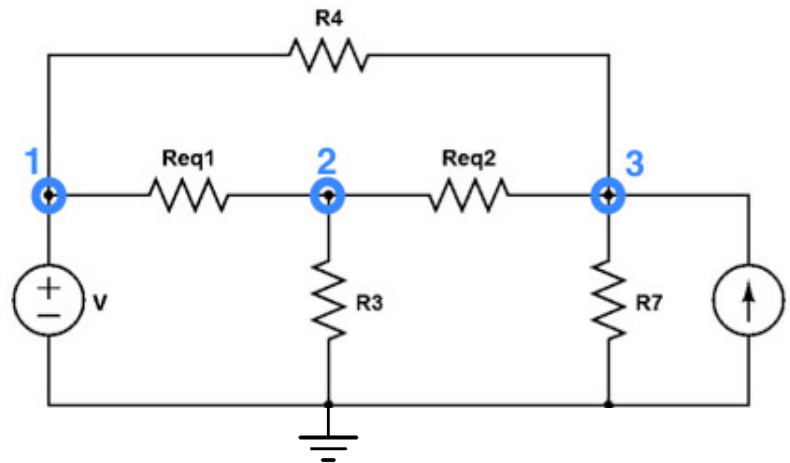


Figura 26: circuito equivalente al de la Figura 25

Este desarrollo hizo que se añadiera una peculiaridad al algoritmo de ordenar. Cuando el elemento a ordenar fuera la matriz de resistencias, además de ordenarse por filas también se haría por columnas, de tal manera que el primer nodo a los que está conectada la resistencia siempre fuera el menor de los dos y que, a partir de ahí, estuvieran ordenadas en orden ascendente de nodos.

Una vez hechas las pruebas correctamente de este algoritmo, se incluyó en el principal de la aplicación y es el primer filtro a pasar, después se pasa por la transformación de fuentes y, finalmente, se agrupan las resistencias, generadores de corriente y tensión para poder pasarlos como parámetros a los algoritmos de cálculo de las tensiones en los nodos de referencia.

El siguiente ejemplo muestra uno de los casos más completos ya que cuenta con una fuente de tensión entre nodos fundamentales, varios casos de resistencias en serie y un generador de tensión en serie con unas resistencias:

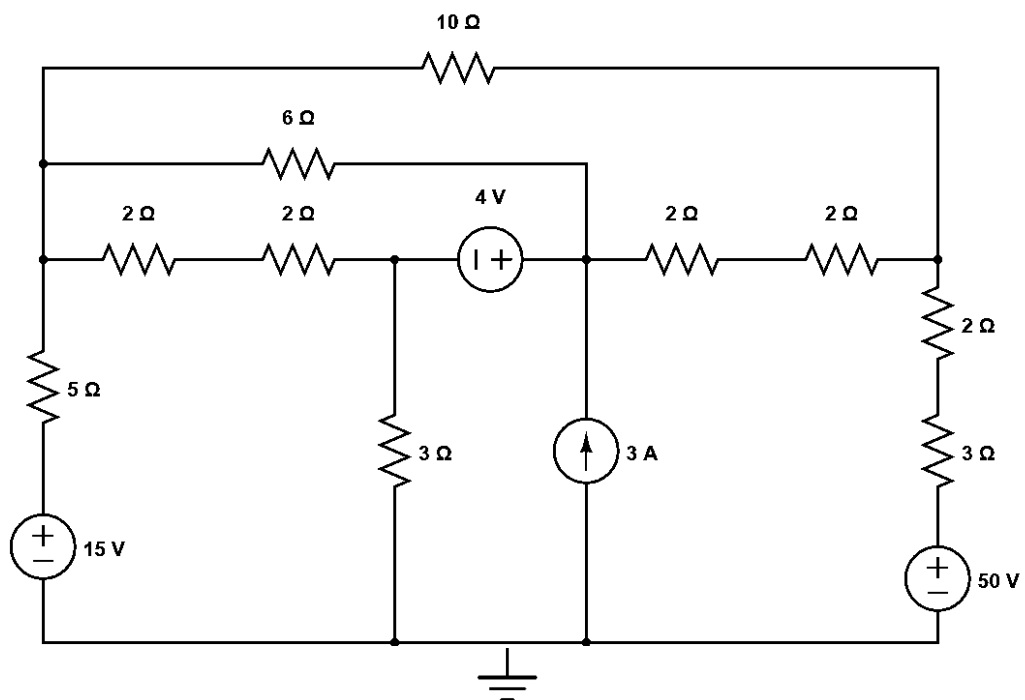


Figura 27: circuito de cuatro nodos fundamentales con resistencias en serie y supernodo

Análisis de circuitos mediante Matlab

Se comprueba las tensiones que tiene que dar en cada nodo fundamental teóricamente, teniendo en cuenta el supernodo que hay entre los nodos B y C, que hay resistencias en serie y que se puede hacer transformación de fuentes. Teniendo en cuenta estos elementos, el análisis se realiza sobre el circuito de la Figura 28.

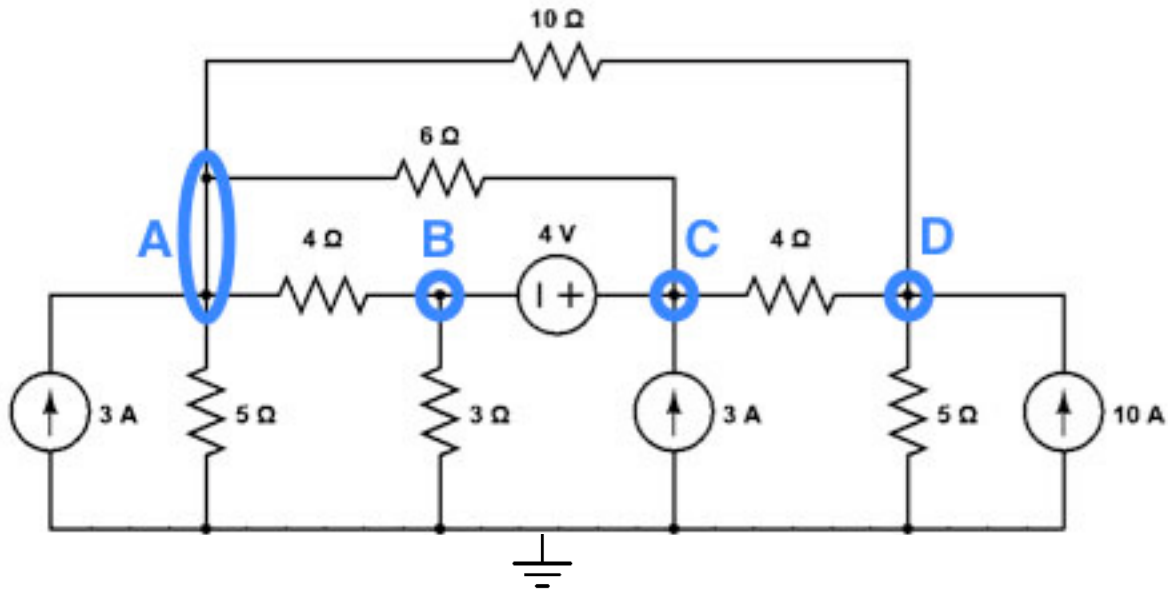


Figura 28: circuito equivalente al de la Figura 27

$V_C = V_B + 4$ Ecuación correspondiente al nodo C.

$$\text{Nodo A: } \frac{V_A}{5} + \frac{V_A - V_B}{4} + \frac{V_A - V_C}{6} + \frac{V_A - V_D}{10} = 3 \rightarrow V_A \cdot \frac{43}{60} - V_B \cdot \frac{5}{12} - \frac{V_D}{10} = \frac{11}{3}$$

$$\text{Nodo B: } \frac{V_B}{3} + \frac{V_B - V_A}{4} + \frac{V_C - V_A}{6} + \frac{V_C - V_D}{4} = 3 \rightarrow -V_A \cdot \frac{5}{12} + V_B - \frac{V_D}{4} = \frac{4}{3}$$

$$\text{Nodo D: } \frac{V_D}{5} + \frac{V_D - V_A}{10} + \frac{V_D - V_C}{4} = 10 \rightarrow -\frac{V_A}{10} - \frac{V_B}{4} + V_D \cdot \frac{11}{20} = 11$$

Se utiliza Matlab para reducir las ecuaciones usando la función rref, como se puede observar en la Figura 29, y así se obtienen las tensiones correspondientes al nodo A, B y D. Para ello hay que escribir las ecuaciones en forma de matriz:

$$\begin{pmatrix} \frac{43}{60} & -\frac{5}{12} & -\frac{1}{10} & \frac{11}{3} \\ \frac{5}{12} & 1 & -\frac{1}{4} & \frac{4}{3} \\ -\frac{1}{10} & -\frac{1}{4} & \frac{11}{20} & 11 \end{pmatrix} - \text{Reducida} \rightarrow \begin{pmatrix} 1 & 0 & 0 & 10,6015 \\ 0 & 1 & 0 & 17,3649 \\ 0 & 0 & 1 & 31,4570 \end{pmatrix}$$

```

Command Window
>> M=[43/60 -5/12 -1/10 11/3;
      -5/12 1 -1/4 4/3;
      -1/10 -1/4 11/20 11];
>> V=rref(M)

V =

    1.0000    0    0    19.6015
         0    1.0000    0    17.3649
         0    0    1.0000    31.4570

fx >> |
    
```

Figura 29: captura de pantalla de la reducción de la matriz del circuito de la Figura 28

Por lo que, aplicando la ecuación del supernodo se obtienen los cuatro valores de tensión:

- $V_A = 19,6015V$
- $V_B = 17,3649V$
- $V_C = 21,3649V$
- $V_D = 31,4570V$

Volviendo al circuito de la Figura 27 sin simplificar, se nombran todos los nodos, como se ve en la Figura 30, y se ponen los elementos del circuito en los tres ficheros como se ve en la Figura 31.

