Matematiikan laitos
Numeeriset menetelmät ja C–kieli
Loppukoe
25.5.2004

Vastaa **viiteen** tehtävään

1. Tee ohjelma, joka generoi yksikköneliöön m satunnaista pistettä (x_i, y_i), $i = 1, \ldots, m$ ja tulostaa taulukon niiden välimatkoista.

2. a) Selosta suoran sovitus datapisteisiin (x_i, y_i), $i = 1, \ldots, p$, pienimmän neliösumman menetelmällä.
   b) Tee ohjelma, joka generoi synteettistä dataa, sovittaa suoran $y = kx + b$ ja tulostaa kertoimet k ja b.

3. a) Muodosta iteraatiokaava yhtälön $f(x) = 0$, missä $f(x) = x^3 - x^2 - x - 1$, ratkaisemiseksi Newtonin menetelmällä alkuarvolla $x_0 = 2$.
   b) Tee ohjelma, joka toteuttaa ko. iteraation.

4. a) Selosta jonkin minimointialgoritmin toimintaa kahden muuttujan funktion minimoimiseksi. Analysoi tehtävään liittyviä vaikeuksia.
   b) Tee ohjelma, joka etsii funktion $f(x, y) = 100(x^2 - y) + (1 - x)^2$ minimin jonkin kurssilla käytetyn ohjelman avulla.

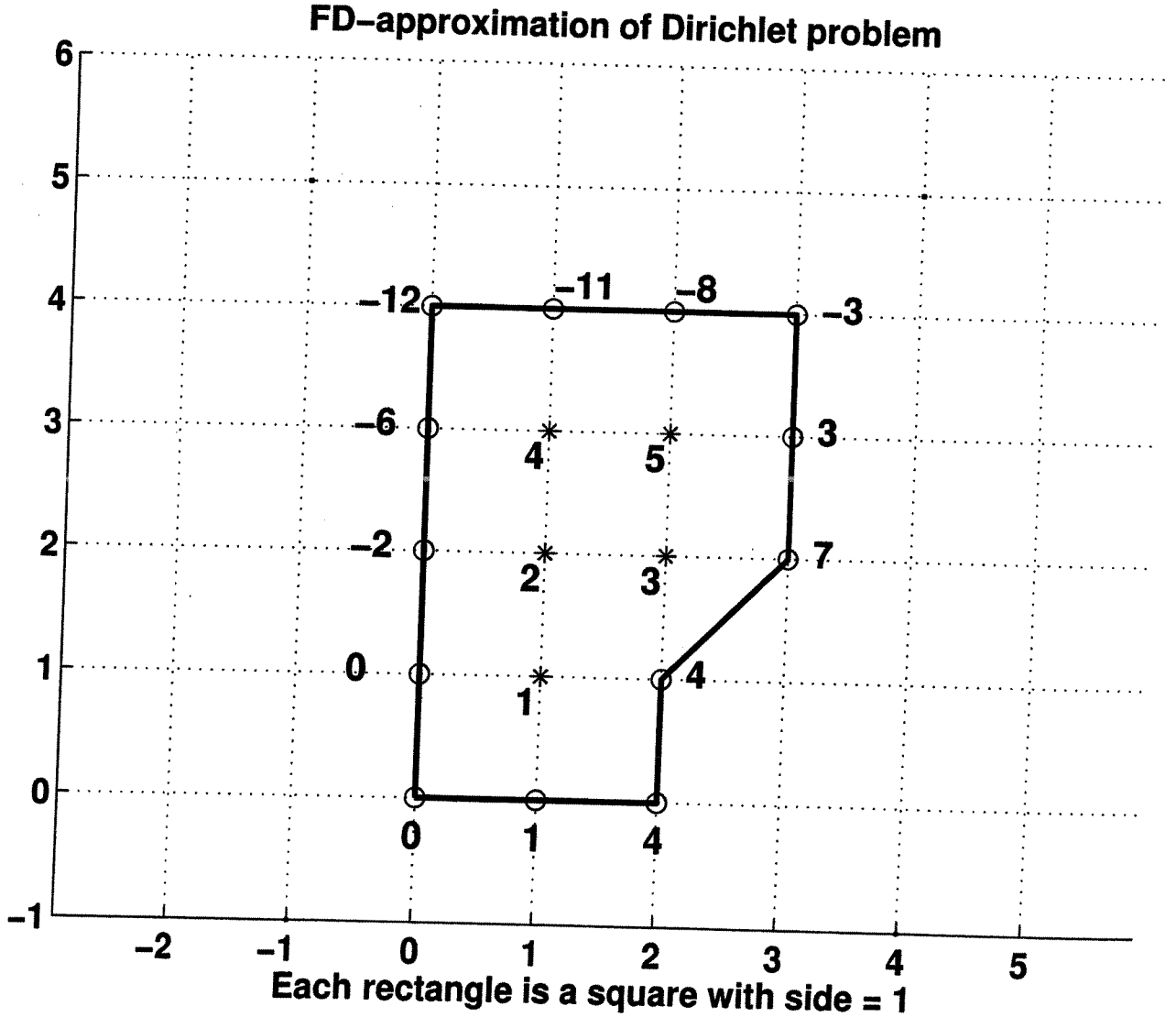
5. Tee ohjelma alkuarvotehtävän $y_1(1) = 3.0$, $y_2(1) = 1.0$

