Marcin Rociek
Informatyka, II rok

Metody Obliczeniowe w Nauce i Technice

laboratorium

zestaw 9: równania różniczkowe

Zadanie 1: Dane jest równanie różniczkowe (zagadnienie początkowe): $y'+ycosx=sinxcosx$, $y(0)=0$.
Znaleźć rozwiązanie metodą Rungego-Kutty i metodą Eulera. Porównać otrzymane rozwiązania z rozwiązaniem dokładnym $y(x)=e^{-sinx}+sinx-1$

W metodzie Eulera korzysta się z wzoru:

$$y_{n+1} = y_n + h \cdot f(x_n, y_n), \qquad n \geq 1$$

W przypadku naszego zadania, funkcja *f* jest określona następująco:

$$f(x, y) = \sin x \cos x - y \cos x = (\sin x - y)\cos x$$

Treść programu realizującego zadanie:

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

/*
 * Dane rownanie rozniczkowe
 */
double func(double x, double y)
{
  return ((sin(x)-y)*cos(x));
}

/*
 * Rozwiazanie dokladne
 */
double propersolution(double x)
{
  return exp(-sin(x))+sin(x)-1.0;
}

/*
 * Funkcja rozwiazujaca metoda Eulera
 */
double *euler(int npoints, double x0, double y0, double h, double (*funct)(double, double))
{
  int i;
  double *y = (double*)malloc(npoints*sizeof(double));

  y[0] = y0;
  for (i=0; i<npoints; i++)
  {
    double x = x0+h*i;
    y[i+1] = y[i]+h*funct(x, y[i]);
  }
  return y;
}

/*
 * Czesc glowna programu
 */
void main(void)
{
  double h = 0.1;                       // krok
  int n = 40;                           // ilosc punktow
  int i;
  double *y = euler(n, 0, 0, h, func);  // rozwiazanie
  double diff, maxdiff = 0;

