**Компютърна графика и визуализация**

**Упражнениe 10,11**

**Интерполация и апроксимация**

Нека е дадена съвкупност от реални абсциси

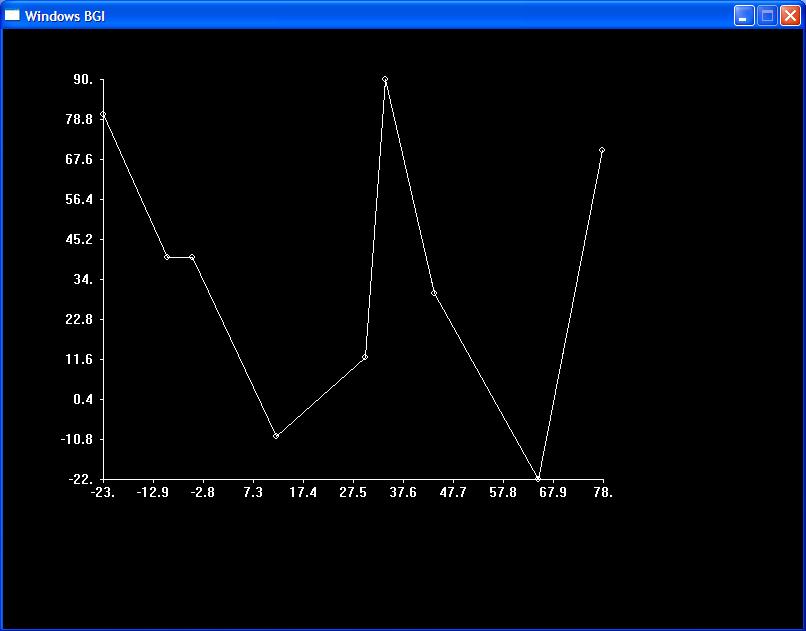
и ординати

и е някакво наблюдавано или математически определено число.

Задачата на едномерната интерполация е да се построи такава функция, че за всяко и да приема “разумни” стойности за всяко   
. Критерият за “разумност” може да се мени и може никога да не бъде разбран задоволително.

Ако се допусне да се различава от , може характерът на данните да се отрази по-правилно и да се коригират някои грешки в тях. Така поставената задача е задача на апроксимацията: за всяко и да приема “разумни” стойности за всяко .

Търсенето на крива, преминаваща през зададено множество точки е задача на интерполацията, а търсенето на крива, минаваща близо до зададено множество точки е задача на апроксимацията. За двете може да се използва общ термин - построяване на крива по точки. Най-известните методи използват кривите на полиноми, линейна комбинация на многочлени, сплайни и различни видове параметрични криви. Същината на задачата за построяване на крива по точки е определението за разумно поведение на функцията между точките, тъй като от безкрайния брой функции, които могат да построят кривата, трябва да се избере една с подходящ критерий. Например - със зададена степен на гладкост или простота на реализация. Задачата на графичното изобразяване на тези криви е да се намери за всяка една от тях най-добре представящото ги множество от пиксели.



Повечето интерполационни и апроксимационни функции се строят посредством линейна комбинация от елементарни функции. Когато елементарната функция е едночлен, интерполацията и апроксимацията е с полиноми. Когато елементарната функция е многочлен, то видът на интерполационните и апроксимационни алгоритми се определят от вида на многочлена.

Функцията, интерполираща данни, представени чрез множество точки с координати (xi, yi), има вида:

, където е едночлен, например: , ,.

Възможни са много други базисни едночлени.

Най-известния полином има вида:

са коефициенти на полинома.

Полиномът се намира след решение на система линейни уравнения, съставени на базата на известните базови точки. Като резултат се определят коефициентите на полинома. Недостатъците на този подход: графиката на полином от голяма степен се характеризират с голямо „вълнение” между базовите точки; извън границите на базовите точки полиномите имат силна тенденция към неограничено нарастване или намаляване; колкото по-висока е степента на полинома толкова повече нараства и грешката от решаването на системата линейни уравнения.

Пример: Дадени )),)

**Многочлени на Безие**

Този клас многочлени се прилагат в интерактивни графични системи за приблизително решаване на задачата за построяване на крива по точки. Вместо самите точки се използват определено множество точки, наречени ориентири или още реперни или управляващи. Многочленът се задава в параметрична форма.

, ,

Друг запис на уравнението

където е крива получена за ориентири от 0 до m.

е крива получена за ориентири от 1 до m.

е крива получена за ориентири от 0 до m-1.

Разполагането на точките ориентири е нещо, което изисква достатъчно висока квалификация. Това обстоятелство се явява сериозен недостатък на метода. Методът е с достатъчна простота за програмиране. Справедлив е не само за двумерни, но и за вектори с произволна размерност и това го прави удобен за описание на пространствени криви. Кривите на Безие приличат на намагнитена лента, закрепена в краищата си и привличана от магнити, поставени в точките ориентири. Кривата се приближава толкова по-близо до точките, колкото по-голяма е тяхната кратност.

**Геометричен алгоритъм за построяване на многочлен на Безие**

Първоначални точки ориентири - Pi

Стари точки ориентири - Ri

Нови точки ориентири - Qi

P3

P2

P4

t=0.5

P0

P1

P6

P5

int m = n;

for(double t = 0; t <= 1; t+=0.0001)

{

for(i = 0; i < m; i++)

{

Rx[i] = Px[i];

Ry[i] = Py[i];

}

n = m;

while(n > 0)

{

for(i=0; i < n-1; i++)

{

Qx[i] = Rx[i] + t\*(Rx[i+1] - Rx[i]);

Qy[i] = Ry[i] + t\*(Ry[i+1] - Ry[i]);

}

n--;

for(i = 0; i < n; i++)

{

Rx[i] = Qx[i];

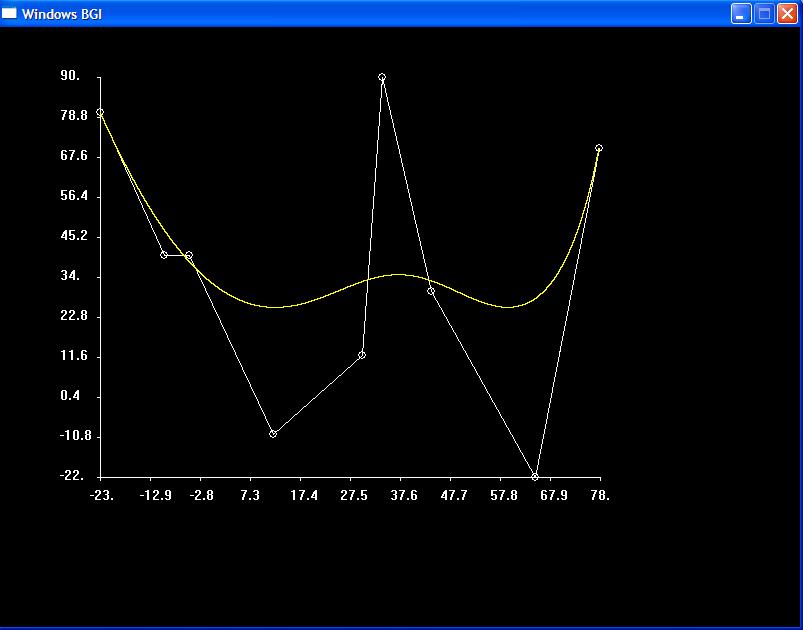
Ry[i] = Qy[i];

