1 SCILAB

Scilab è un prodotto free sviluppato dall'INRIA per applicazioni di controllo di sistema e signals processing.

E' composto di tre parti distinte:

- un interprete
- librerie di funzioni (Scilab procedure)
- librerie di routines in C o Fortran

Una caratteristica chiave di Scilab risulta essere la semplicità di manipolazione delle matrici: operazioni basilari sulle matrici , come concatenazione, estrazione o trasposizione vengono realizzate come semplici operazioni quali addizioni e sottrazioni.

Anche polinomi e matrici polinomiali sono definite e la sintassi usata per manipolare questi oggetti è la stessa di quella usata per vettori costanti e matrici numeriche.

Scilab fornisce una varietà di primitive per l'analisi di sistemi non-lineari. Scicos toolboox permette la definizione grafica e la simulazione di complesse interconnessioni di sistemi ibridi.

Infine, Silab è facilmente interfacciabile con funzioni Fortran o C.

Dopo aver lanciato il programma è possibile editare i commandi dal prompt o lanciare il file in cui sono stati precedentemente scritti i comandi d'interesse (vedi Fig. 1). Affinchè il file venga eseguito correttamente è necessario spostarsi nella directory dove si trova il file,come mostrato nelle figura seguente.

Per eseguirlo basta selezionare " Exec" da "File" e poi scegliere il file da eseguire.

Scilab è fornito di un HELP soddisfacente. Per avere informazioni su qualsiasi funzione chiamare l'Help (come in Fig. 2) e nel campo "Aprop" digitare la funzione d'interesse.

Si può notare che mano a mano che si digitano le lettere nella finestra sopra vengono visualizzate tutte le voci che iniziano con le lettere che stiamo digitando . Ciò è utile quando non si conosce esattamente il nome della funzione che realizza l'operazione di nostro interesse.

Si può anche scegliere un argomento generale (es. Graphic Library) e vedere le funzioni relative.

Selezionata la voce cliccare su "SHOW" e verrà visualizzata una pagina contenente tutte le informazioni necessarie.

Un manuale on-line e alcuni esempi possono essre trovati ai seguenti indiriizzi:

