



**FACULTAD
DE INGENIERIA**

Universidad de Buenos Aires

Departamento de Computación – Análisis Numérico I (75.12)

Trabajo Práctico N°1:

Ecuación de Laplace

Grupo 10

Bacchilega, Johanna – 92058 – johi.bacchilega@gmail.com

Ceres, Micaela – 90908 – micaelaceres@gmail.com

Otegui, Matías – 97263 – mati.otegui@gmail.com



2do Cuat. 2016

Introducción y Objetivo

Una ecuación en derivadas parciales es una relación entre una función de dos o más variables y algunas de sus derivadas parciales. Resolver una ecuación de este tipo significa idealmente encontrar las funciones que la cumplen, tal vez bajo ciertas condiciones adicionales, o en caso de que no se pueda, dar la mayor información posible sobre cómo son estas funciones.

La ecuación a resolver es un tipo de ecuación en derivadas parciales denominada Ecuación de Laplace. En ella (x, y) es una función que depende de dos variables espaciales x e y , que se mueven en una región plana Ω y ∇^2 es el operador laplaciano. El tiempo no aparece como variable independiente, de hecho, esta ecuación se usa para modelar un fenómeno de tipo estacionario en los que el tiempo no interviene. También hay que establecer condiciones de contorno en la frontera de la región Ω , denominadas condiciones de frontera de Dirichlet.

Para la resolución de este trabajo práctico se pide resolver aproximadamente la siguiente ecuación

$$\nabla^2 u(x, y) = \frac{\partial^2 u(x, y)}{\partial x^2} + \frac{\partial^2 u(x, y)}{\partial y^2} = 0$$

donde $u(x, y)$ representa la temperatura en el punto (x, y) del plano.

Desarrollo y Conceptos teóricos

A veces encontrar la solución analítica de una ecuación diferencial resulta muy complicado y muchas veces imposible. Sin embargo prácticamente en todos los casos es posible encontrar valores de nuestra función en un conjunto discreto de puntos, eso lo conseguimos aproximando las derivadas por fracciones de incrementos de la función y de las variables, y pasando de una expresión diferencial a una algebraica.

Se estudió la ecuación diferencial considerándola en el rectángulo $[0,1] \times [0,1]$ y discretizándola con un paso $h = 0,1$. Además se consideraron las siguientes condiciones de contorno:

$$\begin{cases} u(x, 0) = u(x, 1) = 0 \\ u(0, y) = u(1, y) = y(1 - y) = y - y^2 \end{cases}$$

Para construir el sistema de ecuaciones lineales que representen la ecuación se usó la técnica de las diferencias finitas. En primera medida armamos la malla en la cual se encuentran los puntos sobre los cuales vamos a poder obtener el resultado de la ecuación (serán puntos discretos y no sobre un continuo).

Planteamos la discretización de los puntos $u_{i,j}$ como representación de los nodos equidistantes en la malla de la región R para cada punto $(x_i, y_i) = (i \cdot h, j \cdot h)$ con $i = 0, 1, 2, \dots, n+1$, y $j = 0, 1, 2, \dots, m+1$.

La discretización de las derivadas segundas se planteó a partir de diferencias finitas centradas, según:

$$\frac{\partial^2 u(x, y)}{\partial x^2} \cong \frac{u_{i-1,j} - 2 * u_{i,j} + u_{i+1,j}}{h^2}$$

$$\frac{\partial^2 u(x, y)}{\partial y^2} \cong \frac{u_{i,j-1} - 2 * u_{i,j} + u_{i,j+1}}{h^2}$$

Luego sumando ambas aproximaciones obtenemos la ecuación de Laplace discretizada:

$$\frac{\partial^2 u(x, y)}{\partial x^2} + \frac{\partial^2 u(x, y)}{\partial y^2} \cong \frac{u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4 * u_{i,j}}{h^2} = 0$$

De esta expresión podemos despejar el valor de $u_{i,j}$ para cada par i,j , que representa la aproximación para cada $u(x_i, y_j)$:

$$u_{i,j} = \frac{u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}}{4}$$

Esto implica que para conocer el valor de la función en cada nodo interior a la malla se debían conocer los valores en los alrededores (puntos vecinos) tanto en la dirección x como en la dirección y .

