

# Matlab 语言在电路暂态分析中的应用

张文生,刘耀年,张光烈

(东北电力学院电力系,吉林 吉林 132012)

**摘要:** Matlab 是可视化的面向科学与工程计算的大型优秀科技应用软件,语句精练,功能强大,简单实用,用途广泛,它为广大科技工作者提供了一个简便实用的计算工具,可大大提高计算效率,缩短编程时间,在电路计算中有广泛的应用前景。通过一个具体算例探索其在电路暂态分析和计算方面的应用。

**关键词:** Matlab 语言; 电路计算; 二阶电路; 暂态

**中图分类号:** TM711

**文献标识码:** A

**文章编号:** 1003-4897(2000)12-0016-04

## 1 引言

Matlab 语言 80 年代由 MathWorks 公司开发,经过 20 年的不断发展和完善,迄今已推出 5.3 版本,据称在全球已有 40 万用户。Matlab 语言具有的特点可概括如下:

Matlab 语言以矩阵或数组为数据单位进行运算,可直接处理矩阵或数组,这是 Matlab 语言有别于其他语言的一大特色。

Matlab 语言结构紧凑、语句精练。仅需几条简单的语句,就可以完成一大串其他高级语言才能完成的任务,可大大节省编程时间,提高计算效率。

Matlab 语言具有强大的绘图功能,能方便地绘制二维、三维图形,用户只需一条或几条语句,就可绘出复杂的绚丽多彩的图形,这也是其它高级语言力所不能及的。

Matlab 语言含有丰富的常用的内部函数,可直接调用而不需另行编程。

本文列举一个简单的二阶电路,探索 Matlab 语言在电路暂态计算分析中的应用。

## 2 复频域分析的直接拉氏反变换法

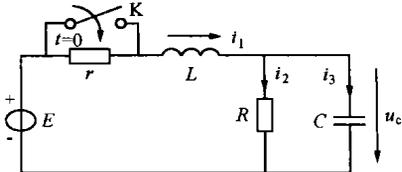
二阶电路如图 1(a) 所示,开关 k 原来是打开的,电路已稳定,已知  $E = 120\text{V}$ ,  $r = 15\ \Omega$ ,  $L = 1\text{H}$ ,  $R = 25\ \Omega$ ,  $C = 1000\ \mu\text{F}$ 。  $i_1(0_-) = 3\text{A}$ ,  $u_c(0_-) = 75\text{V}$ ,在  $t = 0$  时将开关闭合,求  $t > 0$  时的电感电流  $i_1(t)$  及电容电压  $u_c(t)$  的变化规律。

**解:** 此题有多种解法,现用复频域法求解。画出运算电路图 1(b) 所示,应用回路电流法,列出方程如下:

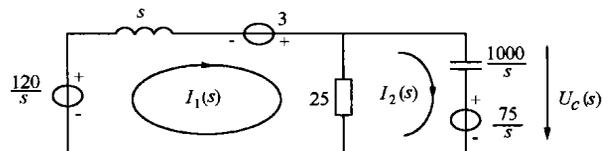
$$\left. \begin{aligned} (25 + s) I_1(s) - 25 I_2(s) &= \frac{120}{s} + 3 \\ -25 I_1(s) + (25 + \frac{1000}{s}) I_2(s) &= -\frac{75}{s} \end{aligned} \right\} \quad (1)$$

又由回路电压定律,有

$$U_c(s) = \frac{120}{s} + 3 - s I_1(s) \quad (2)$$



(a)



(b)

图 1 二阶电路的动态分析

为求原函数  $i_1(t)$  及  $u_c(t)$ , 编制程序如下:

```
A = sym( '[25 + s, - 25; - 25, 25 + 1000/ s]' ); %定义符号矩阵 A, 即由  $I_1, I_2$  的系数构成;
```

```
A1 = sym( '[120/ s + 3, - 25; - 75/ s, 25 + 1000/ s]' ); %定义矩阵 A1;
```

```
II = det(A1) / det(A); %用行列式法求  $I_1(s)$ ;
```

```
以上三条语句用于解出  $I_1(s)$ ;
```

```
syms t s; %声明 t, s 为符号变量;
```

```
i1 = ilaplace(II); %对  $I_1(s)$  进行拉氏反变换;
```

```
i2 = vpa(i1, 4); %原函数  $i_1$  为四位有效数字的近似解;
```

```
Uc = 120/ s + 3 - s * II; %由  $I_1(s)$  解出  $U_c(s)$ ;
```

```
uc = ilaplace(Uc); %对  $U_c$  进行拉氏反变换;
```

```
uc1 = vpa(uc, 4); %原函数  $u_c$  为四位有效数字的近似解;
```

```
disp( 'i1' );
```

```
disp( i2 ); %显示  $i_1$ ;
```

```
disp('uc');
```

```
disp(uc1); % 显示  $U_c$ ;
```