  // wypisanie wynikow
  printf("\t  wyliczone\t\t  dokladne\n");
```

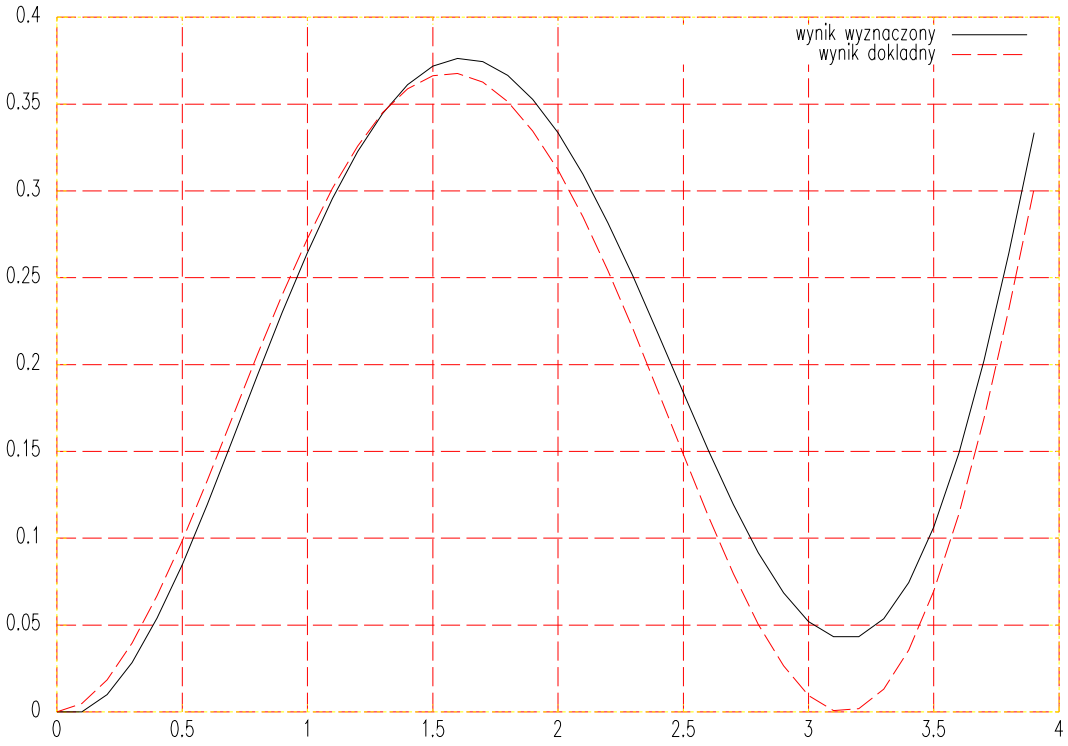
```
  for (i=0; i<n; i++)
  {
    double x = h*i;
    printf("y(%.2f)=%.10f\t\t%.10f\n", x, y[i], propersolution(x));
    diff = fabs(y[i]-propersolution(x));
    if (diff>maxdiff)
      maxdiff = diff;
  }
  printf("maksymalna roznica: %.10f\n", maxdiff);
}
```

Wynik działania programu:

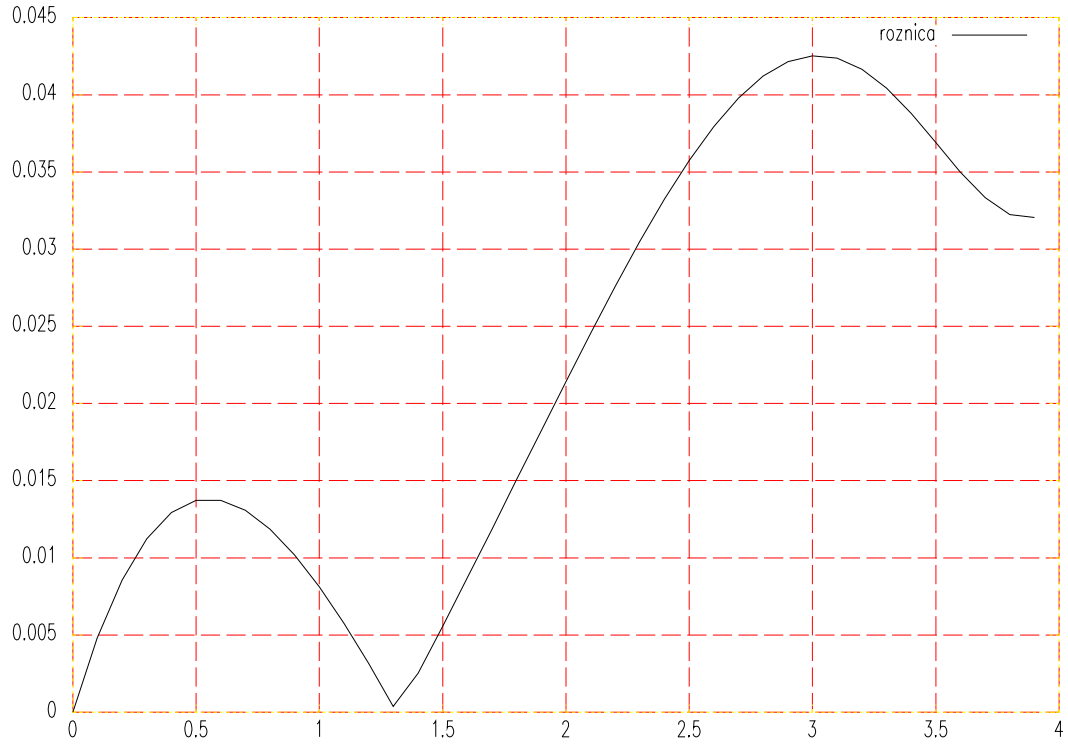
```
         wyliczone             dokladne
y(0.00)=0.0000000000         0.0000000000
y(0.10)=0.0000000000         0.0048215781
y(0.20)=0.0099334665         0.0184902689
y(0.30)=0.0284308378         0.0396645846
y(0.40)=0.0539468598         0.0668691467
y(0.50)=0.0848458295         0.0985644997
y(0.60)=0.1194734567         0.1332058604
y(0.70)=0.1562148411         0.1692908401
y(0.80)=0.1935393575         0.2053969783
y(0.90)=0.2300340208         0.2402103796
y(1.00)=0.2644272583         0.2725469355
y(1.10)=0.2956050639         0.3013676026
y(1.20)=0.3226213531         0.3257890871
y(1.30)=0.3447040772         0.3450910885
y(1.40)=0.3612583521         0.3587210479
y(1.50)=0.3718675546         0.3662971259
y(1.60)=0.3762930680         0.3676099403
y(1.70)=0.3744731186         0.3626234114
y(1.80)=0.3665209434         0.3514748998
y(1.90)=0.3527223539         0.3344746694
y(2.00)=0.3335326050         0.3121045529
y(2.10)=0.3095723341         0.2850155437
y(2.20)=0.2816221842         0.2540238601
y(2.30)=0.2506155775         0.2201048417
y(2.40)=0.2176289423         0.1843838244
y(2.50)=0.1838685333         0.1481229262
y(2.60)=0.1506528297         0.1127024716
y(2.70)=0.1193893685         0.0795956215
y(2.80)=0.0915448043         0.0503347132
y(2.90)=0.0686070284         0.0264679105
y(3.00)=0.0520383749         0.0095051003
y(3.10)=0.0432193601         0.0008526174
y(3.20)=0.0433830881         0.0017374119
y(3.30)=0.0535414594         0.0131227040
y(3.40)=0.0744056383         0.0356189792
y(3.50)=0.1063048295         0.0693962093
y(3.60)=0.1491091463         0.1141052218