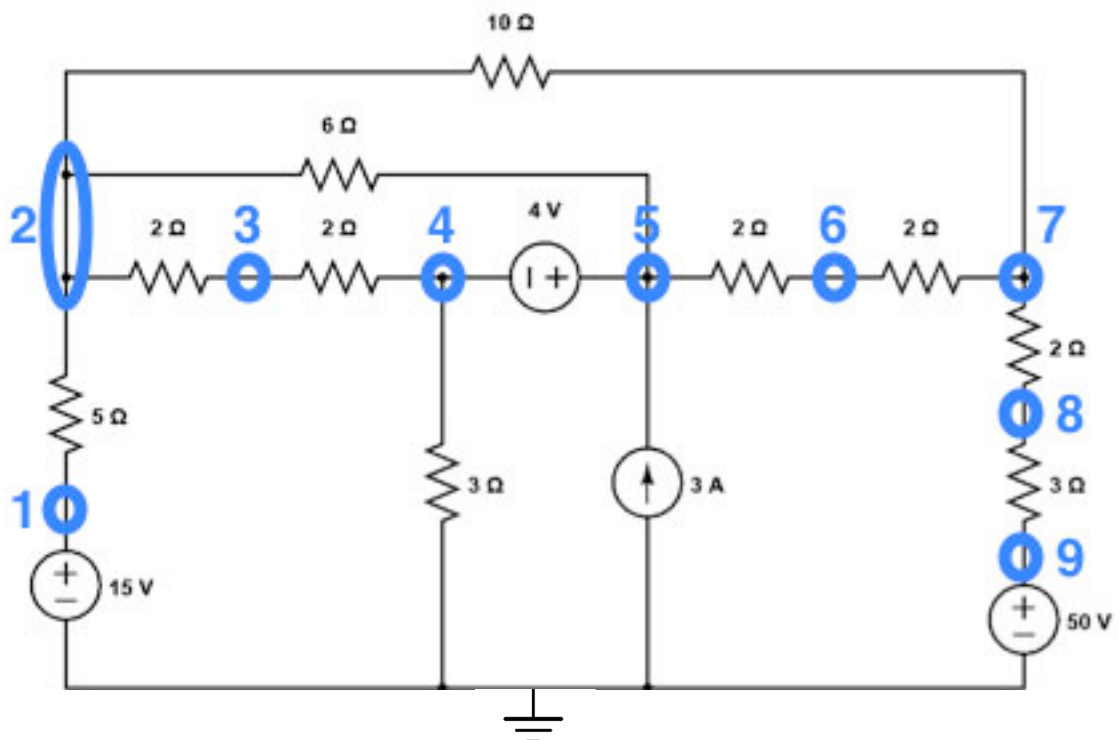


Figura 30: circuito de la Figura 27 identificando todos los nodos

Análisis de circuitos mediante Matlab

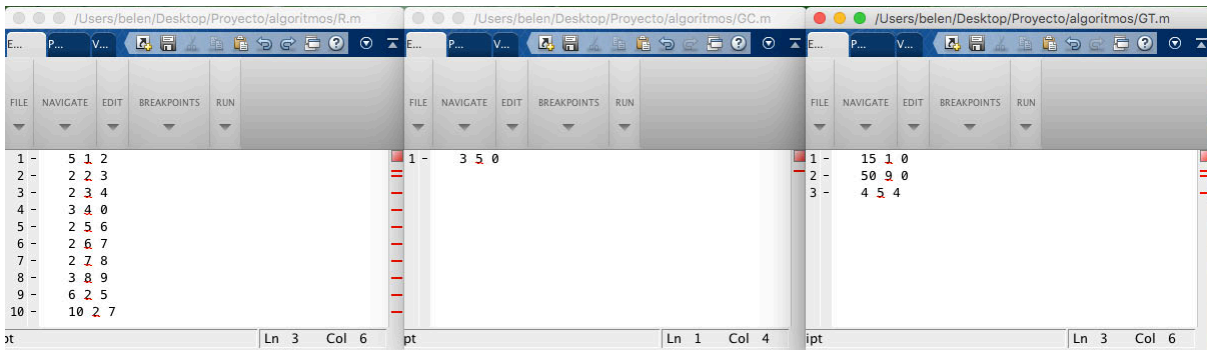


Figura 31: ficheros con la información del circuito de la Figura 30

Se ejecuta el main de la aplicación, traductor.m, sobre la consola de comandos de Matlab:

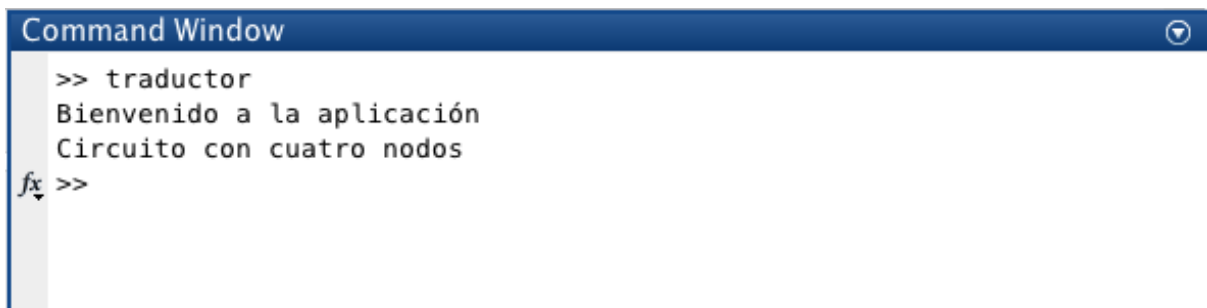


Figura 32: ventana de comandos de Matlab tras ejecutar el programa con los ficheros de la Figura 31

Se comprueba en el fichero de tensiones.txt el resultado de la prueba:

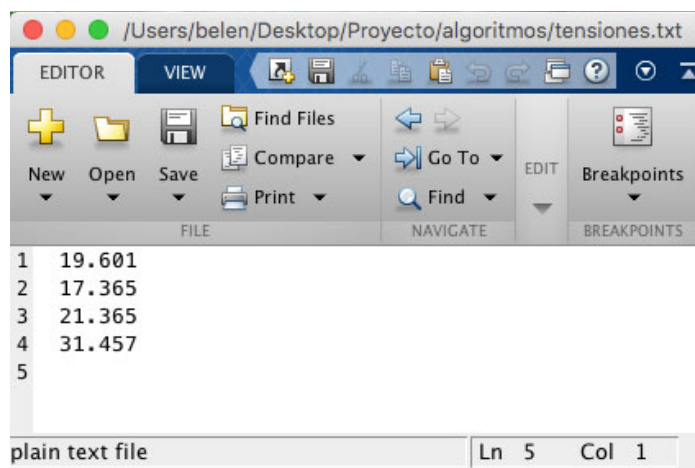


Figura 33: fichero tensiones.txt, solución del circuito de la Figura 30

Se verifica que en ambos casos el resultado es el mismo.

6. MANUAL DE USUARIO

6.1. Inicio de la aplicación

Los archivos necesarios para la ejecución del algoritmo se encuentran en la propia carpeta raíz de la aplicación.