$$\frac{dy_1}{dx} = 10y_1(1.0 - y_2)$$

$$\frac{dy_2}{dx} = y_2(y_1 - 1.0)$$

   ratkaisemiseksi oppikirjan ohjelmilla.

6. Tee ohjelma Dirichletin ongelman ratkaisemiseksi kuvan mukaisessa tilanteessa. Noudata kuvasta ilmenevää tuntemattomien indeksointia. Käytä ratkaisuun kurssin ohjelmaa ja selosta ohjelman syötetiedoston sisältö. Muodosta kyseinen lineaarinen yhtälöryhmä.


FD–approximation of Dirichlet problem
6
5
4
3
2
1
0
−1
−2
−1
0
1
2
3
4
5
−12
−11
−8
−3
−6
3
−2
7
0
4
4
5
2
3
1
0
1
4
Each rectangle is a square with side = 1

Department of Mathematics, University of Helsinki
Numerical methods and the C language, spring 2006
Exam, February 28, 2006 (2 hours)

`Answer four questions! You may use either C++ and matutl04 or C and gmatutl in your work. Header files such as #include <iostream> are not required.`

1. (a) Compare the floating point arithmetic with the properties of the real numbers. What types of errors are likely to occur in floating point arithmetic, and how can they be avoided ?

   (b) Describe some mathematical functionalities usually offered by programming languages and environments. What additional numerical properties would you like the C/C++ language to have, and on what basis ?

   (c) How can one examine how sensitive to errors the numerical solution for an $n \times n$ linear system is ? What methods are there for solving such a system ? Use a method of your choice to solve the system $Ax = b$, where

$$A = \begin{bmatrix} 3.0 & 2.0 & 1.0 \\ 4.0 & 6.0 & 5.0 \\ 7.0 & 8.0 & 9.0 \end{bmatrix} \qquad b = \begin{bmatrix} 2.0 \\ -1.0 \\ -2.0 \end{bmatrix}.$$

2. Make a program that generates a random $n \times n$ matrix A, computes its inverse b with matutl04.cpp, and prints the number $\mathrm{norm}(Id - A * b)$ which should be zero. Here Id is the $n \times n$ identity matrix. Use the programs from `matutl04.cpp`. Writing down the standard header files such as `<stdio.h>` is not required.

3. Suppose that we have two vectors of the same length `x`, `y`. Make a program that prints the value of the scalar product of these vectors.

4. Describe the usage of *one* of the datatypes `Mat_DP` or `gsl_matrix` used on the course, especially considering the following:

   (a) Declaration, memory allocation, component access and printing.

   (b) Show how to handle matrices, including summing two matrices, reading a matrix from a file, and the matrix product of two matrices. You may use the programs from matutl04.cpp/gmatutl.c.

