

# 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_156_IVT                                                                                                                                                     |
| Autor:                                                                       | Ing. Pavel Bezděk                                                                                                                                                         |
| Tematický okruh:                                                             | Algoritmy                                                                                                                                                                 |
| Datum tvorby:                                                                | Září 2013                                                                                                                                                                 |
| Ročník:                                                                      | 4. ročník a oktáva                                                                                                                                                        |
| Anotace:                                                                     | Algoritmus XVI. – Algoritmus řešení soustav lineárních rovnic II.<br>Jacobiho metoda                                                                                      |
| Metodický pokyn:                                                             | Při výuce nutno postupovat individuálně.<br>Části DUM – „Pro hloubavé“, jsou určeny pro zájemce o studium na technických a matematicko-fyzikálních oborech vysokých škol. |
| Pokud není uvedeno jinak, je použitý materiál z vlastních zdrojů autora DUM. |                                                                                                                                                                           |



|                          |                                                                                             |                 |            |
|--------------------------|---------------------------------------------------------------------------------------------|-----------------|------------|
| <b>Autor</b>             | <b>Ing. Pavel Bezděk</b>                                                                    |                 |            |
| <b>Vytvořeno dne</b>     | <b>9. 9. 2013</b>                                                                           |                 |            |
| <b>Odpilotováno dne</b>  | <b>31. 3. 2014</b>                                                                          | <b>ve třídě</b> | <b>8.Y</b> |
| <b>Vzdělávací oblast</b> | <b>Informatika a informační a komunikační technologie</b>                                   |                 |            |
| <b>Vzdělávací obor</b>   | <b>Informatika a výpočetní technika</b>                                                     |                 |            |
| <b>Tematický okruh</b>   | <b>Algoritmus</b>                                                                           |                 |            |
| <b>Téma</b>              | <b>Algoritmus XVI. – Algoritmus řešení soustavy lineárních rovnic II. – Jacobiho metoda</b> |                 |            |
| <b>Klíčová slova</b>     | <b>Algoritmus, Jacobiho metoda, aproximace</b>                                              |                 |            |

# **Řešení soustavy lineárních rovnic**

## **Jacobiho 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  $\mathbf{x}_i$  dané soustavy rovnic.**

**Budeme řešit soustavu rovnic:**

$$11x + 2y + z = 15$$

$$x + 10y + 2z = 16$$

$$2x + 3y - 8z = 1$$

**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 rovnou 0.**

$$y = (16 - x - 2z)/10$$

$$x^{(0)} = 0; y^{(0)} = 0; z^{(0)} = 0$$

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

**První aproximaci již vypočteme z nulové aproximace.**

Z hodnot aproximace  $x^{(0)}, y^{(0)}, z^{(0)}$ , vypočítáváme následující hodnoty aproximací  $x^{(1)}, y^{(1)}, z^{(1)}$ .

Z hodnot aproximace  $x^{(1)}, y^{(1)}, z^{(1)}$ , vypočítáváme následující hodnoty aproximací  $x^{(2)}, y^{(2)}, z^{(2)}$ ,  
atd.

$$x^{(0)} = 0$$

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

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

$$y^{(0)} = 0$$

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

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

$$z^{(0)} = 0$$

$$z^{(1)} = (-1 + 2x^{(0)} + 3y^{(0)})/8$$

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

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

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

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

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

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

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

**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:**

$$x_1^{(k+1)} = (b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)})/a_{11}$$

$$x_2^{(k+1)} = (b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)})/a_{22}$$

$$x_3^{(k+1)} = (b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)})/a_{33}$$

## Pro n - rovnic o n - neznámých

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

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

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

.

.

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

$$\mathbf{x}^{(0)} = (0, 0, 0, 0, \dots\dots\dots 0)$$

**(k+1) aproximace i- té proměnné:** pro  $i = 1, 2, 3, 4, \dots\dots\dots n$

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

# Jednoduchá varianta programu na Jacobiho metodu - Pascal

**Program Jacobi;**

**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:=0.0; y:=0.0; z:=0.0;**

**for i:=1 to N do**

**begin**

**x2:=(-a12\*y-a13\*z+b1)/a11;**

**y2:=(-a21\*x-a23\*z+b2)/a22;**

**z2:=(-a31\*x-a32\*y+b3)/a33;**

**x:=x2; y:=y2; z:=z2;**

**writeln; write(x:4:5,' '); write(y:4:5,' '); write(z:4:5,' '); delay(500);**

**end;**

**writeln; writeln; repeat until keypressed;**

**end.**

# Jednoduchá varianta programu na Jacobiho metodu, již lépe provedená - Pascal

Program Jacobi;

uses crt;

const N=50;

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,j,k,l,m,p: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 i:=1 to 3 do begin write('zadej b[' ,i,']:');readln(b[i]); end ;

    for i:=1 to 3 do x[i]:=0;

    { Chceme-li sami navolit pocet iteraci, zrusime konstantu, deklarujte promenou N.

