

Propuesta sobre la Enseñanza del Lenguaje Octave

Ascheri, María E. - Pizarro, Rubén A.

Facultad de Ciencias Exactas y Naturales – Universidad Nacional de La Pampa

Uruguay 151 – (6300) Santa Rosa – La Pampa – Argentina

mavacheri@exactas.unlpam.edu.ar

ruben@exactas.unlpam.edu.ar

Resumen

En este trabajo presentamos una propuesta que tiene como meta final apoyar la enseñanza y el aprendizaje de temas de Cálculo Numérico utilizando, fundamentalmente, la computadora y un software matemático libre y de código abierto como Octave. Para lograr este objetivo, proponemos inicialmente el dictado de un curso-taller titulado “Introducción al Software Libre Octave: aplicaciones a problemas de Matemática”, para aquellos alumnos que tengan en su Plan de Estudios la asignatura Cálculo Numérico y también para alumnos relacionados con el área de Matemática. Seguidamente, se implementa su uso en el dictado de la mencionada asignatura. En una instancia posterior y con las adaptaciones pertinentes, esta propuesta se ha hecho extensiva a profesionales interesados con la temática aquí abordada, la cual está en vías de efectivizarse.

Describimos aquí, de manera introductoria, algunas de las particularidades de este software a través de su aplicación a problemas de Matemática. Además, presentamos algunas de las actividades propuestas y las características del curso-taller. Por último, describimos ciertas experiencias recabadas durante el desarrollo de la propuesta y el dictado de Cálculo Numérico.

Palabras clave: software libre y de código abierto, enseñanza del Octave, aplicaciones a problemas de Matemática.

1. Introducción

Los avances tecnológicos y la disponibilidad de recursos informáticos se han convertido en

la actualidad en herramientas de apoyo al proceso de enseñanza – aprendizaje en una amplia variedad de disciplinas. No ajenos a esta situación, nos propusimos introducir nuevas herramientas tecnológicas para apoyar la enseñanza y el aprendizaje de temas de Cálculo Numérico, asignatura que se dicta para las carreras: Ingeniería Civil (2° Año), Licenciatura en Física (3° Año) y Profesorado en Matemática (3° Año).

Fundamentaremos el por qué de nuestro interés de implementar un curso-taller sobre el uso del software Octave en el curso de Cálculo Numérico. Para el desarrollo de este curso curricular, tenemos en cuenta las diversas orientaciones de los alumnos sin comprometer el propósito del mismo:

- Ofrecer una introducción a los métodos numéricos.
- Explicar cómo, por qué y cuándo se espera que éstos funcionen.
- Proporcionar una base firme para el estudio posterior de problemas que requieran de la aplicación de métodos numéricos para su solución.
- Capacitar para que se practiquen los métodos numéricos en una computadora.
- Instruir para que se elaboren programas que puedan usarse en aplicaciones científicas.
- Proporcionar software que resulte fácil de comprender y utilizar.
- Posibilitar el acceso a un software libre.

Teniendo como meta el logro de estos objetivos, varios paquetes de programas se han usado en oportunidades anteriores (QW - BASIC, BASIC, QUICK BASIC, PASCAL, MATHEMATICA), según su predominio en el medio académico. Al respecto, la actitud previa era que cualquier lenguaje de programación que los alumnos utilizaran

podría resultar adecuado. Sin embargo, teniendo en cuenta que muchos de los alumnos que se matriculan en este curso no han aprendido o no tienen la práctica suficiente de ningún lenguaje de programación y a efectos de unificar los conocimientos relativos a un software matemático, que el paquete MATLAB se ha convertido en una herramienta para casi todos los campos de la ingeniería y de la matemática aplicada, y que sus versiones nuevas han mejorado los aspectos de programación, se ha pensado que los alumnos emplearán su tiempo de manera más eficiente y productiva si utilizan este paquete. Pero, obviamente, se requiere que los alumnos dispongan, además de los contenidos conceptuales específicos de Cálculo Numérico, de cierta destreza en el manejo de la computadora y de este paquete. Por ello es que, en años anteriores, durante las primeras semanas de iniciado el curso, se les ha dado una introducción sobre el manejo de la computadora y el uso del paquete MATLAB contando, además, con el apoyo de un Cuaderno elaborado para tal fin por los docentes de la Cátedra. Ahora bien, el acceso a este tipo de software se ve dificultado por el costo del mismo. Este problema es solucionable debido a que en la actualidad se puede disponer de software libre y que además, cumple con las necesidades de la asignatura en lo que se refiere al proceso de enseñanza-aprendizaje de los contenidos temáticos involucrados en la misma. Tal es el caso del software Octave [3, 4, 5], que es un lenguaje de alto nivel para realizar cálculos numéricos en la computadora, y también es un programa capaz de interpretar este lenguaje y realizar los cálculos. Otros programas de características similares a Octave, y hasta cierto punto compatibles, son el lenguaje R de la FSF, MATLAB y Scilab [2, 8, 10].

