Digitální učební materiál

Číslo projektu:	CZ.1.07/1.5.00/34.0548
Název školy:	Gymnázium, Trutnov, Jiráskovo náměstí 325
Název materiálu:	VY_32_INOVACE_157_IVT
Autor:	Ing. Pavel Bezděk
Tematický okruh:	Algoritmy
Datum tvorby:	září 2013
Ročník:	4. ročník a oktáva
Anotace:	Algoritmus XVII. – Algoritmus řešení soustav lineárních rovnic III. Gauss – Seidelova metoda
Metodický pokyn:	Při výuce nutno postupovat individuálně. Části DUM – „ Pro hloubavé“, jsou určeny pro zájemce o studium na technických a matematicko-fyzikálních oborech vysokých škol.

esf evropský sociální fond v ČR
EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Autor	**Ing. Pavel Bezděk**		
Vytvořeno dne	**14. 9. 2013**		
Odpilotováno dne	**24. 3. 2014**	**ve třídě**	**8.Y**
Vzdělávací oblast	**Informatika a informační a komunikační technologie**		
Vzdělávací obor	**Informatika a výpočetní technika**		
Tematický okruh	**Algoritmus**		
Téma	**Algoritmus XVII. – Algoritmus řešení soustavy lineárních rovnic III. - Gauss - Seidelova metoda**		
Klíčová slova	**Algoritmus, Gauss - Seidelova metoda, aproximace**		

Řešení soustavy lineárních rovnic

Gauss – Seidelova metoda

Co dělat, když počítač lineární soustavu rovnic není schopen vyřešit v reálném čase!

Řešíme-li soustavy lineárních rovnic na počítači a používáme Gaussovu eliminaci, pak se vzrůstajícím počtem rovnic a neznámých, rostou časy výpočtu a při určitých počtech neznámých (asi 20) a rovnic jsou časy výpočtu již nereálné.

V takových případech používáme metody, jejichž výpočty jsou podstatně rychlejší (probíhají v reálných časech).

Řeší tyto soustavy rovnic metodami postupného přibližování se k přesnému řešení.

Vytváříme tak posloupnost aproximací (přibližných řešení), jejíž limitou je přesné řešení x_i dané soustavy rovnic.

Budeme řešit soustavu rovnic:

$$\begin{aligned} 11x + 2y + z &= 15 \\ x + 10y + 2z &= 16 \\ 2x + 3y - 8z &= 1 \end{aligned}$$

Pro jednoduchost si vzali jen 3 rovnice o 3 neznámých a ukážeme si na nich princip řešení n - rovnic o n - neznámých, kde např. n>20, které již nelze řešit Gaussovou eliminací.

Postupné přibližování se k stále přesnějšímu řešení

$x = (15 - 2y - z)/11$ **Nejprve zvolíme hodnotu nulové aproximace , budou to**

$y = (16 - x - 2z)/10$ **hodnoty na pravé straně rovnice.** $x^{(0)} = 15$; $y^{(0)} = 16$; $z^{(0)} = 1$

$z = (-1 + 2x + 3y)/8$ **První aproximaci již vypočteme z nulové aproximace.**

Z hodnot aproximace $x^{(0)}, y^{(0)}, z^{(0)}$, vypočítáváme následující hodnotu aproximace $x^{(1)}$. Následující hodnotu $y^{(1)}$, ale počítáme z hodnot $x^{(1)}$ a $z^{(0)}$, protože $x^{(1)}$ jsme již vypočetli. Následující hodnotu $z^{(1)}$, ale počítáme z hodnot $x^{(1)}$ a $y^{(1)}$, protože $x^{(1)}$ a $y^{(1)}$ jsme již vypočetli.

$x^{(0)} = 15$

$y^{(0)} = 16$

$z^{(0)} = 1$

$x^{(1)} = (15 - 2y^{(0)} - z^{(0)})/11$

$y^{(1)} = (16 - x^{(1)} - 2z^{(0)})/10$

$z^{(1)} = (-1 + 2x^{(1)} + 3y^{(1)})/8$

$x^{(2)} = (15 - 2y^{(1)} - z^{(1)})/11$

$y^{(2)} = (16 - x^{(2)} - 2z^{(1)})/10$

$z^{(2)} = (-1 + 2x^{(2)} + 3y^{(2)})/8$

$x^{(3)} = (15 - 2y^{(2)} - z^{(2)})/11$

$y^{(3)} = (16 - x^{(3)} - 2z^{(2)})/10$

$z^{(3)} = (-1 + 2x^{(3)} + 3y^{(3)})/8$

$x^{(4)} = (15 - 2y^{(3)} - z^{(3)})/11$

$y^{(4)} = (16 - x^{(4)} - 2z^{(3)})/10$

$z^{(4)} = (-1 + 2x^{(4)} + 3y^{(4)})/8$

Když vezmeme obecný zápis soustavy 3 rovnic

$$a_{11}*x_1 + a_{12}*x_2 + a_{13}*x_3 = b_1$$
$$a_{21}*x_1 + a_{22}*x_2 + a_{23}*x_3 = b_2$$
$$a_{31}*x_1 + a_{32}*x_2 + a_{33}*x_3 = b_3$$