    write ('Zadej kolik iteraci chces provest:'); readln(N);}

```
for i:=1 to N do
  begin
    for k:=1 to 3 do
      begin
        xx[k]:=0;
        for m:=1 to 3 do if k<>m then
          xx[k]:= xx[k]+(-a[k,m]*x[m]);
        xx[k]:= (xx[k]+b[k])/a[k,k];
      end;
    for p:=1 to 3 do x[p]:=xx[p];
    writeln;
    for l:=1 to 3 do write(' ',x[l]:4:5); writeln;
    end;

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

## Pro hloubavé

# Jacobiho metoda pro $m$ lin.rovnic o $n$ neznámých - Pascal

Program Jacobi;

uses crt;

const MaxRadku=30;

const MaxSloupcu=30;

var a: array[1..MaxRadku,1..MaxSloupcu] of real;

    b: array[1..MaxRadku] of real;

    x: array[1..MaxSloupcu] of real;

    xx: array[1..MaxSloupcu] of real;

    i,j,k,l,q,p,M,N,PocetIteraci:integer;

begin

    clrscr;

    writeln('Reseni M linearnich rovnic o N neznamych:');

    write('Zadej pocet lin. rovnic:'); Readln(M);

    write('Zadej pocet neznamych:'); Readln(N);

    for i:=1 to M do

        for j:=1 to N do begin write('zadej a['*i*,'*j*,']);readln(a[*i*,*j*]); end;

    for i:=1 to M do begin write('zadej b['*i*,']);readln(b[*i*]); end ;

```

for i:=1 to N do x[i]:=0;
  writeln;
  write('Zadej pocet iteraci, ktere se maji provest pri vypoctu:');Readln(PocetIteraci);

for i:=1 to PocetIteraci do
  begin
    for k:=1 to N do
      begin
        xx[k]:=0;
        for q:=1 to N do if k<>q then
          xx[k]:= xx[k]+(-a[k,q]*x[q]);
        xx[k]:= (xx[k]+b[k])/a[k,k];
      end;
    for p:=1 to N do x[p]:=xx[p];
    writeln;
    for l:=1 to N do write(' ',x[l]:4:5); writeln;
  end;

writeln; writeln;
repeat until keypressed;
end.

```

# Jacobiho metoda

## C++

```
// Jacobiho 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,a[i];
double eps=0.0001;        //pripustna chyba
int i,j,k,n;
int main()
{ cti:cout << " Zadejte dimenzi soustavy - pocet rovnic ( do 10):\n"; //nacteni vstupu
  cin >> n;                if (n>N)
    {cout << " Pozor, n nesmi byt vetsi nez N! N=10 Znovu zadavej!\n "; goto cti;}
  cout << "Zadejte matici po radcich (leve strany):\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];
    }

    // uprava
cout << "uprava\n";
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]=b[i]; xold[i]=b[i];
}

```

```

// iteracni postup
    cout << "iteracni postup\n";
    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]*xold[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;
}

```

# Jacobiho metoda

## pro hloubavé

C++

```
// Metoda Jacobi
// #include "stdafx.h"
#include <iostream>
#include <math.h>
#include <conio.h>
using namespace std;
#define max 10
//const int max=10;
int main()
{
    double s, s1, eps;
    double a[max + 1][max + 1], b[max + 1], x[max + 1], y[max + 1];
    int n, i, j;
    cout << "Zadej pocet rovnic: " << endl;    cin >> n;
    cout << " Nacti matici soustavy (leve strany ): " << endl;
    for (i = 1; i <= n; i++) for (j = 1; j <= n; j++)
        {
            cout << "a[" << i << j << "]=";    cin >> a[i][j];
        }
    cout << endl;
```

```

cout << "Nacti prave strany rovnic: " << endl;
    for (i = 1; i <= n; i++)
    {
        cout << "b[" << i << "]=";    cin >> b[i];
    }
    cout << endl;
    cout << "Pocatecni aproximace ( napr. x[1]=0): " << endl;
    for (i = 1; i <= n; i++)
    {
        cout << "x[" << i << "]=";                cin >> y[i];
    }
    cout << endl;
cout << " Velikost chyby (např. 0.0001): " << endl;    cin >> eps;    cout << endl;
    s = eps + 1;
    while (s >= eps)
    {
        for (i = 1; i <= n; i++)
        {
            s1 = 0;    for (j = 1; j <= n; j++)
                if (j != i) s1 = s1 + a[i][j] * y[j]; x[i] = (b[i] - s1) / a[i][i];
        }
        s = 0;    for (i = 1; i <= n; i++)    s = s + fabs(x[i] - y[i]);
        for (i = 1; i <= n; i++)    y[i] = x[i];
    }

```

```
cout << "Přibližná řešení " << endl;  
    for (i = 1; i <= n; i++)  
        cout << x[i] << endl;  
    return 0;  
}
```

# Výpočet na počítači

| počet<br>aproximací | x       | y       | z        |
|---------------------|---------|---------|----------|
| (0)                 | 0,00000 | 0,00000 | 0,00000  |
| (1)                 | 1,36364 | 1,60000 | -0,12500 |
| (2)                 | 1,00409 | 1,48864 | 0,81591  |
| (3)                 | 1,01800 | 1,32841 | 0,70426  |
| (4)                 | 1,05808 | 1,35727 | 0,62785  |
| (5)                 | 1,05978 | 1,36862 | 0,64850  |
| (6)                 | 1,05584 | 1,36432 | 0,65318  |
| (7)                 | 1,05620 | 1,36378 | 0,65058  |
| (8)                 | 1,05653 | 1,36426 | 0,65047  |
| (9)                 | 1,05646 | 1,36425 | 0,65073  |
| (10)                | 1,05643 | 1,36421 | 0,65071  |
| (11)                | 1,05644 | 1,36421 | 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  |

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

**Zjišťujeme, již při 4. aproximaci  
jsme blízko přesnému výsledku.**

**Blížíme se stále více k přesnému  
řešení.**

**Při 12. 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.

# Použité zdroje

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