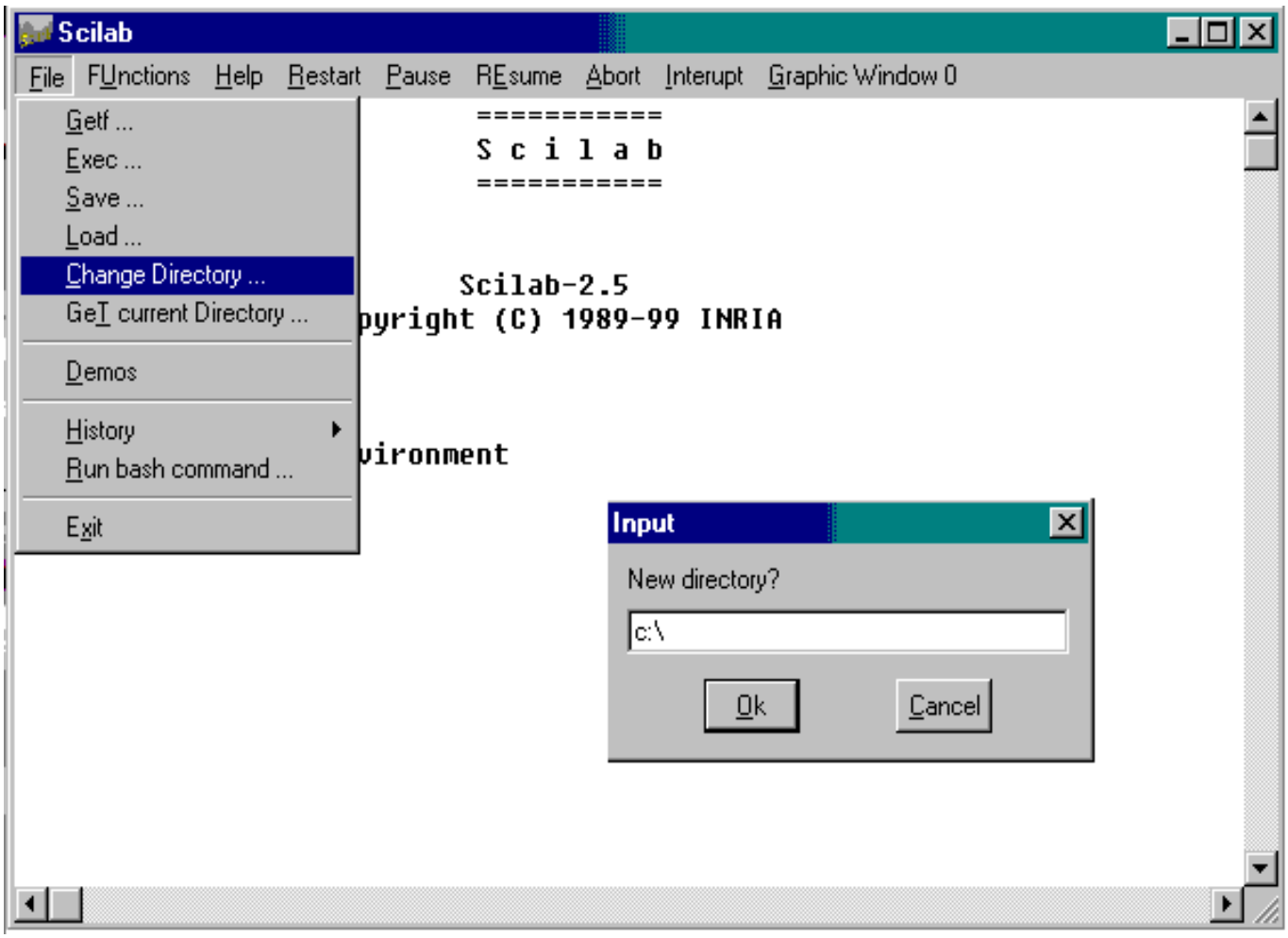

Figura 1: Scelta della Directory di lavoro

- **http://www-rocq.inria.fr/scilab/doc/intro/intro.html**
- **http://poincare.dma.unifi.it/scilab/demos.html**

2 AMBIENTE

Scilab viene caricato con una serie di variabili e primitive. Il comando *who* visualizza la lista delle variabili già definite

Le variabili possono essere salvate in un file esterno utilizzando *save* e queste variabili possono essere caricate in Scilab utilizzando il comando *load*.

Si noti che dopo il comando *clear* < *nome_variabile* > la viaribile non esiste più nell'ambiente in uso e con il comando *clear* senza alcuna variabile si cancellano tutte le viariabili presenti nell'ambiente.

