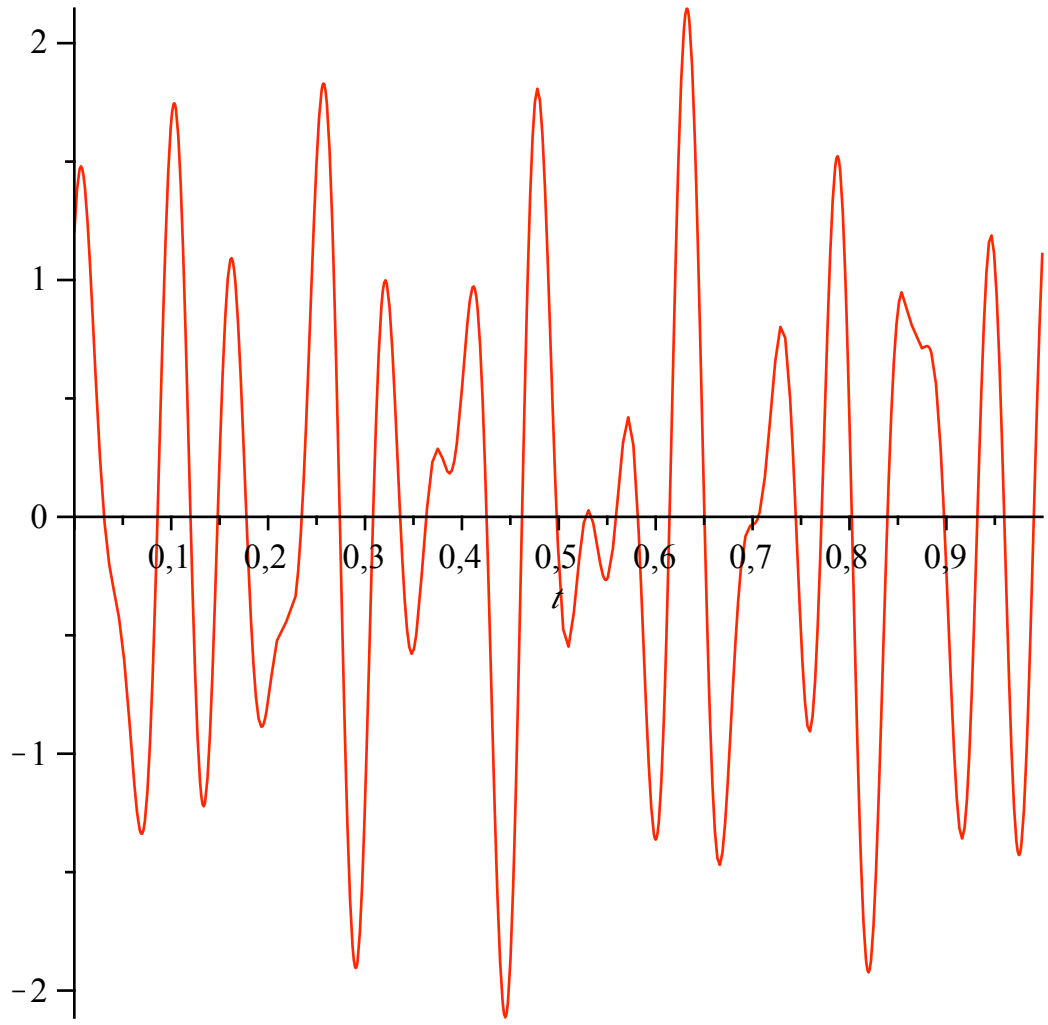
Exemple 1 (somme de sinusoïdes)

```
> restart;
> X:=sin(13*2*Pi*t)+0.5*cos(8*2*Pi*t)+0.7*cos(19*2*Pi*t);
```

$$X := \sin(26\,\pi\,t) + 0.5\cos(16\,\pi\,t) + 0.7\cos(38\,\pi\,t) \qquad \textbf{(1.1)}$$