程序显示结果如下:

$$i_1 = 4.800 - 1.800 * \exp(-20 * t) * \cos(24.49 * t) + 0.3674 * \exp(-20 * t) * \sin(24.49 * t)$$

$$u_c = 120 - 45 * \exp(-20 * t) * \cos(24.49 * t) - 36.74 * \exp(-20 * t) * \sin(24.49 * t)$$

即  $i_1(t) = 4.8 - 1.8e^{-20t} \cos 24.49t + 0.3674e^{-20t} \sin 24.49t$

$$u_c(t) = 120 - 45e^{-20t} \cos 24.49t - 36.74e^{-20t} \sin 24.49t$$

49t

如还需求电感电压  $u_1$  和电容电流  $i_c$  的变化规律,取电压、电流为关联参考方向(见图 1-a),有

$$u_1 = L \frac{di_1}{dt}$$

及

$$i_c = i_3 = C \frac{du_c}{dt}$$

可用下述语句完成:

```
u1 = vpa(diff(i2),4); % 通过求电感电流的导数计算电感电压,精度为四位;
```

```
ic = vpa(1.0e-3 * diff(uc),4); % 计算电容电流;
```

```
disp(u1),disp(ic); % 显示  $u_1, i_c$ ;
```

结果为:

$$u_1 = 45.00 * \exp(-20 * t) * \cos(24.49 * t) + 36.73 * \exp(-20 * t) * \sin(24.49 * t)$$

$$i_c = 1.837 * \exp(-20 * t) * \sin(24.49 * t)$$

即

$$u_1(t) = 45e^{-20t} \cos 24.49t + 36.73e^{-20t} \sin 24.49t$$

49t

$$i_c(t) = 1.837e^{-20t} \sin 24.49t$$

以下程序绘出了  $u_1$  和  $i_c$  随时间变化的曲线(见图 2);

```
t=0:2.0e-3:0.4; % 采样间隔取为 2 ms;
```

```
plot(t,u1,'LineWidth',1.5); % 绘电感电压;
```

```
xlabel('time'),ylabel('u1'); % 标注坐标;
```

```
grid; % 设网格;
```

```
plot(t,ic,'LineWidth',1.5); % 绘电容电流;
```

```
xlabel('time'),ylabel('ic'),grid;
```

由图 2 看出,换路瞬间,电感电压和电容电流均发生了跃变,  $u_1$  由 0 跳变为 45V,电容电流由 0 跳变到约 0.7A,暂态持续时间约为 0.2s,以后两者都衰减

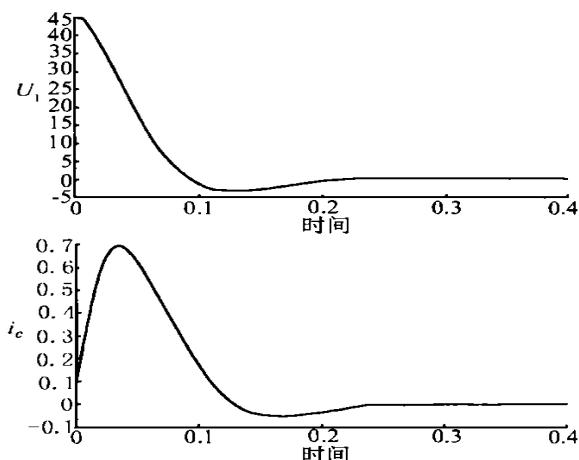


图 2 电感电压和电容电流的变化规律

至零。在暂态过程中,电容电流  $i_c$  出现了最大值,利用 Matlab 语言,很容易求出该最大值。

不难看出,利用 Matlab 所提供的拉氏反变换函数 ilaplace 对象函数求原函数时,只要列出象函数的表达式,即可直接进行反变换。使得计算效率大为提高,此外,Matlab 的函数求导运算及函数绘图功能,通过此例可见一斑。

### 3 复频域分析的留数法

在复频域分析中,设象函数为:

$$F(s) = \frac{b(s)}{a(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} \quad (3)$$

其中  $b(s)$  和  $a(s)$  都是复频域  $s$  的多项式,所有的系数都是实数。电路分析中遇此类问题,为求  $F(s)$  的原函数  $f(t)$ ,传统的方法一般是用部分分式法,即将  $F(s)$  分解为多个  $s$  的一次分式之和:

$$F(s) = \frac{b(s)}{a(s)} = \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2} + \dots + \frac{r_n}{s - p_n} + k(s) \quad (4)$$