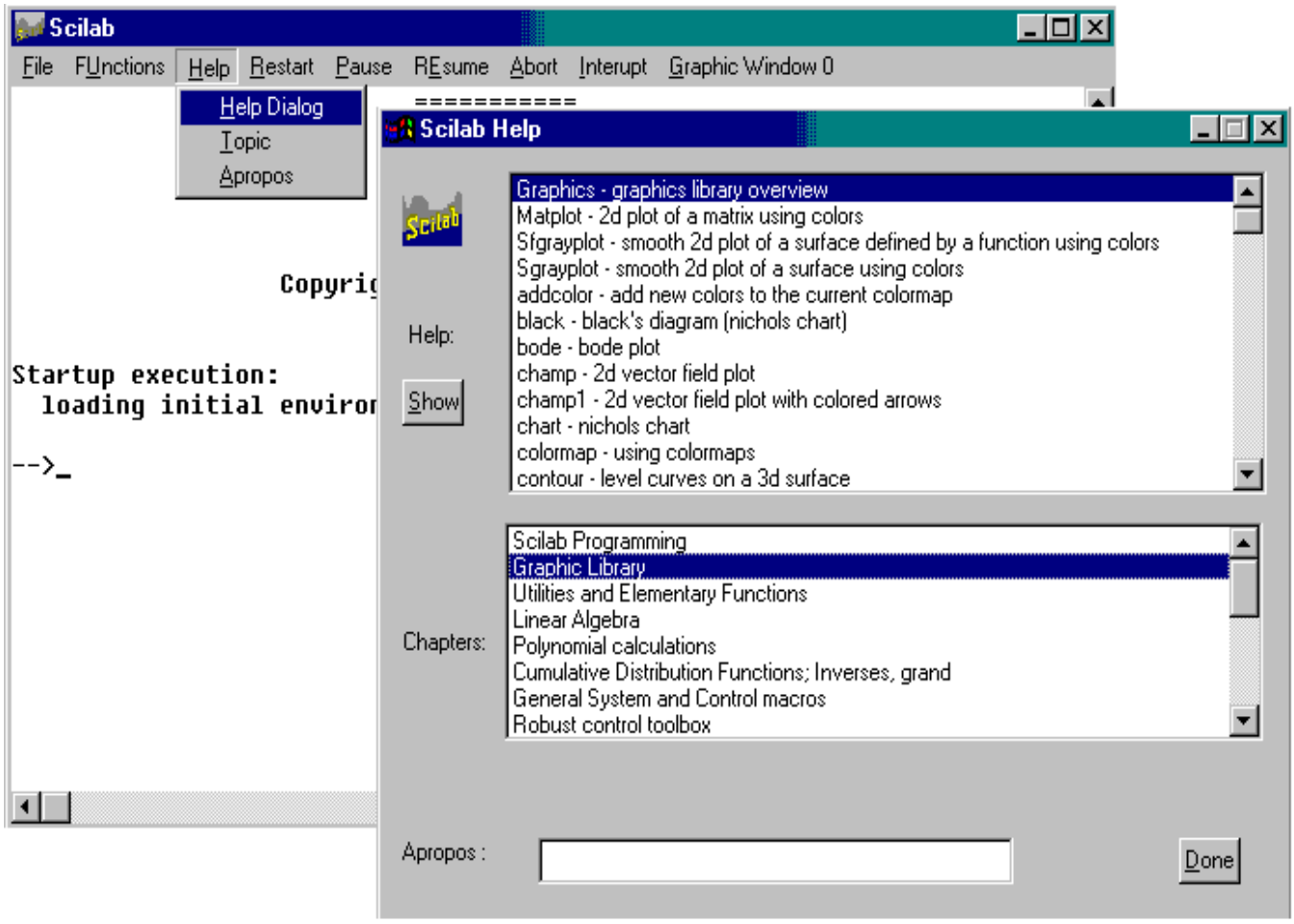

Figura 2: Help

3 MATEMATICA

3.1 SCALARI

Assegnazione di un valore ad una variabile:

$$a = 1;$$

Con il simbolo '==' viene restituito un booleano,senza che venga assegnato il valore:

$$a == 1$$

$$ans = \ T$$

Per gli scalari sono definite tutte le operazioni.

3.2 POLINOMI

Per definire un polinomio si fa uso della funzione 'poly':

$$p = poly([0\ 2\ 3],'x');$$

definisce un polinomio con variabile letteraria x e radici in zero,due e tre,con il coefficiente del termine di grado massimo sempre pari ad uno.