   (c) Make a program which demonstrates the usage of the functions above.

5. Consider the problem of fitting a LSQ line with the data points $(x_i, y_i), i = 1, ..., n$. It is assumed that the line goes through a fixed point (x_0, y_0). Thus we want to minimize the function

$$s(c) = \sum_{i=1}^{n} (y_i - y_0 - c(x_i - x_0))^2.$$

   (a) Write the formula for $s'(c)$ and solve the equation $s'(c) = 0$ for c.

(b) We assume that the data is in the file a.dat (in the standard format of the course). Write a program that reads the data and computes the value of c and writes it on the screen.

Department of Mathematics, University of Helsinki
Numerical methods and the C language, spring 2006
Final Examination, May 9, 2006, 13.00-15.00

Answer four questions! You may use either C++ and matutl02 or C and gmatutl in your work. Header files such as include <iostream> are not required.

1. We consider the system of equations

$$\begin{cases} 5x_1^2 + x_2^2 - 4x_1x_2 - 2x_1 + 1 = 0 \\ x_1^2 + x_2^2 - 2x_1 - 4x_2 + 5 = 0 \end{cases}$$

   (a) Form the Jacobian matrix for the Newton method.

   (b) Write the Newton iterative formula to solve the system with $x_0 = [0,1]$. Write a program to solve this.