y(3.70)=0.2021640277         0.1688178050
y(3.80)=0.2642449643         0.2319960070
y(3.90)=0.3335418714         0.3015007016
maksymalna roznica: 0.0425332746
```

Jak widać, metoda ta nie ma zbyt wielkiej dokładności (zwłaszcza w porównaniu z metodą Rungego-Kutty).

Wykres przedstawiający rozwiązanie wyznaczone i dokładne:



Wykres przedstawiający różnicę między rozwiązaniem wyznaczonym a dokładnym:



Metoda Rungego Kutty:

Wzór ogólny: $y_{n+1} = y_n + \sum_{i=1}^{s} w_i K_i$,

gdzie:

$$K_1 = h \cdot f(x_n, y_n),$$

$$K_i = h \cdot f\left(x_n + a_i \cdot h, y_n + \sum_{j=1}^{i-1} b_{ij} K_j \right), \qquad i > 1$$

w_i, a_i, b_{ij} – stałe.

Jak widać metoda Eulera jest szczególnym przypadkiem metody Rungego-Kutty dla $s = 1$ i $w_1 = 1$.

Dla porównania zaimplementowałem metody Rungego-Kutty rzędu drugiego oraz czwartego.

Dla rzędu drugiego:

$$y_{n+1} = y_n + \left(1 - \frac{1}{2a_2}\right) K_1 + \frac{1}{2a_2} K_2, \quad a_2 \neq 0,$$

gdzie:

$$K_1 = h \cdot f(x_n, y_n), \qquad K_2 = h \cdot f(x_n + a_2 h, y_n + a_2 K_1).$$

Wartość parametru a_2 została dobrana eksperymentalnie na *0.1*, co powoduje mniej więcej dwukrotne zwiększenie dokładności (w porównaniu z $a_2=1$).

Dla rzędu czwartego parametry we wzorze ogólnym wynoszą (tzw „wzór klasyczny”):
$w_1=1/6$, $w_2=1/3$, $w_3=1/3$, $w_4=1/6$,
$a_2=1/2$, $a_3=1/2$, $a_4=1$,
$b_{21}=1/2$, $b_{31}=0$, $b_{32}=1/2$, $b_{41}=0$, $b_{42}=0$, $b_{43}=1$.

Treść progamu realizującego zadanie:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/*
 * Dane rownanie rozniczkowe
 */
double func(double x, double y)
{
  return ((sin(x)-y)*cos(x));
}

/*
 * Rozwiazanie dokladne
 */
double propersolution(double x)
{
  return exp(-sin(x))+sin(x)-1.0;
}

/*
 * Rozwiazanie metoda Rungego-Kutty rzedu drugiego
 */
double *r_k(int npoints, double x0, double y0, double h,
            double (*funct)(double, double))
{
  double *y = (double*)malloc(npoints*sizeof(float));
  int i;
  double a2 = 0.1;

  y[0] = y0;

  for (i=0; i<npoints; i++)
  {
    double x = x0+h*i;
    double k1 = h*funct(x, y[i]);
```

