Utilizzo dei lua script

Nicola Bianchi and Michele Dai Pré*

January 26, 2005

Introduzione

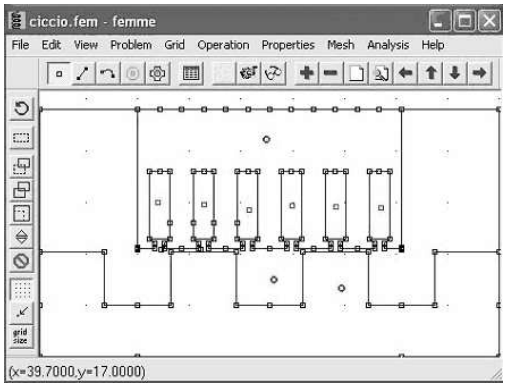
Lo scopo di questa dispensa quello di spiegare l'uso dei file *.lua utili alla creazione di cicli iterativi per il programma femm. Questo tipo di programmazione utile quando si vogliono eseguire molte simulazione al variare di di un parametro, che altrimenti richiederebbero un elevato intervento manuale; per esempio la frequenza o la posizione di un elemento geometrico.

Creazione della geometria

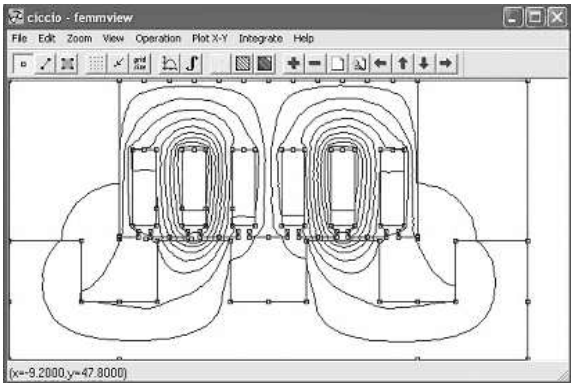
La geometria pu essere creata direttamente attraverso **femm editor**, con **AutoCAD** oppure con un file *.lua di disegno. Successivamente i file di automatizzazione non andranno a variare la geometria.

Creazione di un file lua

Per scrivere il programma non esiste un editor quindi si aprono due nuovi file con il **Blocco Note** di **Windows**, uno per il **pre-processor** Fig.1(a) e uno per il **post-processor** Fig.1(b).



(a) Pre-processor



(b) Post-processor

Figure 1:

Il linguaggio di programmazione molto simile al Basic e C. Sono riportati di seguito alcuni esempi di programmi con spiegazione e un file con il manuale dei comandi Lua.

*Dipartimento di Ingegneria Elettrica, Universitá di Padova

Programma per il pre-processor

```
for n=0,6,1 do
   dt=30*n
   iai=200*sin(90+dt)
   iaf=-iai
   ibi=200*sin(dt-30)
   ibf=-ibi
   ici=200*sin(dt-150)
   icf=-ici
```

Questa parte del programma serve per modificare i dati del problema.Nel caso esemplificato viene cambiata la fase della corrente attraverso un "ciclo for".

```
   openfemmfile("motorelineare1.fem")

   modifycircprop("fase1+",1,iai)
   modifycircprop("fase2+",1,ibf)
   modifycircprop("fase3+",1,icf)
   modifycircprop("fase1-",1,iaf)
   modifycircprop("fase2-",1,ibi)
   modifycircprop("fase3-",1,ici)
```

La seconda parte assegna i valori calcolati nei passaggi precedenti, al disegno fatto prima. Vengono utilizzati dei comandi propri del femm editor che si possono trovare nel men Help di femm, Help topics e quindi nella cartella Lua Scripting Documentation.

```
  savefemmfile("motoretemp.fem ")
   analyse()
```

Viene salvata una copia del file "motorelinare1.fem" con le modifiche apportate in "motoretemp.fem". Questo viene fatto per mantenere inalterate le condizioni poste inizialmente. Il comando successivo (analyse) fa partire il calcolo agli elementi finiti.

```
   handle=openfile("tempfile","w");
   write (handle,dt,"\n");
   closefile(handle);
```

Questa parte indispensabile al funzionamento del programma e va sempre inserita. Essa crea un file temporaneo di scambio tra i dati del pre-processor e quelli del post-processor.

```
   runpost("motlinpost.lua")
end
```

Questa funzione richiama il programma "motlinpost.lua" (Il file pu avere qualsiasi nome purch venga messa l'estensione ".lua") che esegue le operazioni del post-processor.