Surge así nuestra propuesta de un curso-taller introductorio sobre el uso del Octave, especialmente dirigido a aquellos alumnos que tengan en su Plan de Estudios la asignatura Cálculo Numérico y también a alumnos relacionados con el área de Matemática. Una vez concretada esta propuesta se implementa

su uso en el desarrollo de la asignatura antes citada, con la finalidad de que ayude a los alumnos a realizar la componente numérica de los problemas que deban resolver en la sala de computación.

En este trabajo presentamos:

- Descripción breve del lenguaje Octave.
- Algunas aplicaciones a problemas de Matemática (analizadas durante el desarrollo de la propuesta).
- Características del curso-taller.
- Experiencias y conclusiones.
- Acciones futuras.

2. ¿Qué es Octave?

Octave es un programa que ha sido desarrollado dentro del proyecto GNU. Este proyecto fue iniciado por Richard Stallman con el objetivo de crear un sistema operativo completo libre: el sistema GNU. El 27 de septiembre de 1983 se anunció públicamente el proyecto por primera vez en el grupo de noticias net.unix-wizards. Al anuncio original, siguieron otros ensayos escritos por Richard Stallman como el "Manifiesto GNU", que establecieron sus motivaciones para realizar el proyecto GNU, entre las que destaca "retornar al espíritu de cooperación que prevaleció en los tiempos iniciales de la comunidad de usuarios de computadoras".

GNU es un acrónimo recursivo que significa "GNU No es Unix". UNIX es un sistema operativo no libre muy popular. El sistema GNU fue diseñado para ser totalmente compatible con UNIX.

Octave es un lenguaje de alto nivel diseñado originalmente para realizar cálculos numéricos en la computadora. Tiene una interfase de línea de comando para resolver problemas lineales y no lineales, y un lenguaje de programación similar a su contraparte comercial MATLAB, con el que es prácticamente compatible. Se pueden extender las prestaciones de Octave a través de funciones definidas por el usuario escritas en lenguaje propio o usando módulos escritos en C++, C, Fortran u otros lenguajes.

Octave originalmente fue desarrollado para facilitar la tarea a los estudiantes de Ingeniería Química de la Universidad de Texas, sin que estos tuvieran que enfrentarse a las dificultades de la programación. Su flexibilidad enseñada lo hizo popular y su uso se expandió a otros problemas relacionados con el álgebra lineal y con las ecuaciones diferenciales. Su desarrollo se vio favorecido por medio de los aportes de la comunidad de usuarios. Debido a esto, Octave permite abordar numerosos problemas del campo de las ciencias y de la ingeniería: cálculo numérico, estadística, procesamiento de señal, de imagen, entre otros.

Octave entra en la categoría de software libre. Esto significa que se puede usar, distribuir y/o modificar bajo los términos de la licencia GPL (General Public Licence) de GNU, establecida por la Free Software Foundation. La libertad a la que se refiere el software libre es a la que tienen los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software, pues éste es gratuito y su código fuente se encuentra disponible.

En la propuesta utilizamos la versión 2.1.50 de Octave. Si bien no pretendemos dar una descripción completa del lenguaje Octave, sí mostramos de forma introductoria sus principales características a partir de la presentación de algunos ejemplos y actividades propuestas que fueron analizadas durante el desarrollo del curso-taller (ver sección 3). Parte de los contenidos aquí expuestos corresponden a un manual que elaboramos para el dictado del curso-taller [1]. De todas maneras, el manual de referencia más recomendable y difundido de Octave en Internet es el de su creador John W. Eaton [3, 4], y tanto allí como en [2, 5, 6, 7, 8, 9] pueden encontrarse mayores detalles.

2.1 Instalación del lenguaje Octave

Existen varias versiones de Octave, todas disponibles en forma gratuita en Internet. La página principal de octave es <http://www.octave.org>. La versión original se utiliza en LINUX, un sistema operativo gratuito. A pesar de ello, existen varias

versiones para Windows, el sistema operativo propietario de Microsoft. Dos de las páginas desde donde se puede obtener la distribución para Windows son:

<http://prdownloads.sourceforge.net/octave/octave-2.1.50-inst.exe>

<http://math.furman.edu/~tlewis/math13/downloads.html>

Una vez obtenido el instalador, simplemente se debe ejecutar el mismo y se iniciará el proceso de instalación de Octave.