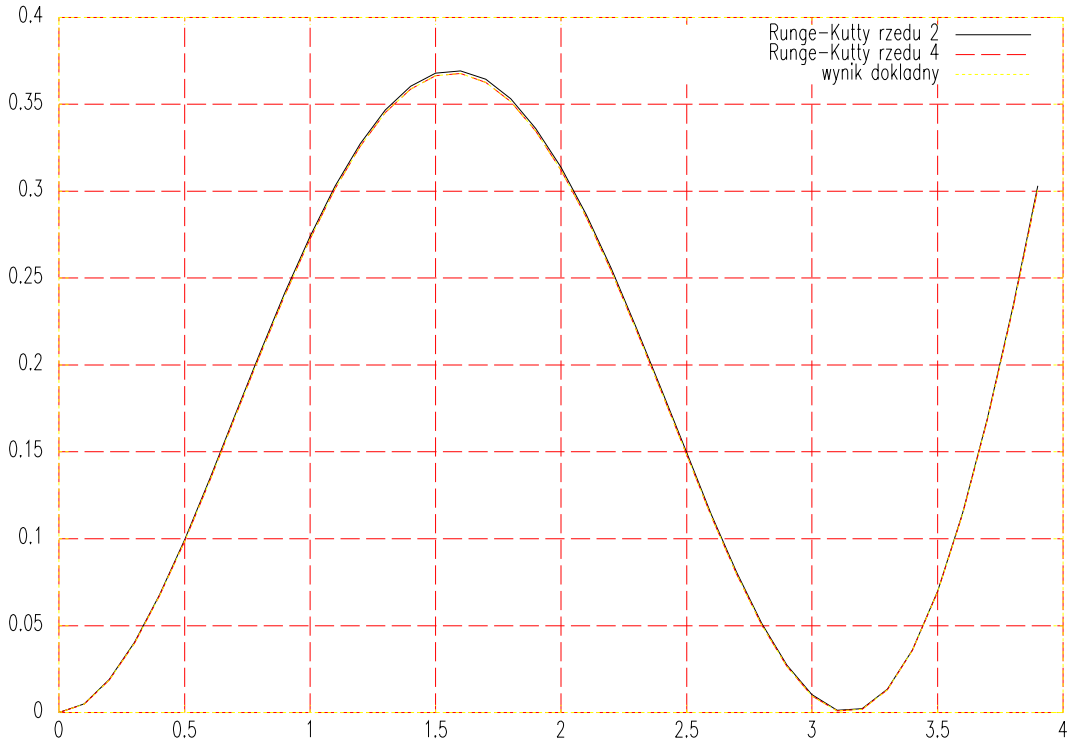
```
    double k2 = h*funct(x+a2*h, y[i]+a2*k1);
    y[i+1] = y[i]+(1.0-(0.5/a2))*k1+(0.5/a2)*k2;
  }
  return y;
}

/*
 * Runge-Kutty czwartego rzedu
 */
double *r_k4(int npoints, double x0, double y0, double h,
             double (*funct)(double x, double y))
{
  double w[4] = { 1.0/6.0, 1.0/3.0, 1.0/3.0, 1.0/6.0 };
  double a[4] = { 0.5, 0.5, 1.0 };
  double b[6] = { 0.5, 0.0, 0.5, 0.0, 0.0, 1.0 };
  int i, j, k, l, bidx;
  double *y, temp, x, K[4], s;

  y = (double*)malloc(npoints*sizeof(double));

  y[0] = y0;
  for (k=0; k<npoints-1; k++)
  {
    x = x0+h*k;
    K[0] = h*( funct(x, y[k]) );
    for (i=1, bidx=0; i<4; i++)
    {
      for (j=0, s=0; j<i; j++, bidx++)
        s += b[bidx]*K[j];
      K[i] = h*funct(x+a[i-1]*h, y[k]+s);
    }
    for (l=0, temp=0; l<4; l++)
      temp += w[l]*K[l];
    y[k+1] = y[k]+temp;
  }
  return y;
}

/*
 * Czesc glowna programu
 */
void main(void)
{
  double h = 0.1;                       // krok
  int n = 40;                           // ilosc punktow
  int i;
  double *y;
  double diff, maxdiff = 0.0;

  printf("rzad 2\n");
  y = r_k(n, 0, 0, h, func);
  printf("\t  wyliczone\t\t  dokladne\n");
  for (i=0; i<n; i++)
  {
    double x = h*i;
    printf("y(%.2f)=%.10f\t\t%.10f\n", x, y[i], propersolution(x));
    diff = fabs(y[i]-propersolution(x));
    if (diff>maxdiff)
      maxdiff = diff;
  }
  printf("maksymalna roznica: %.10f\n", maxdiff);
  free(y);

  maxdiff = 0.0;
  printf("\nrzad 4\n");
  y = r_k4(n, 0, 0, h, func);
  printf("\t  wyliczone\t\t  dokladne\n");
  for (i=0; i<n; i++)
  {
    double x = h*i;
    printf("y(%.2f)=%.10f\t\t%.10f\n", x, y[i], propersolution(x));
    diff = fabs(y[i]-propersolution(x));
    if (diff>maxdiff)
      maxdiff = diff;
  }
  printf("maksymalna roznica: %.10f\n", maxdiff);
  free(y);
}
```