Se non si conoscono le radici allora si definisce la variabile letteraria di nostro interesse (ad es. 'x') come un polinomio con radice in zero:

$$x = poly(0,'x');$$

Si può ora scrivere qualsiasi polinomio in x nel seguente modo:

$$p = 2 + 3 * x + x^2$$

Per conoscere le radici del polinomio p appena dichiarato si utlilizza la funzione 'roots':

$[r] = roots(p)$
$r =$
$-1.$
$-2.$

sono possibili tutte le normali operazioni fra polinomi: divisioni, moltiplicazioni,addizioni e sottrazioni come fra scalari

$r = x/p$

$$r = \frac{x}{2 + 3x + x^2}$$

$r = x * p$

$$r = 2x + 3x^2 + x^3$$

3.3 MATRICI

DEFINIZIONE DI UN VETTORE

$$vet = [0 : 1 : 10]$$

Definisce un vettore con elementi da 0 a 10 con passo 1

$$vet = 012345678910$$

Nella seguente tabella vengono riportate le possibili operazioni sulle matrici:

[ ]	matrix definition, concatenation
;	row separator
( )	extraction m=a(k)
( )	insertion: a(k)=m
'	transpose
+	addition
-	subtraction
*	multiplication
\	left division
/	right division
^	exponent
.*	elementwise multiplication
.\	elementwise left division
./	elementwise right division
.\^	elementwise exponent
.*.	kronecker product
./.	kronecker right division
.\.	kronecker left division

DEFINIZIONE DI UNA MATRICE

$A = [123; 479; 1830]$

$A =$

1.2.3.

4.7.9.

18.3.0.

DETERMINANTE

$det(A)$;

INVERSA

$invr(A)$;

oppure

$inv(A)$;

ESTRAZIONE DELLA PRIMA RIGA

$A(1,:)$

$ans = 1.2.3.$

ESTRAZIONE DELL'ULTIMA COLONNA

$A(:,\$)$

$ans =$

3.

9.

0.

4 GRAFICI

Per visualizzare tutte le funzioni utilizzabili per la visualizzazione di un grafico nella finestra dell'Help evidenziare la voce " Graphic Library". In generale le funzioni utilizzate sono le seguenti,per ognuna vanno specificati il vettore delle x (ascisse) e quello delle y (ordinate):

- plot(x,y)
- plot2d(x,y)
- plot2d1(x,y) //assi logaritmici

Per aprire una seconda finestra grafica basterà dare i comandi

$xset('window', 2); xselect(2);$

che rispettivamente istanziano una seconda finestra e la selezionano. Per tornare alla prima finestra (che si apre automaticamente con l'uso dei comandi di plot visti prima) basterà impartire il comando

$xselect(1)$

Invece, l'uso della funzione $xbasc()$ permette di "pulire" l'eventuale finestra-grafico già aperta evitando così la sovrapposizione di più grafici.

5 FUNZIONI

Le funzioni sono insiemi di comandi che vengono eseguiti in un nuovo ambiente isolando così le variabili delle funzioni da quelle iniziali. Esse possono essere create e eseguite in diversi modi. Alle funzioni possono essere passati dei parametri;esse possono contenere cicli $while$, for..., possono essere richiamate da altre funzioni...

Per definire una funzione direttamente dal prompt di Scilab si usa il comando $deff$

Esempio

$$deff('[x] = foo(y)','if \quad y > 0 \quad then, x = 1; else, x = -1; end')$$

$foo(5)$

$ans = 1.$

$foo(-3)$

$ans = -1.$

E' però più comodo usare un editor di testo e caricare la funzione in Scilab con $getf('filename')$. La prima riga del file dev essere : $function[y1, ..., yn] = nome\_funzione(x1, ..., xk)$ dove le yi sono gli output e le xi sono le variabili di input. Ovviamente, il nome della funzione che dovrà essere utilizzato è quello definito come $nome\_funzione$ e non il nome del file. Un file, peraltro, può contnere più definizioni di funzioni.

- CICLI

  in Scilab esistono due tipi di cicli : for e while.

