Lucrarea 9

REZOLVAREA NUMERICĂ A ECUAȚIILOR DIFERENȚIALE

1. SCOPUL LUCRĂRII

Studiul şi însuşirea unor metode de rezolvare a ecuațiilor diferențiale cu ajutorul unor algoritmi care vor fi implementați într-un limbaj de nivel înalt (în particular, C) şi aplicarea acestora în rezolvarea unor probleme de electronică.

2. PREZENTAREA TEORETICĂ

În acest capitol sunt prezentate metodele de rezolvare a ecuațiilor diferențiale ordinare, prin metode directe de *tip Runge—Kutta* şi prin metode indirecte de tip *predictor-corector*.

2.1. REZOLVAREA NUMERICĂ A ECUAȚIILOR DIFERENȚIALE ORDINARE DE ORDINUL ÎNTÂI

Se consideră ecuațiile de tipul:

$$y' = f(x, y) \tag{9.1}$$

cu condiția inițială:

$$y(x_0) = y_0 \tag{9.2}$$

Ecuația diferențială (9.1) defineşte o curbă în planul xOy .

Soluția unei ecuații diferențiale reprezintă o *familie de curbe*, şi aceasta deoarece în rezolvarea unei astfel de ecuații apare operația de integrare, pentru integrale de tip *Riemann*, fără limite de integrare la stânga sau la dreapta. Apare aşadar, inevitabil, o constantă de integrare. Condiția inițială are rolul de a particulariza această constantă de integrare, în acest fel determinând una dintre curbele din familie, în funcție de problemă.

Rolul metodelor numerice folosite este acela de a *aproxima prin puncte* curba soluție aleasă.

2.1.1. METODA LUI EULER ÎMBUNĂTĂȚITĂ (media pantelor)

Această metodă face parte din grupul de metode de tip *Runge-Kutta de ordin 2*. deoarece dă aceleaşi soluţii ca şi metoda lui Taylor până la h^2 .

Utilizează media pantelor din punctele (x_m, y_m) şi (x_{m+1}, y_{m+1}) unde $x_{m+1} = x_m + h$ iar $y_{m+1} = y_m + hy'_m$ este ordonata obținută din ecuația tangentei în punctul M la curbă cu panta $y'_m = f(x_m, y_m)$. Grafic metoda este prezentată în figura 9.1. În acest punct H, de coordonate $(x_{m+1}, y_m + hy'_m)$ se calculează panta la curbă, anume:

$$\overline{y}_{m+1} = f\left(x_m+h, y_m + hy'_m\right) \tag{9.3}$$

Se face media dintre panta dreptei prin punctul M şi panta dreptei prin punctul H, şi se notează cu:

$$z_m = \frac{1}{2}\left(f\left(x_m, y_m\right) + f\left(x_m+h, y_m + hy'_m\right)\right) \tag{9.4}$$

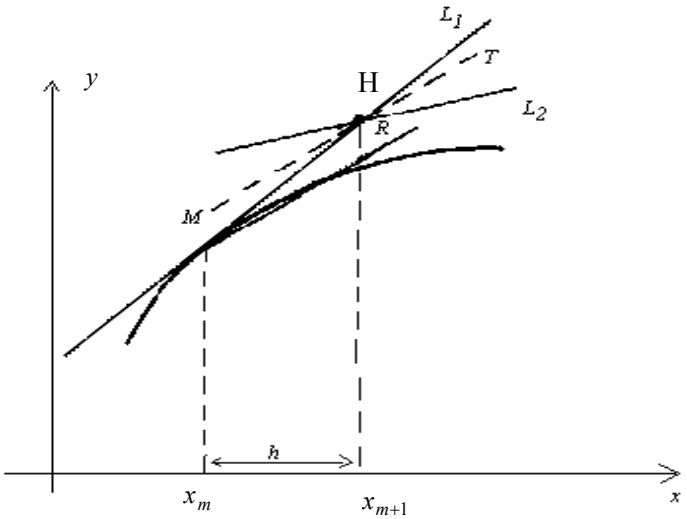

Fig. 9.1. - Calculul grafic al soluției prin metoda lui Euler îmbunătățită.

