

Guía para aprender Matlab - MAT237 CÁLCULO NUMÉRICO (1º período 2018)

Es una adaptación de la guía del Prof. Carlos H. Q. Forster (Instituto Tecnológico de Aeronáutica – Brasil)

Iniciar el Matlab, escribir los siguientes comandos y descubrir lo que sucede en cada caso.

<p>1. Matlab como calculadora</p> <pre>2+2; 2+2 ans disp(5) 2*ans x=2 exp(x) x=x+1 sin(pi/2) y=[1 2 3; 4 5 6; 7 8 9] who whos why</pre> <p>2. Comandos del área de trabajo</p> <pre>help exp lookfor float help diary pwd mkdir mat237 cd mat237 diary diario.txt diary on help pinv type pinv disp('Hello world') % this is a comment diary off edit diario.txt save datos clear whos load datos whos pause pause(3) quit</pre>	<p>3. Punto flotante</p> <pre>help format format long e 0.12-0.1 eps realmax realmin format rat pi 1: 30 format compact 1: 30</pre> <p>4. Manipulación de matrices</p> <pre>vector=[1 2 -2 4 1] vectcol=[1; 2; -2; 4; 1] hcat=[vectcol, vectcol] vcac=[vectcol; vectcol] vector' vector*vectcol vectcol*vector 2*vector norm(vector) vector.*vector matriz=[1 1 3; 3 4.0 2; 1 5 1] matriz' matriz*matriz matriz^2 matriz.*matriz matriz.^2 inv(matriz) det(matriz) trace(matriz) eig(matriz) rank(matriz) size(matriz) eye(3) zeros(3) ones(3,4)</pre>
--	--

<p>5. Intervalos e índices</p> <pre> matriz(2,3) matriz(2,3)=7 matriz(1,:) matriz(1,:)=[3 4 5] matriz(1:2, 2:3) matriz(:,2) matriz(2:end, 1:end-1) matriz(:) 1:10 1:0.5:10 10:-0.5:1 linspace(1, 10, 13) logspace(1, 2, 10) max(1:10) prod(1:10) sum(1:10) mean(1:10) max(matriz) max(max(matriz)) sum(sum(matriz)) sin(0:pi/16:pi) 6. Gráficos t=0:pi/16:2*pi plot(cos(t),sin(t)) axis square grid title('grafico de la circunferencia') whitebg('b') xlabel('eje x') ylabel('eje y') hold on plot(cos(3*t),sin(t),'m:') plot(cos(t),sin(5*t),'g--') legend('circ','curva 3', 'curva 5') figure subplot(2,1,1); plot(t,cos(t),'ro'); title('coseno'); subplot(2,1,2); plot(t,sin(t),'ro'); title('seno'); [x,y]=ginput(1) </pre>	<p>7. Gráficos en 3 dimensiones</p> <pre> figure; [x,y]=meshgrid(-1:0.4:2, 0:0.4:4) d= sqrt(x.^2+y.^2); z=sin(d)./d; surf(x,y,z); figure; [x,y]=meshgrid(-10:0.4:20, 0:0.4:40); d= sqrt(x.^2+y.^2); z=sin(d)./d; surf(x,y,z); pcolor(x,y,z); mesh(x,y,z); surfc(x,y,z,abs(cos(d))); colormap(bone); plot3(x,y,z) contour3(x,y,z) </pre> <p>Rote los gráficos utilizando los controles de la ventana gráfica. Exporte los gráficos como imágenes BMP, por ejemplo.</p> <p>8. Scripts</p> <p style="text-align: center;">edit grafico.m</p> <p>Copie la lista de comandos de los itens 6 ó 7, péguelos en el editor del archivo grafico.m y guarde. Ahora, basta llamar:</p> <p style="text-align: center;">grafico</p> <p>Vea el manual de los comandos estructurados:</p> <pre> help if help while help relop help for </pre> <p>Observe que, en el editor de Matlab, el script puede ser ejecutado paso a paso y se pueden consultar las variables.</p>
---	--

<p>9. Funciones</p> <p>type factorial</p> <p>Vea como se define la función factorial. El valor de retorno es atribuido a una variable de retorno definida en la primera línea del archivo. La función puede retornar más de un valor u objeto:</p> <p>help function</p> <p>No se hace un control de parámetros de entrada, que deben ser testados por el propio programa (ver nargin e narginout). Intente dejar la función más general, que pueda aceptar vectores y matrices como parámetros, y no apenas valores.</p> <p>Intente reescribir la función factorial para que pueda recibir matrices como parámetros.</p> <p>help global</p> <p>Las variables definidas dentro de funciones tienen ámbito definido dentro de la función. Las variables globales deben ser declaradas como global dentro y fuera de la función.</p> <p>Vea ahora una función más compleja que realiza la eliminación gaussiana con pivoteamiento parcial de una matriz.</p> <p>type rrefmovie</p>	<p>10. Observaciones sobre el Matlab</p> <p>Experimente crear un script con la siguiente secuencia de comandos:</p> <pre>clear; tic; for i=1:1000 for j=1:1000 f(i,j)=5; end; end; toc</pre> <p>Vea el tiempo que llevó construir una matriz. Ahora pruebe nuevamente con:</p> <pre>clear; tic; f=zeros(1000,1000); for i=1:1000 for j=1:1000 f(i,j)=5; end; end; toc</pre> <p>Y nuevamente con:</p> <pre>clear; tic; f=5*ones(1000,1000); toc</pre> <p>Por lo tanto, se deben evitar operaciones de manipulación de la memoria (como la extensión del tamaño de vectores), y lazos de repetición muy internos (como el segundo for del primer script de arriba). Preferentemente se deben usar operaciones de matrices, buscar con “find”, sumar con “sum”, comparar con “any” y “all”, entre otras, en vez de construir lazos de repetición.</p>
---	--