  - FOR

    $$x = 1;\ for\ k = 1 : 4,\ x = x * k,\ end$$

    x =1.
    x =2.
    x =6.
    x =24.
  - WHILE

    Il ciclo while esegue ripetutamente una sequenza di comandi fino a quando la condizione non è soddisfatta

    $$x = 1;\ while\ x < 14,\ x = 2 * x,\ end$$

x=2.

x=4.

x=8.

x=16.

- ISTRUZIONI CONDIZIONALI

  Si possono utilizzare due tipi di istruzioni condizionali : "IF-THEN-ELSE" e "SELECT-CASE"

  - IF THEN ELSE

    x=1;

    $$if\ x\ 0then,\ y = -x,\ else,\ y = x,\ end$$

    y= - 1.

    x=-1;

    $$ifx > 0then,\ y = -x,\ else,\ y = x,\ end$$

    y= - 1.

  - SELECT CASE

    x=-1

    x =- 1.

    $$select\ x,\ case\ 1,\ y = x + 5,\ case\ -1,\ y = sqrt(x),\ end$$

    y= i

6 CONTROL TOOLBOX

Scilab è provvisto di molteplici funzioni utili nell'analisi dei segnali

DEFINIZIONE DI UN SISTEMA LINEARE (Funzione di trasferimento)

$s = poly(0,'s')$

$sl = syslin('c', 1/(s * s + 0.2 * s + 1))$

sl è un istema lineare continuo (la lettera c sta ad indicare proprio che si tratta di un sistema continuo; per il caso discreto si deve specificare d)

$$sl = \frac{1}{s * s + 0.2 * s + 1}$$

Del sistema lineare possono essere visualizzati il diagramma di BODE, di NYQUIST e di NICHOLS utilizzando le seguenti funzioni:

- bode(sl)
- nyquist(sl)
- chart(list(1,0,2,3))
- black(sl,0.01,100)

Gli ultimi due servono per graficare la carta di Nichols e il diagramma rispettivamente.
Tutte le funzioni graficano direttamente i rispettivi diagrammi, come si può vedere dalla Fig. 3:

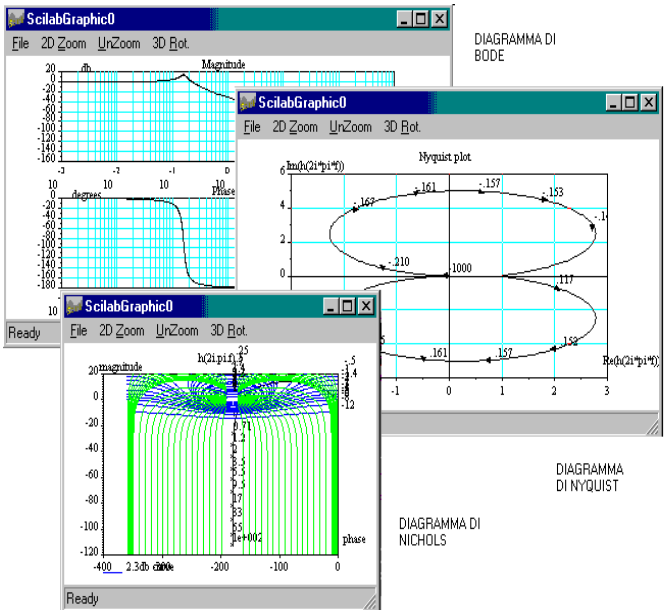

Figura 3: Diagrammi di Bode Nyquist Nichols

Per gli esempi che seguono definiamo un vettore degli istanti di tempo :

instants=0:0.05:20;
vettore degli istanti di tempo

- risposta al gradino

  $y = csim('step', instants, sl);$
- risposta al gradino traslato

  $deff('[in] = u(t)','if \quad t < 3 \quad then \quad in = 0; else \quad in = 1; end');$

  $y1 = csim(u, instants, sl);$

  $plot2d(instants', y1');$

- Risposta impulsiva
  $yi = csim('imp', instants, sl);$
  $xbasc();$
  $plot2d(instants', yi');$
- Discretization
  $dt = 0.05;$
  $ld = dscr(tf2ss(sl), 0.05);$
  Dove sl è il sistema nel tempo continuo