Wynik działania programu:

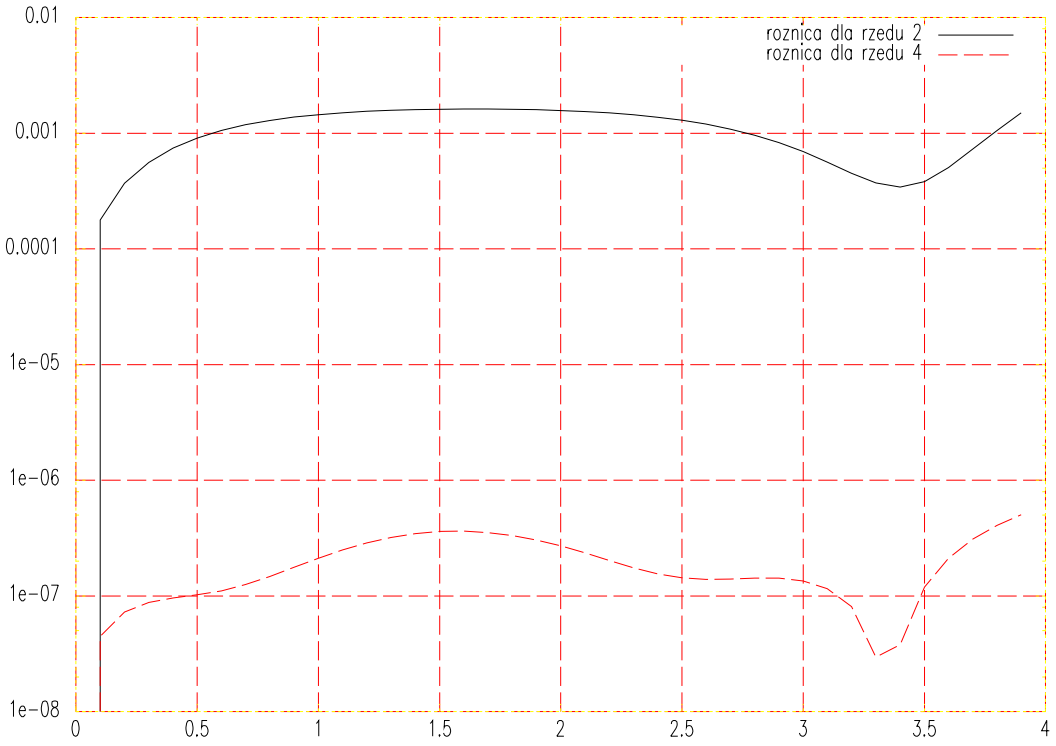
```
rzad 2
         wyliczone              dokladne
y(0.00)=0.0000000000         0.0000000000
y(0.10)=0.0049996667         0.0048215781
y(0.20)=0.0188594090         0.0184902689
y(0.30)=0.0402249310         0.0396645846
y(0.40)=0.0676114512         0.0668691467
y(0.50)=0.0994732733         0.0985644997
y(0.60)=0.1342620720         0.1332058604
y(0.70)=0.1704740524         0.1692908401
y(0.80)=0.2066869140         0.2053969783
y(0.90)=0.2415879699         0.2402103796
y(1.00)=0.2739949318         0.2725469355
y(1.10)=0.3028708452         0.3013676026
y(1.20)=0.3273345163         0.3257890871
y(1.30)=0.3466675762         0.3450910885
y(1.40)=0.3603191089         0.3587210479
y(1.50)=0.3679085555         0.3662971259
y(1.60)=0.3692274107         0.3676099403
y(1.70)=0.3642400500         0.3626234114
y(1.80)=0.3530838675         0.3514748998
y(1.90)=0.3360687512         0.3344746694
y(2.00)=0.3136757790         0.3121045529
y(2.10)=0.2865548604         0.2850155437
y(2.20)=0.2555208794         0.2540238601
y(2.30)=0.2215477095         0.2201048417
y(2.40)=0.1857592593         0.1843838244
y(2.50)=0.1494164949         0.1481229262
y(2.60)=0.1138991748         0.1127024716
y(2.70)=0.0806808689         0.0795956215
y(2.80)=0.0512957544         0.0503347132
y(2.90)=0.0272957634         0.0264679105
y(3.00)=0.0101969617         0.0095051003
y(3.10)=0.0014146614         0.0008526174
y(3.20)=0.0021877702         0.0017374119
y(3.30)=0.0134942938         0.0131227040
y(3.40)=0.0359617086         0.0356189792
y(3.50)=0.0697779912         0.0693962093
y(3.60)=0.1146111818         0.1141052218
y(3.70)=0.1695471389         0.1688178050
y(3.80)=0.2330561461         0.2319960070
y(3.90)=0.3029987972         0.3015007016
maksymalna roznica: 0.0016174704
```

```
rzad 4
y(0.00)=0.0000000000         0.0000000000
y(0.10)=0.0048216227         0.0048215781
y(0.20)=0.0184903414         0.0184902689
y(0.30)=0.0396646725         0.0396645846
y(0.40)=0.0668692428         0.0668691467
y(0.50)=0.0985646018         0.0985644997
y(0.60)=0.1332059711         0.1332058604
y(0.70)=0.1692909653         0.1692908401
y(0.80)=0.2053971255         0.2053969783
y(0.90)=0.2402105563         0.2402103796
y(1.00)=0.2725471474         0.2725469355
y(1.10)=0.3013678528         0.3013676026
y(1.20)=0.3257893747         0.3257890871
y(1.30)=0.3450914090         0.3450910885
y(1.40)=0.3587213935         0.3587210479
y(1.50)=0.3662974860         0.3662971259
y(1.60)=0.3676103030         0.3676099403
y(1.70)=0.3626237645         0.3626234114
y(1.80)=0.3514752326         0.3514748998
y(1.90)=0.3344749735         0.3344746694
y(2.00)=0.3121048233         0.3121045529
y(2.10)=0.2850157788         0.2850155437
y(2.20)=0.2540240623         0.2540238601
y(2.30)=0.2201050165         0.2201048417
y(2.40)=0.1843839792         0.1843838244
y(2.50)=0.1481230693         0.1481229262
y(2.60)=0.1127026105         0.1127024716
y(2.70)=0.0795957615         0.0795956215
y(2.80)=0.0503348559         0.0503347132
y(2.90)=0.0264680530         0.0264679105
y(3.00)=0.0095052351         0.0095051003
y(3.10)=0.0008527327         0.0008526174
y(3.20)=0.0017374926         0.0017374119
y(3.30)=0.0131227336         0.0131227040
y(3.40)=0.0356189414         0.0356189792
y(3.50)=0.0693960900         0.0693962093
y(3.60)=0.1141050107         0.1141052218
y(3.70)=0.1688174962         0.1688178050
y(3.80)=0.2319955993         0.2319960070
y(3.90)=0.3015001980         0.3015007016
maksymalna roznica: 0.0000005036
```