众所周知,求  $p_1, p_2, \dots, p_n$  及  $r_1, r_2, \dots, r_n$  的计算工作量非常大,同时,在进行拉氏反变换时,需记忆很多公式,很为不便。Matlab 提供了一个非常好用的留数函数 residue,可以直接求出  $p_n, r_n$ ,用法如下:

$$[r, p, k] = \text{residue}(b, a)$$

其中  $a, b$  分别为分母多项式  $a(s)$  及分子多项式  $b(s)$  按降幂排列的各项系数向量,而  $p, r$  分别为  $F(s)$  的极点向量和留数向量,若  $n > m$  时,(3) 式为真分式,  $k$  为空阵。

仍以图 1 所示电路为例,在(1)式中,解出  $I_1(s)$

$$I_1(s) = \frac{3s^2 + 165s + 4800}{s^3 + 40s^2 + 1000s} = \frac{b_1(s)}{a_1(s)}$$

及

$$U_c(s) = \frac{75s^2 + 3000s + 120000}{s^3 + 40s^2 + 1000s} = \frac{b_2(s)}{a_2(s)}$$

键入如下程序:

```
a1 = [1, 40, 1e+3, 0], b1 = [3, 165, 48e+2];
a2 = [1, 40, 1e+3, 0], b2 = [75, 3e+3, 12e+4];
[r1, p1, k1] = residue(b1, a1), [r2, p2, k2]
= residue(b2, a2);
```

程序运行结果如下:

$$r1 = \begin{bmatrix} -0.9000 - 0.1837i \\ -0.9000 + 0.1837i \\ 4.8000 \end{bmatrix} \quad p1 = \begin{bmatrix} -20.0000 + 24.4949i \\ -20.0000 - 24.4949i \\ 0 \end{bmatrix}$$

$$r2 = \begin{bmatrix} -22.5 + 18.37i \\ -22.5 - 18.37i \\ 120 \end{bmatrix} \quad p2 = \begin{bmatrix} -20 + 25i \\ -20 - 25i \\ 0 \end{bmatrix}$$

而  $k1 = []$   $k2 = []$

均为空矩阵。

由此可知原函数  $i_1(t)$ ,  $u_c(t)$  为:

$$i_1(t) = r_{11}e^{p_{11}t} + r_{12}e^{p_{12}t} + r_{13}e^{p_{13}t} =$$

$$(-0.9 - j0.1837)e^{(-20 + j25)t} + (-0.9 +$$

$$j0.1837)e^{(-20 - j25)t} + 4.800 =$$

$$1.836e^{-20t} \cos(24.5t - 168.4^\circ) + 4.800$$

$$u_c(t) = r_{21}e^{p_{21}t} + r_{22}e^{p_{22}t} + r_{23}e^{p_{23}t} =$$

$$(-22.5 + j18.37)e^{(-20 + j25)t} + (-22.5 -$$

$$j18.37)e^{(-20 - j25)t} + 120 =$$

$$58.1e^{-20t} \cos(25t + 140.7^\circ)$$

可见 residue 函数可直接对  $F(s)$  进行部分分式分解并给出  $r_n, p_n$ 。

#### 4 电路状态方程的数值解法

图1所示电路也可在时频域内求解,由图1-(a),容易得出:

$$LC \frac{d^2 u_c}{dt^2} + \frac{L}{R} \frac{du_c}{dt} + u_c = E \quad (5)$$

此为二阶微分方程,为方便求解,取电感电流  $i_1$  和电容电压  $u_c$  为状态变量,可列出状态方程如下:

$$\left. \begin{aligned} \frac{di_1}{dt} &= -\frac{1}{L}u_c + E \\ \frac{du_c}{dt} &= \frac{1}{C}i_1 - \frac{1}{RC}u_c \end{aligned} \right\} \quad (6)$$

代入数据并写成矩阵形式,有

$$\begin{bmatrix} \frac{di_1}{dt} \\ \frac{du_c}{dt} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} i_1 \\ u_c \end{bmatrix} + \begin{bmatrix} E \\ 0 \end{bmatrix} =$$

$$\begin{bmatrix} 0 & -1 \\ 1000 & 40 \end{bmatrix} \begin{bmatrix} i_1 \\ u_c \end{bmatrix} + \begin{bmatrix} 120 \\ 0 \end{bmatrix} \quad (7)$$