Dostaneme pro výpočet aproximací:

$$x_1^{(k+1)} = (b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)})/a_{11}$$
$$x_2^{(k+1)} = (b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)})/a_{22}$$
$$x_3^{(k+1)} = (b_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)})/a_{33}$$

Pro n - rovnic o n - neznámých

$$a_{11}*x_1 + a_{12}*x_2 + a_{13}*x_3 \ \ldots\ldots\ldots\ldots\ a_{1n}*x_n = b_1$$

$$a_{21}*x_1 + a_{22}*x_2 + a_{23}*x_3 \ \ldots\ldots\ldots\ldots\ a_{2n}*x_n = b_2$$

$$a_{31}*x_1 + a_{32}*x_2 + a_{33}*x_3 \ \ldots\ldots\ldots\ldots\ a_{3n}*x_n = b_3$$

.

$$a_{n1}*x_1 + a_{n2}*x_2 + a_{n3}*x_3 \ \ldots\ldots\ldots\ldots\ a_{nn}*x_n = b_n$$

$x^{(0)} = (b_1, b_2, b_3, \ \ldots\ldots\ldots b_n)$

(k+1) aproximace i- té proměnné: pro i= 1,2,3,…….n

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}\, x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}\, x_j^{(k)}\right)$$

Jednoduchá varianta programu na GaussSeidelovu metodu - Pascal

```
Program Gauss_Seidel;
   uses crt;
const N=20;
var  a11,a12,a13,a21,a22,a23,a31,a32,a33,b1,b2,b3,x,y,z,x2,y2,z2:real;
      i:integer;
  begin
    read(a11, a12, a13, b1, a21, a22, a23, b2, a31, a32, a33, b3);
    x:=b1;  y:=b2;  z:=b3;
    for i:=1 to N  do
      begin
        x2:=(-a12*y-a13*z+b1)/a11;
        y2:=(-a21*x2-a23*z+b2)/a22;
        z2:=(-a31*x2-a32*y2+b3)/a33;
        x:=x2; y:=y2; z:=z2;
        writeln;  write(x:4:5,'  ');  write(y:4:5,'  ');   write(z:4:5,'  ');
      end;
    writeln;  writeln;
repeat until keypressed;
  end.
```

Jednoduchá varianta programu na Gauss-Seidelovu metodu, již lépe provedená - Pascal

```
Program GaussSeidel2;
 uses crt;
var a: array[1..3,1..3] of real;
    b: array[1..3] of real;
    x: array[1..3] of real;
    xx: array[1..3] of real;
   i,ii,j,k,q,p,N:integer;
  begin
    clrscr;
    writeln('Reseni 3 linearnich rovnic o 3 neznamych:');
    for i:=1 to 3 do
      for j:=1 to 3 do begin write('zadej a[',i,j,']:');readln(a[i,j]); end;
    for ii:=1 to 3 do  begin write('zadej b[',ii,']:');readln(b[ii]); end ;
    for ii:=1 to 3 do x[ii]:=b[ii];
    write ('Zadej kolik iteraci chces provest:'); readln(N);
```

```
 for k:=1 to N  do
      begin

      for i:=1 to 3 do
                  begin
                   xx[i]:=0;
                  for j:=1 to 3 do

                              if i<>j then
                                    if i<j then
                           xx[i]:= xx[i]+(-a[i,j]*x[j])
                                      else
                           xx[i]:= xx[i]+(-a[i,j]*xx[j]);
                  xx[i]:= (xx[i]+b[i])/a[i,i];
                  end;
      for p:=1 to  3 do x[p]:=xx[p];
      writeln;
      for q:=1 to 3 do write('  ',x[q]:4:5);
writeln;
      end;

    writeln;   writeln;
    repeat until keypressed;
  end.
```

Pro hloubavé

Gauss-Seidelova metoda pro *m* lin.rovnic o *n* neznámých - Pascal

```
Program GaussSeidel3;
 uses crt;
 const MM=30;
var a: array[1..MM,1..MM] of real;
    b: array[1..MM] of real;
    x: array[1..MM] of real;
    xx: array[1..MM] of real;
   i,ii,j,k,q,p,N,M:integer;
  begin
    clrscr;
    writeln('Reseni M linearnich rovnic o M neznamych:');
     write('Zadej pocet neznamych (max.30):');readln(M);
    for i:=1 to  M DO
      for j:=1 to M  do begin write('zadej a[',i,j,']:');readln(a[i,j]); end;
    for ii:=1 to M do  begin write('zadej b[',ii,']:');readln(b[ii]); end ;
    for ii:=1 to M do x[ii]:=b[ii];
    write ('Zadej kolik iteraci chces provest:'); readln(N);
```

```
writeln;   writeln;
for k:=1 to N  do
      begin
      for i:=1 to M do
                  begin
                   xx[i]:=0;
                  for j:=1 to M do

                              if i<>j then
                                    if i<j then
                           xx[i]:= xx[i]+(-a[i,j]*x[j])
                                     else
                           xx[i]:= xx[i]+(-a[i,j]*xx[j]);

                  xx[i]:= (xx[i]+b[i])/a[i,i];
                  end;
      for p:=1 to  M do x[p]:=xx[p];
      writeln;
      for q:=1 to M do write('  ',x[q]:4:6); writeln;
      end;
   writeln;   writeln;
   repeat until keypressed;
 end.
```