Wykres przedstawiający rozwiązania wyznaczone metodami Rungego-Kutty, oraz rozwiązanie dokładne:



Wykres przedstawiający różnicę pomiędzy rozwiązaniami wyznaczonymi metodami Rungego-Kutty, a rozwiązaniem dokładnym (skala logarytmiczna):



Zadanie 2: Dane jest zagadnienie brzegowe: $y''+y=x,\ y(0)=1,\ y(p/2)=p/2-1$. Znaleźć rozwiązanie metodą strzałów. Porównać otrzymane rozwiązanie z rozwiązaniem dokładnym $y(x)=cosx-sinx+x$.

Treść programu realizującego zadanie:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#ifndef M_PI
  #define M_PI 3.14159265358979323846
#endif

/*
 * Pomocnicza funkcja do alokacji tablicy dwuwymiarowej
 */
double **tab2(int a, int b)
{
  int i;
  double **tab = (double**)malloc(a*sizeof(double*));
  for (i=0; i<a; i++)
    tab[i] = (double*)malloc(b*sizeof(double));
  return tab;
}

/*
 * Pomocnicza funkcja do dealokacji tablicy dwuwymiarowej
 */
void freetab2(double **tab, int a, int b)
{
  int i;
  for (i=0; i<a; i++)
  {
    free(tab[i]);
  }
  free(tab);
}