Como ya se ha visto, los ficheros de entrada son R.m, GC.m y GT.m, se tendrán que crear en el caso de que no estén en la carpeta raíz. Se completarán el siguiente formato:

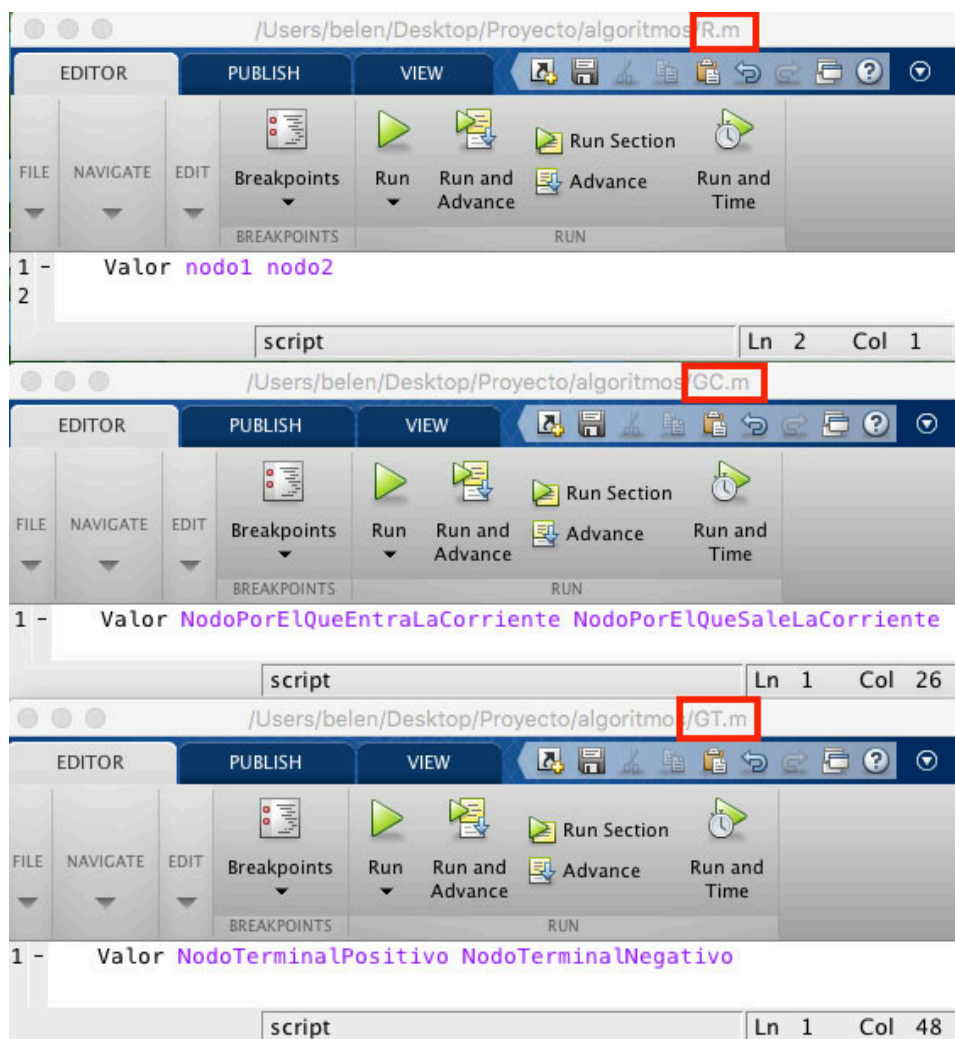


Figura 34: formato de los ficheros de las resistencias, fuentes de corriente y fuentes de tensión

Una vez guardados los ficheros se iniciará la aplicación usando el main de la misma, traductor.m. Cuando acabe de ejecutarse se creará un nuevo fichero, o se sobrescribirá en el caso de que no sea la primera vez en usarse la aplicación, con el nombre tensiones.txt. Al abrirlo aparecen en columnas las tensiones de los nodos fundamentales como se muestra en la siguiente figura:

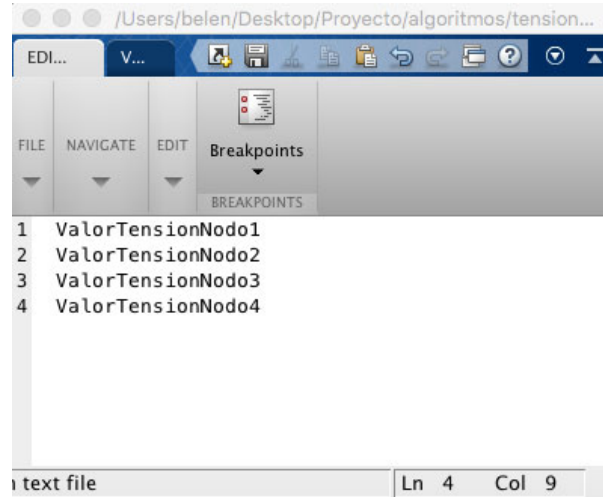


Figura 35: formato del fichero tensiones.txt

6.2. Ejemplos

A continuación se expondrán algunos ejemplos de uso: se pondrá el circuito, el contenido que hay que poner en cada fichero de entrada y la solución escrita en el fichero de salida. Además, para comprobar su que está bien el resultado, se hará el análisis del circuito usando las técnicas descritas en el apartado 3.2. *Conceptos básicos de circuitos eléctricos* de este documento.

6.2.1. Ejemplo de circuito con un nodo fundamental

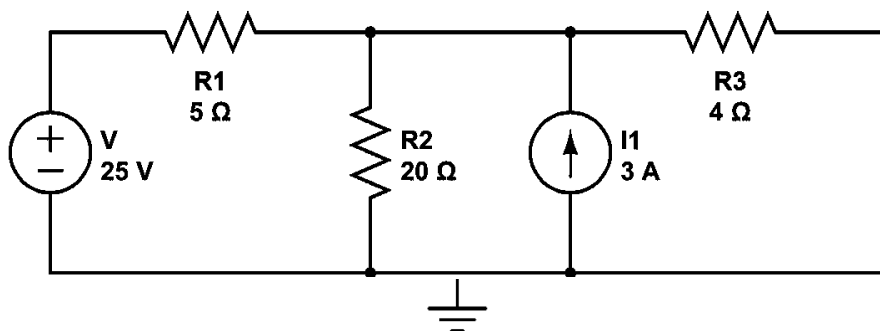


Figura 36: ejemplo de circuito con un nodo fundamental

El circuito de ejemplo podría inducir a error por parecer que tiene más de dos nodos distintos al de referencia, pero, como se ve en la Figura 37, identificamos que solamente hay dos.

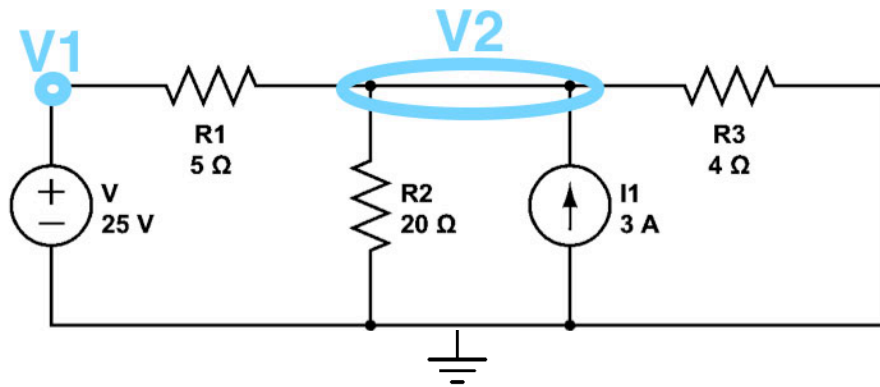


Figura 37: identificación de los nodos de la Figura 36

Una vez identificados, se observa que entre el nodo de referencia y el nodo 1 hay una fuente independiente de tensión, por lo que la tensión en este nodo es la misma del generador, es decir, 25V. Pero se puede aplicar la transformación de fuentes para simplificar el circuito ya que entre el nodo 1 y el nodo 2 únicamente hay una resistencia, por lo que la fuente de tensión y la resistencia están en serie.

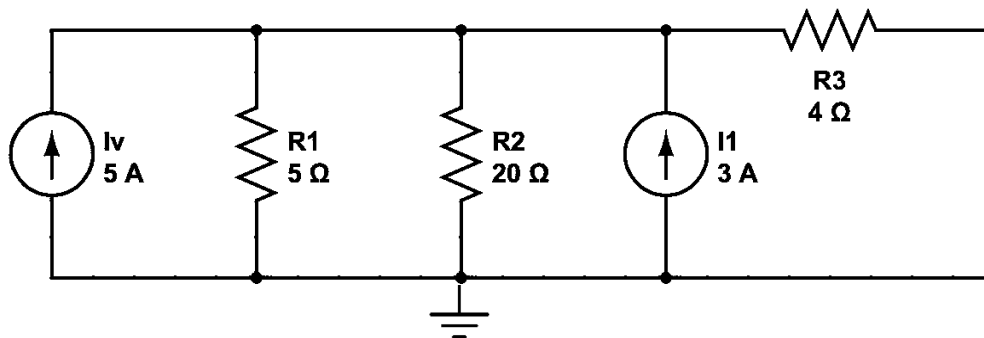


Figura 38: circuito equivalente al de la Figura 36

Nuevamente, se identifican los nodos y se observa que hay un único nodo fundamental distinto al de referencia.

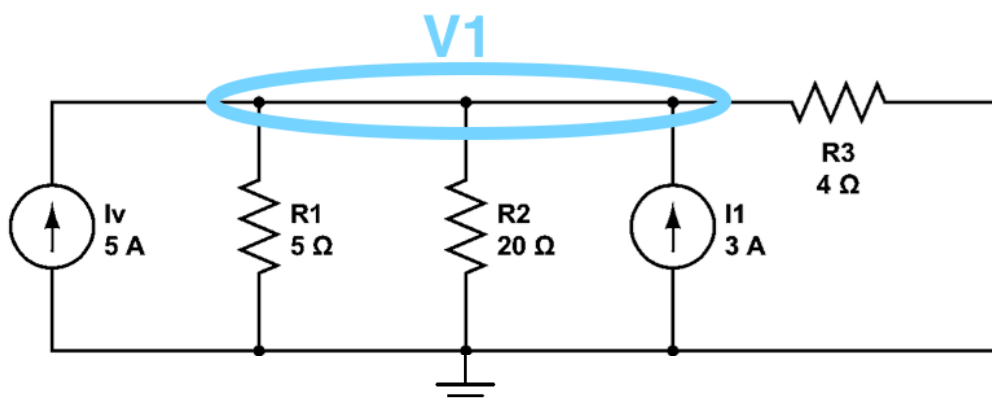


Figura 39: identificación del nodo fundamental del circuito de la Figura 38

Análisis de circuitos mediante Matlab

Para el análisis de circuitos teórico o manual es mucho más cómodo hacer el paralelo de la resistencia R1 y R2 pero la aplicación lo realiza automáticamente.