Gauss-Seidelova metoda C++

```
// Gauss-Seidelova metoda
#include <iostream>
#include <math.h>
#include <conio.h>
#define N 10
using namespace std;

double a[N][N];                    //matice soustavy
double b[N];                       //vektor pravych stran
double x[N];                       //reseni rovnice
double xold[N];                    //predchozi priblizeni
double delta,aii;
double eps=0.0001;          //pripustna chyba
int i,j,k,n;
```

```
int main()
{       //nacteni vstupu
cti:cout << "Zadejte dimenzi soustavy:\n";
            cin >> n;
            if (n>N)
            {
                        cout << "n nesmi byt vetsi nez N\n";
                        goto cti;
            }
            cout << "Zadejte matici po radcich:\n";
            for(i=0;i<n;i++)
                        for(j=0;j<n;j++)
                                    cin >> a[i][j];
            cout << "Zadejte prave strany rovnic:\n";
            for(i=0;i<n;i++)
            {
                        cin >> b[i];
            }
```

```
cout << "uprava\n";                                          // uprava
for(i=0;i<n;i++)
{           aii=a[i][i];  b[i]=b[i]/aii;   for(j=0;j<n;j++)
            {       a[i][j]=a[i][j]/aii ;          }
            x[i]=0; xold[i]=0;
}
x[0]=b[0]; xold[0]=b[0];
cout << "iteracni postup\n";                                 // iteracni postup
do
{           delta=0;    for(i=0;i<n;i++)
            {           x[i]=b[i];    for(j=0;j<n;j++)
                        {   if (i!=j)  {     x[i]=x[i]-a[i][j]*x[j];    }
                        }
                        if (delta < fabs(x[i]-xold[i])) delta=fabs(x[i]-xold[i]);
                        xold[i]=x[i];
            }
            for(i=0;i<n;i++)        cout << "x" <<i <<"=" << x[i] << endl;
}
while (delta > eps);
```

```
// tisk vysledku
for(i=0;i<n;i++)
        cout << "x" <<i <<"=" << x[i] << endl;
        getch();
        return 0;
}
```

počet aproximací	x	y	z
(0)	15,00000	16,00000	1,00000
(1)	-1,63636	1,56364	0,05227
(2)	1,07459	1,48209	0,69943
(3)	1,03058	1,35706	0,64154
(4)	1,05858	1,36583	0,65183
(5)	1,05603	1,36403	0,65052
(6)	1,05649	1,36425	0,65072
(7)	1,05644	1,36421	0,65069
(8)	1,05644	1,36422	0,65069
(9)	1,05644	1,36422	0,65069
(10)	1,05644	1,36422	0,65069
(11)	1,05644	1,36422	0,65069
(12)	1,05644	1,36422	0,65069
(13)	1,05644	1,36422	0,65069
(14)	1,05644	1,36422	0,65069
(15)	1,05644	1,36422	0,65069

Výpočet na počítači

Přesné hodnoty řešení soustavy rovnic:
x = 1,05644 y = 1,36422 z = 0,65069

Zjišťujeme, již při 4. aproximaci jsme blízko přesnému výsledku.
A pohybujeme se prakticky už jen velmi blízko přesného řešení.
Při 8. aproximaci se výpočet stabilizuje na přesném výsledku.

Ne vždy, bude posloupnost aproximací řešení soustavy rovnic dobře konvertovat.

Rozbor, pro kterou soustavu lineár. rovnic se nám nepodaří dojít k řešení, je však už záležitostí vyšší matematiky, proto se jím nebudeme zabývat.

Pro vytvoření představy, jak se obecně řeší soustavy n rovnic o n neznámých, které nejsou řešitelné Gaussovou eliminací, nám to stačí.

Na závěr něco, pro hloubavé.

Aitkenova extrapolační formule

Pokud hodnoty aproximací konvergují pomalu, lze zlepšit konvergenci, tak že budeme vylepšovat aproximace Aitkenovou extrapolační formulí.

$$x_i \approx x_i^{(k+1)} - (x_i^{(k+1)} - x_i^{(k)})^2/(x_i^{(k+1)} - 2x_i^{(k)} + x_i^{(k-1)})$$

Když hodnotu aproximace $x_i^{(k+1)}$ snížíme o $(x_i^{(k+1)} - x_i^{(k)})^2/(x_i^{(k+1)} - 2x_i^{(k)} + x_i^{(k-1)})$ dostaneme hodnotu aproximace $x_i^{(k+1)}$, která se více blíží přesné hodnotě x_i než původní $x_i^{(k+1)}$.

Použité zdroje

MÍKA, Stanislav. *Numerické metody algebry*. Vyd. 2. Praha: SNTL, 1985, 169 s. Matematika pro vysoké školy technické, sešit IV.