EQUAZIONI DIFFERENZIALI

E' possibile risolvere sistemi di equazioni differenziali del tipo

$\dot{y} = f(t, y), y(t0) = y0$

dove y può essere anche un vettore, mediante l'uso della funzione *ode*;

$y = ode(y0, t0, t, f)$

dove:

- y0 : vettore che esprime le condizioni iniziali
- t0 : istante iniziale
- t : vettore degli istanti di tempo per i quali è valutato y
- f : funzione esterna di cui è stata definita la sintassi

Per maggiori informazioni vedere l'Help sulla voce *ode*

Esempio

$\dot{y} = y^2 - ysin(t) + cos(t), y(0) = 0$

$deff('[ydot] = f(t, y)', 'ydot = y^2 - y * sin(t) + cos(t)')$

f è il nome della funzione ; t ed y sono le variabili di input di f, e viene restutito $ydot$ che contiene le derivate;

$y0 = 0; t0 = 0; t = 0 : 0.1 : \%pi;$

$y = ode(y0, t0, t, f)$

$plot(t, y)$

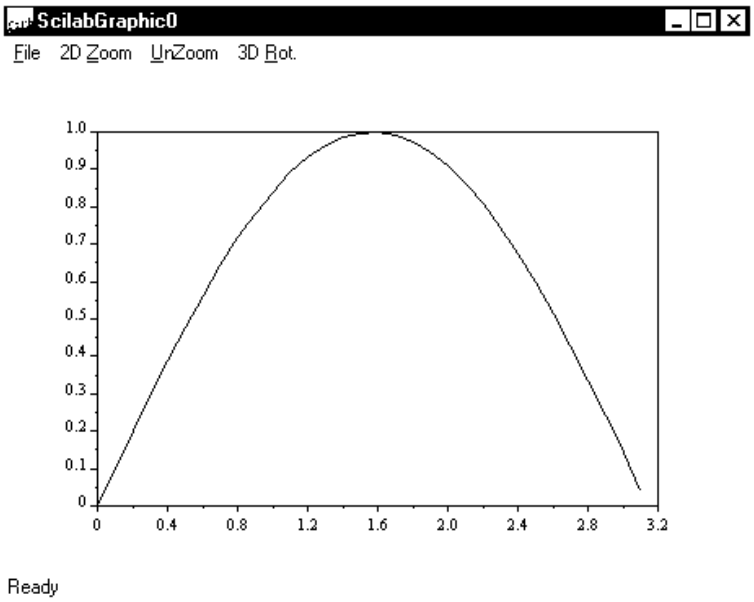

Figura 4: Grafico della funzione calcolata con *ode*

7 INTERFACCIA CON PROGRAMMI C E FORTRAN

Scilab può interfacciarsi con programmi scritti in C o in Fortran. Naturalmente la prima cos ada fare è un link a questi programmi . Per ciò si usa il comando *link* seguito dal comando $fort$ che trasmette le variabili definite in scilab (Matrici, stringhe..) ai programmi linkati e trasforma in variabili di Scilab gli output di tali funzioni

Il comando $link('path/pgm.o','pgm', flag)$ connette il programma già compilato pgm a Scilab. Qui pgm.o é un object file che si trova nella directory path e pgm è il punto d'ingresso (program name) nel file pgm.o. Il valore di $flag$ deve essere settato a 'C' per un programma in C e a 'F' per il Fortran. ('F' è di default e può essere omesso). Se l'operazione di link è OK Scilab ritorna un intero n associato a questo programma . Per eliminare il link digitare $ulink(n)$. Il comando c_link('pgm') ritorna true se pgm correttamente linkato a Scilab e false altrimenti. Per maggiori informazioni vedere l' help di *link* e di $fort$.