Aplicando los métodos de análisis de circuitos, exactamente la Ley de Kirchhoff de las corrientes:

$$\frac{V1}{R1} + \frac{V1}{R2} + \frac{V1}{R3} = Iv + I1 \rightarrow V1 \left(\frac{1}{5} + \frac{1}{20} + \frac{1}{4} \right) = 5 + 3 \rightarrow V1 = 8 \cdot 2 = 16V$$

Poniendo los datos del circuito de la Figura 37, es decir, el circuito sin transformaciones de fuentes de la Figura 36 con los nodos identificados, en los tres ficheros correspondientes a las fuentes de corriente, de tensión y a las resistencias y ejecutando el programa se obtiene:

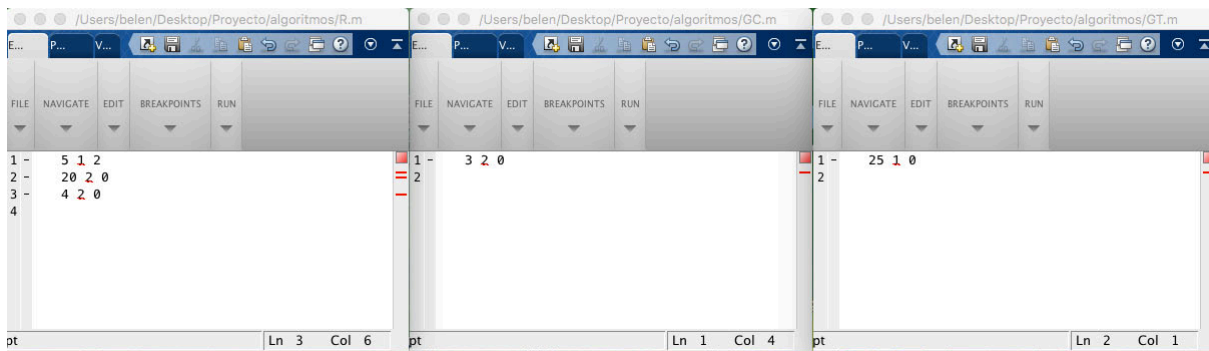


Figura 40: ficheros con la información del circuito de la Figura 37

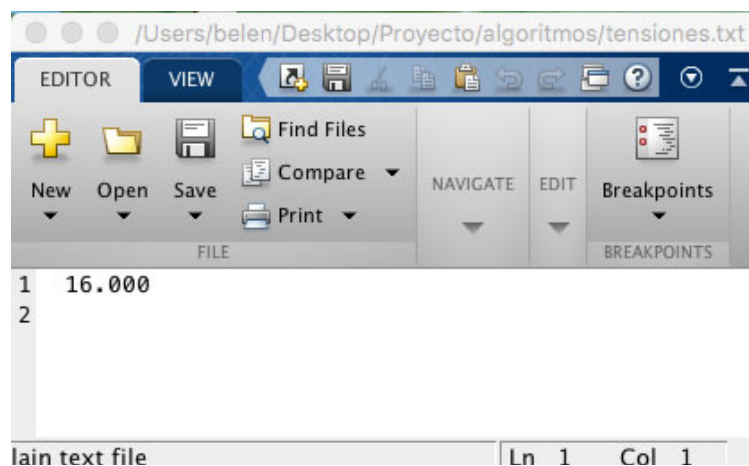


Figura 41: resolución del circuito de la Figura 40 (fichero tensiones.txt)

6.2.2. Ejemplo de circuito con dos nodos fundamentales

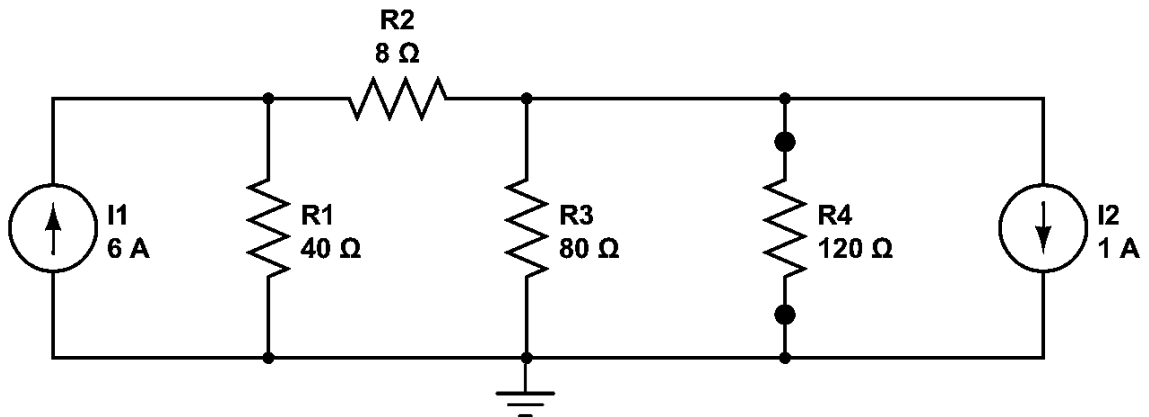


Figura 42: ejemplo de circuito con dos nodos fundamentales

De la misma forma, se identifican los nodos del circuito de la Figura 42:

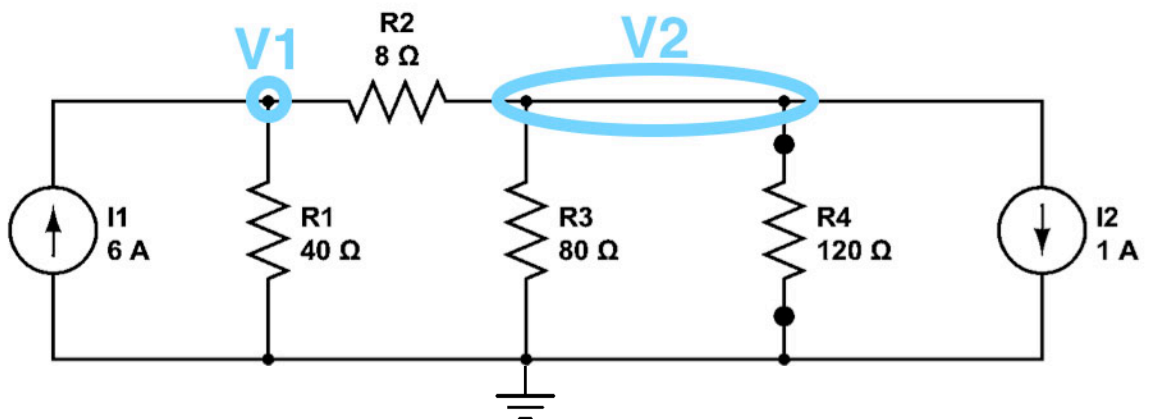


Figura 43: identificación de los nodos de la Figura 42

En este caso, se obtienen dos nodos diferentes al de referencia y no hay posibilidad de reducir su número aplicando modelos equivalentes.

Se calcula el valor en los dos nodos:

$$\text{Nodo 1: } \frac{V1}{R1} + \frac{V1 - V2}{R2} = I1 \quad \rightarrow \quad \frac{V1}{R1 + R2} - \frac{V2}{R2} = I1$$

$$\text{Nodo 2: } \frac{V2 - V1}{R2} + \frac{V2}{R3} + \frac{V2}{R4} = -I2 \quad \rightarrow \quad -\frac{V1}{R2} + \frac{V2}{R2 + R3 + R4} = -I2$$

Análisis de circuitos mediante Matlab

De la ecuación del nodo 1 se despeja $V1$ obteniendo $V1 = \left(I1 + \frac{V2}{R2}\right) \cdot \frac{(R1 \cdot R2)}{R1+R2}$ y se sustituye en la ecuación del nodo 2 y despejando $V2$:

$$-\frac{1}{R2} \cdot (I1 + V2) \cdot \frac{R1 \cdot R2}{R1 + R2} + \frac{V2}{R2 + R3 + R4} = -I2$$

$$V2 \cdot \left(\frac{1}{R2} + \frac{1}{R3} + \frac{1}{R4} - \frac{R1}{R2 \cdot (R1 + R2)}\right) = -I2 + I1 \cdot \frac{R1}{R1 + R2}$$

Se sustituyen los valores de los componentes y se calculan las tensiones de los dos nodos fundamentales:

$$V2 \cdot \left(\frac{7}{48} - \frac{5}{48}\right) = 1 - 6 \cdot \frac{5}{6} \quad \rightarrow \quad V2 = \frac{4}{\frac{1}{24}} = 96V$$

$$V1 = \left(6 + \frac{96}{8}\right) \cdot \left(\frac{320}{48}\right) = 120V$$

Por lo tanto, el resultado que mostrará la aplicación desarrollada deberá ser $V1=120V$ y $V2=96V$, se comprueba usando el programa:

