Přímka procházející zadanými body

```
> A:=[0.5,1];
B:=[1,7.5];
>
:
> primka:=y=p*x+q;
> rceA:=subs(x=A[1],y=A[2],primka);
rceB:=subs(x=B[1],y=B[2],primka);
param:=solve({rceA,rceB},{p,q});
op(param);
> primka:=subs(op(param),primka);
> primka:=rhs(primka);
> p:=unapply(primka,x);
>
> p(A[1])=A[2];p(B[1])=B[2];
> p(.75);
> with(plots):
pointplot({[0,1],[1,-1],[3,0],[4,-3]},axes=BOXED,
color=RED,symbol=CROSS);
A;B;

> with(plots):
> plotA:=
pointplot({A,B},color=RED,symbol=CROSS):
plotB:=
plot(primka,x=A[1]-.5..B[1]+.5,color=NAVY):
display(plotA,plotB);

>
pointplot([A,B],color=RED,symbol=CROSS,symbolsize=100,connect=tru
e);
```

Funkce, která dvľma zadaněm bodôm pťiťadí pťímku, jež jimi prochází

```
> line:=proc(A,B)
local x;
if A[1]=B[1] then
     print(`primka je svisla, nejde o funkci`)
  else
unapply( (B[2]-A[2])/(B[1]-A[1])*x-(-A[2]*B[1]+A[1]*B[2])/(B[1]-
A[1]),x );
fi;
```

```
end;
> x:='x';
line([1,0],[2,1])(x);
> line([1,0],[2,1])(1.5);
> #X:=Body;
#Y:=Hodnoty;
Plin:=proc(X,Y)
local N,i,x;
N:=nops(X);
param:=x<X[1],Y[1];
for i from 1 to N-1 do
  param:=param, x<X[i+1],line([X[i],Y[i]],[X[i+1],Y[i+1]])(x);
od;
  param:=param, Y[N];
unapply(piecewise(param),x);
end;

xxx:=Plin(Body,Hodnoty)(x);


>
> plot(xxx,x=0..30);
> i:='i';
Body:= [i $i=1..27];
Hodnoty:=[evalf(sin(i*1.)) $i=1..27];
> Plin(Body,Hodnoty);
>
```

Lagrangeuv polynom

```
> n:=5;
for i from 1 to n do
X[i]:=i;
Y[i]:=evalf(ln(i));
od;
```

$$n := 5$$

$$X_1 := 1$$

$$Y_1 := 0.$$

$$X_2 := 2$$

$$Y_2 := 0.6931471806$$

$$X_3 := 3$$

$$Y_3 := 1.098612289$$

$$X_4 := 4$$

$$Y_4 := 1.386294361$$

$$X_5 := 5$$

$$Y_5 := 1.609437912$$

```
>
> i:='i';j:='j';

>
```

$$i := i$$

$$j := j$$

```
> for j from 1 to n do

> citatel[j]:=simplify(product(x-X[i],i=1..n)/(x-X[j]));

> jmenovatel[j]:=subs(x=X[j],citatel[j]);
> clen[j]:=citatel[j]/jmenovatel[j]*Y[j];
od;
```

$$citatel_1 := (x-2)(x-3)(x-4)(x-5)$$

$$jmenovatel_1 := 24$$

$$clen_1 := 0.$$

$$citatel_2 := (x-1)(x-3)(x-4)(x-5)$$

$$jmenovatel_2 := -6$$

$$clen_2 := -0.1155245301\,(x-1)(x-3)(x-4)(x-5)$$

$$citatel_3 := (x-1)(x-2)(x-4)(x-5)$$

$$jmenovatel_3 := 4$$

$$clen_3 := 0.2746530722\,(x-1)(x-2)(x-4)(x-5)$$

$$citatel_4 := (x-1)(x-2)(x-3)(x-5)$$

$$jmenovatel_4 := -6$$

$$clen_4 := -0.2310490602\,(x-1)(x-2)(x-3)(x-5)$$

$$citatel_5 := (x-1)(x-2)(x-3)(x-4)$$

$$jmenovatel_5 := 24$$

$$clen_5 := 0.06705991300\ (x-1)\,(x-2)\,(x-3)\,(x-4)$$

```
> Lagrange:=simplify(sum(clen[i],i=1..n));
```

$$Lagrange := -0.004860605100\ x^4 + 0.07692255710\ x^3 - 0.4838612513\ x^2 + 1.679182111\ x - 1.267382812$$

```
> with(plots);
Warning, the name changecoords has been redefined
```

