

パルスパワー工学補足資料

2010/11/12 竹内

Scilab の使い方

1. scilab を起動する

-->

と表示されるはず。現在のディレクトリを見るには `pwd` を入力しリターン(Enter)。

日本語のファイル名やフォルダ名があるとエラーになる場合が多いので、例えば D ドライブのルートに `scilab` フォルダを作ると良い。このフォルダに移動するには

-->`cd D:\scilab`

2. scilab での LCR 回路計算その 1: コマンドラインで動作させる

`scilab` を立ち上げる。-->となっている状態

次の行を入力しリターン(RT)

`deff('dydt=f(t,y)','dydt=[-y(2);y(1)/(L1*C1)-R*y(2)/L1]') (RT)`

以下同様に進める。

`R=180; L1=500; C1=0.005;`

(RT)

`y0=[100;0];`

(RT)

`t0=0;Dt=0.1;tn=100;t=[t0:Dt:tn];`

(RT)

`y=ode(y0,t0,t,f);`

(RT)

`min([y(1,:)]),max([y(1,:)])`

(RT)

`plot2d([t],[y(1,:)],[10],'111','q1',[0 -120 100 120])`

(RT)

`xtitle('CLR-charge','Time [s]','Charge [C]')`

(RT)

`min([y(2,:)]),max([y(2,:)])`

(RT)

`scf();plot2d([t],[y(2,:)],[20],'111','I1',[0 -50 100 50])`

(RT)

`xtitle('CLR-current','Time [s]','Current [A]')`

電荷と電流の変化を示す二つのグラフが表示されていれば正常です。

微分方程式を解いているのは `y=ode(y0,t0,t,f)` だけです。

3. scilab を sce ファイルで動作させる

コマンドラインでいちいち入力するのは大変なので、スクリプトをファイルにして実行するようにします。

-->`scipad`

で `scilab` テキストエディタが起動。

2 で入力した下記の内容を `CLR.sce` というテキストファイルとして `D:\scilab` に保存。//のあとは無効なのでコメント等に使用可。なくても動きます。

//scilab CLR

`deff('dydt=f(t,y)','dydt=[-y(2);y(1)/(L1*C1)-R*y(2)/L1]')`

//ベクトル `dy/dt` を `t,y` の関数として定義。 `y(1)=q1`, `y(2)=I1`

`R=180; L1=500; C1=0.005;`

```

//素子値を設定
t0=0;Dt=0.1;tn=100;t=[t0:Dt:tn];
//t=0 から t=tn まで Dt 毎に値を保存
y0=[100;0];
//y の初期値を設定
y=ode(y0,t0,t,f);
//Runge-Kutta 法で y の時間変化を解く
min([y(1,:)]),max([y(1,:)])
//y(1)つまり q1 の最大値・最小値を出力
plot2d([t],[y(1,:)],[10],'111','q1',[0 -120 100 120])
//q1 の時間変化を 2 次元のグラフに出力。横軸 0～100, 縦軸-120～120。
xlabel('CLR-charge','Time [s]','Charge [C]')
//グラフタイトル, 横軸, 縦軸のタイトルを追加
min([y(2,:)]),max([y(2,:)])
//I1 の最大値・最小値を出力
scf();plot2d([t],[y(2,:)],[20],'111','I1',[0 -50 100 50])
//I1 の時間変化を別グラフに表示
xlabel('CLR-current','Time [s]','Current [A]')
//

```

CLR.sce をダブルクリックすると, scilab のエディタが起動します。これを使って中身を見たり変更したりできます。ファイルの内容はテキストなので, 他のエディタを使っても構いません。

CLR.sce を保存したら, scilab の画面に戻って,

```
-->exec('CLR.sce')
```

とすると, 電荷と電流のグラフが二つ書かれたと思います。scilab 上には CLR.sce のスクリプトが実行された様子が出ています。毎回これが出るのは面倒なので,

```
-->exec('CLR.sce',0)
```

とすると, 今度はグラフを書くために求めた最大値・最小値だけが表示されて, 他のスクリプトは表示されません。

scilab では, 拡張子 sci で保存したファイルに記述された関数を読み込むことができます。ここでは, 一次元の熱伝導方程式を解くためのプログラムを HeatPDE1.sci というファイルに記述し, 関数として読み込みます。

scilab のテキストエディタを起動し, 下記を HeatPDE1.sci として D:\scilab に保存します。

```

//scilab HeatPDE1
function [u]=HeatPDE1(xrange,trange,uinit,u0bound,uLbound,K)
//u は xrange,trange,uinit,u0bound,uLbound,K の関数
n=length(xrange);
//n は x ベクトルの長さ(列数)
m=length(trange);
u=zeros(n,m);
//u は n×m 行列で全ての要素は零
Dx=(xrange(n)-xrange(1))/(n-1);
Dt=(trange(m)-trange(1))/(m-1);
u(:,1)=uinit';
//行列 u の 1 列目をベクトル uinit の転置行列とする
u(1,:)=u0bound;

```

```

//行列 u の 1 行目をベクトル u0bound とする(境界条件)
u(n,:)=uLbound;
//行列 u の n 行目をベクトル uLbound とする(境界条件)
alpha=K*Dt/(Dx)^2;
for j=1:m-1
    for i=2:n-1
        u(i,j+1)=u(i,j)+alpha*(u(i-1,j)-2*u(i,j)+u(i+1,j));
    end;
end;
//差分法で解く
//

```

scilab で

```
--> getf('HeatPDE1.sci')
```

とし、関数 HeatPDE1 を読み込みます。

HeatPDE1 は位置 x , 時間 t , 初期条件 ui , 境界条件 $u0$ および uL , 定数 K を変数とする関数なので、それぞれ以下のようにコマンドラインで定義します。

```

K=1;
(RT)
x=[0:0.1:1];t=[0:0.005:1.5];ui=zeros(x);
(RT)
u0=exp(-2*t).*sin(50*t);uL=exp(-3*t).*cos(50*t);
(RT)
[u]=HeatPDE1(x,t,ui,u0,uL,K);
(RT)
plot3d(x,t,u,45,45,'x@t@u(x,t)');
(RT)
xtitle('Heat equation solution')
(RT)

```

温度 u が位置 x , 時間 t の関数としてグラフに表示されたと思います。初期条件や境界条件が変わったときは、 ui や $u0$ などを変更してください。