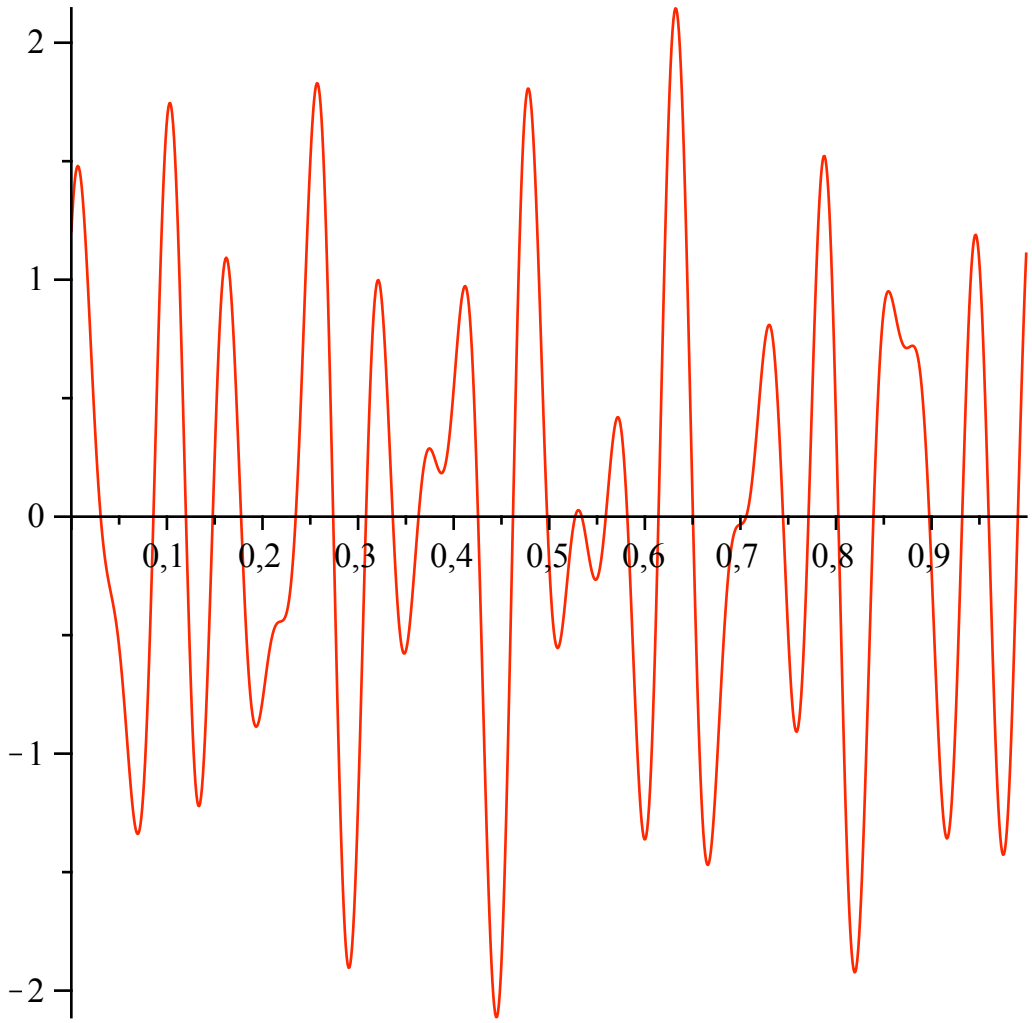
```
> plot(X,t=0..1023/1024);
```




```
>
> t:=[evalf(($ 0..1023)/1024)]:
> X:=[seq([t[i],evalf(sin(13*2*Pi*t[i])+0.5*cos(8*2*Pi*t[i])
  +.7*cos(19*2*Pi*t[i]))],i=1..1024)]:
> X[1];
```