Programma per il post-processor

```
  handle=openfile("tempfile","r")
  dt=read(handle,"*n")
  closefile(handle)
```

La prima parte dello script legge i dati posti nel file temporaneo, creato dal pre-processor ed indispensabile per il funzionamento del programma.

```
seteditmode(contour)
selectpoint(0,20.75)
selectpoint(51,20.75)
zoomnatural()
fx, k, fy, r=lineintegral(3)
```

Viene selezionata una linea, sulla quale viene fatto un integrale per determinare la forza.

```
handle=openfile("datipost.txt","a")
write(handle,dt,"       ",fx,"  ",fy,"\n")
closefile(handle)
```

Questa funzione apre un file di testo dove saranno posti i risultati della simulazione, nel caso specifico i valori di posizione e forza, come nel listato riportato di seguito.

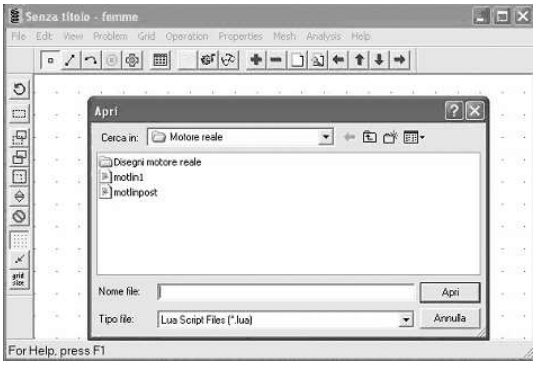
```
exitpost()
```

Con questo comando si esce dal post-processor.

Esecuzione dei file del pre-processor (motlin.lua)

Per eseguire i programmi creati, si deve aprire il programma femm editor e dal men "File" Fig.2(a) selezionare "Open Lua Script". Quindi apparir una finestra dove vi sono i file *.lua (nell'esempio saranno "motlin.lua" e "motlinpost.lua"), posti nella stessa cartella dei file *.fem, che riporta la geometria e i parametri della macchina da simulare.

(a) Barra di femm



(b) finestra di apertura file *.lua

Figure 2:

Si seleziona il file per il Preprocessor (nell'esempio "motlin.lua") e si preme apri. La simulazione parte.

Visualizzazione dei dati

I risultati sono nel file "datipost.txt", pronti ad essere elaborati con un foglio di calcolo o attraverso Matlab. Il formato impostato di uscita il seguente:

0	9,971019874172834	530,7128613192226
3,6	28,19792633342082	536,1208282765389
7,2	45,69312111519717	550,6880777078027
10,8	62,51519266367021	574,0560219623116

Esempi

Attuatore

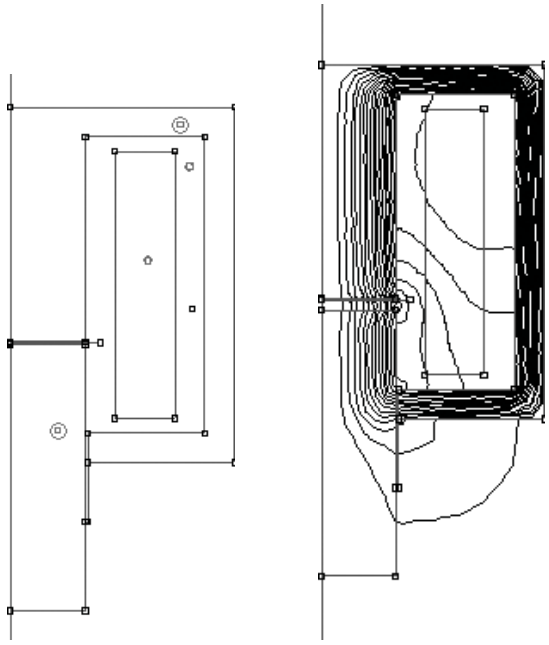
PRE_ATT.LUA

```
--definizione della densit di corrente massima e salvataggio su file
--per impaginazione risultati col post.lua
jm=3
handle=openfile("cont","w");
write(handle,jm,"\n");
closefile(handle)

--simulazioni parametrizzate per ampiezza del traferro.
--dz  la variazione dalla posizione iniziale
for dz=0,-0.6,-0.2 do
   handle=openfile("tempfile2","w");
   write(handle,dz,"\n");
   closefile(handle)

--     simulazioni parametrizzate per densit di corrente j
       for j=1,jm,1 do
       open_femm_file("attuatore.fem")
       modifymaterial("Copper",4,j)
       seteditmode("group")
       selectgroup(1)
       move_translate(0,dz)
       save_femm_file("temp.fem")
       handle=openfile("tempfile1","w");
       write(handle,j,"\n");
       closefile(handle);

--     eventuali pause nelle simulazioni utili per il debug
--     if dz==-0.2 then pause() end

--     lancio simulazione 0=finestra in vista, 1=finestra nascosta
       analyse(1)

--     lancio postprocessor. "-windowhide" lo fa partire
--     con finestra nascosta
       runpost("post_att.lua","-windowhide")
   end
end
```