Dreapta de pantă z_m este HT. Cu această pantă z_m se trasează coarda prin M (echivalent cu a spune că se translatează dreapta HT în punctul M) şi se intersectează cu dreapta (verticala) din punctul $x = x_{m+1}$, rezultând punctul R, de coordonate (x_{m+1}, y_{m+1}) considerat al doilea punct de pe curba soluție, primul fiind considerat $M(x_m, y_m)$. Ecuația coardei prin M este:

$$y = y_m + 0.5 \cdot \left(f\left(x_m, y_m\right) + f\left(x_m+h, y_m + hy'_m\right)\right)\left(x - x_m\right) \tag{9.5}$$

Punctul R, care reprezintă soluția numerică a ecuației diferențiale, are ordonata:

$$y_{m+1} = y_m + 0.5 \cdot h \cdot \left(f\left(x_m, y_m\right) + f\left(x_m+h, y_m + hy'_m\right)\right) \tag{9.6}$$

unde $y'_m = f(x_m, y_m)$.

Această formulă (9.6) reprezintă formula de calcul a soluțiilor numerice pentru o ecuație diferențială ordinară de ordinul întâi.

2.1.1.1. Algoritmul 9.1. Metoda lui *Euler îmbunătățită*

```
void Euler_Imb
(
Adr_functie,   /* Adresa unei funcţii reale de două
               variabile reale ce reprezintă ecuaţia
               diferenţială (pointer la o funcţie)*/
real x0,       /* abscisa punctului iniţial prin care trece
               graficul soluţiei */
real y0,       /* ordonata punctului iniţial prin care trece
               graficul soluţiei */
real h,        /* pasul între abscisele punctelor de calcul
               ale soluţiei întreg n, numărul de puncte în
               care se calculează soluţia */
real *y        /* vectorul ordonatelor (ce vor fi calculate)
               ale punctelor prin care se aproximează
               soluţia ecuaţiei diferenţiale */
)
{ // declaraţiile şi definiţiile variabilelor locale
real x[N];     // vectorul absciselor calculate ale soluţiei
               //numerice
întreg i;      // indice pentru bucla 'for()'.

   // corpul de instrucţiuni al funcţiei
   y[0]= y0;  x[0]=x0; // se porneşte cu punctul iniţial.
   pentru i= 1÷ n
   {/* calculul absciselor şi ordonatelor punctelor prin
    care se aproximează soluţia ecuaţiei */
   x[i]= x[0]+i*h;
   y[i]= y[i-1] + 0.5*h*( *Adr_functie( x[i-1],y[i-1]) +
   +  *Adr_functie(x[i],  y[i-1]  +  h*(*Adr_functie(  x[i-1],
y[i-1]) );
   }
}
```

Observații:

1) Notațiile '*Adr_functie*' și '**Adr_functie*' sunt *simbolice*. Ele reprezintă, respectiv, adresa unei funcții (deci un pointer la o funcție) și apelul unei funcții prin pointer. Vedeți *Anexa B* - "*Noțiuni de C necesare desfășurării lucrărilor de laborator*" pentru lămuriri.

2) Alegerea numărului 'N' de componente al vectorilor 'x[ ]' și 'y[ ]' se face în funcție de numărul 'n' al punctelor prin care se dorește aproximarea soluției.

Între *N* şi *n* există legătura: *N=n+1*, deoarece în C indexarea vectorilor se face de la valoarea 0.

2.1.2. METODA LUI EULER MODIFICATĂ (media punctelor)

Face parte din clasa metodelor de rezolvare de tip *Runge-Kutta de ordin 2*, deoarece, ca şi Euler îmbunătățită, oferă aceleaşi soluții ca şi metoda lui Taylor până la h^2.