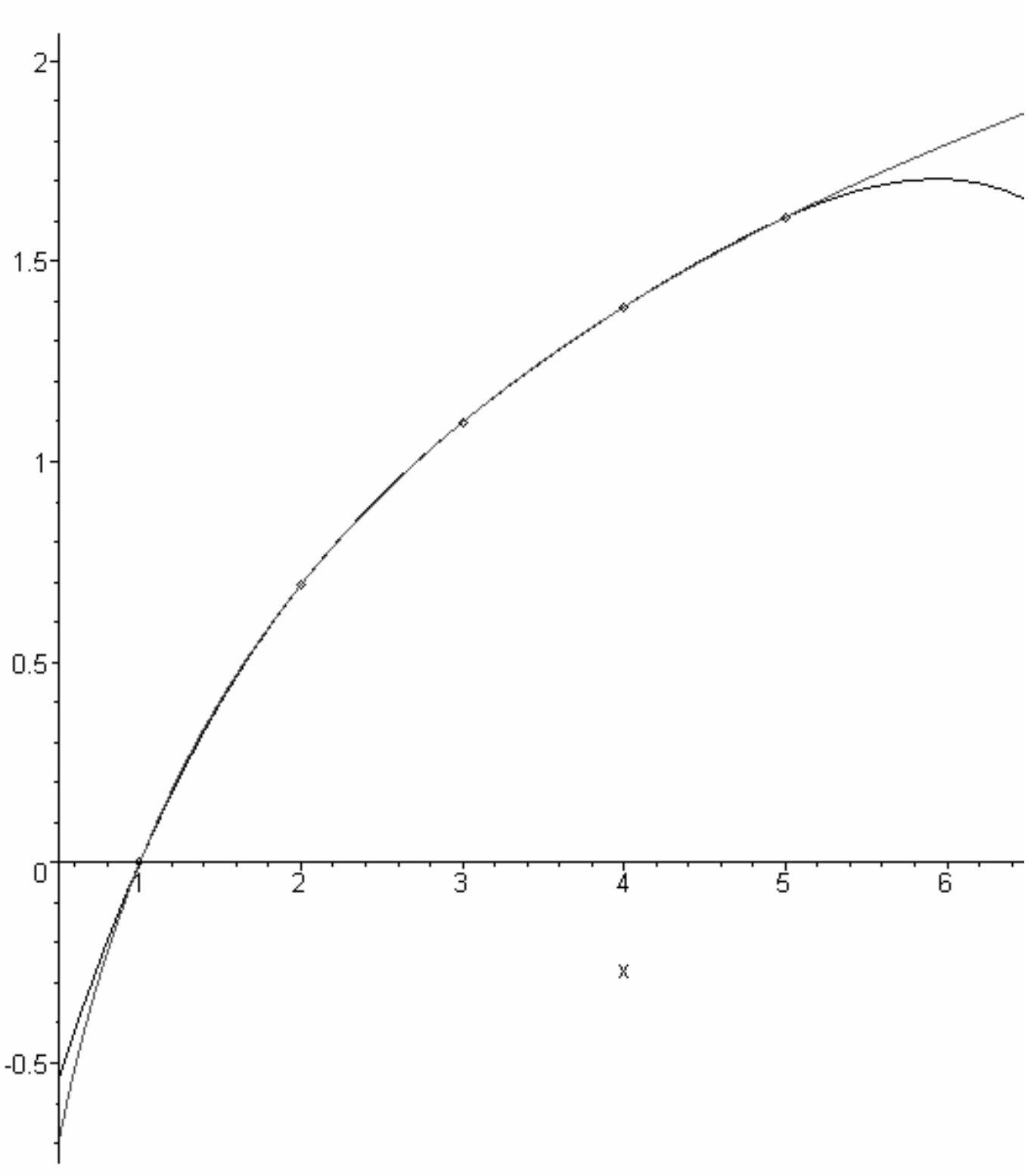
[*animate*, *animate3d*, *animatecurve*, *arrow*, *changecoords*, *complexplot*, *complexplot3d*, *conformal*, *conformal3d*, *contourplot*, *contourplot3d*, *coordplot*, *coordplot3d*, *cylinderplot*, *densityplot*, *display*, *display3d*, *fieldplot*, *fieldplot3d*, *gradplot*, *gradplot3d*, *graphplot3d*, *implicitplot*, *implicitplot3d*, *inequal*, *interactive*, *listcontplot*, *listcontplot3d*, *listdensityplot*, *listplot*, *listplot3d*, *loglogplot*, *logplot*, *matrixplot*, *odeplot*, *pareto*, *plotcompare*, *pointplot*, *pointplot3d*, *polarplot*, *polygonplot*, *polygonplot3d*, *polyhedra_supported*, *polyhedraplot*, *replot*, *rootlocus*, *semilogplot*, *setoptions*, *setoptions3d*, *spacecurve*, *sparsematrixplot*, *sphereplot*, *surfdata*, *textplot*, *textplot3d*, *tubeplot*]

```
> i:='i';
body:=[[X[i],Y[i]] $i=1..n];
Aplot:=pointplot(body):
Bplot:=plot(ln(x),x=.5..7.5):
Cplot:=plot(Lagrange,x=.5..7.5,color=BLACK):
```