2.2 Ejecutando Octave por primera vez

Para ejecutar el programa, simplemente se debe hacer click en el icono correspondiente. Cuando el software se inicia, se muestra una ventana donde podemos escribir los comandos que necesitemos. Octave muestra un mensaje inicial y un prompt indicando que está esperando órdenes del usuario:

```
GNU Octave, version 2.1.50 (i686-pc-cygwin).
```

```
Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003 John W. Eaton.
```

```
This is free software; see the source code for copying conditions.
```

```
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. For details, type `warranty'.
```

```
Please contribute if you find this software useful. For more information, visit http://www.octave.org/help-wanted.html
```

```
Report bugs to <bug-octave@bevo.che.wisc.edu>.
```

```
>>
```

Tipeando los comandos a utilizar, el software responderá a cada uno de ellos mostrando los resultados a través de la pantalla. Es decir, cada vez que tipeamos algo, el software nos responderá, excepto que finalicemos lo que escribimos con un punto y coma (;).

Para poder obtener información de Octave es necesario conocer el nombre de la orden que se quiere usar. Este nombre no tiene por qué

ser obvio. Un buen sitio para empezar es tipeando **help** y luego presionando **ENTER**. Si ya se conoce el nombre del comando, simplemente hay que pasarlo como parámetro:

```
>> help sqrt
sqrt is a built-in function

- Mapping Function: sqrt (X)
  Compute the square root of X. If X is negative, a complex result is returned. To compute the matrix square root, see *Note Linear Algebra::.

Overloaded function

gsqrt(galois,...)

Additional help for built-in functions, operators, and variables is available in the on-line version of the manual. Use the command `help -i <topic>' to search the manual index.

Help and information about Octave is also available on the WWW at http://www.octave.org and via the help-octave@bevo.che.wisc.edu mailing list.
>>
```

De todas formas, el texto completo de ayuda está disponible a partir del momento que se instala Octave en la carpeta:

```
C:\Archivos de programa\GNU Octave 2.1.50\opt\octave\doc\octave_toc.html
```

pudiendo acceder en este archivo a una tabla que nos brinda acceso a las diferentes funciones que necesitamos.

Si ocurre algún tipo de problema, se puede interrumpir la tarea que está realizando Octave con **Control-C**.

Para editar los programas se utiliza el bloc de notas. Los archivos creados deben ser guardados con extensión **m** para poder ser invocados luego desde Octave.

Para salir de Octave bastará con escribir **exit** en el prompt de Octave y presionar **ENTER**.

3. Algunas aplicaciones a problemas de Matemática

Sin entrar en detalles sobre la sintaxis de Octave (tipos de datos, variables, operadores, funciones, expresiones de control de flujo) y otras cuestiones específicas de este lenguaje, a continuación presentamos algunos de los ejemplos dados en el curso-taller y que se encuentran desarrollados en [1], los cuales permiten ilustrar las principales características del lenguaje Octave y su utilidad en la resolución de problemas matemáticos. Tanto para la selección de éstos como de las actividades propuestas, algunas de las cuales se muestran en la sección 3.1, se tuvo en consideración hacia quién estaba dirigida la propuesta (ver sección 4).

Ejemplo 1

Si queremos implementar una función que obtenga los primeros **n** múltiplos de un número entero **x**, podemos definir la siguiente función:

```
function multiplos(x,n)
  for i=1:n
    x*i
  endfor
endfunction
```

Si a esta función la guardamos con el nombre **multiplo.m**, deberemos invocarla con dos argumentos:

```
>> multiplo(13,4)
ans = 13
ans = 26
ans = 39
ans = 52
```

Ejemplo 2

Definimos una función donde se guardarán el elemento mayor de un vector y el índice correspondiente:

```
function [max, idx] = vmax (v)
  idx = 1; max = v (idx);
  for i = 2:length (v)
```

```

if (v (i) > max)
    max = v (i); idx = i;
endif
endfor
endfunction

```

A esta función la guardamos en un archivo con el nombre **mayor.m**. Luego, la invocamos desde Octave de la siguiente forma:

```

>> v=rand(1,7)
v =
    0.69649    0.10844    0.45652    0.25819
    0.48350    0.82244    0.98909
>> [elementomayor,indice]=mayor(v,i)
elementomayor = 0.98909
indice = 7

