

## “Modelado de Placa mediante Diferencias Finitas”

Para el uso en Windows del simulador se necesita tener instalado Python2.7, Octave 4.2.1, y la utilidad de python “sympy-1.1.1” en las siguientes direcciones:

Python2.7: <https://www.python.org/downloads/>

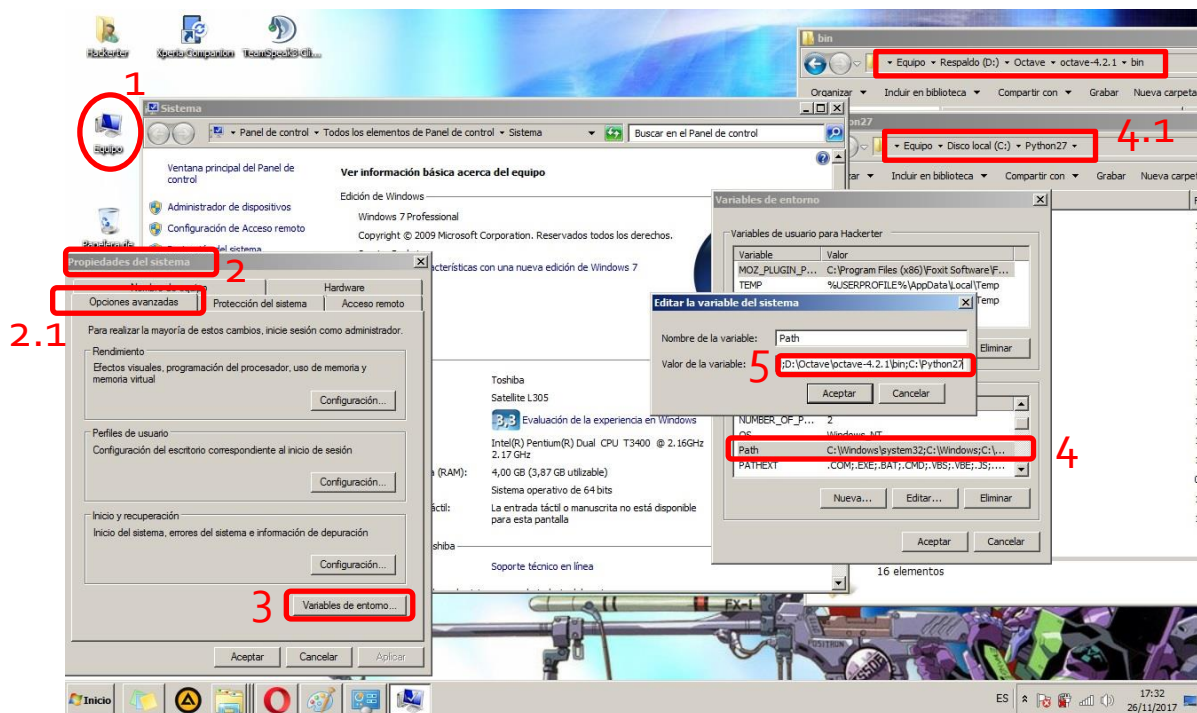
Octave 4.2.1: <https://www.gnu.org/software/octave/download.html>

Sympy-1.1.1: <https://github.com/sympy/sympy/releases>

El proceso de instalación completa partirá asumiendo tener instalado python y octave en el computador, ya que éstos no presentan complejidad.

1<sup>º</sup> Integraremos primero python y octave como variables de entorno del sistema, para ejecutarlas en “CMD” o “Línea de comandos de Windows”.

Por lo que debemos identificar donde está alojada la carpeta de python y octave, ubicando sus directorios `"/bin"` de octave y `"/python27"` de python (Paso 4.1), y agregarlos de la siguiente forma:



Pasos demostrados en la imagen:

- 1.- Ir a equipo, clickear con izquierdo, escoger propiedades, seguido de "Configuración avanzada del Sistema"
- 2.- Luego escoger "Propiedades del Sistema"
- 2.1.- Seleccionar la pestaña "Opciones Avanzadas"
- 3.- Lanzar "Variables de Entorno"
- 4.- Ubicar la variable de entorno "*PATH*" y seleccionar "editar".
- 4.1.- Seleccionar las rutas de ubicación de "*/python27*" de *python* y "*/bin*" de *octave*
- 5.- En la ventana "Editar variable de entorno", se editará "Valor de variable" ingresando al final ";" seguido de la primera ruta + ";" ingresando la segunda ruta, de la siguiente forma sin importar el orden (cabe mencionar que lo siguiente es un ejemplo):

```
%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\;C:\Python27;D:\Octave\octave-4.2.1\bin
```

Presionando "aceptar" en todas las opciones, pasaremos a la instalación de la paquetería "sympy"

Para la instalación de la paquetería "sympy" es necesario haber realizado los pasos anteriores, debido que usaremos "*CMD*" o "*Línea de comandos de Windows*", para esto teclearemos el acceso directo "Windows"+R, en la ventana ejecutar escribimos "*CMD*".