2. Correct the programming mistakes in one of the programs example.c or example.cpp below.

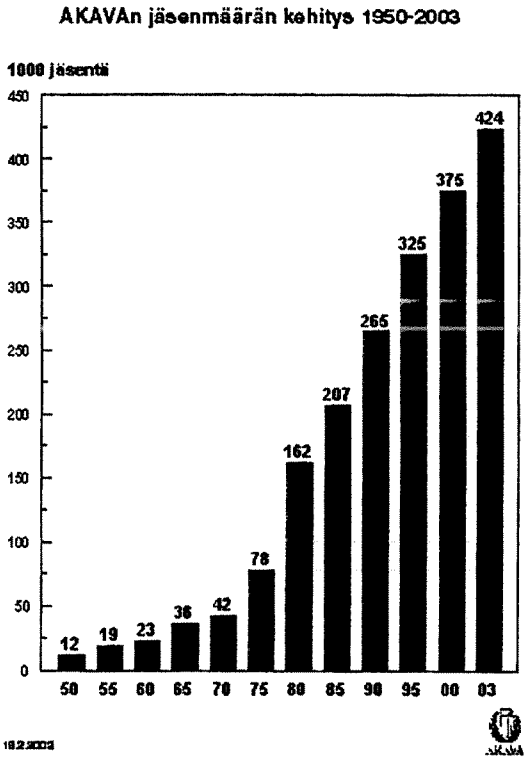
```
1   /* FILE: example.c /NRC03  begins                    */
2   /* gcc  example.c -o example -lgslcblas -lgsl -lm */
3
4   #define NP 10000
5   #include <studio.h>
6   #include <standlib.h>
7   #include <limes.h>
8   #include <time.h>
9   #include <mathem.h>
10  #include <gsl/gsl_sort_vector.h>
11  #include "../../util/gnatutl.c"
12  #include "../../Gnuplot02/Gnuplt1.c"
13
14  gsl_vector *a,*c;
15  gsl_matrix *b;
16  int m;
17
18  double fun(double x)
19  {
20    double v=0.0;
21    int j;
22    for (j=1;j<=(long)pow(2,m);j++)
23      if ((gsl_vector_get(a,j)<x)& (x <gsl_vector_get(b,j)))
24        v++;
25    return v*pow(2,-m);
26  }
```

```
27
28  double int_of_fun(int m)
29  {
30    double s=0.0;
31    int j;
32    for (j=1;j<=2*m-1;j++)
33      s+=(gsl_vector_set(c,j+1)-gsl_vector_get(c,j))*
34         fun((gsl_vector_get(c,j+1)+gsl_vector_get(c,j))/2 );
35    return s;
36  }
37  int main(void)
38  {
39    int i;
40    double tmp;
41    a=gsl_matrix_alloc(NP,5);
42    b=gsl_vector_alloc(NP);
43    c=gsl_vector_alloc(2*NP);
44    for (m=1;m<=10; m++)
45    {
46      for (i=1;i<=(int)pow(2,m);i++)
47      {
48        tmp=rdm(0.0,1.00);
49        gsl_vector_set(b,i,rdm(0.0,1.00));
50        if (tmp< b[i])
51          gsl_vector_set(a,i,tmp);
52        else {
53          gsl_vector_set(a,i,gsl_vector_get(b,i));
54          gsl_vector_set(b,i,tmp);
55        }
56      }
57      for (i=1;i<=(int)pow(2,m);i++)
58      {
59        gsl_vector_set(c,i,gsl_vector_get(a,i));
60        gsl_vector_set(c,(int)pow(2,m)+i,gsl_vector_get(b,i);
61      ]
62      gsl_sort_vector(c);
63      printf("%5d %10.5lf\n",(int)pow(2,m), int_of_fun((int)pow(2,m)));
64      gnuplt1(fun,"fun(x)", 0,NULL);
65      }
66    return 1.0;
67  }
68  /* FILE: example.c ends */
```

```
1  /* FILE: example.cpp /NRC03  begins                    */
2  /* gcc  example.cpp -o example -lgslcblas -lgsl -lm */
3
```

```
4   #define NP 10000
5   #include <cstudio>
6   #include <cstandlib>
7   #include <climes>
8   #include <ctime>
9   #include <cmathem>
10  using namespace std
11  #include "../../util/mr.h"
12  #include "../../util/natutl02.cpp"
13  #include "../../Gnuplot02/Gnuplt1.c"
14
15  Vec_DP a,c;
16  Mat_DP b;
17  int m;
18
19  double fun(double x)
20  {
21    double v=0.0;
22    for (int j=1;j<=(long)pow(2,m);j++)
23      if ((a[j]<x) & (x<b[j]))
24        v++;
25    return v*pow(2,-m);
26  }
27
28  double int_of_fun(int m)
29  {
30    double s=0.0;
31    for (int j=1;j<=2*m-1;j++)
32      s+=(c(j+1)-c[j])*fun((c[j+1]+c[j])/2);
33    return s;
34  }
35  int main(void)
36  {
37    double tmp;
38    a(NP,5);
39    b(NP);
40    c(2*NP);
41    for (m=1;m<=10; m++)
42    {
43      for (int i=1;i<=(int)pow(2,m);i++)
44      {
45        tmp=rdm(0.0,1.00);
46        b[i]=rdm(0.0,1.00));
47        if (tmp< b[j])
48          a[i]=tmp;
49        else {
```

```
50          a[i]=b[i];
51          b[i]=tmp;
52        }
53      }
54      for (int i=1;i<=(int)pow(2,m);i++)
55      {
56        c[i]=a[i];
57        c[(int)pow(2,m)+i=b[i];
58      ]
59      NR::sort(c);
60      printf("%5d %10.5lf\n",(int)pow(2,m), int_of_fun((int)pow(2,m)));
61      gnuplt1(fun,"fun(x)", 0,NULL);
62      }
63    return 1.0;
64  }
65  /* FILE: example.cpp ends */
```



3. Write a C/C++ program with the following properties.

   (a) It reads with a subprogram (`GetData(char *myfile, Vec_DP &x, Vec_DP &y)` for NR or `GetData(char *myfile, gsl_vector *x, gsl_vector *y)` for GSL) the (x, y) pairs as indicated in the above picture and returns in x the values of x scaled to the range $[0, 5.3]$ and the values of y scaled to the range $[0, 4.12]$.