```

Ejemplo 3 Sistemas de ecuaciones lineales

Para hallar la solución del siguiente sistema lineal

$$\begin{cases} x_1 - 2x_2 + 3x_3 = 1 \\ 4x_1 + x_2 - 2x_3 = -1 \\ 2x_1 - x_2 + 4x_3 = 2 \end{cases}$$

se puede hacer de varias maneras:

```

>> A=[1 -2 3;4 1 -2;2 -1 4]; % Introducimos
la matriz cuadrada de coeficientes A
>> b=[1;-1;2]; % Introducimos la matriz
columna de términos independientes b
>> x=A\b % Vector solución según el
método de eliminación de Gauss
x =
   -0.04167
    0.41667
    0.62500
>> x=rref([A b]) % Vector solución según
el método de Gauss-Jordan
x =
    1.00000    0.00000    0.00000   -0.04167
    0.00000    1.00000    0.00000    0.41667
    0.00000    0.00000    1.00000    0.62500
>> x=inv(A)*b % Vector solución según
el método de la inversa
x =
   -0.04167
    0.41667
    0.62500

```

Ejemplo 4 Sistemas de ecuaciones no lineales

Resolvamos el siguiente sistema de ecuaciones no lineales, ingresando el valor inicial:

$$\begin{aligned} 4x^3 - 27xy^2 + 25 &= 0 \\ 4x^2 - 3xy^3 - 1 &= 0 \end{aligned}$$

Primero escribimos la función para calcular el valor de la función dada:

```

>> function y=f(x)
y(1)=4*x(1)^3-27*x(1)*x(2)^2+25;
y(2)=4*x(1)^2-3*x(1)*x(2)^3-1;
endfunction

```

Luego, usamos **fsolve** con una condición inicial especificada para hallar las raíces del sistema dado:

```

>> [x,info]=fsolve('f',[1;1]) % Solución
para el valor inicial [1;1]
x =
    1.04429
    1.02382
info = 1

```

Un valor de **info = 1** indica que la solución es convergente.

Ejemplo 5 Gráfica en dos dimensiones

Con la función **plot** y definiendo un vector **x**, con su valor inicial, el incremento y el valor final, podemos realizar la gráfica de una función.

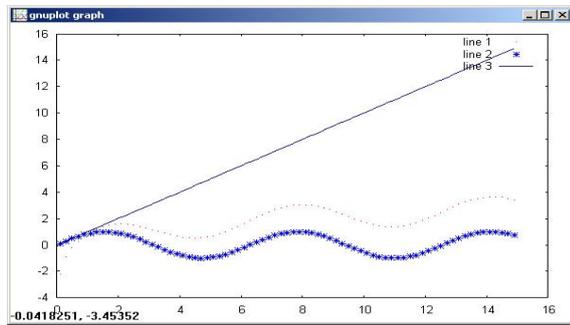
También podemos graficar varias funciones en la misma ventana, simplemente colocando la variable con respecto a la cual obtendremos la gráfica y además separarlas por coma. En este caso, en el cual graficaremos varias funciones juntas, es necesario poder distinguirlas. Para ello, podremos cambiar el tipo de línea o trazo y color de cada una, colocando después de la función y separado por coma, el símbolo y eventualmente el número que nos indicará el color. En [1] se pueden encontrar mayores detalles sobre los posibles tipos de líneas del comando **plot**, la relación entre el número y el color, y el estilo de línea.

```

>> x=0.5:0.1:1.5; % Se introduce el vector x

```

```
>> plot(x,
sin(x)+log(x),'.1',x,sin(x),'*3',x,abs(x),'-5')
```

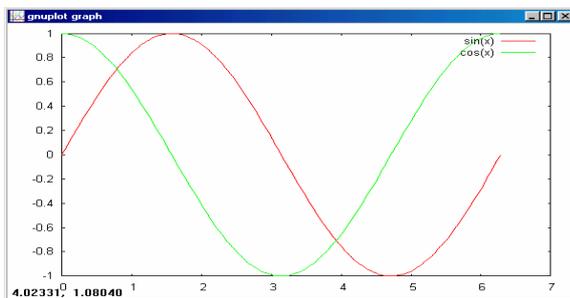


Ejemplo 6 Gráfica en dos dimensiones. Funciones para el control de la gráfica

De igual forma, por medio de la función **fplot** podemos obtener la gráfica de una función determinada a la que se debe indicar el intervalo en el cual se quiere obtener la gráfica. También podemos indicar el intervalo que debe considerar en el eje y.