Desarrollo

Para resolver el SEL obtenido se trabajó con diversos métodos numéricos. El sistema fue resuelto tanto por método directo (método de Gauss) como por métodos iterativos (Gauss – Seidel, Jacobi, y sobrerelajación).

El método directo más importante para resolver un sistema lineal general cualquiera es el método de eliminación de Gauss, siendo la idea eliminar las incógnitas una a una de forma sistemática para transformar el sistema original en uno equivalente triangular, a resolverse con sustitución inversa. Utilizando la descomposición LU de la matriz A se procedió a la resolución según los siguientes pasos:

$$Ax = b \rightarrow A = LU \rightarrow LUx = b$$

Proponemos $Ux = y$, luego $Ly = b$, siendo este sistema resuelto por sustitución directa (hacia adelante), y una vez obtenido y , resolvemos $Ux = y$ por sustitución inversa (hacia atrás).

Para la resolución en forma iterativa se usaron tres métodos distintos. Todo método iterativo comienza con una aproximación de la solución, en este caso se usó un vector con valor cero en todas sus componentes. Para cada paso de iteración se genera un vector x que va convergiendo gradualmente a la solución. Se definió un error de tolerancia y un número máximo de iteraciones con la finalidad de que el programa pueda concluir las operaciones y los ciclos tengan definido tanto principio como final.

El método de Jacobi consiste en usar fórmulas como iteración de punto fijo. A efectos prácticos si el algoritmo se detiene después de un número finito de pasos se llega a una aproximación al valor de x de la solución del sistema. Se toma una aproximación inicial $x^{(0)}$ y se construye una secuencia de aproximaciones utilizando la expresión:

$$x^{(k+1)}_i = \frac{(-\sum_{j \leq n; j \neq i} a_{i,j} * x_j^{(k)} + b_i)}{a_{i,i}} \quad i = 1, 2 \dots n$$

Tomando límite con $k \rightarrow \infty$, si el límite existe, el método converge y se obtiene una solución del sistema.

Cabe destacar que al calcular $x_i^{(k+1)}$ se necesitan todos los elementos en $x^{(k)}$, excepto el que tenga el mismo i . Al contrario que en el método Gauss-Seidel, no se puede sobrescribir $x_i^{(k)}$ con $x_i^{(k+1)}$, ya que su valor será necesario para el resto de los cálculos. En el método de Gauss-Seidel se utilizan los valores mejorados apenas calculados (lo que en general acelera el proceso de convergencia), es decir:

$$x^{(k+1)}_i = \frac{(-\sum_{j<i} a_{i,j} * x_j^{(k+1)} - \sum_{j>i} a_{i,j} * x_j^{(k)} + b_i)}{a_{i,i}} \quad i = 1, 2 \dots n$$

La diferencia entre este método y el de Jacobi es que, en este último, las mejoras a las aproximaciones no se utilizan hasta completar las iteraciones.

Se puede introducir una modificación al método de Gauss-Seidel obteniendo una mejora sustancial en la velocidad de convergencia. Si reescribimos la ecuación del método como

$$x^{(k+1)}_i = x^{(k)}_i + r^{(k)}_i$$

Donde $r^{(k)}_i$ es el residuo de la i -ésima ecuación:

$$r^{(k)}_i = \frac{(-\sum_{j<i} a_{i,j} * x_j^{(k+1)} - \sum_{j\geq i} a_{i,j} * x_j^{(k)} + b_i)}{a_{i,i}}$$

El método de Sobrerrelajaciones Sucesivas o SOR se caracteriza por la fórmula:

$$x^{(k+1)}_i = x^{(k)}_i + \omega * r^{(k)}_i$$

El parámetro de relajación ω debe ser elegido de modo que permita obtener la mayor velocidad de convergencia y debe cumplirse que $0 < \omega < 2$, además de que cuanto más se acerca a 2 más inestable es.

El SEL del presente trabajo se resolvió mediante la aplicación de todos los métodos mencionados previamente, siendo estos programados en lenguaje de Octave.