În cadrul acestei metode nu se mediază pantele, ci se evaluează panta într-un punct care reprezintă *media a două puncte*.

Considerăm curba din figura 9.2 în care se dă punctul inițial M prin care să treacă curba soluție a ecuației diferențiale şi dorim să determinăm cel de al doilea punct al soluției. Prin punctul (x_m, y_m) dat, cunoscut sau determinat, se duce tangenta prin M şi se determină pe această tangentă punctul H de coordonate (x_{m+1}, y_{m+1}). Până în acest punct nu există deosebire cu metoda Euler îmbunătățită.

Tangenta prin punctul M are ecuația

$$y = y_m + f(x_m, y_m)(x - x_m) \tag{9.7}$$

Prin intersecția cu dreapta (verticala) în punctul $x = x_m + h$ rezultă coordonatele punctului H:

$$H(x_{m+1} = x_m + h, \quad y_{m+1} = y_m + hf(x_m, y_m) \tag{9.8}$$

prezentat în figura 9.2.

Metoda face media coordonatelor punctelor M şi H şi determină punctul:

$$P\left(x_p = x_m + \frac{h}{2}, \quad y_p = y_m + \frac{h}{2} f\left(x_m, y_m\right)\right) \tag{9.9}$$

Se calculează panta soluției *y* în acest punct P şi rezultă:

$$z_m = f\left(x_m + \frac{h}{2}, \quad y_m + \frac{h}{2} f\left(x_m, y_m\right)\right) \tag{9.10}$$

Se scrie ecuația dreptei care trece prin P, de pantă z_m, şi se intersectează cu verticala în punctul următor pe abscisă:

$$x_{m+1} = x_m + h$$

Dreapta MN are ecuația:

$$y = y_m + f\left(x_m + \frac{h}{2}, \quad y_m + \frac{h}{2} f\left(x_m, y_m\right)\right)\left(x - x_m\right) \tag{9.11}$$

Prin intersecția dreptei (8.11) cu verticala în punctul $x = x_{m+1} = x_m + h$ se determină ordonata R:

$$y_{m+1} = y_m + hf\left(x_m + \frac{h}{2}, \quad y_m + \frac{h}{2} f\left(x_m, y_m\right)\right) \tag{9.12}$$

care reprezintă formula de calcul a ordonatelor soluției ecuației diferențiale.

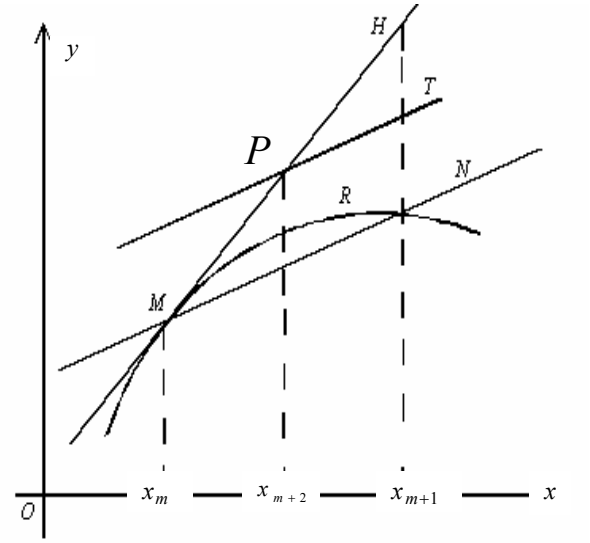

Fig. 9.2. - Calculul grafic al soluției ecuației diferențiale prin metoda lui Euler modificată.