Por otro lado, existe un conjunto de funciones por medio de las cuales podemos modificar el aspecto de las gráficas realizadas, añadirles títulos, carteles, etc. Para mayores detalles remitirse a [1].

```
>> fplot('sin(x)',[0 2*pi]) % Dibuja la
función seno en el intervalo [0, 2*pi]
>> hold on % Mantiene
en la ventana gráfica el dibujo anterior
>> fplot('cos(x)',[0 2*pi]) % Dibuja la
función coseno sobre la gráfica anterior
```

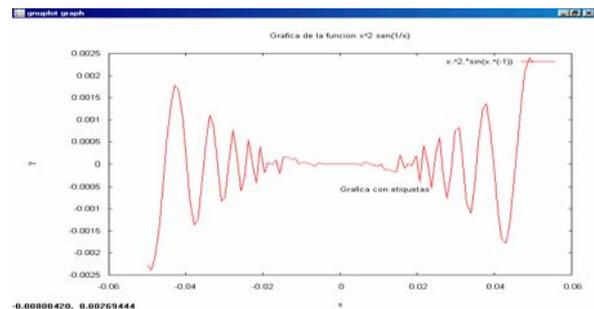


Ejemplo 7 Etiquetas en la gráfica

Si a una gráfica queremos agregarle algunas etiquetas, para que se visualicen estas modificaciones debemos ingresar cada vez la función **replot**.

```
>> fplot('x.^2.*sin(x.^(-1))',[-0.05 0.05])
```

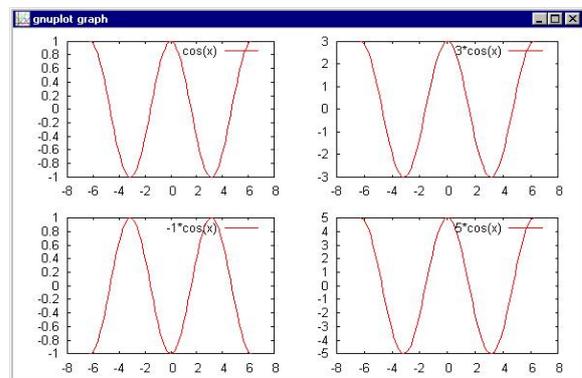
```
>> title('Gráfica de la función x^2 sin(1/x)')
% Coloca título a la gráfica
>> replot % Agrega los cambios
>> xlabel('x') % Coloca un cartel en el eje x
>> ylabel('y') % Coloca un cartel en el eje y
>> replot % Agrega los cambios
>> text(0,12,'Gráfica con etiquetas') % Co-
loca el texto indicado en el punto indicado
>> replot % Agrega los cambios
```



Ejemplo 8 Múltiples gráficas

La sentencia **subplot** nos permite dividir la ventana en la cual aparecerá la gráfica en varias sub ventanas.

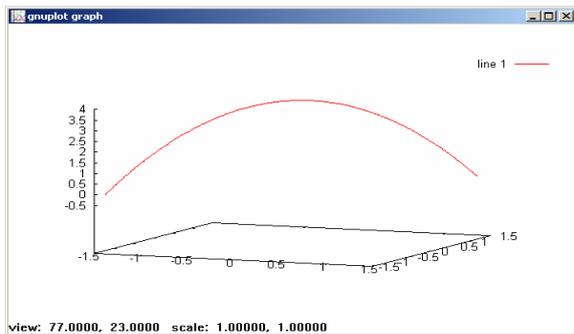
```
>> subplot(2,2,1)
>> fplot('cos(x)',[-2*pi,2*pi])
>> subplot(2,2,2)
>> fplot('3*cos(x)',[-2*pi,2*pi])
>> subplot(2,2,3)
>> fplot('-1*cos(x)',[-2*pi,2*pi])
>> subplot(2,2,4)
>> fplot('5*cos(x)',[-2*pi,2*pi])
```



Ejemplo 9 Gráficas tridimensionales. Gráficas de línea, de malla y de superficie

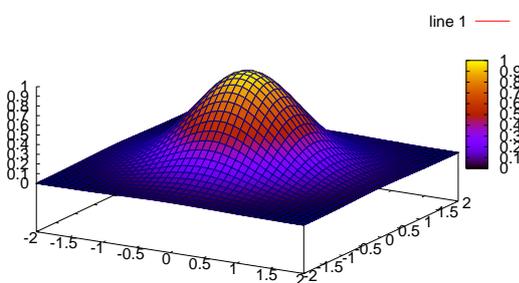
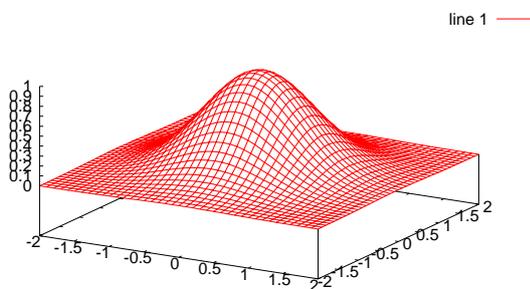
Primero generamos una gráfica de línea. Para ello, a **x** y **y** le damos valores entre $-\sqrt{2}$ y $\sqrt{2}$, y graficamos la función $z = 4 - (x^2 + y^2)$:

```
>> x=-sqrt(2):0.01:sqrt(2); % Intervalo
en el eje x
>> y=x; % Intervalo en el eje y
>> plot3(x,y,4-(x.^2+y.^2)) % Graficamos
la función, que representa la coordenada z
```