$$i := i$$

$$body := [[1, 0.], [2, 0.6931471806], [3, 1.098612289], [4, 1.386294361], [5, 1.609437912]]$$

```
> display(Aplot,Bplot,Cplot);
```



Obecnl':

```
> Lagrange:=proc(X,Y)
local C,J,N,x,i,j,substitut,xxx;
```

```
i:='i';
Xpom:=convert(X,list);
N:=nops(Xpom);
for j from 1 to N do
C[j]:=product((x-X[i]),i=1..N)/(x-X[j]);
J[j]:=subs(x=X[j],C[j]);
#print(C[j],J[j]);
od;
i:='i';
xxx:=sum(C[i]/J[i]*Y[i],i=1..N);
unapply(xxx,x)
end;
```

```
Warning, `Xpom` is implicitly declared local to procedure `Lagrange`
```

$$Lagrange := \mathbf{proc}(X, Y)$$
$$\mathbf{local}\ C, J, N, x, i, j, substitut, xxx, Xpom;$$
$$i := 'i';$$
$$Xpom := \mathrm{convert}(X, list);$$
$$N := \mathrm{nops}(Xpom);$$
$$\mathbf{for}\ j\ \mathbf{to}\ N\ \mathbf{do}$$
$$C[j] := \mathrm{product}(x - X[i], i = 1 .. N)/(x - X[j]);$$
$$J[j] := \mathrm{subs}(x = X[j], C[j])$$
$$\mathbf{end\ do};$$
$$i := 'i';$$
$$xxx := \mathrm{sum}(C[i] \times Y[i]/J[i], i = 1 .. N);$$
$$\mathrm{unapply}(xxx, x)$$
$$\mathbf{end\ proc}$$

```
> xxx:=Lagrange(X,Y);
Lagrange(X,Y)(3.1)=ln(3.1);
```

$$xxx := x \rightarrow -0.1155245301\ (x-1)(x-3)(x-4)(x-5)$$
$$+ 0.2746530722\ (x-1)(x-2)(x-4)(x-5)$$
$$- 0.2310490602\ (x-1)(x-2)(x-3)(x-5)$$
$$+ 0.06705991300\ (x-1)(x-2)(x-3)(x-4)$$

