

Εργαστήριο 6

Χρησιμοποιώντας τις ρουτίνες που υλοποιήσατε στο Εργαστήριο 4 για την επίλυση ενός γραμμικού συστήματος με τη μέθοδο της απαλοιφής Gauss, να γράψετε ένα πρόγραμμα το οποίο θα υλοποιεί τη μέθοδο προσέγγισης δεδομένων με πολυώνυμο m -οστού βαθμού. Ο βαθμός να είναι ένα από τα δεδομένα εισόδου από το πληκτρολόγιο. Να εφαρμόσετε το πρόγραμμα χρησιμοποιώντας το αρχείο που κατασκευάσατε για τη συνάρτηση του ημιτόνου για την Άσκηση 1 του 4^{ου} Κεφαλαίου των σημειώσεων στο διάστημα $[2,4]$. Προσεγγίστε τη συνάρτηση $\sin(x)$ με ένα πολυώνυμο 6^{ου} βαθμού.

Λύση σε C

```
//Erg6-Eksetash
//Polywnymikh proseggish n-ostou vathmou
//Epilysh grammikou systhmatos me
//apaloifh Gauss (merikh odhghsh)
//xrhsimopieie dynamikh diaxeirish mnhmhs

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#define EPS 1.e-5
#define ZERO 1.e-10

void triang(double *a[ ], double b[ ], int N)
```

```

{
    int i,j,i1,j_d,imax, s;
    double dummy,mult, MAX;

    s=0;
    for (i=0;i<N;i++)
    {
        //do pivoting
        MAX = fabs(a[i][i]);
        imax = i;
        for (i1=i+1;i1<N;i1++)
        {
            if ( fabs(a[i1][i]) > MAX )
            {
                MAX = fabs(a[i1][i]);
                imax = i1;
            }
        }

        //do the change-over
        if(fabs( MAX-fabs(a[i][i]) ) > EPS )
        {
            for (j_d=0;j_d<N; j_d++)
            {
                dummy = a[imax][j_d];
                a[imax][j_d] = a[i][j_d];
                a[i][j_d] = dummy;
            }
        }
    }
}

```

```

    }
    dummy = b[imax];
    b[imax] = b[i];
    b[i] = dummy;
    s++; //metrhts gia thn orizousa
}

```

```

//trigwnopoihsh kathe grammhs

```

```

for(i1=i+1;i1<N;i1++)
{
    mult = a[i1][i]/a[i][i];
    for(j = i;j<N;j++)
    {
        a[i1][j] = a[i1][j] - mult* a[i][j];
        if (fabs(a[i1][j]) < ZERO) a[i1][j]=0.0;
    }
    b[i1] = b[i1]-mult*b[i];
    if (fabs(b[i1]) < ZERO) b[i1]=0.0;
}

```

```

} //end

```

```

return;

```

```

}

```

```
void back(double *a[ ], double b[ ], double x[ ], int N)
```

```
{
```

```
    int i,j;
```

```
    double sum;
```

```
    x[N-1] = b[N-1]/a[N-1][N-1];
```

```
    for (i=N-2; i>=0; i--)
```

```
    {
```

```
        sum=0.0;
```

```
        for(j=N-1;j>=i+1;j--)
```

```
            sum = sum+a[i][j]*x[j];
```

```
        x[i] = (b[i]-sum) / a[i][i];
```

```
    }
```

```
    return;
```

```
}
```

```
void Gauss(double *a[ ], double b[ ], double x[ ], int N)
```

```
{
```

```
    triang(a,b,N);
```

```
    back(a,b,x,N);

    return;
}

int main(int argc, char *argv[ ])
{

    int i, j, k, NMAX, vathmos;

    double *X, *Y, **A, *B, *x_a, *S, x_in, y_out;

    FILE *dat_in, *dat_out;

    if (argc !=3) {printf("%d\n",argc); exit (1);}

    if( (dat_in=fopen(argv[1], "r"))==NULL) exit(1);

    fscanf(dat_in, "%d", &NMAX);

    X = malloc(NMAX*sizeof(double));
    Y = malloc(NMAX*sizeof(double));

    for(i=0;i<NMAX;i++)
```

```

{
    fscanf(dat_in, "%lf%lf", &X[i], &Y[i]);
}

fclose(dat_in);

printf("Dwse vathmo polywnymou gia proseggish\n");
scanf("%d", &vathmos);

x_a = malloc((vathmos+1)*sizeof(double));
B = malloc((vathmos+1)*sizeof(double));
A = malloc((vathmos+1)*sizeof(double));
for(i=0;i<=vathmos;i++) A[i] = malloc((vathmos+1)*sizeof(double));
S = malloc(2*(vathmos+1)*sizeof(double));

//ftiakse ton pinaka toy systhmatos
for(k=0;k<=2*vathmos;k++)
{
    S[k] = 0.0;
    if(k<=vathmos) B[k]=0.0;
    for(i=0;i<NMAX;i++)
    {
        S[k] = S[k]+pow(X[i], k);
        if (k<=vathmos)
            B[k]= B[k] + pow(X[i],k)*Y[i];
    }
}
}

```

```

for (i=0;i<=vathmos;i++)
{
    for (j=0;j<=vathmos;j++)
    {
        A[i][j] = S[i+j];
        printf("%g ", A[i][j]);
    }
    printf("%g \n", B[i]);
}

```

```
Gauss(A,B,x_a,vathmos+1);
```

```

for(i=0;i<vathmos+1;i++)
    printf("a%d = %10.5f\n", i, x_a[i]);

```

```
if( (dat_out=fopen(argv[2], "w"))==NULL) exit(1);
```

```
x_in = 2.0;
```

```
while(x_in<=4.0)
```

```
{
```

```
    y_out = 0.0;
```

```
    for (i=0;i<=vathmos;i++)
```

```
        y_out = y_out + x_a[i]*pow(x_in,i);
```

```
        fprintf(dat_out, "%10.5lf %10.5lf %10.5lf\n", x_in, y_out,
```

```
sin(x_in) );
```

```
x_in = x_in + 2./100.;
```

```
}
```

```
fclose(dat_out);
```

```
free(X); free(Y); free(A); free(B); free(x_a); free(S);
```

```
return 0;
```

```
}
```