POST_ATT.LUA

```
-- lettura j
handle=openfile("tempfile1","r")
j=read(handle,"*n")
closefile(handle)

-- lettura dz
handle=openfile("tempfile2","r")
dz=read(handle,"*n")
```

```
closefile(handle)

--lettura contatore, in questo caso jm, utilizzato per
--impaginazione dei risultati
handle=openfile("cont","r")
cont=read(handle,"*n")
closefile(handle)

-- apertura file in cui vengono salvati i risultati
handle=openfile("ris attuatore.txt","a");

--selezione linea di integrazione e calcolo
seteditmode(contour)
selectpoint(0,10.5)
selectpoint(3,10.5)
totflux,angflux=lineintegral(1)

-- eventuale deselezione
--clearcontour()

-- impaginazione risultati in funzione di dz e j
if j==1 then
   write(handle,-dz,"  ")
end
write(handle,totflux," ")
if j==cont then
   write(handle,"\n")
end

-- chiusura file risultati
closefile(handle)

exitpost()
```



(a) Pre-processor (b) Post-processor

Figure 3: Attuatore

Motore trifase monospira

PRE.LUA

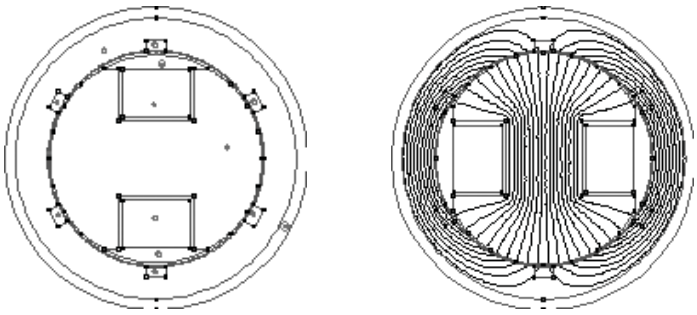
```
 -- N.B. Le seguenti righe di comando sono ottimizzate per Femm 3.3
 -- Il motore  trifase monospira. la profondit  di 1 m. 2p=2.

 for dz=0,180,3 do
    open_femm_file("sinc2p.fem")

 -- Le seguenti righe servono a modificare le caratteristiche dei materiali
 -- in particolare le densit di corrente di degli avvolgimenti di statore.
 -- Il rotore deve essere prima di tutto posizionato con l'angolo di sfasamento
 -- tra i flussi di statore e rotore desiderato.
 -- jrap=2.4*cos(dz)
 -- jrbp=2.4*cos(dz+120)
 -- jrcp=2.4*cos(dz-120)
 -- jram=-jrap
 -- jrbm=-jrbp
 -- jrcm=-jrcp
 -- modifymaterial("ap",4,jrap)
 -- modifymaterial("bp",4,jrbp)
 -- modifymaterial("cp",4,jrcp)
 -- modifymaterial("am",4,jram)
 -- modifymaterial("bm",4,jrbm)
 -- modifymaterial("cm",4,jrcm)

 -- Modalit di selezione
    seteditmode("group")
```

```
    selectgroup(1)

 -- Azione da compiere; in questo caso la rotazione dell'angolo dz
    move_rotate(0,0,dz)
    save_femm_file("temp.fem")

 -- Eventuale pausa per il controllo dei calcoli in corso. In questo caso ci
 -- ci sar una pausa dopo 5 gradi di rotazione del rotore.
 -- if dz==5 then pause() end

 -- Lancio del fkern. Valore: 0 per visualizzare la finestra; 1 per avvio a
 -- icona ridotta.
    analyse(1)
 -- hendle  una variabile di c alla quale  associato fisicamente il file. "\n" a capo.
    handle=openfile("tempfile","w");
    write(handle,dz,"\n");
    closefile(handle);

 -- apertura del post.lua
    runpost("post.lua")
 end
```