   (b) It fits the following model to the data `Model(coef,x) = coef[0]*pow(abs(x),`

4. We consider a quadrature formula of the form

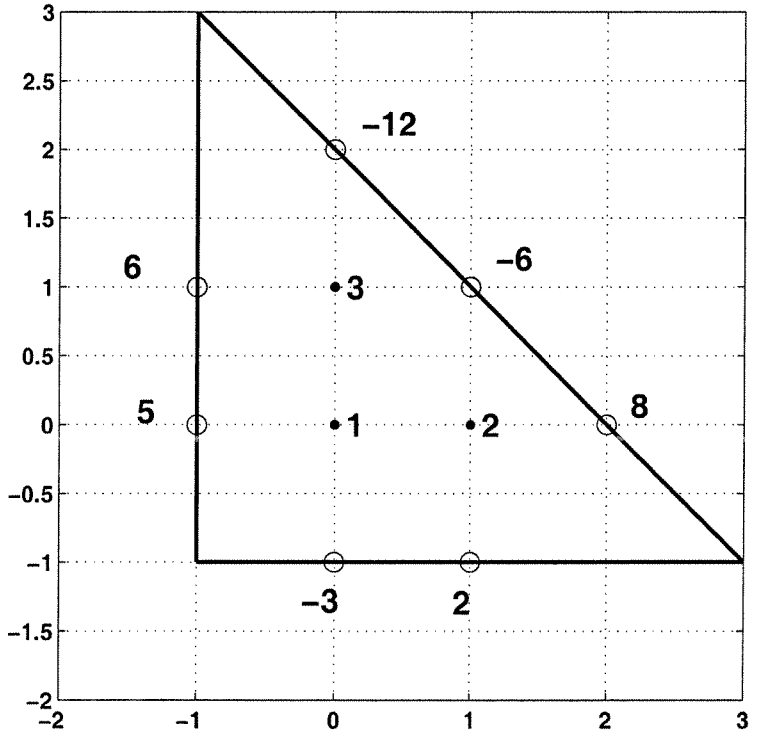
$$\int_0^3 f(x)dx = c_1 f(0) + c_2 f(1) + c_3 f(3)\,.$$

Determine the coefficients c_1, c_2, c_3 so that the formula holds as equality for $f(x) = x^j, j = 0, 1, 2$.

5. We consider the Dirichlet problem in the situation of the picture.

   (a) Form the difference equations for solving the problem numerically by indexing the variables as in the picture.

   (b) Explain how a numerical solution is obtained by the program `mylu2.c`, and write the contents of the input file for the program in the case of the picture.



coef[1]) by minimizing the sum

$$f(coef[0], coef[1]) = \sum_{i=1}^{m} (coef[0] * pow(x[i], coef[1]) - y[i])^2$$

with the help of some hypothetical minimizing function `NRC06Min(Vec_DP &c0, DP myf(Vec_DP , Vec_DP & ), int MAXIT)` (or `void NRC06Min(gsl_vector *xsta, double (*myff)(gsl_vector *),int ITMAX)` for GSL). Here myf map an object `x(m)` of class `Vec_DP` to an object `y(n)` of the same class and in our case $n = 1, m = 2$.

The main program could be e.g. for NR

```
int main()
{
  init_srand();
  char *fname="akava.dat";
  GetData(fname,xdata,ydata);
  Vec_DP coef(2);
  for (int i=0;i<coef.size();i++) coef[i]= 0.5+0.5*rdm(0.0, 1.0);
  NRC06Min(coef,myf,20);
  PlotData(xdata,ydata,coef);
}
```

or for GSL

```
int main()
{
  int i;
  gsl_vector *coef=gsl_vector_alloc(2);
  xdata=gsl_vector_alloc(NDATA); ydata=gsl_vector_alloc(NDATA);
  char *fname="akava.dat";
  init_srand();
  GetData(fname,xdata,ydata);
  for (i=0;i<coef->size;i++)
    gsl_vector_set(coef,i, 0.5+0.5*rdm(0.0, 1.0));
   printf("M: %f\n",myf(coef));
  NRC06Min(coef,myf,20);
  PlotData(xdata,ydata,coef);
  return 0;
}
```

In particular, write GetData and myf. Give an outline of what the program NRC06Min should do, so as to make this main program to have the desired properties. Explain what the program PlotData should do in order to make a picture of the data and the fitted function.