$$1.130887918 = 1.131402111$$

```
> whattype(X);print(X);
```

$$symbol$$

$$\mathrm{table}([1 = 1, 2 = 2, 3 = 3, 4 = 4, 5 = 5])$$

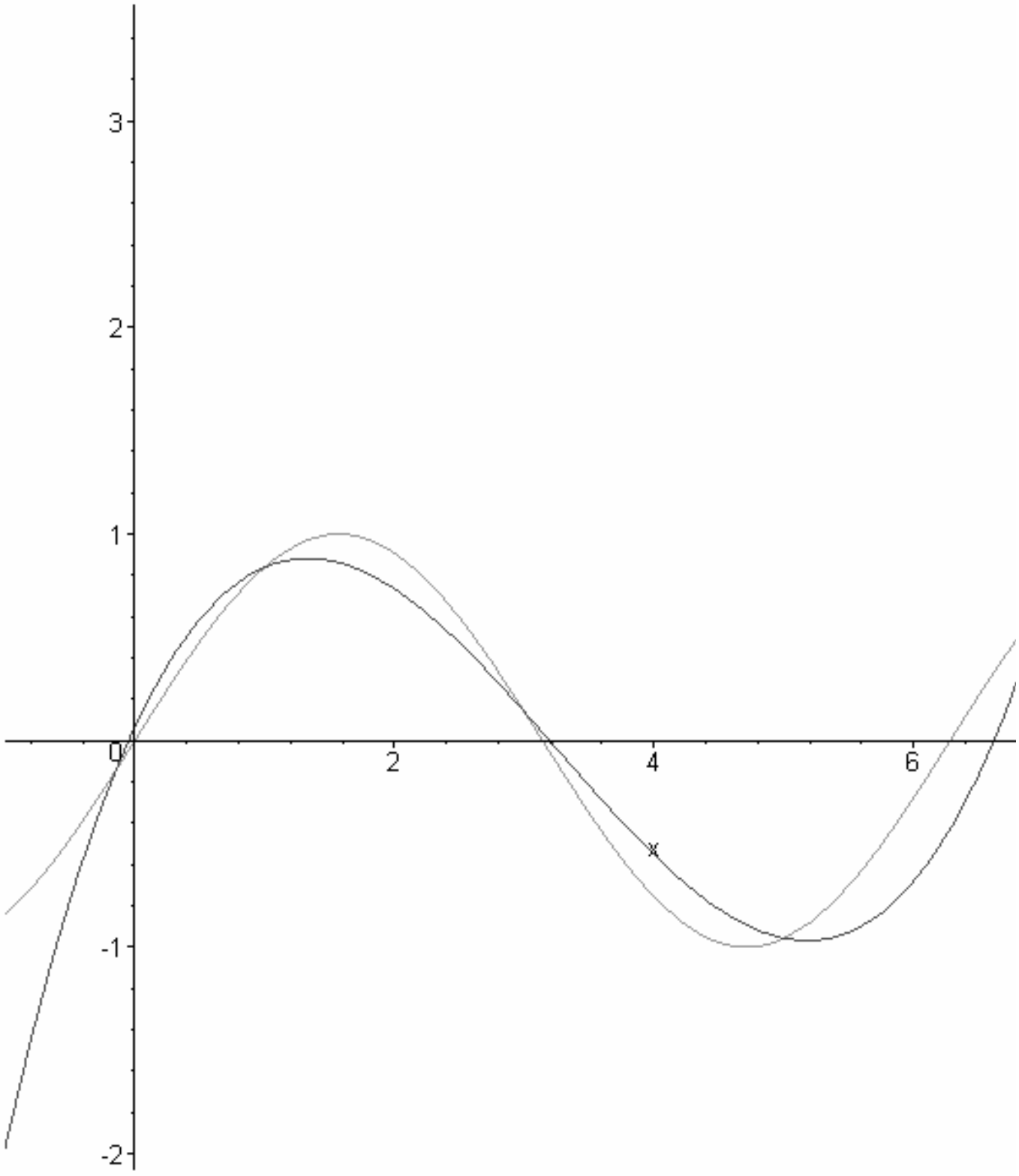
```
> yyy:=interp(convert(X,list),convert(Y,list), x);
```