POST.LUA

```
 handle=openfile("tempfile","r")
 dz=read(handle,"*n")
 closefile(handle)
 handle=openfile("sinc2p ris.txt","a");

 -- Modalit di selezione
    seteditmode(area)
    selectblock(0,0)

 -- Calcolo della coppia con weighted stress tensor
    ta=blockintegral(22)

 -- Deselezione
    clearblock()

 -- Calcolo del flusso concatenato con una singola spira
    seteditmode(area)
    selectblock(0,53)
    ap=blockintegral(1)
    clearblock()
    selectblock(0,-53)
    am=blockintegral(1)
    s=blockintegral(5)
    clearblock()
    flux=(am-ap)/s
    clearblock()

 -- Calcolo dell'induzione a traferro in punto
    Are,Aim,B1re,B1im,B2re,B2im,
    Sig,E,H1re,H1im,H2re,H2im,
    Jere,Jeim,Jsre,Jsim,Mu1re,Mu1im,Mu2re,Mu2im,
```

```
    Pe,Ph=getpointvalues(49.5,0)

 -- Calcolo della coppia con tensore Maxweel su linea
    seteditmode(contour)
    selectpoint(49.5,0)
    selectpoint(-49.5,0)
    selectpoint(49.5,0)
    tlr,tli=lineintegral(4)
    clearcontour()

 -- Scrittura dei risultati. Rispettivamente: coppia weighted, coppia
 -- Maxwell, differenza tra le due, flusso, Bx, By, |B|.
    write(handle,dz,"   ",ta,"  ",tlr," ",tlr-ta,"  ",flux,"    ",B1re,"    ",
    B2re,"    ",sqrt(B1re*B1re+B2re*B2re),"\n")

 closefile(handle)
 exitpost()
```



(a) Pre-processor (b) Post-processor

Figure 4: Motore trifase monospira

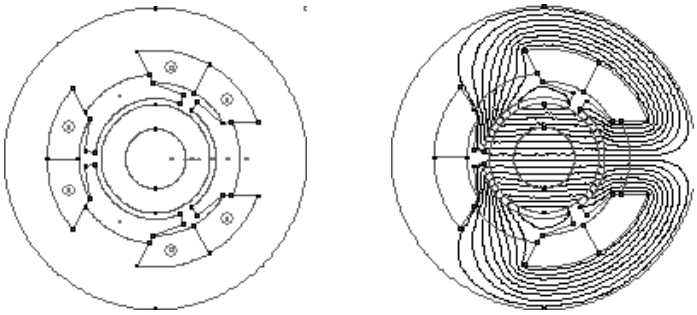
Motore per alte velocitá

```
PRE_COG.LUA
 --Programma per il calcolo della forza e della coppia sul rotore  al variare del
 --suo spostamento
 for n=0,360,1 do
    dz=n
    open_femm_file("Motore.fem")

    seteditmode("group")
    selectgroup(2)
    move_rotate(0,0,dz)

    save_femm_file("tempMotore.fem")
    analyse()

    handle=openfile("tempfile","w");
    write(handle,dz,"\n");
    closefile(handle);
```

```
    runpost("post_cog.lua")
 end
```

POST_COG.LUA

```
 handle=openfile("tempfile","r")
 dz=read(handle,"*n")
 closefile(handle)

 seteditmode("area")
 selectblock(0,0)
 selectblock(0,8)
 coppia=blockintegral(23)
 Forzax=blockintegral(18)
 Forzay=blockintegral(19)

 handle=openfile("coggingRisultato.txt","a");
 write(handle,dz," ",Forzax," ", Forzay," ", coppia,"\n")
 closefile(handle)

 exitpost()
```