2.1.2.1. Algoritmul 9.2. Metoda lui *Euler modificată*

```
void Eul_Mod
(
Adr_functie,    /* Adresa unei funcţie reală de două
                variabile reale ce reprezintă ecuaţia
                diferenţială */
real x0 ,       /* abscisa punctului iniţial prin care trece
                graficul soluţiei */
real y0 ,       /* ordonata punctului iniţial prin care trece
                graficul soluţiei */
real h,         /* pasul între abscisele punctelor de calcul
                ale soluţiei  */
întreg n,       /*numărul de puncte în care se calculează
                soluţia */
real *y         /* vectorul ordonatelor (ce vor fi calculate)
                ale punctelor prin care se aproximează
                soluţia ecuaţiei diferenţiale */
)
{ // declaraţiile şi definiţiile variabilelor locale
real x[N];      //   vectorul absciselor calculate ale
                //soluţiei numerice.
întreg i;       // indice pentru bucla 'for()'.

   // corpul de instrucţiuni al funcţiei
   x[0]= xo;  y[0]= y0;   // se începe cu punctul iniţial
   pentru i= 1÷n
   {     // se calculează abscisele şi ordonatele punctelor
         // prin care se aproximează soluţia ecuaţiei.
   x[i]= x[i -1] + h;
```

```
    y[ i ]= y[i-1] + h*( *Adr_functie( x[i-1] + 0.5*h, y[i-1]
    + 0.5*h * (*Adr_functie( x[i-1], y[i-1]) );
    }
}
```

Observații:
1) Notațiile '*Adr_functie*' şi '**Adr_functie*' sunt *simbolice*. Ele reprezintă, respectiv, adresa unei funcții (deci un pointer la o funcție) şi apelul unei funcții prin pointer. Vedeți *Anexa B* - "*Noțiuni de C necesare desfăşurării lucrărilor de laborator*" pentru lămuriri.
2) Alegerea numărului 'N' de componente al vectorilor 'x[ ]' şi 'y[ ]' se face în funcție de numărul 'n' al punctelor prin care se doreşte aproximarea soluției. Între *N* şi *n* există legătura: *N=n+1*, deoarece în C indexarea vectorilor se face de la valoarea 0.

2.1.3. METODA RUNGE-KUTTA DE ORDINUL PATRU

Formula de calcul numeric a soluției ecuației diferențiale (9.1) este dată de expresia (9.13)

$$\begin{aligned} x_{m+1} &= x_m + h \\ y_{m+1} &= y_m + \frac{h}{6}\left[k_1 + 2k_2 + 2k_3 + k_4\right] \end{aligned} \tag{9.13}$$

unde

$$\begin{aligned} k_1 &= f(x_m, y_m) \\ k_2 &= f\left(x_m + \frac{h}{2}, y_m + \frac{h}{2}k_1\right) \\ k_3 &= f\left(x_m + \frac{h}{2}, y_m + \frac{h}{2}k_2\right) \\ k_4 &= f\left(x_m + h, y_m + hk_3\right) \end{aligned} \tag{9.14}$$

Eroarea de trunchiere a metodei este:

$$e_T \approx k \cdot h^5 \tag{9.15}$$

ceea ce demonstrează clasa din care face parte această metodă (ordin 4).