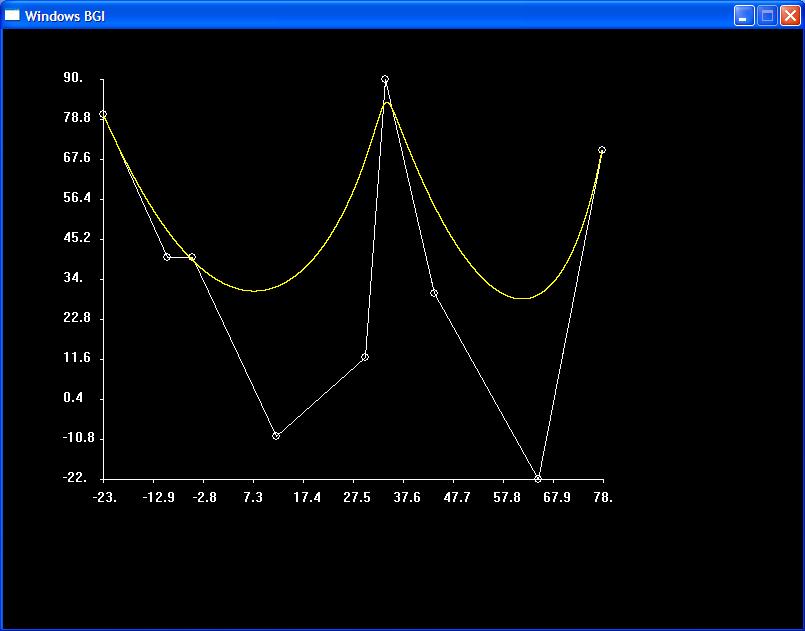
}

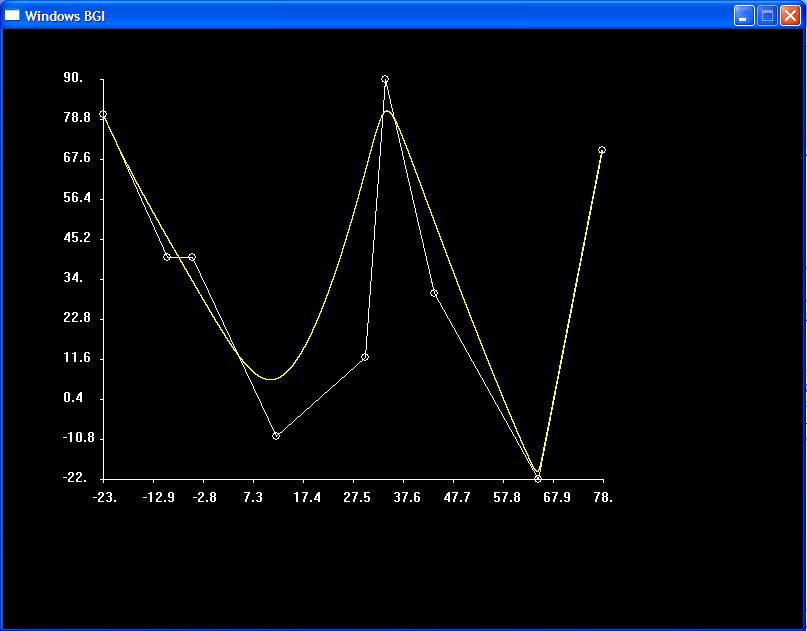
}

putpixel(Rx[0], Ry[0], 14);

}







**Многочлен на Лагранж**

Теорема на Лагранж: Парабола от ред , която преминава през точка с координати и пресича абсцисата в точките има уравнение:

X0,Y0

X2

X1

Xn

Ако се приложи това уравнение за парабола, която прeминава през точки , то уравнението на параболата ще е:

Или:

Пример: Дадени )),)

//многочлен на Лагранж

double yy,pr,xx;

for(double xx=Px[0];xx<=Px[n-1];xx+=0.0001)

{

yy=0;

for (i=0;i<n;i++)

{

pr=1;

for(j=0;j<n;j++)

{

if(i!=j)

pr=pr\*(xx-Px[j])/(Px[i]-Px[j]);

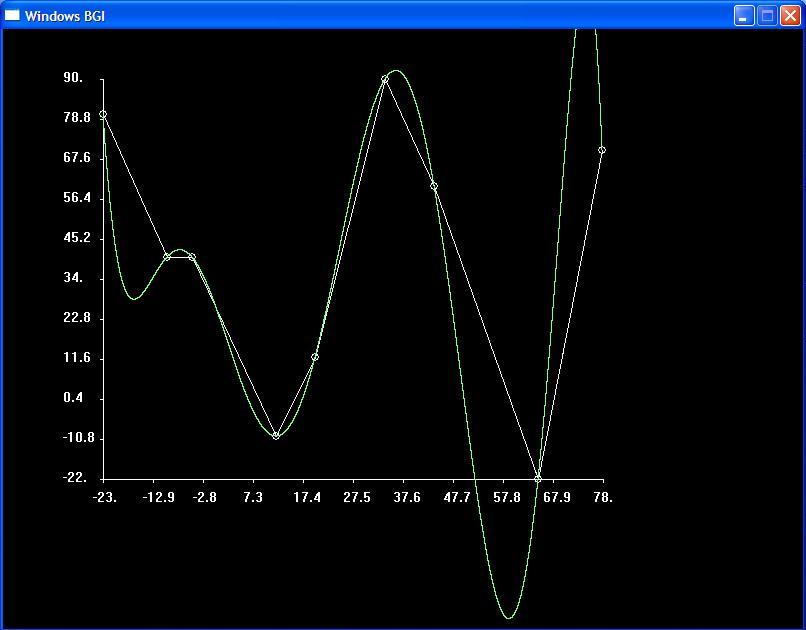
}

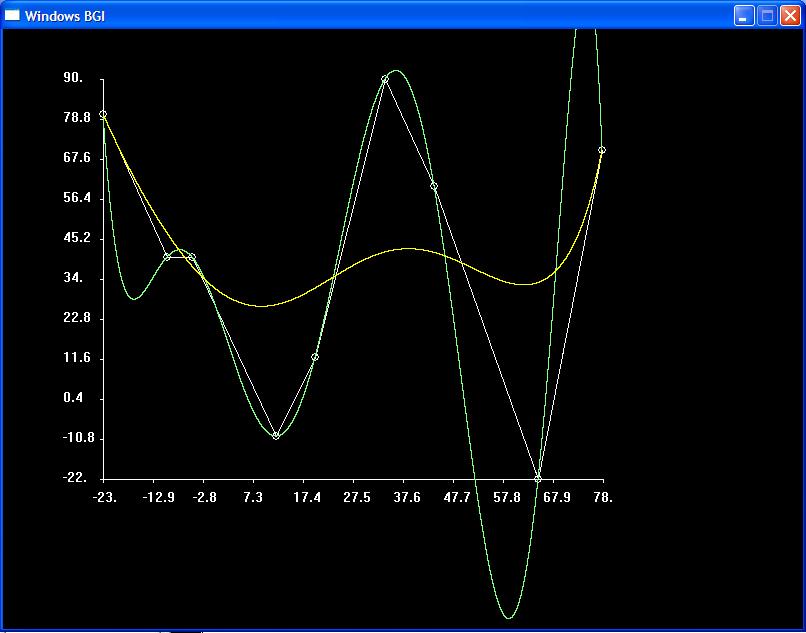
yy=yy+Py[i]\*pr;

}

putpixel(xx,yy,LIGHTGREEN);

}



****