$$yyy := -0.004860604917\ x^4 + 0.07692255500\ x^3 - 1.267382806 - 0.4838612431\ x^2$$
$$+ 1.679182099\ x$$

```
> simplify(xxx(x)-yyy);
```

$$0.1230000000\ 10^{-7}\ x - 0.1830000000\ 10^{-9}\ x^4 + 0.2100000000\ 10^{-8}\ x^3 - 0.6000000000\ 10^{-8} - 0.8200000000\ 10^{-8}\ x^2$$

```
>
plot({sin(x),Lagrange([1,3,5,7],[sin(1),sin(3),sin(5),sin(7)])(x)
},x=-1..8);
```




```
>
> A[1]:=7;A[2]:=2;
```

$$A_1 := 7$$

$$A_2 := 2$$

```
> print(A);
```

table([1 = 7, 2 = 2])

```
> nops(A);
```

1

```
> A:=convert(A,list);
```

$$A := [7, 2]$$

```
> A[1];
```

7

```
> whattype(A);
```

list

```
> A:='A';
```

$$A := A$$

```
> xxx:=x^2-1;
```

$$xxx := x^2 - 1$$

```
> subs(x=2,xxx);
```

3

```
> f:=unapply(xxx,x);
```

$$f := x \rightarrow x^2 - 1$$

```
> f(2);
```

3

```
> g:=t->t-1;
```

$$g := t \rightarrow t - 1$$

```
> g(2);
```

1

```
> g(x);
```

$$x - 1$$

Fourierova t'ada

```
> Int(sin(x),x=0..Pi/4)=int(sin(x),x=0..Pi/4);
> fourier:=proc(f,n)
local i,N;
if n mod 2 = 1 then N:=(n+1)/2
  else
    N:=n/2;
```

```
fi;
for i from 0 to N do
  a[i]:=evalf(1/Pi*int(f(x)*cos(i*x),x=-Pi..Pi));
  b[i]:=evalf(1/Pi*int(f(x)*sin(i*x),x=-Pi..Pi));
od;
i:='i';
if n mod 2 = 1 then
  unapply(a[0]/2+sum(a[i]*cos(i*x)+b[i]*sin(i*x),i=1..N-1),x)
else
  unapply(a[0]/2+sum(a[i]*cos(i*x)+b[i]*sin(i*x),i=1..N-
1)+a[N]*cos(N*x),x);
fi
end;
>
> f:=x->x^2-2^x;


for i from 1 to 2 do
print(i,fourier(f,i));
od;
> plot({f(x),fourier(f,15)(x)},x=-Pi/2..Pi/2);
> f:=x->abs(x);
K:=21;
ll:=t<1,(t)*fourier(f,1);
for i from 1 to K do
ll:=ll, t<i+1 ,(1-(t-i))*fourier(f,i)+(t-i)*fourier(f,i+1);
print(i);
od:
F:=piecewise(ll):
G:=unapply(F,t):
#subs();

> animate({G(t)(x),f(x)},x=-Pi..Pi,t=1..K,frames=K+1);
> plot({G(12.5)(x),f(x)},x=-Pi..Pi);
> Fourier:=proc(f,n,beta)
local i,N;
if n mod 2 = 1 then N:=(n+1)/2
  else
    N:=n/2;
fi;
for i from 0 to N do

a[i]:=evalf(2/beta*evalf(int(f(x)*cos(2*i*Pi*x/beta),x=0..beta)))
;

b[i]:=evalf(2/beta*evalf(int(f(x)*sin(2*i*Pi*x/beta),x=0..beta)))
;
```

```
od;
i:='i';
if n mod 2 = 1 then

unapply(a[0]/2+sum(a[i]*cos(2*i*Pi*x)+b[i]*sin(2*i*Pi*xx),i=1..N-
1),x)
else

unapply(a[0]/2+sum(a[i]*cos(2*i*Pi*x)+b[i]*sin(2*i*Pi*x),i=1..N-
1)+a[N]*cos(N*2*Pi*x),x);
fi
end;

Aproximace:=proc(X,Y,F) #vstup: pole bodu, pokle hodnot, pole
funkci. Vystup: linearni kombinace zadanych funkci, ktera nejlepe
aproximuje zadane hodnoty.
global MaticeSoustavy,PraveStrany;
local i,j,PX,PF,v
;
PF:=nops(F);PX:=nops(X);
i:='i';j:='j';
v:=evalf
([ unapply(F[i],x)(X[j])  $j=1..PX] $i=1..PF);

#print(v);
MaticeSoustavy:=
matrix(PF,PF,
[[sum((v[i][k]*v[j][k]),k=1..PX) $j=1..PF] $i=1..PF]
);
PraveStrany:=vector([sum(Y[j]*v[i][j],j=1..PX) $i=1..PF]);

Koeficienty:=linsolve(MaticeSoustavy,PraveStrany):
unapply(sum(Koeficienty[i]*F[i],i=1..PF),x);
end;
> Fourier(exp(x),4,1);
> AZ:=Fourier(xxx,2,10);
plot(AZ(x),x=0..30);
> plot({Fourier(exp(x),6,1)(x),exp(x)},x=-0.1..1.1);
```