2.1.3.1. Algoritmul 9.3. Metoda *Runge-Kutta de ordinul 4*

```
void RGK4
(
Adr_functie,   /* Adresa unei funcții reale de două
                  variabile reale ce reprezintă ecuația
                  diferențială (pointer la funcție) */
real x0 ,      /* abscisa punctului inițial prin care trece
                  graficul soluției
```

```
real y0 ,       /* ordonata punctului iniţial prin care trece
                graficul soluţiei.
real h,         /* pasul între abscisele punctelor de calcul
                ale soluţiei
întreg n,       /* numărul de puncte prin care se aproximează
                numeric soluţia ecuaţiei.
real *y         /*  vectorul  ordonatelor  (  ce  vor  fi
                calculate) ale punctelor prin care se face
                aproximarea soluţiei ecuaţiei diferenţiale*/
)
{ // declaraţiile şi definiţiile variabilelor locale
real x[N];      // vectorul absciselor calculate
întreg i;       // indice pentru bucla 'for()'.

   // corpul de instrucţiuni al funcţiei
   x[0]= x0;  y[0]= y0;     // se începe cu punctul iniţial .
   pentru i= 1÷n
   {     /* se calculează coordonatele punctelor prin care
         se aproximează soluţia ecuaţiei diferenţiale date*/
   x[i] = x0 + i*h;
   k1 = *Adr_functie( x[ i-1 ], y[ i-1 ]);
   k2 = *Adr_functie( x[ i-1 ] + 0.5*h, y[ i-1 ] + 0.5*h*k1);
   k3 = *Adr_functie( x[ i-1 ]+ 0.5*h, y[ i-1 ] + 0.5*h*k2);
   k4 = *Adr_functie( x[ i ], y[ i-1 ] + h*k3);
```

$$y[i] = y[i-1] + \frac{h}{6}(k_1 + 2*k_2 + 2*k_3 + k_4);$$

```
   }
}
```

Observaţii:

1) Notaţiile '*Adr_functie*' şi '**Adr_functie*' sunt *simbolice*. Ele reprezintă, respectiv, adresa unei funcţii (deci un pointer la o funcţie) şi apelul unei funcţii prin pointer. Vedeţi *Anexa B* - "*Noţiuni de C necesare desfăşurării lucrărilor de laborator*" pentru lămuriri.

2) Alegerea numărului 'N' de componente al vectorilor 'x[ ]' şi 'y[ ]' se face în funcţie de numărul 'n' al punctelor prin care se doreşte aproximarea soluţiei. Între *N* şi *n* există legătura: *N=n+1*, deoarece în C indexarea vectorilor se face de la valoarea 0.

2.1.1. METODE PREDICTOR-CORECTOR

Aceste metode prezic o valoare pentru soluţie la pasul $m+1$, y_{m+1}, apoi se utilizează o formulă pentru corecţia ei. Formula de corecţie se poate utiliza de mai multe ori pentru re-corectări. Acest proces de iteraţie poate fi făcut eficient dacă se alege dimensiunea intervalului pentru un număr minim de iteraţii.

Considerăm formula predictor - corector ***de ordinul doi***:

$$y_{m+1}^{(0)} = y_{m-1} + 2 * h * f\left(x_m, y_m\right) \tag{9.16}$$

în care indicele (0) arată prima estimare a lui y_{m+1}.

Deoarece este necesar să se cunoască un punct anterior lui x_0 se utilizează pentru pornirea metodei o aproximație inițială prin metoda Runge-Kutta de ordinul doi.

2.2.1. Algoritm 9.4. Metoda predictor-corector

```
void Pred_Cor
(
Adr_functie,   /* Adresa unei funcţii reale de două
               variabile reale ce reprezintă ecuaţia
               diferenţială */
real x0 ,      /* abscisa punctului iniţial prin care trece
               graficul soluţiei
real y0 ,      /* ordonata punctului iniţial prin care trece
               graficul soluţiei */
real h,        /* pasul între abscisele punctelor de calcul
               ale soluţiei */
real eps,      /* eroarea de calcul */
întreg n,      /* numărul de puncte în care se calculează
               soluţia */
real *y        /* vectorul ordonatelor soluţiei numerice (ce
               vor fi calculate de funcţie */
)
{ /* declaraţiile şi definiţiile variabilelor locale */
 real x[N];    /*   vectorul   absciselor   calculate   ale
               punctelor prin care se aproximează soluţia
               ecuaţiei diferenţiale */
 întreg i;     /* indice pentru ciclul 'for()' */
 real prec;    /* o aproximaţie precedentă a soluţiei */

/* corpul de instrucţiuni al funcţiei */
 y[0] = y0;  x[0] = x0; x[1] = x0+h;
 y[1] = y[0] + 0.5*h*( *Adr_functie(x0, y[0])
   + (*Adr_functie(x[1],y[0]+h*(*Adr_functie(x[1],y[0])))));
 pentru i= 2÷ n
 {
   x[i] = x[i-1] + h;
   y[i] = y[i-2] + 2*h*(*Adr_functie( x[i-1], y[i-1]));

   execută
   {
   prec = y[i];
   y[i] = y[i-1] + 0.5*h*(*Adr_functie( x[i-1], y[i-1])+
                              + (*Adr_functie(x[i], prec)));
   } cât timp( fabs(prec-y[i]) >= eps);
 }
}
```