Generamos gráficas de malla y de superficie:

```
>> x=-2:0.1:2; y=-2:0.1:2; % Intervalo
en el eje x e intervalo en el eje y
>> [X,Y]=meshgrid(x,y); % Genera las
matrices X e Y para la función mesh
>> z=exp(-X.^2-Y.^2); % Genera z
>> mesh(X,Y,z) % Gráfica de malla
>> surf(X,Y,Z) % Gráfica de superficie
```



3.1 Algunas de las actividades propuestas

Seguidamente presentamos algunas de las actividades propuestas en el curso-taller. La lista completa puede verse en [1].

1. Resuelva el siguiente problema utilizando Octave como una herramienta para el cálculo de las operaciones necesarias. Cada línea de comando que se utilice debe estar comentada (% o #). Luego copie la sesión y péguela en un archivo de texto.

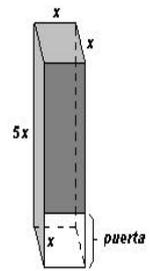
Se lanza un proyectil cuya trayectoria está dada por la siguiente función:

$$y = -x^2 + 4.25x$$

Indique en que momento x alcanza la mayor altura y de cuanto es la misma.

2. Resuelva el siguiente problema:

Las empresas de subtes desean construir ascensores para permitir el acceso de personas con discapacidad. Para cada ascensor se debe excavar un hueco desde la vereda con forma de prisma con base cuadrada de x metros de lado y profundidad igual al quintuplo de la base. La base y las paredes del hueco deben estar totalmente cubiertas con cemento, y deben llevar además a lo largo de sus cuatro aristas verticales, potentes vigas de hierro para evitar su distorsión. Una superficie cuadrada igual a la base debe quedar libre para la puerta. Para presupuestar la obra se consideran los siguientes gastos:



Máquina excavadora \$60 el m^3 , cemento \$100 el m^2 y vigas de hierro \$30 el metro. Calcular las dimensiones del hueco del ascensor si su costo fue de \$27900.

3. Resuelva el siguiente sistema de ecuaciones lineales de tres formas distintas utilizando el operador \, la función **rref** y la función **inv**():

$$\begin{cases} 3x + 5y - 6z = -7 \\ x + y + z = 6 \\ 7x - y - z = 0 \end{cases}$$

4. Utilizando la sentencia **fsolve**, resolver las siguientes ecuaciones no lineales:

- $y = \log(x) - 3x^2 + 2$
- $y = \sin(x) - x^2 - 4$

5. Utilizando la función **plot**, represente en un mismo gráfico las siguientes funciones variando los colores y tipo de trazo:

- $y = 3\text{sen}(x)$
- $y = \text{sen}(3x)$

6. Utilizando **fplot**, obtenga en un mismo gráfico las siguientes funciones:

$$y = x^2 - 4 \quad y = -x^2 - 4$$

7. Realice los siguientes programas que permitan:

- Ingresar un número natural y luego muestre todos los números pares menores que él.
- Calcular las raíces de una ecuación de segundo grado, ingresando los valores de a, b y c.
- Calcular el valor de la hipotenusa de un triángulo ingresando los valores de sus catetos.
- Generar números aleatorios y que el usuario decida cuando terminar ingresando su opción desde el teclado.
- Ingresar una matriz y muestre:
 - El mayor de sus elementos.
 - El menor de sus elementos.
 - El promedio de sus elementos.

4. Características del curso -taller

Nombre del curso-taller: Introducción al Software Libre Octave: aplicaciones a problemas de Matemática.

Institución responsable: Departamento de Matemática de la Facultad de Ciencias Exactas y Naturales de la Universidad Nacional de La Pampa, Santa Rosa, Argentina.

Docentes a cargo: Cátedra de Cálculo Numérico integrada por un profesor asociado, un profesor adjunto y un ayudante.

Destinatarios: Especialmente dirigido a aquellos alumnos que tengan en su Plan de Estudios la asignatura Cálculo Numérico y también a alumnos relacionados con el área de Matemática.