初始条件:

$$\begin{bmatrix} i_1(0) \\ u_c(0) \end{bmatrix} = \begin{bmatrix} 3 \\ 75 \end{bmatrix} \quad (8)$$

对于状态方程的求解,Matlab 提供了采用 Runge - Kutta - Fehlberg 算法的数值积分 ODE 函数,并将其解以图形方式输出。

编制程序如下:

```
function dy = myfile(t, y); % 将式(6)定义为函数文件,供 ODE 调用;
dy = zeros(2, 1); % 变量 y 为两行一列向量;
dy(1) = - y(2) + 120;
dy(2) = (1000 * y(1) - 40 * y(2));
[t, y] = ode45('myfile', [0, 0.4], [3, 75]); % 调用 ODE 函数,并代入初始条件;
plot(t, y(:, 1), 'LineWidth', 1.5); % 根据计算结果输出 i1 波形;
xlabel('time'), ylabel('i1'); % 标注坐标;
grid; % 绘网格
plot(t, y(:, 2), 'LineWidth', 1.5); % 输出 uc 波形;
xlabel('time'), ylabel('uc');
grid;
```

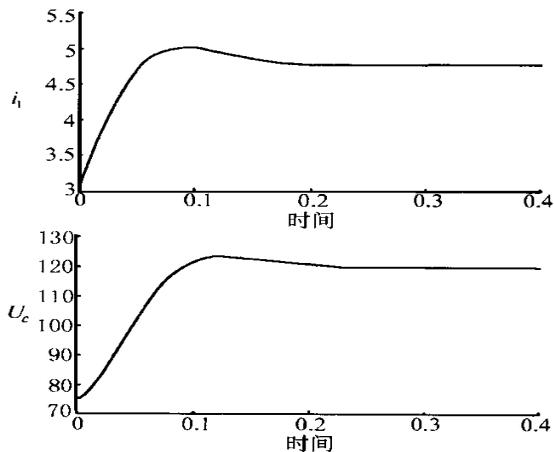


图3 电感电流  $i_1$  和电容电压  $u_c$  的变化规律

图3示出了程序运行的结果。由此例看出,用 Matlab 自身提供的数值积分函数 ODE 求解状态方程简便快捷,大大节省了编程时间,采用同一算法的 Fortran 语言程序,语句却多达 200 条。

#### 5 结语

本文仅用一个简单的二阶电路的算例说明 Matlab 语言在电路暂态分析方面的应用,应该说是很不

全面的,实际上,Matlab 提供的大量而丰富的内部函数,高效简洁的语句,能满足电路计算的各种需要,Matlab 语言还允许复数直接参与运算,满足了交流电路分析中复数计算的需求。其次,用 Matlab 语言还可以直接绘制向量图,而向量图是分析交流电路必不可少的重要工具,Matlab 所独有的矩阵或数组运算功能,可以允许用户成组、大批地处理复杂电路的电压、电流、功率等物理量,例如三相对称或不对称电路的计算、谐波分析、潮流分析、不对称电路中的对称分量分解等等,从而可大大地提高计算精度,缩短编程时间,提高工作效率。尽快熟悉并掌握这种语言,将其用于电路分析和计算,必将起到事半功倍之效。

### 参考文献:

- [1] The Mathworks Inc. Using MATLAB Version 5. 1997,1.
- [2] John O Attia. Electronics and Circuit Analysis Using MATLAB. CRC Press,1999.
- [3] 楼顺天,于卫,阎华梁. MATLAB 程序设计语言. 西安:西安电子科技大学出版社,1997.
- [4] 张志涌,刘瑞桢,杨祖樱. 掌握和精通 MATLAB. 北京:北京航空航天大学出版社,1997.

收稿日期: 2000-06-22

作者简介: 张文生(1952-),男,副教授,从事电路理论,计算机在电路计算中的应用研究; 刘耀年(1951-),男,教授,硕士,从事电路理论的研究。

### Application of language Matlab in analyzing circuit transient

ZHANG Wen - sheng, LIU Yao - nian, ZHANG Guang - lie

(Northeast China Institute of Electric Power Engineering, Jilin132012, China)

**Abstract:** Matlab is a visual and science and engineering calculation oriented quality application software. It features simple language, powerful functions, practical and wide application scope. It is a simple and practical calculation tool for engineers to improve the calculation speed and save programming time. Therefore, it will be widely used in circuit calculation in the future. In this paper, an actual calculation is taken as an example to research the application of Matlab in circuit transient analysis and calculation.

**Keywords:** language Matlab; circuit calculation; second circuit; transient