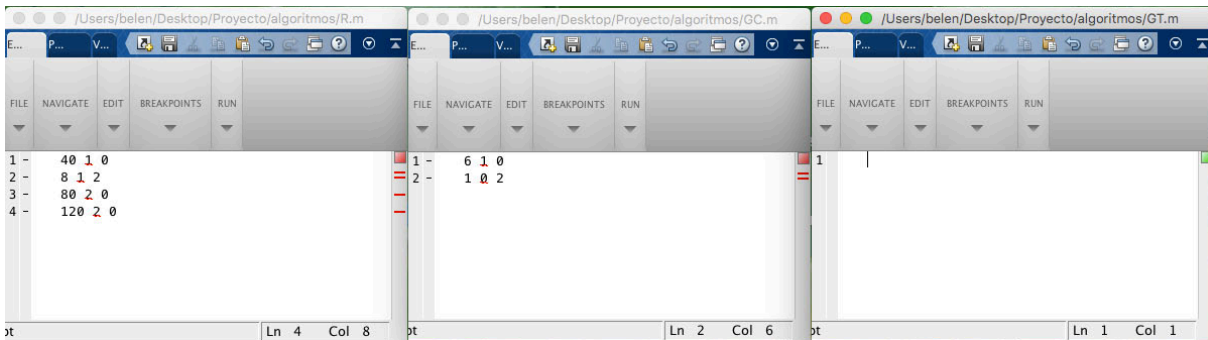


Figura 44: ficheros con la información del circuito de la Figura 43

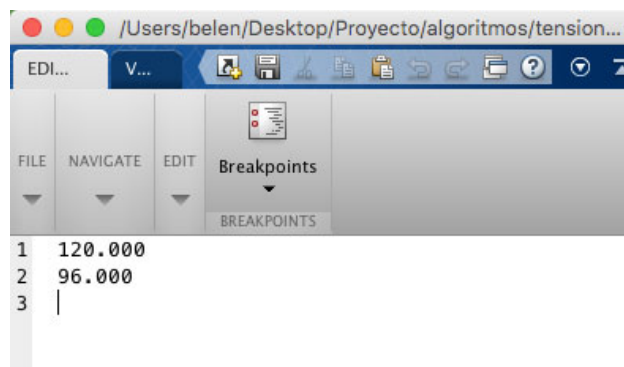


Figura 45: resolución del circuito de la Figura 43 (tensiones.txt)

6.2.3. Ejemplo de circuito con tres nodos fundamentales

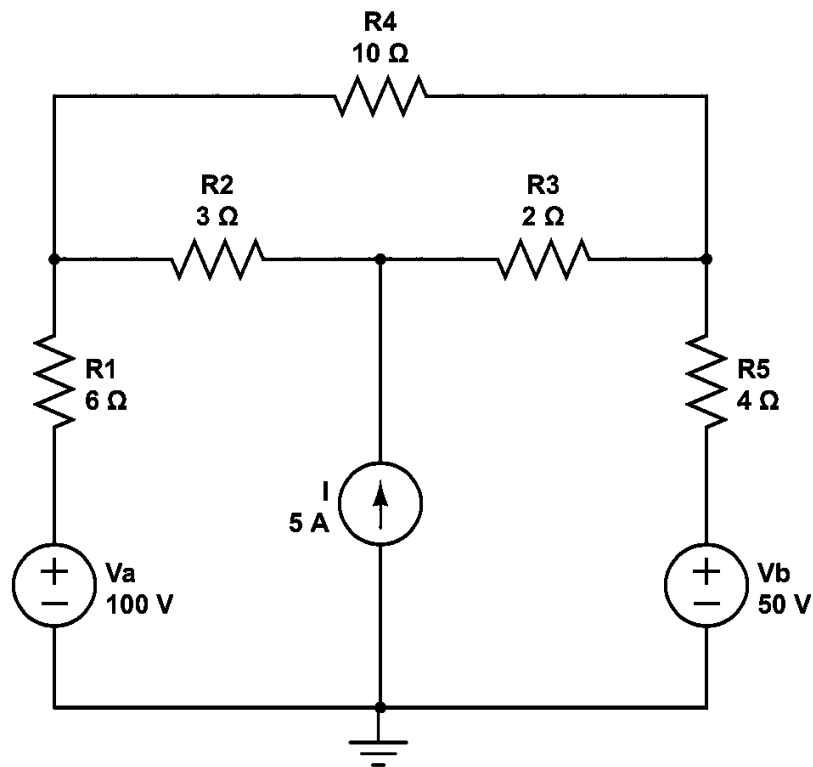


Figura 46: ejemplo de circuito con tres nodos fundamentales

Se identifican los nodos fundamentales del circuito mostrado en la Figura 46:

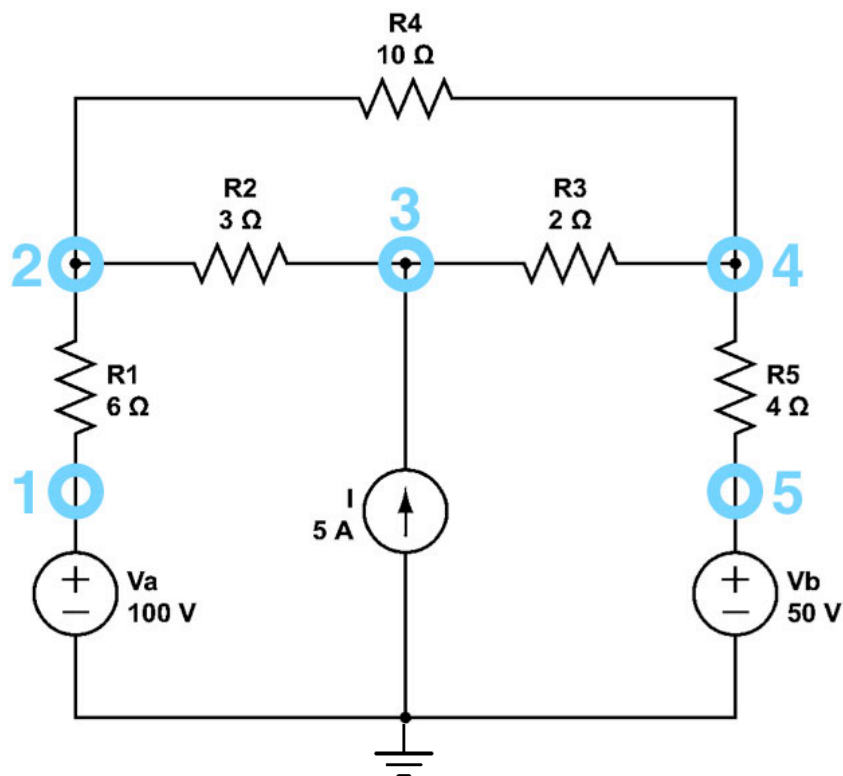


Figura 47: identificación de los nodos de la Figura 46.

De los cinco nodos localizados, se puede reducir en tres nodos fundamentales ya que la fuente V_a situada entre el nodo de referencia y el nodo 1 y la resistencia R_1 están en serie y ocurre lo mismo con la fuente V_b y R_5 . Si se aplica la técnica de transformación de fuentes en ambos casos, obtenemos el circuito de la Figura 48.

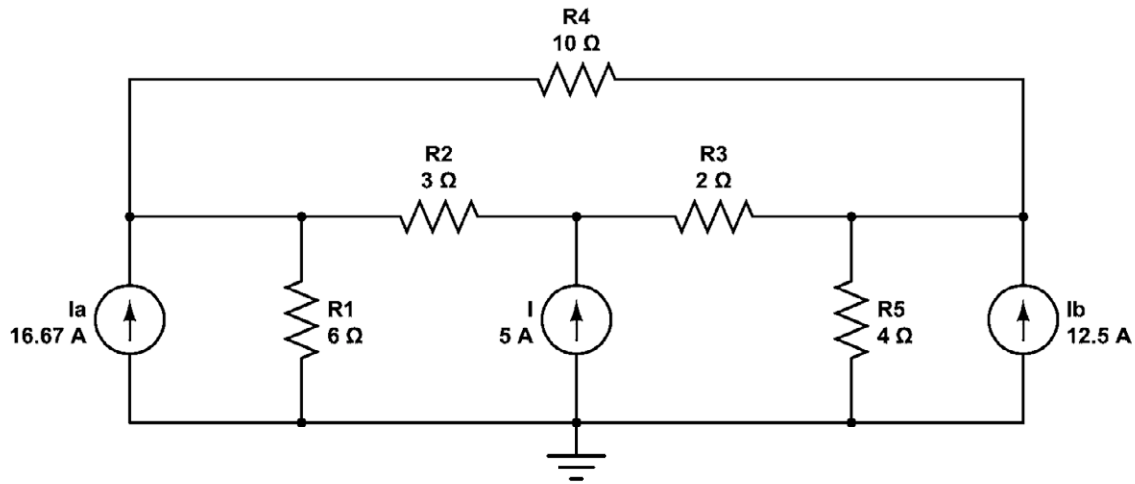


Figura 48: circuito equivalente al de la Figura 46

Se identifican los nodos en el circuito ya reducido, en cuanto a nodos se refiere.