Objetivos

- Aportar una nueva herramienta adecuada a los avances tecnológicos y a la disponibilidad de recursos informáticos.
- Iniciar, capacitar y motivar a los participantes en el uso del lenguaje Octave.

- Mostrar su utilidad en la resolución de problemas matemáticos.
- Capacitar a los participantes para la realización de proyectos de trabajo relacionados con sus áreas de estudio.
- Proporcionar una base sólida sobre el uso de este lenguaje para el estudio posterior de problemas que requieran de la programación de métodos numéricos para su solución.

Programa de contenidos mínimos

- Introducción
- Sintaxis de Octave
- Archivos de funciones y de scripts
- Polinomios
- Sistemas de ecuaciones lineales y no lineales
- Gráficas
- Funciones de entrada / salida
- Otras funciones de interés
- Evaluación

Metodología: Las clases serán de carácter teórico-práctico. Se darán los fundamentos teóricos mínimos, prevaleciendo el enfoque práctico. Los participantes deberán realizar una serie de actividades en la computadora utilizando el software Octave.

Conocimientos previos: Manejo básico de PC.

Duración: Diez encuentros de dos horas de duración cada uno, en una de las salas de cómputos de la Facultad de Ciencias Exactas y Naturales de la UNLPam.

Carga horaria total: Cuarenta horas reloj, distribuidas en horas de aula y horas de trabajo del participante.

Cupo: Hasta 20 participantes (uno por computadora).

Infraestructura e insumos: Sala con 20 computadoras donde se tenga instalado el software Octave y una PC con cañón de proyección.

Condiciones de aprobación: Para aprobar el curso-taller será necesario:

1. Asistir al 80 % de los encuentros.
2. Presentar en forma individual y/o grupal (no más de dos participantes por grupo), un disquete con la resolución de todas las actividades propuestas durante el desarrollo del curso-taller.
3. Presentar y defender ante el grupo, un proyecto de trabajo de carácter individual integrador de los contenidos desarrollados y acorde a las áreas de cada uno.

Certificación: Se emitirán certificados de aprobación por un total de cuarenta horas reloj para los participantes que hayan cumplido con las instancias de 1 a 3 antes señaladas, y de asistencia por un total de cuarenta horas reloj para aquellos participantes que hayan cumplido sólo con 1 y 2.

5. Experiencias y Conclusiones

En el primer semestre del año 2006 hemos dictado este curso-taller, con un buen registro de inscripciones. Como el cupo máximo fue superado, debimos hacer una selección de aspirantes según el número de materias cursadas, aprobadas, promedio general y año de la carrera en que se encontraba cada participante en ese momento. De los 20 participantes seleccionados, sólo uno no recibió el certificado de aprobación, pero sí de asistencia, por no haber entregado el proyecto de trabajo.

La presentación y defensa de estos proyectos fue muy enriquecedora para todo el grupo, ya que cada participante presentó una posible implementación de una situación problemática relacionada con sus diferentes perfiles, esto es, acorde con las carreras de cada uno. Por ello es que esta presentación y defensa de trabajos resultó altamente positiva, debido a que actuó como punto de encuentro entre alumnos de diferentes áreas.

También hemos experimentado el uso del lenguaje Octave en la asignatura Cálculo Numérico en el segundo semestre del año

2006. En una primera instancia, nos animamos a afirmar que los resultados obtenidos fueron satisfactorios debido a que pudimos alcanzar uno de los objetivos propuestos en la asignatura: que los alumnos tuvieran acceso a un software libre y de código abierto para poder desarrollar sus propios programas sin demasiada dificultad, desde el punto de vista académico. Se logró que trabajaran de manera activa y participativa en las actividades que debían realizar en la Sala de Cómputos, haciendo uso de un manual confeccionado por la Cátedra con numerosos ejemplos. Además, usaron sus programas en las evaluaciones parciales de la asignatura con resultados positivos, en general.

Por otro lado, el haber llevado a cabo este curso-taller antes de comenzar con el desarrollo de los contenidos temáticos de Cálculo Numérico, ha resultado de apoyo para el proceso de enseñanza y de aprendizaje de dichos contenidos. También ha favorecido en lo que se refiere a la profundización en la aplicación de métodos numéricos a situaciones problemáticas de la vida real. Si bien un estudio sobre evaluación de herramientas de software libre para cálculo numérico se puede ver en [10] (entre las que se incluye a Octave, entre otros, y se hace una comparación con MATLAB), consideramos que aun debemos hacer un análisis más profundo acerca de las ventajas y desventajas del lenguaje Octave con respecto al paquete MATLAB (que era el que se utilizaba anteriormente en este curso), en lo que se refiere al desarrollo global de la asignatura y a los resultados académicos obtenidos.