(a) Pre-processor (b) Post-processor

Figure 5: Motore per alte velocit

Motore Asincrono trifase

PRE.LUA

```
 --Programma per il calcolo delle induttanze di cava
 for j=1.5,1.5,0.1 do

    handle=openfile("tempfile","w");
    write(handle,j,"\n");
    closefile(handle);

    RepA=j*4800
    RemA=-j*4800
    RepB=-j*2400
    RemB=j*2400
```

```
    ImpB=-j*4161
    ImmB=j*4161
    RepC=-j*2400
    RemC=j*2400
    ImpC=j*4161
    ImmC=-j*4161

    open_femm_file("motore.fem")
    modifycircprop("+Ia",1,RepA)
    modifycircprop("-Ia",1,RemA)
    modifycircprop("+Ib",1,RepB)
    modifycircprop("-Ib",1,RemB)
    modifycircprop("+Ib",2,ImpB)
    modifycircprop("-Ib",2,ImmB)
    modifycircprop("+Ic",1,RepC)
    modifycircprop("-Ic",1,RemC)
    modifycircprop("+Ic",2,ImpC)
    modifycircprop("-Ic",2,ImmC)

    save_femm_file("mot_e.fem")
    analyse()
    runpost("post.lua")

 end
```

POST.LUA

```
 handle=openfile("tempfile","r")
 j=read(handle,"*n")
 closefile(handle)

 seteditmode("area")

 selectblock(-11, 41.1)
 selectblock(0, 43)
 selectblock(11,41.1)
 selectblock(21.4,37)
 lam_p=blockintegral(1)

 clearblock()

 selectblock(-21.4,-37)
 selectblock(-11,-41.1)
 selectblock(0,-43)
 selectblock(11,-41.1)
 lam_m=blockintegral(1)

 handle=openfile("540_results.txt","a");
 write(handle,j," ",lam_p," ",lam_m,"\n")
 closefile(handle)

 exitpost()
```

Vengono presentate delle altre simulazioni per lo stesso motore, con due file per il pre-processor il primo per una singola simulazione e poi per pi cicli (50 simulazioni al variare della frequenza e poi altre 44 sempre al variare della frequenza).

PRE_SINGOLA.LUA

```
-- Singola simulazione

-- fisso la frequenza
  freq=50

-- apro il file .FEM che contiene
-- il disegno del motore asincrono
  open_femm_file("motore.fem")

-- definizione del problema
-- fisso la frequenza di lavoro
  probdef(freq,"millimeters","planar",1e-8,(700))

-- scrivo il file e risolvo
  save_femm_file("temp.fem")
  analyse()

-- scrivo in un file temporaneo il valore della
-- frequenza in modo che sia letta da altri file
  handle=openfile("frequenza.txt","w");
  write(handle,freq,"\n");
  closefile(handle);

-- lancio la soluzione
  runpost("post.lua")
```

PRE.LUA

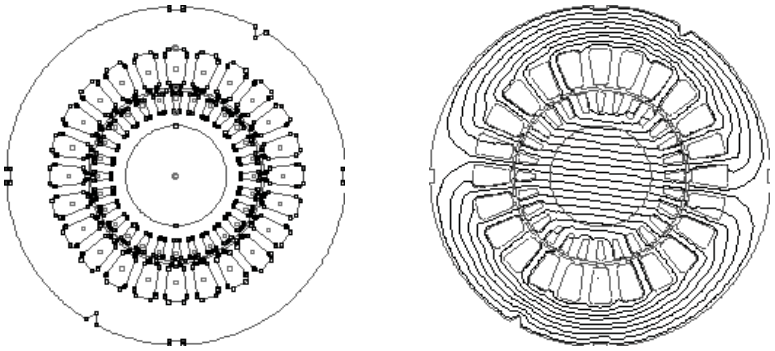
```
-- Ciclo di simulazioni

-- ciclo 1: frequenza 0.01-0.5 Hz
  for k=1,50,2 do
      -- fisso la frequenza
      freq=k/100

      -- apro il file .FEM che contiene
      -- il disegno del motore asincrono
      open_femm_file("motore.fem")

      -- definizione del problema
      -- fisso la frequenza di lavoro
      probdef(freq,"millimeters","planar",1e-8,(700))

      -- scrivo il file e risolvo
      save_femm_file("temp.fem")
      analyse()

      -- scrivo in un file temporaneo il valore della
```

```
      -- frequenza in modo che sia letta da altri file
      handle=openfile("frequenza.txt","w");
      write(handle,freq,"\n");
      closefile(handle);

      -- lancio la soluzione
      runpost("post.lua")
  end

 -- ciclo 2: frequenza 0.6-5 Hz
  for k=6,50,1 do
      freq=k/10
      open_femm_file("coverco540.fem")
      probdef(freq,"millimeters","planar",1e-8,(700))
      save_femm_file("temp.fem")
      analyse()
      handle=openfile("tempfile","w");
      write(handle,freq,"\n");
      closefile(handle);
      runpost("post.lua")
      end
```