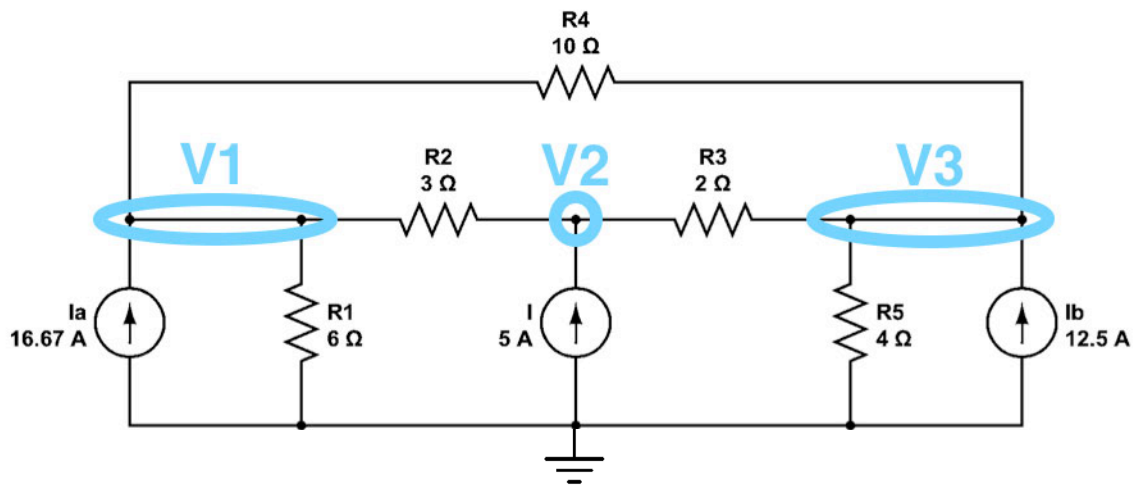


Figura 49: identificación de los nodos fundamentales del circuito de la Figura 48

Siguiendo la misma metodología que en los casos anteriores, primero se calcula el valor teórico de las tensiones que debería dar la aplicación:

$$\text{Nodo 1: } \frac{V_1}{R_1} + \frac{V_1 - V_2}{R_2} + \frac{V_1 - V_3}{R_4} = I_a \rightarrow V_1 \cdot \left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_4} \right) - \frac{V_2}{R_2} - \frac{V_3}{R_4} = I_a$$

$$\text{Nodo 2: } \frac{V_2 - V_1}{R_2} + \frac{V_2 - V_3}{R_3} = I \rightarrow -\frac{V_1}{R_2} + V_2 \cdot \left(\frac{1}{R_2} + \frac{1}{R_3} \right) - \frac{V_3}{R_3} = I$$

$$\text{Nodo 3: } \frac{V_3 - V_1}{R_4} + \frac{V_3 - V_2}{R_3} + \frac{V_3}{R_5} = I_b \rightarrow -\frac{V_1}{R_4} - \frac{V_2}{R_3} + V_3 \cdot \left(\frac{1}{R_3} + \frac{1}{R_4} + \frac{1}{R_5} \right) = I_b$$

Se despeja de la ecuación del nodo 2 V_1 y se sustituye en las otras dos ecuaciones agrupando nuevamente los términos V_2 y V_3 :

$$\text{Nodo 2: } V_1 = \left(V_2 \cdot \left(\frac{1}{R_2} + \frac{1}{R_3} \right) - \frac{V_3}{R_3} - I \right) \cdot R_2$$

$$\text{Nodo 1: } \left(V_2 \cdot \left(\frac{1}{R_2} + \frac{1}{R_3} \right) - \frac{V_3}{R_3} - I \right) \cdot R_2 \cdot \left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_4} \right) - \frac{V_2}{R_2} - \frac{V_3}{R_4} = I_a$$

$$\begin{aligned} V_2 \cdot \left(R_2 \cdot \left(\frac{1}{R_2} + \frac{1}{R_3} \right) \cdot \left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_4} \right) - \frac{1}{R_2} \right) - V_3 \left(\frac{R_2}{R_3} \cdot \left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_4} \right) + \frac{1}{R_4} \right) = \\ = I_a + I \cdot R_2 \cdot \left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_4} \right) \end{aligned}$$

$$\text{Nodo 3: } -\frac{R_2}{R_4} \cdot \left(V_2 \cdot \left(\frac{1}{R_2} + \frac{1}{R_3} \right) - \frac{V_3}{R_3} - I \right) - \frac{V_2}{R_3} + V_3 \left(\frac{1}{R_3} + \frac{1}{R_4} + \frac{1}{R_5} \right) = I_b$$

$$-V_2 \left(\frac{R_2}{R_4} \cdot \left(\frac{1}{R_2} + \frac{1}{R_3} \right) + \frac{1}{R_3} \right) + V_3 \left(\frac{1}{R_3} + \frac{1}{R_4} + \frac{1}{R_5} + \frac{R_2}{R_3 \cdot R_4} \right) = I_b - I \cdot \frac{R_2}{R_4}$$

Puesto que las expresiones que salen son muy largas, queda mucho más simplificado si, a partir de este momento, se sustituyen los valores de los componentes del circuito:

$$\text{Nodo 1: } V_2 \cdot \left(\frac{5}{2} \cdot \frac{3}{5} - \frac{1}{3} \right) - V_3 \cdot (1) = \frac{100}{6} + 5 \cdot 3 \cdot \frac{3}{5} \rightarrow \frac{7 \cdot V_2}{6} - V_3 = \frac{77}{3}$$

$$\text{Nodo 3: } -V_2 \cdot \left(\frac{1}{4} + \frac{1}{2} \right) + V_3(1) = \frac{50}{4} - \frac{3}{10} \cdot 5 \rightarrow -\frac{3 \cdot V_2}{4} + V_3 = 11$$

Se despeja V_3 de la ecuación del nodo 1 y se sustituye en la ecuación del nodo 3:

$$\text{Nodo 1: } V_3 = \frac{7}{6}V_2 - \frac{77}{3} \rightarrow \text{Nodo 3: } -\frac{3}{4}V_2 + \frac{7}{6}V_2 - \frac{77}{3} = 11 \rightarrow V_2 = 88V$$

Ahora se sustituye V_2 sobre la ecuación del nodo 1 para obtener V_3 :

$$\text{Nodo 1: } V_3 = \frac{7}{6} \cdot 88 - \frac{77}{3} = 77V \rightarrow V_3 = 77V$$

Finalmente, se sustituye sobre la ecuación del nodo 2 los valores obtenidos de V_2 y V_3 para obtener la tensión en el nodo 1:

$$\text{Nodo 2: } V_1 = 3 \cdot \left(88 \cdot \left(\frac{1}{3} + \frac{1}{2} \right) - \frac{77}{2} - 5 \right) = 89,5V \rightarrow V_1 = 89,5V$$

Análisis de circuitos mediante Matlab

Al igual que con el primer ejemplo, los datos introducidos en el fichero corresponden con el del circuito sin reordenar ni usar transformación de fuentes ya que el programa realiza estos cambios automáticamente, es decir, el circuito de la Figura 47. Por lo que el contenido de los ficheros iniciales de resistencias, fuentes independientes de tensión y de corriente además del que muestra el resultado final son los siguientes:

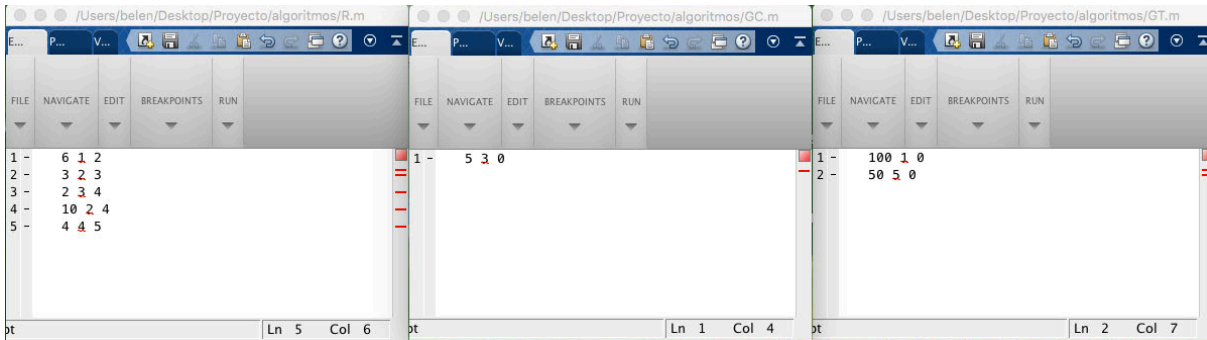


Figura 50: ficheros con la información del circuito de la Figura 47

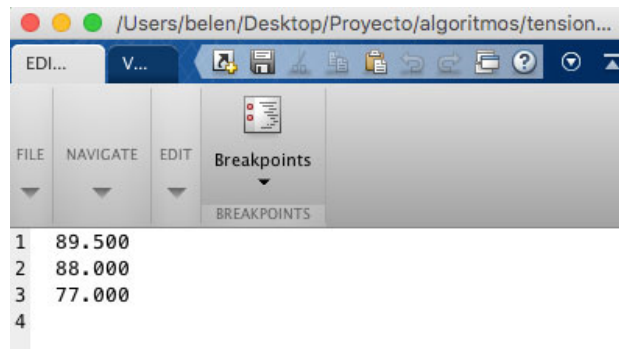


Figura 51: resolución del circuito de la Figura 47 (tensiones.txt)

6.2.4. Ejemplo de circuito con cuatro nodos fundamentales

Las posibilidades de circuitos con cuatro nodos fundamentales aumentan en gran medida respecto a los otros tres anteriores. Se verá solamente un ejemplo en el que, además, al haber fuentes de tensión independientes entre nodo de referencia y nodo fundamental, se obtienen directamente las tensiones buscadas para la solución.