/*
 * Runge-Kutty czwartego rzedu
 */
double **r_k4(double y0[2], double h, double x0, int npoints,
                        double (*f1)(double x, double y1, double y2),
                        double (*f2)(double x, double y1, double y2))
{
  double w[4] = { 1.0/6.0, 1.0/3.0, 1.0/3.0, 1.0/6.0 };
  double a[4] = { 0.5, 0.5, 1.0 };
  double b[6] = { 0.5, 0.0, 0.5, 0.0, 0.0, 1.0 };
  int i, j, k, l, bidx;
  double **y, temp[2], x, K[4][2];

  y = tab2(npoints, 2);

  y[0][0] = y0[0];
  y[0][1] = y0[1];
  for (k=0; k<npoints-1; k++)
  {
    x = x0+h*k;
    K[0][0] = h*( f1(x, y[k][0], y[k][1]) );
    K[0][1] = h*( f2(x, y[k][0], y[k][1]) );
    bidx = 0;
    for (i=1; i<4; i++)
    {
      double s0 = 0, s1 = 0;
      for (j=0; j<i; j++)
      {
        s0 += b[bidx]*K[j][0];
        s1 += b[bidx]*K[j][1];
        bidx++;
      }
      K[i][0] = h*f1(x+a[i-1]*h, y[k][0]+s0, y[k][1]+s1);
      K[i][1] = h*f2(x+a[i-1]*h, y[k][0]+s0, y[k][1]+s1);
    }
    temp[0] = temp[1] = 0;
    for (l=0; l<4; l++)
    {
      temp[0] += w[l]*K[l][0];
      temp[1] += w[l]*K[l][1];
    }
    y[k+1][0] = y[k][0]+temp[0];
```

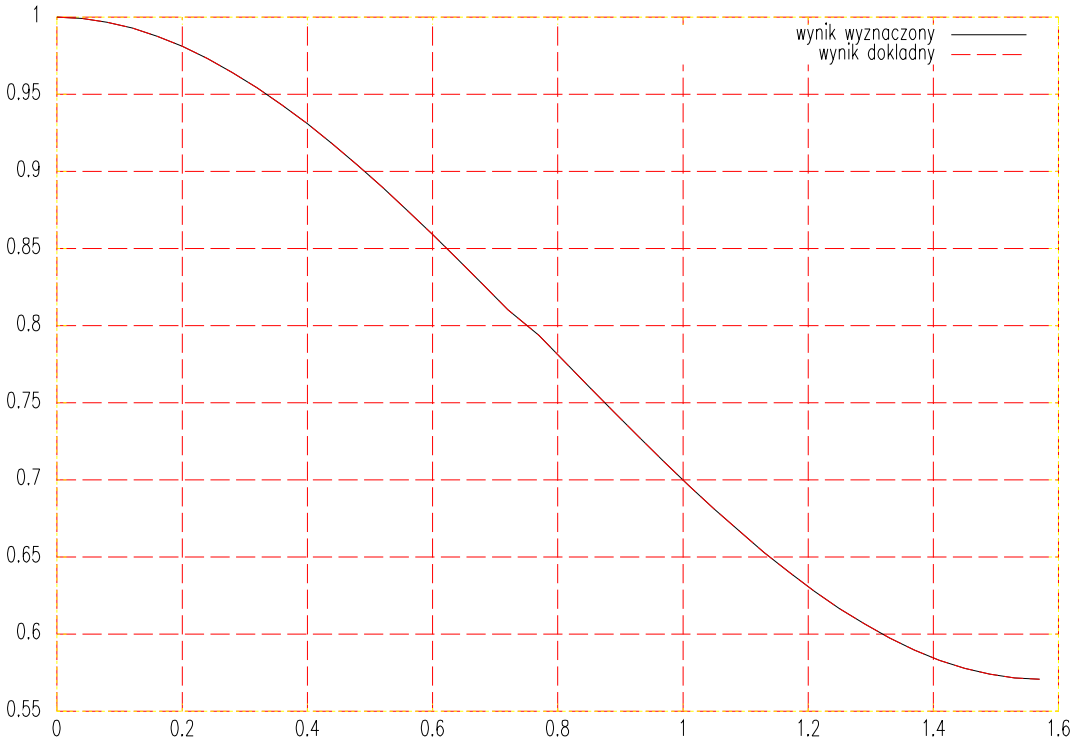
```
    y[k+1][1] = y[k][1]+temp[1];
  }
  return y;
}