POST.LUA

```
 -- Elaborazione di risultati

 -- leggo il file dove e' scritta la frequenza
  handle=openfile("frequenza.txt","r")
  freq=read(handle,"*n")
  closefile(handle)

 -- seleziono l'area del rotore (gruppo 3)
  seteditmode("area")
  groupselectblock(3)

 -- Steady-state weighted stress tensor torque
  t=blockintegral(22)

 -- Total losses
  Prot=blockintegral(6)
-- seleziono l'area dello statore (gruppo 1)
 -- e quella del traferro (gruppo 2)
  groupselectblock(1)
  groupselectblock(2)

 -- Magnetic field energy
  Energia=blockintegral(2)

 -- scrivo i risultati su file
  handle=openfile("output_540.txt","a");
  write(handle,freq,"  ",t,"   ",Prot,"   ",Energia,"\n")
  closefile(handle)
```

```
-- uscita
  exitpost()
```



(a) Pre-processor (b) Post-processor

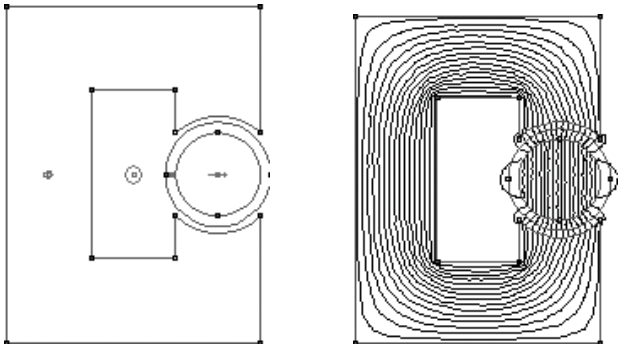
Figure 6: Motore asincrono trifase

Motore monofase per pompa

```
PRE.LUA
--Programma per il calcoli dell'induuzione su una linea
for n=0,6,1 do
   dz=n*15
   open_femm_file("motorino.fem")
   seteditmode("group")
   selectgroup(1)
   move_rotate(0.06,0.05,dz)
   save_femm_file("motortemp.fem")
   analyse()

   handle=openfile("tempfile","w");
   write(handle,dz,"\n");
   closefile(handle);

   runpost("post.lua")
end
```

```
PRE.LUA
--Programma per il calcoli dell'induuzione su una linea
for n=0,6,1 do
   dz=n*15
   open_femm_file("motorino.fem")
   seteditmode("group")
   selectgroup(1)
   move_rotate(0.06,0.05,dz)
   save_femm_file("motortemp.fem")
   analyse()

   handle=openfile("tempfile","w");
   write(handle,dz,"\n");
```

```
    closefile(handle);

    runpost("post.lua")
end
```



(a) Pre-processor (b) Post-processor

Figure 7: Motore monofase per pompa

Funzioni matematiche

$sin(x)$	seno di x
$cos(x)$	coseno di x
$tan(x)$	tangente di x
$asin(x)$	arcoseno di x $[-\pi/2, \pi/2]$, x ϵ $[-1, 1]$
$acos(x)$	arcocoseno di x $[0, \pi]$, x ϵ $[-1, 1]$
$atan(x)$	arcotangente di x $[-\pi/2, \pi/2]$
$atan2(x, y)$	arcotangente di x/y $[-\pi, \pi]$
$sinh(x)$	seno iperbolico di x
$cosh(x)$	coseno iperbolico di x
$tanh(x)$	tangente iperbolica di x
$esp(x)$	funzione esponenziale e^x
$log(x)$	logaritmo naturale di x, con x¿0
$log_{10}(x)$	logaritmo in base 10 di x, con x¿0
$esp(x)$	funzione esponenziale e^x
$pow(x, y)$	x^y
$sqrt(x)$	$\sqrt{x}$
$ceil(x)$	il pi piccolo intero non inferiore a x
$floor(x)$	il pi grande intero non superiore a x
$abs(x)$	valore assoluto di x

Pu essere utile sapere come si pu definire un vettore di dati e richiamarlo:

$Vettore = \{num1, num2\}$ definizione di un vettore;
$Vettore[indice]$ modalit di richiamo di un dato del vettore.

Un'altra funzione importante quella di poter salvare o utilizzare delle variabili o nomi di file con una radice comune ma numero diverso:

"Var"..$indice$ in questo modo vi una parte del nome sempre fisso.