Resultados y Conclusiones

Para obtener la solución del SEL operamos con las siguientes instrucciones en Octave:

Mediante la función generarSEL definimos la matriz A y el vector b en función de la malla propuesta, con número de filas y columnas igual a 11.

```
[A, b] = generarSEL(11, 11)
```

Luego, llamamos x_i para $i=1...4$ a las soluciones obtenidas con cada uno de los métodos. Para los métodos iterativos se propuso un error de tolerancia de 10^{-4} , y un número máximo de iteraciones de 1000.

```
x1 = resolverPorMetodoDirecto(A, b)
```

```
x2 = resolverPorJacobi(A, b, 0.0001, 1000)
```

```
x3 = resolverPorGaussSeidel(A, b, 0.0001, 1000)
```

```
x4 = resolverPorSOR(A, b, 0.0001, 1000, 1.0002)
```

Adicionalmente, se generó la solución mediante la resolución del SEL con la función de Octave $A \setminus b$, que permite la resolución del sistema lineal en forma numérica.

```
c = A \ b
```

Esta solución se utilizó para comparar los resultados obtenidos mediante los métodos programados, y obtener conclusiones acerca de la precisión de los mismos. Mediante el uso de la función de Octave norm se calculó la norma infinito de la diferencia de cada uno de los vectores solución con la solución c, y con eso tener noción del error cometido en cada caso.

```
ni = norm (c - xi, Inf)
```

Cabe destacar que la solución mediante el método directo fue instantánea y en el caso de los métodos iterativos Octave demoró algunos segundos en completar la ejecución. La resolución por el método de Jacobi fue la que más demoró en ejecutarse (aproximadamente el doble del tiempo que tardó la resolución por el método Gauss Seidel, y a su vez el triple del tiempo que tardó la resolución por SOR). La ejecución por Jacobi tardó aproximadamente 15 segundos.

En relación a los errores obtenidos, con el método directo se obtuvo la mejor solución, con una cota de error del orden de 10^{-16} . Con los métodos iterativos el error fue sensiblemente mayor, obteniendo errores del orden de 10^{-4} tanto como para Jacobi y Gauss-Seidel. Sin embargo la cota para Jacobi fue el doble de la obtenida para Gauss-Seidel. Con el método SOR, para un $\omega = 1.0002$ el error fue del orden de 10^{-4} , y si se continúa buscando un ω más aproximado el error puede continuar haciendo más pequeño.

Gráficos comparativos

A continuación se presentan algunos gráficos que permiten visualizar mejor las soluciones obtenidas con los distintos métodos, notar que se han utilizado diferentes colores para diferenciar cada solución, los cuales son:

b de la solución → negro

b por método directo → amarillo

b por Jacobi → azul

b por Gauss-Seidel → rojo

b por SOR → Magenta

Gráfico de la solución b verdadera:

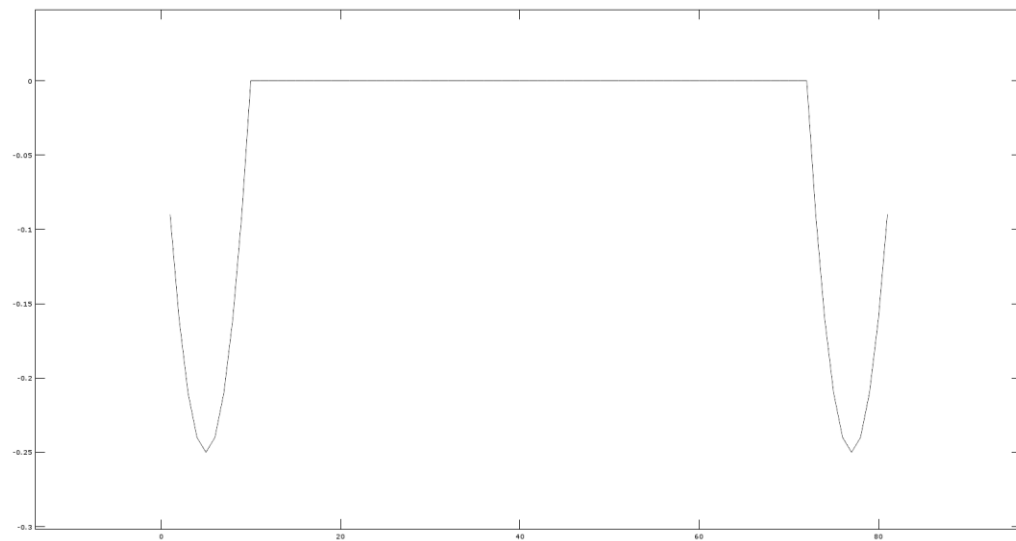


Gráfico de todas las soluciones visto sin zoom:

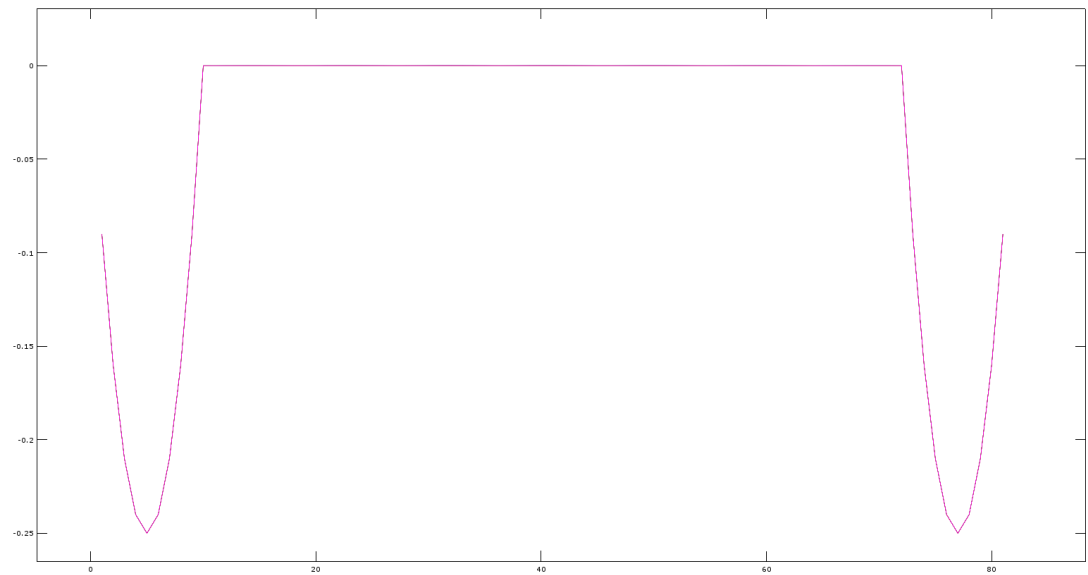


Gráfico de todas las soluciones visto con zoom:

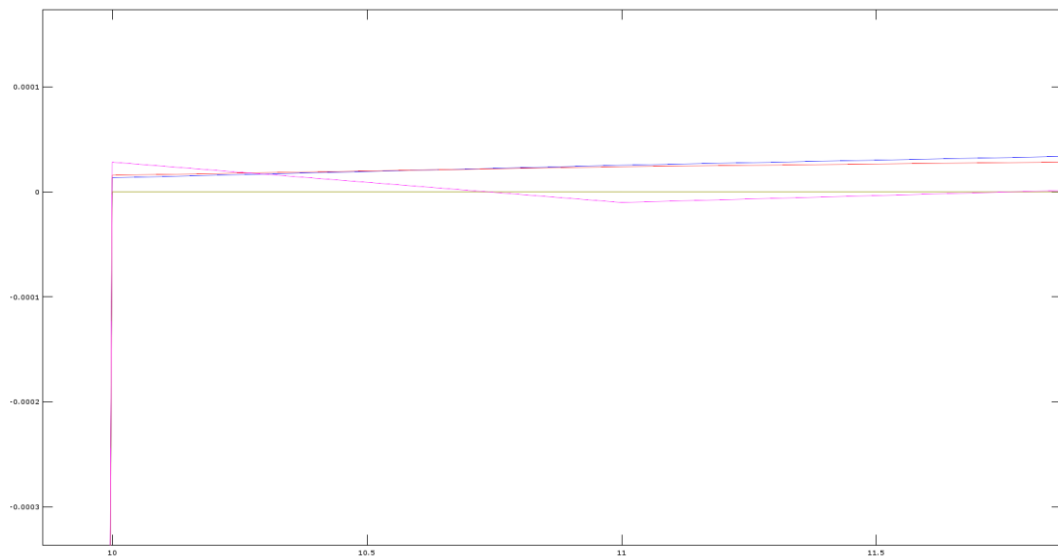


Gráfico comparativo de solución verdadera con solución obtenida por método directo:

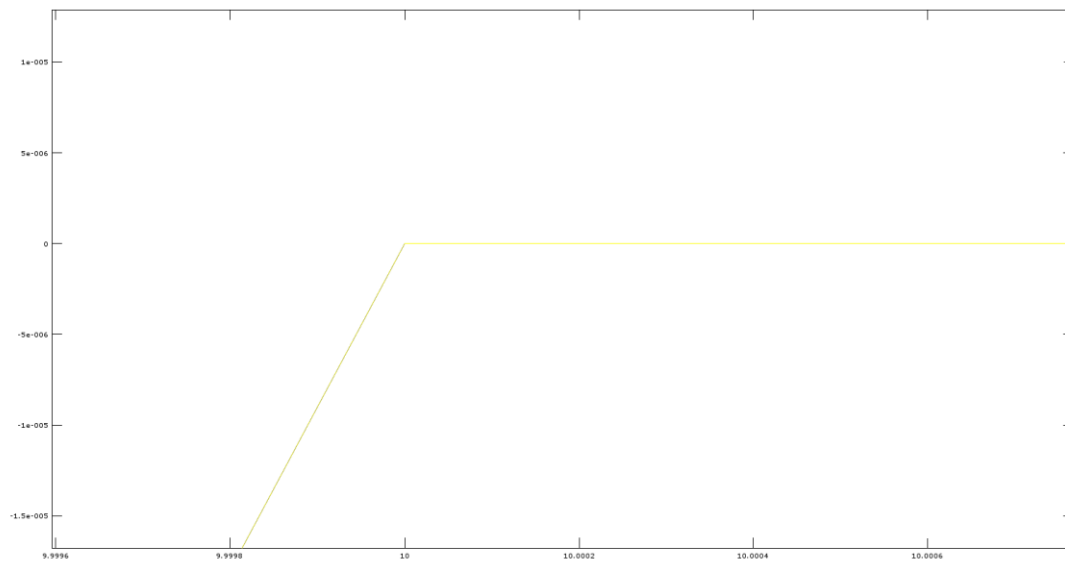


Gráfico comparativo de solución verdadera con solución obtenida por Jacobi:

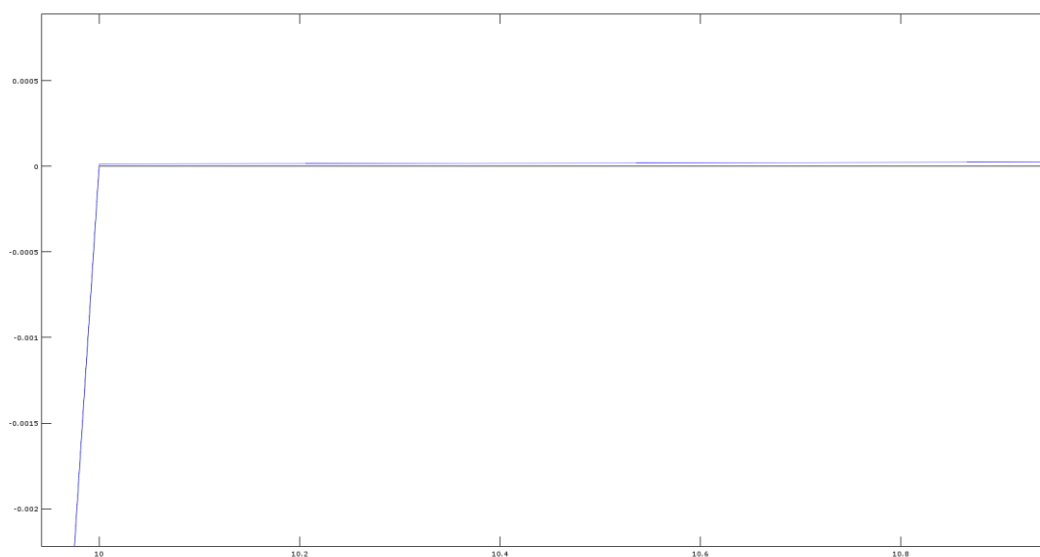


Gráfico comparativo de solución verdadera con solución obtenida por Gauss-Seidel:

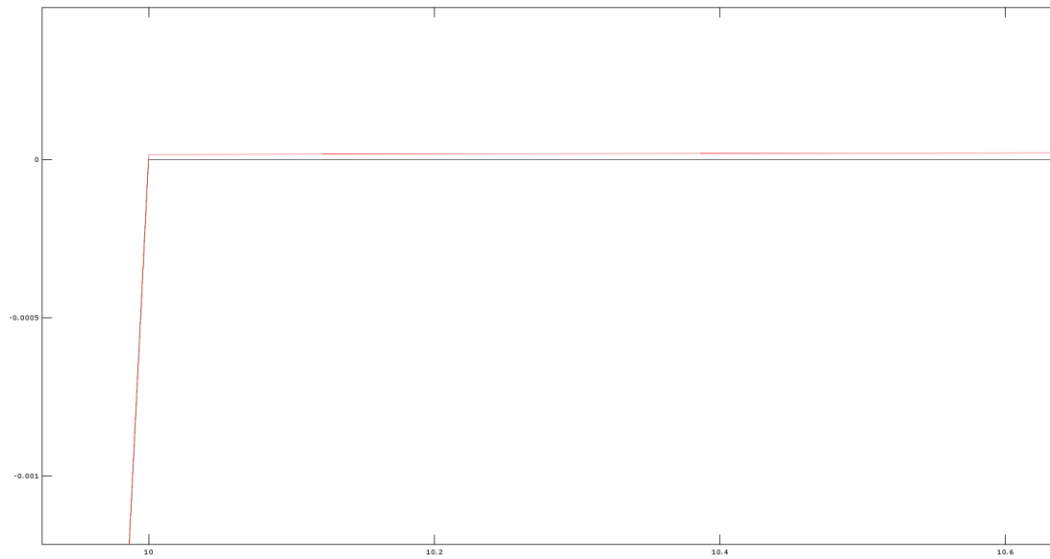
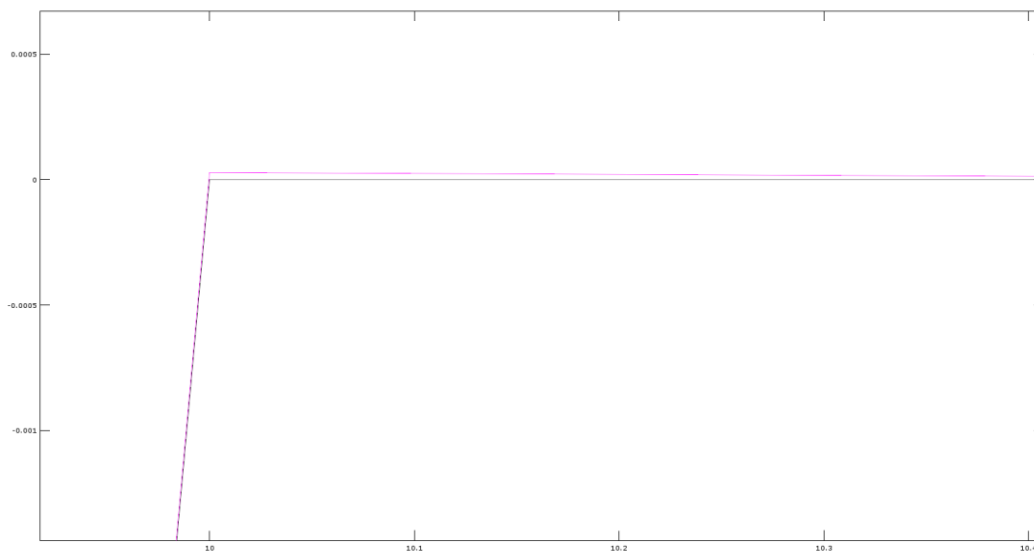


Gráfico comparativo de solución verdadera con solución obtenida por Sobrerrelajamiento:



Referencias

- BURDEN Richard L., FAIRES J. Douglas. *Análisis numérico*. 7ª Ed. México: Thomson Learnig, 2002. 839 págs. ISBN 970-686-134-3.
- GONZALEZ Hernán, *Análisis numérico Primer Curso*. 2da Ed. Buenos Aires: Nueva Librería, 2011. 404 págs. ISBN 978-987-1104-90-1.
- [GNU Octave] <https://www.gnu.org/software/octave/doc/v4.0.1/index.html#Top>