/*
 * Funkcja rozwiazujaca zagadnienie brzegowe metoda strzalow
 */
double **strzaly(double xl, double xp, double yl, double yp, int npoints, double precyzja,
                 double (*f1)(double x, double y1, double y2),
                 double (*f2)(double x, double y1, double y2))
{
  double u[3];
  double h = (double)(xp-xl)/(double)(npoints-1);
  int i;
  double y1, y2;
  double y0[2];
  double **y;
  int licznik;

  u[1] = 0;
  u[2] = 1;
  y0[0] = yl;
  y0[1] = u[1];

  y = r_k4(y0, h, xl, npoints, f1, f2);

  y1 = y[npoints-1][0];

  if (fabs(y1-yp)<=precyzja)
    return y;

  freetab2(y, npoints, 2);

  for (i=0; i<100; i++)
  {
    y0[1] = u[2];
    y = r_k4(y0, h, xl, npoints, f1, f2);
    y2 = y[npoints-1][0];
    if (fabs(y2-yp)<=precyzja)
      return y;
    u[0] = u[1];
    u[1] = u[2];
    u[2] = u[0]-(y1-yp)*(double)(u[1]-u[0])/(double)(y2-y1);
    y1 = y2;
    freetab2(y, npoints, 2);
  };

  return NULL;
}

double fun1(double x, double y1, double y2)
{
  return y2;
}

double fun2(double x, double y1, double y2)
{
  return (-y1+x);
}

/*
 * Dokladne rozwiazanie
 */
double propersolution(double x)
{
  return cos(x)-sin(x)+x;
}

/*
 * Czesc glowna programu
 */
void main(void)
{
  double xl = 0;
  double xp = 0.5*M_PI;
  double yl = 1;
  double yp = 0.5*M_PI-1.0;
  int npoints = 40;
  double precyzja = 0.0000000001;
  double h = (xp-xl)/(double)(npoints-1);
  double **y = strzaly(xl, xp, yl, yp, npoints, precyzja, fun1, fun2);
```

```
  double diff, maxdiff;
  int i;

  for (i=0; i<npoints; i++)
  {
    double x = h*i;
    printf("y(%.10f)=%.10f\t\t%.10f\n", x, y[i][0], propersolution(x));
    diff = fabs(y[i][0]-propersolution(x));
    if (diff>maxdiff)
      maxdiff = diff;
  }
  printf("maksymalna roznica: %.10f", maxdiff);
  freetab2(y, npoints, 2);
}
```

Wynik działania programu:

```
y(0.0000000000)=1.0000000000        1.0000000000
y(0.0402768289)=0.9991998864        0.9991998870
y(0.0805536578)=0.9968443962        0.9968443972
y(0.1208304867)=0.9930026791        0.9930026805
y(0.1611073156)=0.9877462956        0.9877462973
y(0.2013841445)=0.9811491007        0.9811491027
y(0.2416609734)=0.9732871243        0.9732871265
y(0.2819378022)=0.9642384476        0.9642384500
y(0.3222146311)=0.9540830768        0.9540830793
y(0.3624914600)=0.9429028131        0.9429028157
y(0.4027682889)=0.9307811202        0.9307811227
y(0.4430451178)=0.9178029885        0.9178029910
y(0.4833219467)=0.9040547979        0.9040548003
y(0.5235987756)=0.8896241771        0.8896241794
y(0.5638756045)=0.8745998618        0.8745998639
y(0.6041524334)=0.8590715506        0.8590715525
y(0.6444292623)=0.8431297597        0.8431297614
y(0.6847060912)=0.8268656759        0.8268656774
y(0.7249829201)=0.8103710088        0.8103710100
y(0.7652597490)=0.7937378418        0.7937378428
y(0.8055365778)=0.7770584833        0.7770584840
y(0.8458134067)=0.7604253164        0.7604253168
y(0.8860902356)=0.7439306493        0.7439306494
y(0.9263670645)=0.7276665655        0.7276665654
y(0.9666438934)=0.7117247746        0.7117247742
y(1.0069207223)=0.6961964635        0.6961964629
y(1.0471975512)=0.6811721482        0.6811721474
y(1.0874743801)=0.6667415275        0.6667415265
y(1.1277512090)=0.6529933369        0.6529933358
y(1.1680280379)=0.6400152054        0.6400152041
y(1.2083048668)=0.6278935125        0.6278935111
y(1.2485816957)=0.6167132490        0.6167132475
y(1.2888585245)=0.6065578783        0.6065578768
y(1.3291353534)=0.5975092017        0.5975092003
y(1.3694121823)=0.5896472254        0.5896472241
y(1.4096890112)=0.5830500306        0.5830500294
y(1.4499658401)=0.5777936473        0.5777936463
y(1.4902426690)=0.5739519303        0.5739519296
y(1.5305194979)=0.5715964402        0.5715964398
y(1.5707963268)=0.5707963268        0.5707963268
maksymalna roznica: 0.0000000025
```

Wykres przedstawiający rozwiązanie wyznaczone metodą strzałów oraz dokładne:



Wykres przedstawiający różnicę między rozwiązaniem dokładnym a wyznaczonym: