8. Аппроксимация функций ортогональными многочленами

8.1 Введение в библиотеку orthopoly

Для решения многих инженерных задач требуется приблизить некоторые исходные сигналы специальными функциями. Во многих случаях в качестве таких функций используют ортогональные многочлены. В Maple включена библиотека orthopoly для работы с ортогональными многочленами.

Многочлены, содержащиеся в библиотеке orthopoly и некоторые их свойства приведены в таблице.

Любой полином можно использовать без применения команды with. В этом случае команда будет выглядеть следующим образом:

orthopoly[function](args), где function - имя полинома, args - аргументы.

Ортогональные полиномы библиотеки orthopoly

Функция	Полином	Вес w(x)	Условие ортогональности (m≠n)
G(n, a, x)	многочлены Гегенбауэра	$w(x) = (1-x^2)^{\left(a-\frac{1}{2}\right)}$	$\int_{-1}^{1} w(x)\,G(n,a,x)\,G(m,a,x)\,dx = 0$
H(n, x)	многочлены Эрмита	$w(x) = e^{(-x^2)}$	$\int_{-\infty}^{\infty} w(x)\,H(n,x)\,H(m,x)\,dx = 0$
L(n, a, x)	обобщенные многочлены Лагерра	$w(x) = e^{(-x)}\,x^a$	$\int_{0}^{\infty} w(x)\,L(n,a,x)\,L(m,a,x)\,dx = 0$
L(n, x)	многочлены Лагерра	также как и для обобщенных полиномов Лагерра при a=0	
P(n, a, b, x)	многочлены Якоби	$w(x) = (1-x)^a\,(1+x)^b$	$\int_{-1}^{1} w(x)\,P(n,a,b,x)\,P(m,a,b,x)\,dx = 0$
P(n, x)	многочлены Лежандра	также как и для полиномов Якоби при a=0, b=0	
T(n, x)	многочлены Чебышева первого рода	$w(x) = (1-x^2)^{\left(-\frac{1}{2}\right)}$	$\int_{-1}^{1} w(x)\,T(n,x)\,T(m,x)\,dx = 0$
U(n, x)	многочлены Чебышева второго рода	$w(x) = (1-x^2)^{\left(\frac{1}{2}\right)}$	$\int_{-1}^{1} w(x)\,U(n,x)\,U(m,x)\,dx = 0$

Такая нотация бывает необходима в случае конфликта вызова библиотечной функции и функции, определенной в сессии Maple под таким же именем.

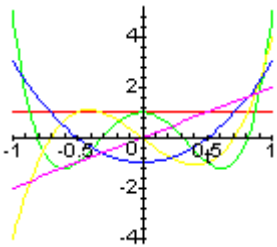
```
> with(orthopoly);
```

$$[G, H, L, P, T, U]$$

```
> G(3,5,x);
```

$$280\,x^3 - 60\,x$$

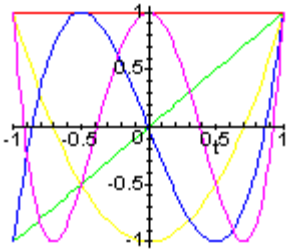
```
> plot({G(0,1,t), G(1,1,t), G(2,1,t), G(3,1,t),
G(4,1,t)},t=-1..1);
```




```
> T(5,y);
```

$$16\,y^5 - 20\,y^3 + 5\,y$$

```
> plot({T(0,t),T(1,t),T(2,t),T(3,t),T(4,t)},t=-
1..1);
```



8.2 Разложение функций по многочленам Чебышева

В задачах разложения функций по ортогональному базису необходимо определить коэффициенты разложения. Maple содержит, к сожалению, всего лишь одну функцию, предоставляющую такую возможность. Функция chebyshev представляет исходную функцию в форме ряда относительно многочленов Чебышева первого рода, причем количество членов ряда выбирается исходя из требуемой точности приближения.

Формат вызова:
chebyshev(expr, eq/nm, eps),
где expr - алгебраическое выражение или функция, eq/nm - уравнение вида x=a..b или некоторое имя x, eps - численное значение (необязательный параметр).

Функция chebyshev вычисляет разложение выражения expr через ортогональные многочлены Чебышева первого рода относительно переменной x на интервале [a,b] с точностью eps.

Если eq/nm задано просто как имя переменной x, то по умолчанию берется интервал разложения [-1,1].

Если присутствует третий аргумент eps, то он определяет точность разложения. По умолчанию используется значение $eps = 10^{(-Digits)}$, т.е. устанавливается ошибка приближения меньше, чем $10^{(-Digits)}$.

Выражение или процедура expr должна принимать числовое значение при подстановке вместо x конкретного числа. Кроме того, expr должна представлять функцию, которая является аналитической на промежутке [a,b].

Функция chebyshev в результате своей работы выдает ряд относительно многочленов Чебышева T(k,x), k=0,1,2,... с коэффициентами в форме чисел с плавающей запятой. Чтобы представить данный ряд в форме обычного многочлена достаточно загрузить из библиотеки orthopoly функцию T. Делается это командой with(orthopoly,T).

Следует быть внимательным, чтобы переменная T нигде раньше не была использована, т.к. данное имя зарезервировано под функцию, представляющую многочлены Чебышева первого рода.

Примеры:

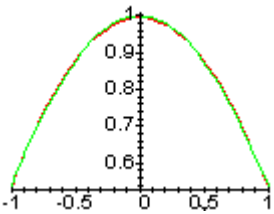
```
> r1:=chebyshev(cos(x),x);
```

$$r1 := .7651976865\,T(0, x) - .2298069699\,T(2, x) + .004953277929\,T(4, x) - .00004187667601\,T(6, x) + .1884468835\,10^{-6}\,T(8, x) - .5261230555\,10^{-9}\,T(10, x)$$

```
> r2:=chebyshev(cos(x),x,0.1);
```

$$r2 := .7651976865\,T(0, x) - .2298069699\,T(2, x)$$

```
> with(orthopoly,T):
> plot({cos(x),r2},x=-1..1);
```



Вычислим наибольшую абсолютную ошибку приближения.

```
> max1:=0: max2:=0:
> for x from -1 to 1 by 0.01 do
>    n1:=evalf(abs(r1-cos(x))):
>    n2:=evalf(abs(r2-cos(x))):
>   if max1<n1 then max1:=n1: fi:
>   if max2<n2 then max2:=n2: fi:
> od:
> max1;
```

$$.2\,10^{-9}$$

```
> max2;
```

$$.0049953436$$

Полученные результаты согласуются с заданной точностью разложения.

8.3 Разложение функций на произвольном интервале

Хотя Maple не содержит кроме единственной функции chebyshev других средств работы с ортогональными полиномами, но при необходимости можно разложить заданную функцию на произвольном интервале на ортонормированном базисе.

Надо подчеркнуть, что для разложения функции необходимо иметь ортонормированный базис. Например, содержащиеся в библиотеке orthopoly многочлены Лежандра не являются ортонормированными. Пояснить это можно следующим примером.

```
> with(orthopoly):
> a:=array(1..4,1..4):
> for i to 4 do
>   for j to 4 do
>      a[i,j]:=int(P(i,t)*P(j,t),t=-1..1):
>   od:
> od:
> print(a);
```

$$\begin{bmatrix} \frac{2}{3} & 0 & 0 & 0 \\ 0 & \frac{2}{5} & 0 & 0 \\ 0 & 0 & \frac{2}{7} & 0 \\ 0 & 0 & 0 & \frac{2}{9} \end{bmatrix}$$

Как видно, матрица "a" не является единичной. Данная матрица будет являться единичной только в случае ортонормированных функций.