Para la navegación en "*CMD*" se usarán los siguientes comandos:

cd : comando acceso para un directorio, en caso de tener espacio el nombre usar ""

Ejemplo: >cd "directorio"

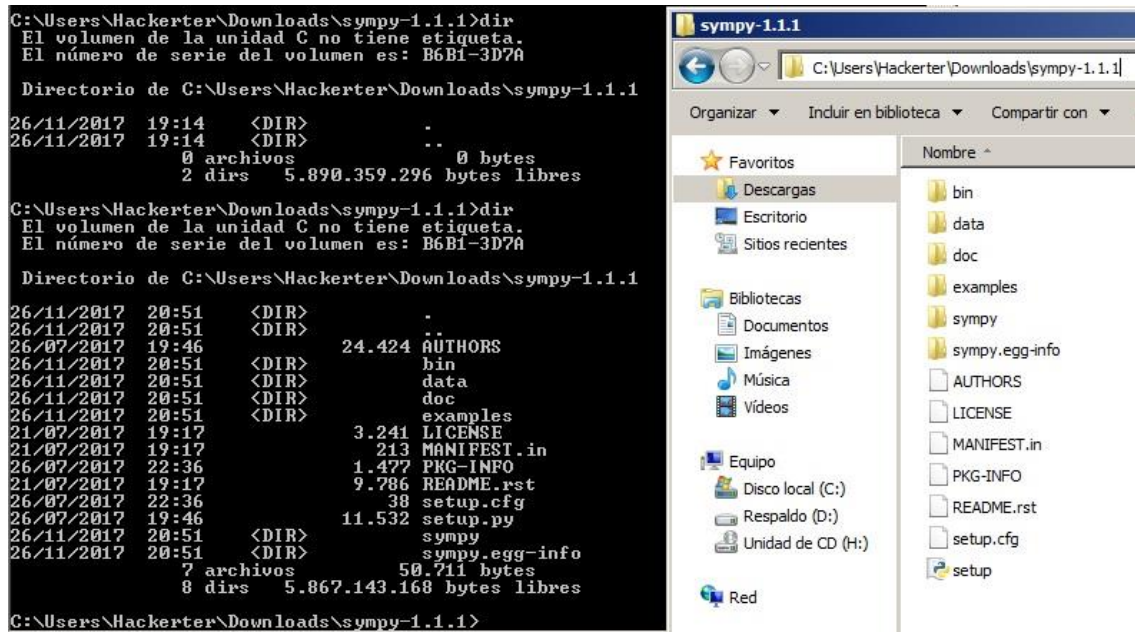
cd.. : comando de retroceso para directorio.

Ejemplo: c:\Users\User\Desktop>cd..

c:\Users\User\>

dir : comando que revisa directorios y archivos localizados en directorio actual

Bien, ubicamos por consola el directorio donde se encuentran los archivos de “sympy”



Ejecutamos en el *prompt* `> python setup.py install` lo que empezará a instalar la paquetería correspondiente en python.



Ahora, para la ejecución del programa de modelado, el orden de los archivos es de la siguiente forma:

1º Dff.py

2º descomposición.py

3º cholesky.m

4º cholesky\_2.m

5º GausSeidel.m

6º GausSeideo\_2.m

Los pasos 3,4 pertenecen al desarrollo de la factorización LU de cholesky realizando  $Ly=b$  con 3º para luego dar la solución de  $Ux=y$  en 4º

Los pasos 5,6 pertenecen al desarrollo del sistema iterativo Gauss Seidel el cual ejecutando 5º, pide parámetros de la "Cota error" y la norma subordinada a utilizar para el cálculo de las iteraciones, siendo "1"- "2"- "inf" como parámetros de entrada, se recomienda la norma 2 por ser la más exacta además de calcular menos número de iteraciones a realizar.

#### Para ejecutar 1º y 2º:

Buscamos el directorio donde aloja los archivos, y ejecutamos en "CMD"

(1º) -> `>python Dff.py`

Lo que entregará en archivos de texto los resultados de su proceso

(2º) -> `>python desconposicion.py`

Lo que entregará en archivos de texto los resultados de su proceso igualmente

#### Para ejecutar 3º,4º,5º,6º:

Se puede hacer usando la interfaz gráfica de octave o la línea de comandos de la siguiente forma:

(5º) -> `>octave GausSeidel.m`

Lo que entregará en archivos de texto los resultados de su proceso

(6º) -> `>octave GausSeidel_2.m`

**Observación Importante:** Es necesario que los archivos se ejecuten en orden, sobre todo de prioridad 1º y 2º, ya que esos generan la matriz "A" y el vector "b", que serán de parámetros de entrada a los métodos de Cholesky y Gauss Seidel. Estos 2 métodos pueden ser ejecutados sin restricción del otro.

Si se ejecuta en Linux, las instrucciones están dadas para instalar paquetería en "Dff.py" junto con un "bash" que realiza la ejecución automáticamente aplicando el comando en terminal `chmod +x Codigo.sh` y ejecutando `sh Codigo.sh` o `./Codigo.sh` en terminal.

Gráfico de ejecución del software de modelado, según su prioridad de ejecución:

