



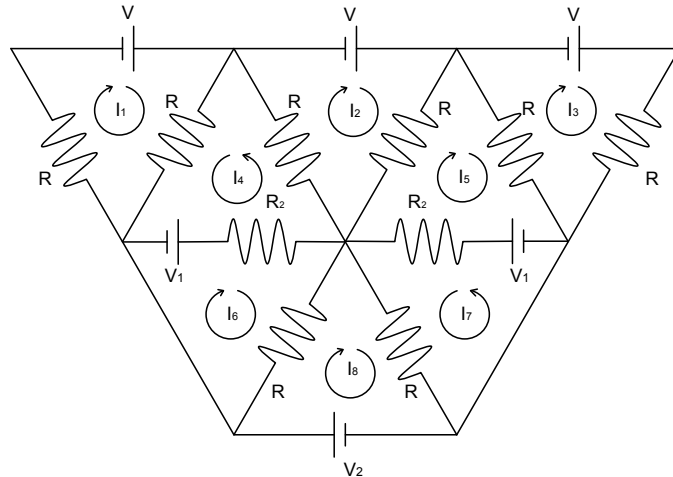
TAREA 04

Fecha de entrega: 15/09/2019 23:59 hrs

Problema

Considere el circuito de la figura. Todas las resistencias que aparecen dibujadas oblicuas son $R=4\Omega$ mientras que las horizontales son $R_2=3\Omega$. Las fuentes de voltage tienen valores $V=20V$, $V_1=30V$ y V_2 es una fuente variable. El circuito es tal que se quema si alguna de las corrientes pasando por el circuito supera los 20 amperes (el problema no es necesariamente muy realista).

Determine el valor máximo que puede tener V_2 de modo que el sistema no se queme.



¿Cómo resolver el problema?

Ud. debe escribir la ecuación de Kirchhoff para circuitos ($V = R \times I$) para cada uno de los circuitos elementales cerrados que componen el circuito más complejo (es decir, cada uno de los triángulos en el dibujo). Al hacerlo, Ud. obtendrá un sistema de 8 ecuaciones para 8 incógnitas (las corrientes I_{1-8}). Escriba este sistema de ecuaciones de forma matricial de modo que se pueda expresar como:

$$R \times \vec{I} = \vec{V}$$

donde R es una matriz de 8×8 e \vec{I} es el vector de las corrientes, $(I_1, \dots, I_8)^t$.

Ayuda. Si no está familiarizado con cómo hacer este tipo de análisis, hay varios videos en youtube que le pueden ayudar. Por ejemplo [este](#), pero hay muchos que le pueden ayudar.

Para V_2 explore el rango de valores $10V - 30V$, variando V_2 y calculando la corriente máxima que pasa por el circuito cada vez (por simplicidad sólo preocúpese del valor máximo del vector \vec{I} y recuerde que lo importante es el valor absoluto de cada uno de sus elementos).

Ahora utilice algún algoritmo de búsqueda de ceros para encontrar el valor de V_2 para el cual el circuito se quema.

Requerimiento. Se pide que elija un método para la búsqueda de ceros y para ese mismo método compare dos alternativas para resolver el sistema de ecuaciones. Por ejemplo, dado que la matriz R no cambia al cambiar V_2 , Ud. podría intentar una descomposición PLU de la matriz R y utilizarla en el algoritmo de búsqueda de ceros. Otra alternativa podría ser invertir cada vez la matriz. Por último, la matriz R no está densamente poblada por lo que podría intentar alguno de los algoritmos para **sparse matrices**, sea creativo. Compare los dos mecanismos elegidos en términos de tiempo de ejecución y discuta sobre las diferencias observadas.

Instrucciones importantes.

- Utilice `git` durante el desarrollo de la tarea para mantener un historial de los cambios realizados. La siguiente [cheat sheet](#) le puede ser útil. **Esta vez revisaremos el uso apropiado de la herramienta y asignaremos una fracción del puntaje a este ítem.** Realice cambios pequeños y guarde su progreso (a través de *commits*) regularmente. No guarde código que no corre o compila (si lo hace por algún motivo deje un mensaje claro que lo indique). Escriba mensajes claros que permitan hacerse una idea de lo que se agregó y/o cambió de un `commit` al siguiente.
- También comenzaremos a revisar su uso correcto de `python`. Si define una función relativamente larga o con muchos parámetros, recuerde escribir el *docstring* que describa los parámetros que recibe la función, el output, y el detalle de qué es lo que hace la función. Recuerde que generalmente es mejor usar varias funciones cortas (que hagan una sola cosa bien) que una muy larga (que lo haga todo). Utilice nombres explicativos tanto para las funciones como para las variables de su código. El mejor nombre es aquel que permite entender qué hace la función sin tener que leer su implementación.
- Para `python` existe una guía de estilo sintáctico (PEP8) que entrega un set de reglas simples para crear código ordenado y fácilmente legible por otras personas. Por ejemplo, se recomienda no usar líneas más largas que 79 caracteres. En el futuro revisaremos que su código apruebe `pep8`, por ahora le sugerimos que lea la guía [aquí](#) se familiarice con las reglas y trate de implementarlas en esta tarea a modo de ejercicio. En [esta página](#) puede chequear si su código aprueba PEP8. También hay utilidades en la línea de comando que permiten hacer la prueba directamente en sus computadores:

```
> pip install pycodestyle  
> pycodestyle <filename>
```
- La tarea se entrega subiendo su trabajo a github. Cuando termine asegúrese de hacer un último `commit` y luego un `push` para subir todo su trabajo a github.
- El informe debe ser entregado en formato `pdf`, este debe ser claro sin información de más ni de menos. **Esto es muy importante, no escriba de más, esto no mejorará su nota sino que al contrario.** Por ejemplo, la presente tarea probablemente no requiere informes de más de 4 páginas en total (esto no es una regla estricta, sólo una referencia útil). Asegúrese de utilizar figuras efectivas y tablas para resumir sus resultados. Revise su ortografía.
- Repartición de puntaje: 50 % implementación y resolución del problema (independiente de la calidad de su código); 40 % calidad del reporte entregado: demuestra comprensión del problema y su solución, claridad del lenguaje, calidad de las figuras utilizadas; 5 % uso apropiado de `git`; 5 % diseño del código: modularidad, uso efectivo de nombres de variables y funciones, *docstrings*, etc.