Чтобы сделать многочлены ортонормированными, необходимо выполнить процедуру нормировки по аналогии с алгоритмом, использованном в следующем примере.

```
> normaF:=array(0..4):
> for i from 0 to 4 do
>    normaF[i]:=sqrt(int(P(i,t)^2,t=-1..1)):
> od:
> for i to 4 do
>   for j to 4 do  a[i,j]:=int(P(i,t)/
>      normaF[i]*P(j,t)/normaF[j],t=-1..1):
```

```
>    od:
> od:
> print(a);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Теперь зададим функцию для ортонормированных многочленов Лежандра и попробуем разложить sin(t) на промежутке [-1,1].

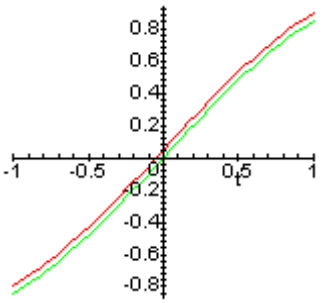
```
> pp1:=(i,t)->P(i,t)/normaF[i];
```

$$pp1 := (i, t) \rightarrow \frac{\mathrm{P}(i, t)}{normaF_i}$$

```
> fun1:=sin(t):
> c1:=array(0..4):
> for i from 0 to 4 do
>    c1[i]:=int(fun1*pp1(i,t),t=-1..1):
> od:
> s:=sort(evalf(sum(c1['i']*pp1('i',t),
'i'=0..4)));
```

$$s := -.1576151193\,t^3 + .9980751086\,t$$

```
> plot({fun1+0.05,s},t=-1..1);
```



В последнем примере вектор "c1" - вектор коэффициентов Фурье; s - разложение исходной функции по ортонормированному базису (уп-

рощенное выражение); при построении графика к исходной функции было добавлено 0.05 для того, чтобы на графике можно было различить две кривые, в противном случае первая кривая становится незаметной из-за наложения графика второй функции.

Как видно из построения, достигается удовлетворительная аппроксимация даже при пяти членах разложения.

Очень часто требуется разложить функцию на произвольном интервале [a,b]. В этом случае необходимо исходные ортонормированные функции переопределить на интервал [a,b]. Данная процедура проделывается в следующем примере.

```
> restart;
> with(orthopoly):
> n:=8:
> normaf:=array(0..n):
> a:=0: b:=6:
> pp:=(i,t)->simplify(subs(x=(2*t-a-b)/(a-
b),P(i,x)));
```

$$pp := (i, t) \rightarrow \mathrm{simplify}\left(\mathrm{subs}\left(x = \frac{2\,t - a - b}{a - b}, \mathrm{P}(i, x)\right)\right)$$

```
> for i from 0 to n do
> normaf[i]:=sqrt(int(pp(i,x)*pp(i,x),x=a..b)):
> od:
> ppn:=(ii,t)->pp(ii,t)/normaf[ii];
```

$$ppn := (ii, t) \rightarrow \frac{\mathrm{pp}(ii, t)}{normaf_{ii}}$$

```
> c:=array(0..n):
> fun:=sin(t):
> for i from 0 to n do
> c[i]:=evalf(int(fun*ppn(i,t),t=a..b)):
> od:
```

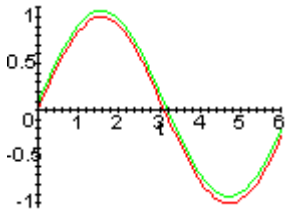
```
> print(c);
```

```
array( 0 .. 8, [
    ( 0 ) = .0162604124
    ( 1 ) = 1.451908566
    ( 2 ) = -.2308306940
    ( 3 ) = -.9755459264
    ( 4 ) = .058228154
    ( 5 ) = .131880629
    ( 6 ) = -.00495342
    ( 7 ) = -.00774684
    ( 8 ) = .0002136
    ])
```

```
> s:=sort(evalf(simplify(sum(c['i']*ppn('i',t),
'i'=0..n))));
```

$$s := .27549811251\,10^{-5}\,t^8 + .00008405043098\,t^7 - .002649997472\,t^6 + .01967875061\,t^5 - .02661563555\,t^4 - .1321078636\,t^3 - .02298865617\,t^2 + 1.006415598\,t - .0004346590000$$

```
> plot({fun+0.07,s},t=a..b);
```



Таким образом, разложить функцию на ортонормированном базисе на любом промежутке не представляет трудности.