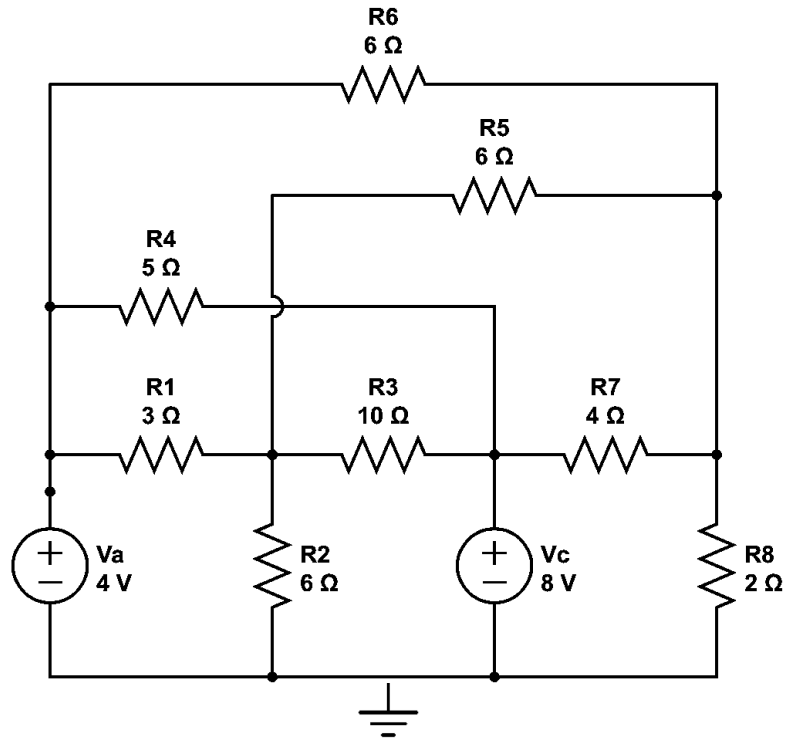


Figura 52: ejemplo de circuito con cuatro nodos fundamentales

Identificando los nodos:

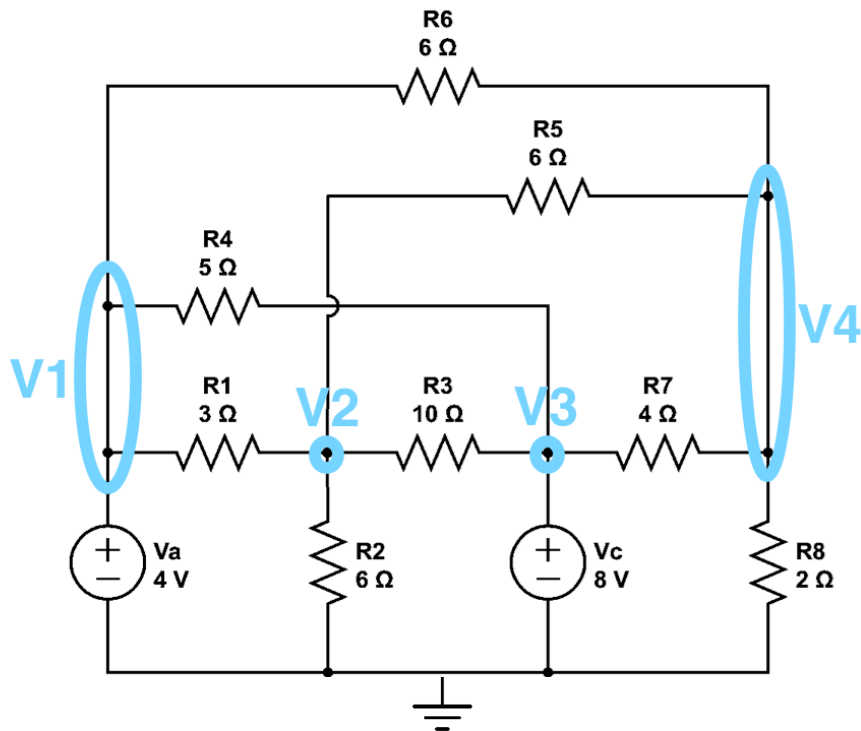


Figura 53: identificación de los nodos de la Figura 52

Análisis de circuitos mediante Matlab

Como ya se había comentado, en este ejemplo se buscan conocer las tensiones correspondientes a los nodos 1, 2, 3 y 4, de las cuáles ya se pueden saber la del nodo 1 y la del nodo 3. Además, en este caso, estos cuatro nodos son los nodos fundamentales del circuito, por lo que no se puede simplificar más.

Se plantea el sistema de ecuaciones:

$$\text{Nodo 1: } V1 = Va = 4V$$

$$\begin{aligned} \text{Nodo 2: } \frac{V2 - V1}{R1} + \frac{V2}{R2} + \frac{V2 - V3}{R3} + \frac{V2 - V4}{R5} &= 0 \\ -\frac{V1}{R1} + V2 \cdot \left(\frac{1}{R1} + \frac{1}{R2} + \frac{1}{R3} + \frac{1}{R5} \right) - \frac{V3}{R3} - \frac{V4}{R5} &= 0 \end{aligned}$$

$$\text{Nodo 3: } V3 = Vc = 8V$$

$$\begin{aligned} \text{Nodo 4: } \frac{V4 - V1}{R6} + \frac{V4 - V2}{R5} + \frac{V4 - V3}{R7} + \frac{V4}{R8} &= 0 \\ -\frac{V1}{R6} - \frac{V2}{R5} - \frac{V3}{R7} + V4 \cdot \left(\frac{1}{R5} + \frac{1}{R6} + \frac{1}{R7} + \frac{1}{R8} \right) &= 0 \end{aligned}$$

Se ponen a la izquierda de las ecuaciones de los nodos 2 y 4 las tensiones $V2$ y $V4$ y a la derecha de la igualdad $V1$ y $V3$, ya que son las tensiones ya conocidas, y se opera con ellas hasta ordenarlas para sustituir los valores numéricos de los componentes:

$$\text{Nodo 2: } V2 \cdot \left(\frac{1}{R1} + \frac{1}{R2} + \frac{1}{R3} + \frac{1}{R5} \right) - \frac{V4}{R5} = \frac{V1}{R1} + \frac{V3}{R3}$$

$$\text{Nodo 4: } -\frac{V2}{R5} + V4 \cdot \left(\frac{1}{R5} + \frac{1}{R6} + \frac{1}{R7} + \frac{1}{R8} \right) = \frac{V1}{R6} + \frac{V3}{R7}$$

Se despeja $V4$ de la ecuación del nodo 2 y se sustituye en la del nodo 4:

$$V4 = R5 \cdot \left[V2 \cdot \left(\frac{1}{R1} + \frac{1}{R2} + \frac{1}{R3} + \frac{1}{R5} \right) - \frac{V1}{R1} - \frac{V3}{R3} \right]$$

$$\begin{aligned} \text{Nodo 4: } -\frac{V2}{R5} + R5 \cdot \left[V2 \cdot \left(\frac{1}{R1} + \frac{1}{R2} + \frac{1}{R3} + \frac{1}{R5} \right) - \frac{V1}{R1} - \frac{V3}{R3} \right] \cdot \left(\frac{1}{R5} + \frac{1}{R6} + \frac{1}{R7} + \frac{1}{R8} \right) &= \\ = \frac{V1}{R6} + \frac{V3}{R7} \end{aligned}$$

Ahora se despeja de la ecuación del nodo 4 V_2 y se sustituyen los componentes por sus valores numéricos:

$$\begin{aligned} \text{Nodo 4: } V_2 \cdot \left(\left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_5} \right) \cdot \left(\frac{1}{R_5} + \frac{1}{R_6} + \frac{1}{R_7} + \frac{1}{R_8} \right) \cdot R_5 - \frac{1}{R_5} \right) + R_5 = \\ = \left[\frac{V_1}{R_1} + \frac{V_3}{R_3} \right] \cdot \left(\frac{1}{R_5} + \frac{1}{R_6} + \frac{1}{R_7} + \frac{1}{R_8} \right) \cdot R_5 + \frac{V_1}{R_6} + \frac{V_3}{R_7} \end{aligned}$$

$$\text{Nodo 4: } V_2 \cdot \left(\frac{23}{30} \cdot \frac{13}{12} \cdot 6 - \frac{1}{6} \right) + 6 = \left[\frac{4}{3} + \frac{8}{10} \right] \cdot \frac{13}{12} \cdot 6 + \frac{4}{6} + \frac{8}{4} \rightarrow \frac{289}{60} V_2 = \frac{248}{15}$$

$$V_2 = \frac{992}{289} = 3,4325V$$

Se sustituye el valor de la tensión del nodo fundamental 2 en la ecuación del nodo 2 en la que estaba despejada V_4 :

$$\text{Nodo 2: } V_4 = 6 \cdot \left[\frac{992}{289} \cdot \frac{23}{30} - \frac{4}{3} - \frac{8}{10} \right] = \frac{864}{289} = 2,9896V$$