Con respecto a Octave, podemos concluir que:

- Es un software de distribución gratuita y a código abierto y por lo tanto, de fácil acceso.
- Es un lenguaje de alto nivel para realizar cálculos numéricos en la computadora, y también es un programa capaz de interpretar este lenguaje y realizar los cálculos.
- Ofrece una interfaz de usuario interactiva orientada a líneas de comandos, pero también puede ser utilizado en modo no interactivo leyendo sus órdenes de fichero.

- Las versiones de Windows se instalan sin problemas, es muy flexible, de fácil uso y con alta compatibilidad con MATLAB.

6. Acciones futuras

6.1 Nuevas implementaciones

Debido a que muchos de los alumnos interesados en realizar este curso-taller no pudieron hacerlo por cuestiones de cupo, continuaremos en próximas instancias con la implementación de esta propuesta.

Adicionalmente, puesto que consideramos que nuestros pares deben conocer y estar en condiciones de usar esta herramienta tecnológica en sus respectivas áreas de estudio y de trabajo, hemos previsto extender el dictado de este curso-taller con las modificaciones pertinentes a docentes del área de Matemática. Hemos propuesto el desarrollo de esta actividad al Consejo Directivo de la Facultad de Ciencias Exactas y Naturales de la UNLPam, y nos ha otorgado el aval académico para la concreción del mismo durante el presente ciclo lectivo.

6.2 Implementación de un editor

Se prevé la implementación de un editor que permita al usuario contar con más recursos al momento de editar sus programas. En el caso de lenguajes como C, Pascal o MATLAB, se cuenta con editores que distinguen palabras reservadas y organizan las diferentes estructuras, permitiendo de esta manera detectar fácilmente errores que puedan aparecer en la compilación del programa. Actualmente, trabajando con el bloc de notas es muy difícil que el usuario pueda descubrir errores durante la edición, y en algunos casos se le dificulta corregir los errores detectados.

7. Bibliografía

[1] Ascheri, M. E., Pizarro, R. A. y Culla, M. E., 2006, *Aplicaciones del lenguaje Octave*

a problemas de matemática, Edición interna, FCEyN, UNLPam, Santa Rosa, Argentina.

[2] Borrelli Noguerras, G., 2005, Manual: *Introducción informal a Matlab y Octave*, Calella, España. En línea:

<http://torroja.dmt.upm.es/~guillem/matlab/index.html>

[3] Eaton, J. W., 1996, 1997, Manual y software: *Octave: (octave). Interactive language for numerical computations*, Versión 2.1.x. En línea:

http://www.octave.org/doc/octave_toc.html

[4] Eaton, J. W., 2002, *GNU Octave Manual. A high-level interactive language for numerical computations*, Publisher: Network Theory Ltd., Free Lic.: GNU General Public License, Edition 3 for Octave version 2.0.13.

[5] García Rojo, J. J., 2003, Manual: *Herramientas en GNU/ Linux para estudiantes universitarios. GNU/Octave: Cálculo Numérico por ordenador*, Free Software Foundation. En línea:

<http://softwarelibre.unsa.edu.ar/docs/index.html>

[6] Hamilton Castro, A. F., 2004, *Introducción al Octave*, Grupo de Computadoras y Control, Dpto. de Física, Elect. y Sistemas de la ULL, España. En línea:

<http://cyc.dfis.ull.es/asignaturas/Curso2004-2005/octave/ApuntesOctave/ApuntesOctave.html>

[7] Long, P. J., 2005, *Introduction to Octave*, Department of Engineering, University of Cambridge. En línea:

www.mdp.eng.cam.ac.uk/CD/engapps/octave/octavetut.pdf

[8] Sánchez de la Rosa, L. L., Manual: *MATLAB / Octave*. En línea:

<http://nereida.deioc.ull.es/~pcgull/ihiu01/cdrom/matlab/contenido/matlab.html>

[9] Storti, M., 2002, *Introducción a Octave*, Id: metnums.text, v 1.1. En línea:

<http://venus.ceride.gov.ar/twiki/pub/Cimec/MethodosNumericosYSimulacion/metnums.pdf>

[10] Medrano, C., Valiente, J. M., Plaza L. y Ramos, P., 2006, *Evaluación de Herramientas de Software Libre para Cálculo Numérico*. En línea:

<http://www.euitt.upm.es/taee06/papers/S1/p46.pdf>