Legenderovy polynomy tvoťí úplně ortonormální systém na intervalu <-1,1>.

```
> Legender:=n-
>sqrt(n+1/2)*`if`(n=0,1,simplify(diff(1/(2^n*n!)*(x^2-
1)^n,x$n)));
> for i from 0 to 5 do
print(i,Legender(i));
od;
>
```

Metoda nejmensich ctvercu

```
> X:=[1,2,3];Y:=[2,4,6];
RegresniPrimka:=proc(X,Y)
local Primka,x,p,q;
Primka:=x->p*x+q;
SoucetCtvercu:=sum(((Y[i]-Primka(X[i]))^2),i=1..nops(X));
Rce:=diff(SoucetCtvercu,p)=0,diff(SoucetCtvercu,q)=0;
Param:=solve({Rce},{p,q});
unapply(subs(Param,Primka(x)),x);
end;
>
> RegresniPrimka([i $i=1..17],[2*i+sin(i) $i=1..17]);
> with (plots):
Obr:=proc(X,Y)
A:=pointplot({[X[i],Y[i]] $i=1..nops(X)});
B:=plot(RegresniPrimka(X,Y)(x),x=0..X[nops(X)]);
display(A,B)
end;
> Obr([i $i=1..17],[2*i+sin(i) $i=1..17]);
> with(stats):
fit[leastsquare[[x,y]]]([[1,2,3,5],[2,4,6,8]]);
>
> RegresniPrimka([1,2,3,5],[2,4,6,8]);
```

Aproximace namerenych hodnot linearni kombinbaci zvolenych funkci:

```
>
>

> with(linalg):
> Aproximace:=proc(X,Y,F)
global MaticeSoustavy,PraveStrany;
local i,j,PX,PF,v
;
PF:=nops(F);PX:=nops(X);
i:='i';j:='j';
v:=evalf
([ unapply(F[i],x)(X[j])  $j=1..PX] $i=1..PF);

#print(v);
MaticeSoustavy:=
matrix(PF,PF,
[[sum((v[i][k]*v[j][k]),k=1..PX) $j=1..PF] $i=1..PF]
```

```
);
PraveStrany:=vector([sum(Y[j]*v[i][j],j=1..PX) $i=1..PF]);

Koeficienty:=linsolve(MaticeSoustavy,PraveStrany):
unapply(sum(Koeficienty[i]*F[i],i=1..PF),x);
end;
>
> i:='i';
X:=[i $i=1..5];
Y:=[sin(i)-cos(i)+(sin(cos(i))) $i=1..5];
(Aproximace(X,Y,[sin(x),cos(x)]));
>
> print(MaticeSoustavy);
print(PraveStrany);
> linsolve(MaticeSoustavy,PraveStrany);
>
>
Warning, `fce` is implicitly declared local to procedure `PrumetFunkce`
```