Por lo que la tensión en los cuatro nodos fundamentales es:

$$\begin{aligned} V_1 &= 4V \\ V_2 &= 3,4325V \\ V_3 &= 8V \\ V_4 &= 2,9896V \end{aligned}$$

Finalmente se comprueba el resultado obtenido utilizando la aplicación:

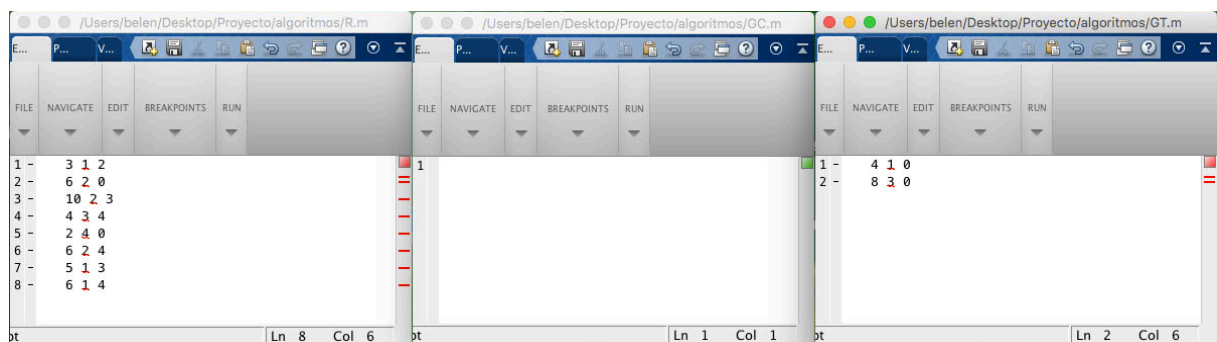


Figura 54: ficheros con la información del circuito de la Figura 53

Análisis de circuitos mediante Matlab

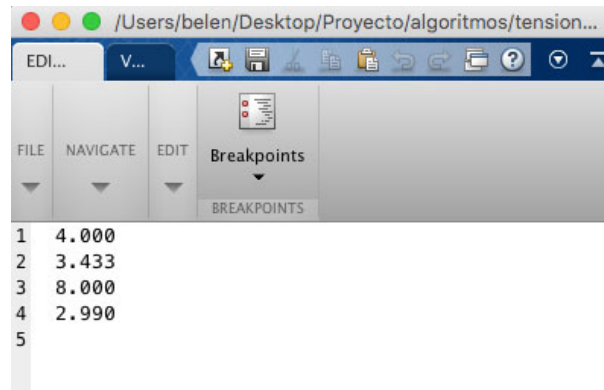


Figura 55: resolución del circuito de la Figura 53

Como se ve en la Figura 55, el resultado obtenido numéricamente coincide con el que da la herramienta.

7. CONCLUSIONES Y LÍNEAS FUTURAS

7.1. Conclusiones

El objetivo de este proyecto es facilitar al aprendizaje de los métodos de circuitos eléctricos para personas que estén empezando en este campo. Libros como el Nilsson, el cuál ha servido de apoyo para la elaboración de este proyecto, explican en qué consisten los métodos pero no siempre resultan fáciles para gente que se esté iniciando en el tema. Esta herramienta permite comprobar si los ejercicios realizados de forma práctica son correctos, proporcionando únicamente el resultado final. Por lo tanto, es una aplicación que corrobora resultados y no que “hace los deberes”, ya que en ese caso, mucha gente haría un mal uso de la herramienta.

Como ya se ha visto en el apartado 2. *Estado del Arte* hay otros programas que actualmente tienen una funcionalidad parecida pero éste es sin duda el más sencillo ya que, una vez se una con la interfaz de usuario desarrollada en el proyecto paralelo a éste, basta con sólo una foto para obtener las tensiones nodales. Todos los alumnos que han pasado por la escuela han tenido errores que no tienen que ver con el análisis de circuitos propiamente dicho como por ejemplo el uso de diferentes masas que, una vez se entra en circuitos que combinan la electrónica analógica y digital es muy útil, pero para un primer contacto lo único que hace es provocar quebraderos de cabeza y pérdidas de tiempo innecesarias. Por lo que, más que una ayuda acaba resultado ser otra cosa más que aprender.

Aún no se ha hecho pruebas en alumnos que estén empezando a estudiar, pero se espera que sea una herramienta didáctica que facilite el aprendizaje de los mismos y que haga que rentabilicen su tiempo.

Se espera que esta parte de la aplicación sea sólo el inicio y que poco a poco se vaya implementando hasta hacer un programa tan potente como es el PSpice, por ejemplo, y que permita el análisis de cualquier tipo de circuito.

7.2. Líneas futuras

Hay muchas posibles líneas futuras tomando como base estos algoritmos, ya que son eso, la base de un proyecto que podría llegar a ser mucho mayor. A continuación se explicarán tres ideas de poca complejidad.

La primera de ellas podría ser añadir más nodos fundamentales para ampliar la diversidad de circuitos analizados. Para ello sólo habría que hacer un nuevo algoritmo matemático por cada ampliación de nodo que se quiera hacer, teniendo en cuenta que muchos casos ya se repiten en los algoritmos desarrollados hasta ahora.

Otra podría ser la posibilidad de añadir elementos capacitivos e inductivos. En esta parte no se ha hecho ya que sólo se ha planteado el análisis en corriente continua y la equivalencia de estos elementos son o cortocircuitos, en el caso de la bobina, o de circuitos abiertos, en el del condensador. Sería relativamente fácilmente implementable añadiendo a las fuentes un tercer parámetro, su frecuencia de oscilación, para poder hacer el equivalente de frecuencia de estos elementos. Por lo tanto, si en una tercera columna no aparece nada o cero el análisis sería en corriente continua y si, por el contrario hay un número, querría decir que el análisis es en corriente alterna.

Otra ampliación, que sería la idónea para dar soporte a los alumnos de Análisis de Circuitos I, sería la de incorporar fuentes de corriente y tensión dependientes de otros parámetros del circuito. En este caso habría que crear nuevos elementos y tener en cuenta las ecuaciones de dependencia, modificando las cuatro funciones de análisis nodal.

8. REFERENCIAS

- [1] J.M. Miró Sans, “Método nodal modificado” en Análisis y Diseño de circuitos con PC, Marcombo. A. Puerta Notario, 1989.
- [2] James W. Nilsson y Susan A. Riedel, Circuitos electrónicos, 7ª edición, Pearson Prentice Hall.
- [3] John O. Attia, Electronics and Circuit Analysis using Matlab, CRC Press. Department of Electrical Engineering Prairie View A&M University, New York, Washington D.C. .
- [4] Orcad [En línea]. <<http://www.orcad.com/about/orcad-history>> . [Consultado: 10-Marzo-2016].
- [5] Mathworks [En línea]. <<http://es.mathworks.com/help/physmod/sps/functionlist.html>>. [Consultado: 5-Marzo-2016].
- [6] Profesor en línea [En línea]<<http://www.profesorenlinea.cl/fisica/ElectricidadCargayCorriente.htm>>. Profesor en línea con N^o registro: 188540. [Consultado: 10-Marzo-2016].
- [7] Blogspot: puntocomnoesunlenguaje, entrada “Algoritmos de ordenación. Método de la Burbuja”. [En línea]. <<http://puntocomnoesunlenguaje.blogspot.com.es/2012/07/metodo-de-ordenacion-burbuja.html?m=1>>. [Consultado: 2-Diciembre-2015].
- [8] Área de Ingeniería eléctrica de la Universidad de Castilla-La Mancha. Ecuaciones con nodos y generadores dependientes. [En línea]. <http://iele.edii.uclm.es/Estudios/ITIE/Albacete/Asignaturas/CI_archivos/A_Descarga/Transparencias/Tema11/Tema11.pdf> [Consultado: 23-Septiembre-2015].
- [9] “Problemas de circuitos Ejemplos resueltos.pptx”, Universidad de Castilla-La Mancha.<https://www.uclm.es/profesorado/ajbarbero/S_Campo_Electrico/Problemas%2520de%2520circuitos%2520Ejemplos%2520resueltos.pptx> [Consultado: 2-Diciembre-2015].
- [10] Ayuda en línea de Matlab. [En línea] <<http://es.mathworks.com/help/>> [Consultado: 25-Mayo-2016].
- [11] Anthony García González en Panama Hitek, *Leyes de Kirchhoff: El método ZBUS* [En línea] <<http://panamahitek.com/leyes-de-kirchhoff-el-metodo-zbus/>> [Consultado: 20-October-2015].