Observaţii:

1) Notaţiile '*Adr_functie*' şi '**Adr_functie*' sunt *simbolice*. Ele reprezintă, respectiv, adresa unei funcţii (deci un pointer la o funcţie) şi apelul unei funcţii prin

pointer. Vedeți *Anexa B* - "*Noțiuni de C necesare desfăşurării lucrărilor de laborator*" pentru lămuriri.
2) Alegerea numărului 'N' de componente al vectorilor 'x[ ]' şi 'y[ ]' se face în funcție de numărul 'n' al punctelor prin care se doreşte aproximarea soluției. Între *N* şi *n* există legătura: *N=n+1*, deoarece în C indexarea vectorilor se face de la valoarea 0.

2.1.2. METODA MILNE

Această metodă foloseşte o pereche de formule de precizare şi corectare, utilizează o integrare după *metoda lui Simpson*.

$$\begin{aligned} y_{k+1} &= y_{k-3} + \left(\frac{4h}{3}\right)\left(2y'_{k-2} - y'_{k-1} + 2y'_k\right) \\ y_{k+1} &= y_{k-1} + \left(\frac{h}{3}\right)\left(y'_{k-1} + 4y'_k + y'_{k+1}\right) \end{aligned} \tag{9.17}$$

Pentru aplicarea metodei din start trebuie cunoscute patru valori $y_k, y_{k-1}, y_{k-2}, y_{k-3}$ care se obțin cu o metodă directă.

Corecția soluției se face până când diferența dintre două valori consecutive ale soluției într-un punct devine mai mică decât o eroare ε impusă.

2.2.2.1. Algoritmul 9.5. Metoda lui *Milne*

```
void Milne
(
Adr_functie,   /* Adresa unei funcţii reală de două
                  variabile reale ce reprezintă ecuaţia
                  diferenţială */
real x0 ,      /* abscisa punctului iniţial prin care trece
                  graficul soluţiei */
real y0 ,      /* ordonata punctului iniţial prin care trece
                  graficul soluţiei */
real h,        /* pasul între abscisele punctelor de calcul
                  ale soluţiei */
real eps,      /* eroarea de calcul */
întreg n,      /* numărul de puncte în care se calculează
                  soluţia */
real *y        /* vectorul ordonatelor (ce vor fi calculate)
                  ale punctelor prin care se aproximează
                  soluţia ecuaţiei diferenţiale */
)
{ /* declaraţiile şi definiţiile variabilelor locale */
real x[N];     /*  vectorul absciselor calculate ale
                  punctelor prin care se aproximează soluţia
                  ecuaţiei */
întreg i;      /* indice pentru ciclurile 'for()' */
real prec;     /* o aproximaţie a soluţiei */
```

```
/* corpul de instrucţiuni al funcţiei */
y[0] = y0; x[0] = x0;    /*se pleacă de la punctul iniţial*/
   pentru i= 1÷ n
      x[i]=x[i-1] + h; // se stabilesc valorile absciselor.
   pentru i= 1÷ 3
   y[i]= y[i-1]+0.5*h*(*Adr_functie( x[i-1],y[i-1])+
               +(*Adr_functie(x[i],y[i-1]
               +h*(*Adr_functie(x[i-1],y[i-1])))));
   pentru i= 4÷ n
   {y[i]= y[i-4]+4*h*( 2*(*Adr_functie( x[i-3]*h, y[i-3])) -
      -  (*Adr_functie(x[i-2]*h,y[i-2]))+
      +  2*(*Adr_functie( x[i-1], y[i-1])) )/3;
   execută
   {
   prec = y[i];
   y[i] = y[i-2] + h*( *Adr_functie( x[i-3], y[i-3])) +
                     + 4*(*Adr_functie( x[i-2], y[i-2])) +
                     + (*Adr_functie( x[i], y[i])) )/3;
   } cât timp( fabs( prec-y[i]) >= eps);
   }
}
```

Observaţii:
1) Notaţiile '*Adr_functie*' şi '**Adr_functie*' sunt *simbolice*. Ele reprezintă, respectiv, adresa unei funcţii (deci un pointer la o funcţie) şi apelul unei funcţii prin pointer. Vedeţi *Anexa B* - "*Noţiuni de C necesare desfăşurării lucrărilor de laborator*" pentru lămuriri.
2) Alegerea numărului 'N' de componente al vectorilor 'x[ ]' şi 'y[ ]' se face în funcţie de numărul 'n' al punctelor prin care se doreşte aproximarea soluţiei. Între *N* şi *n* există legătura: *N=n+1*, deoarece în C indexarea vectorilor se face de la valoarea 0.

3. DESFĂŞURAREA LUCRĂRII

3.1. ÎNTREBĂRI

1. Care este deosebirea dintre metodele Euler îmbunătăţită şi Euler modificată?
2. Găsiţi ecuaţii diferenţiale de ordinul I pentru care atât metoda Euler îmbunătăţită cât şi metoda Euler modificată dau acelaşi rezultat, în toate punctele. Explicaţi.
3. Cum se interpretează precizia metodelor de rezolvare a ecuaţiilor diferenţiale?
4. Comparaţi metodele Euler prezentate cu metoda Euler de bază, în care nu apar corecţii de tip *medie*.

5. Ce legătură există între mărimea pasului *h* şi generalitatea metodelor prezentate?

3.2. PROBLEME LABORATOR

1. Să se implementeze algoritmii metodelor 2.1.1—2.1.3 şi 2.2.1—2.2.2 în limbajul C.
2. Se consideră un circuit *R,L* în serie alimentat de o sursă de curent alternativ $e = 10\cos(100\pi t)$ *V*, f *=50 Hz*, prezentat în fig.(8.3).

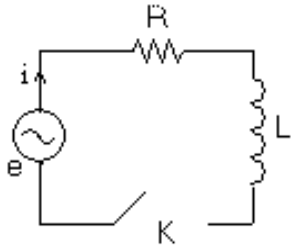

Fig. 9.3 - Circuitul R,L

Cunoscând R=*10Ω,* L=10mH şi că la *t=0* avem *i=0,* să se calculeze valorile curentului la *t* = 0.001; 0.002; 0.003; 0.004; 0.005; 0.006; 0.007; 0.008; 0.009; 0.01*s*.

3.3. TEME DE CASĂ

1. Să se rezolve ecuaţia diferenţială:
   `y'=2x+y ,`
   dacă: `x(0)=1, h=0.01` şi `N=7`.
   Să se găsească apoi forma analitică a acestei curbe folosind o metodă adecvată de optimizare. Discutaţi rezultatul. Dar dacă `h=0.25`?

2. Găsiţi un exemplu de aplicaţie din electronică astfel încât prin formularea problemei să ajungeţi la necesitatea rezolvării unei ecuaţii diferenţiale de ordinul I. Rezolvaţi apoi această ecuaţie printr-o metodă numerică, la alegere.

BIBLIOGRAFIE

I. Rusu – "*Metode numerice în electronică. Aplicaţii în limbaj C*", Editura Tehnică, Bucureşti, 1997.