$$[0.,\ 1.2] \qquad \textbf{(1.2)}$$

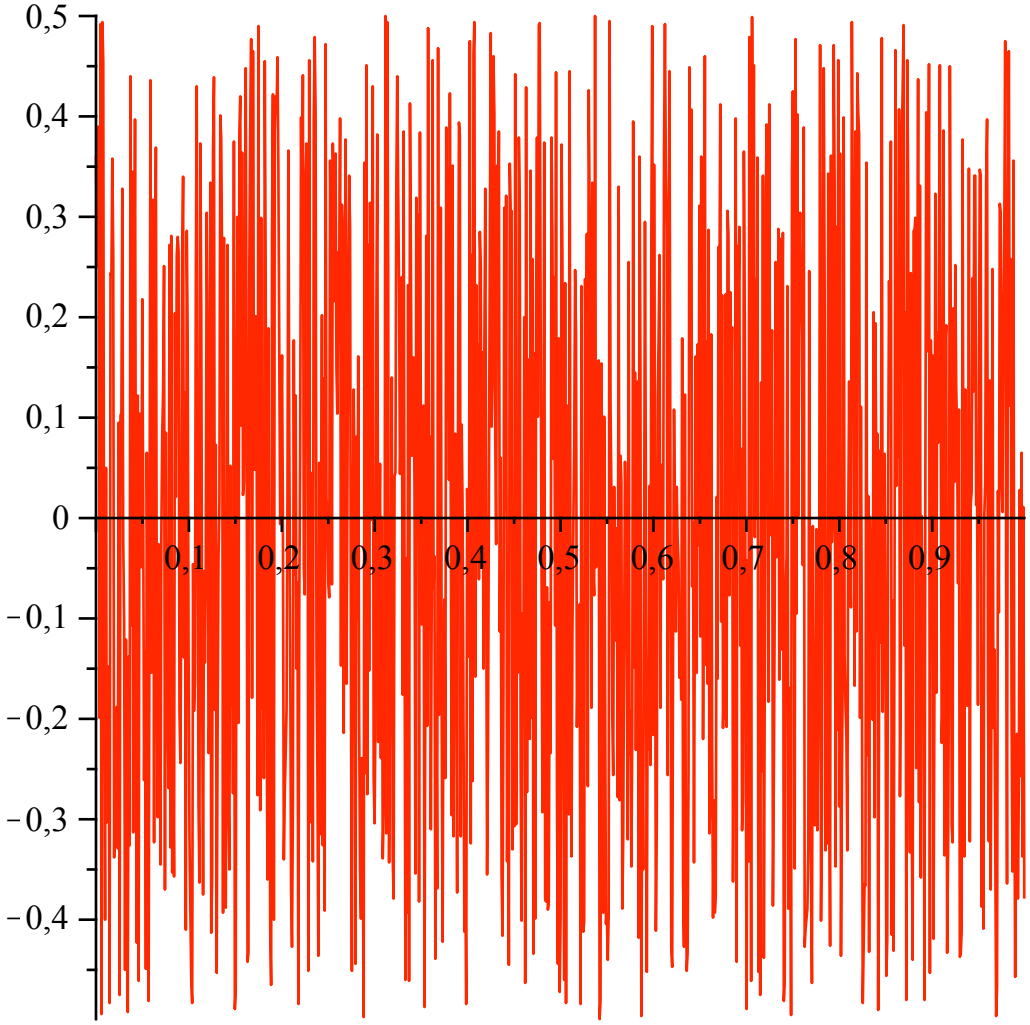
```
> plot(X);
```



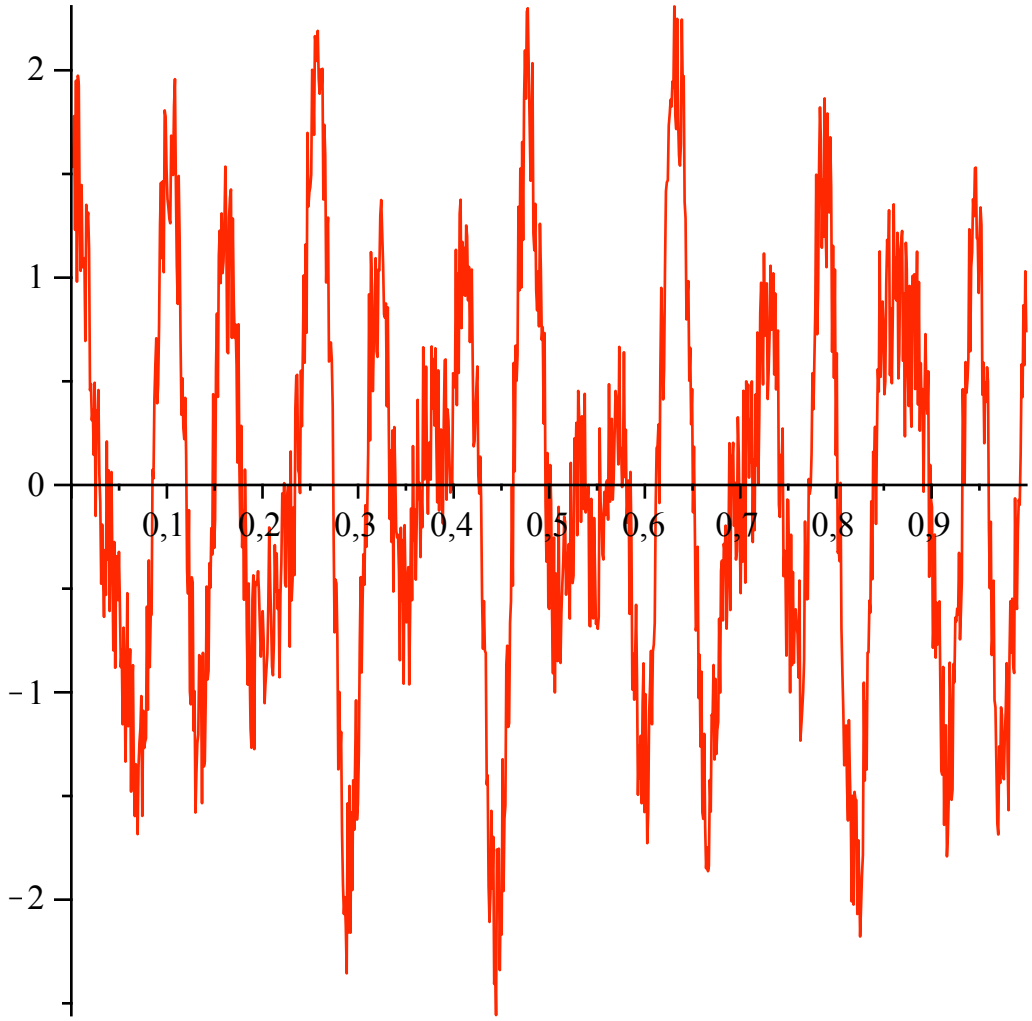

```
> h:=evalf(rand(-499..500)/1000);
```

$h := 0.001000000000$ **proc**( ) **proc**( ) **option** *builtin* = *RandNumberInterface*; **end proc**(6, 1000, 10) − 499 **end proc** **(1.3)**

```
> Y:=[seq([t[i],h()],i=1..1024)]:
> Z:=[seq([t[i],X[i][2]+Y[i][2]],i=1..1024)]:
> plot(Y);
```




```
> plot(Z);
```




```
> ZZ:=[seq(Z[i][2],i=1..1024)]:
> with(DiscreteTransforms);
```

[*FourierTransform*, *InverseFourierTransform*] **(1.4)**

```
> ZZ:=convert(ZZ,Vector):
> for i from 1 to 10 do print(ZZ[i]) od;
```

1.561000000
1.533328530
1.588508473
1.779233318
1.230331391
1.949764686
0.9806259650
1.974134224
1.929628584
1.258560707 **(1.5)**

```
> F:=FourierTransform(ZZ);
```

$$F := \begin{bmatrix} 1 .. 1024\ Vector_{column} \\ Data\ Type: anything \\ Storage: rectangular \\ Order: Fortran_order \end{bmatrix} \tag{1.6}$$

```
> for i from 1 to 20 do print(F[i]) od;
```

```
                    -0.279249995909375158 + 0. I
          -0.00627542862629427122 - 0.262298879438103482 I
           -0.342342517460109352 + 0.266139789292381190 I
          -0.0310861118121861652 + 0.0876641229229681851 I
           -0.0925413357818133526 + 0.467914373478336365 I
          0.00588703977980710448 - 0.147148057705133172 I
          -0.0455327905451952953 - 0.0267176507323262091 I
           0.124955349939043739 - 0.0603759911925317583 I
           7.99102398856521212 + 0.00868661624969278845 I
          -0.0957392435713840251 + 0.00245956362614596636 I
          0.0286444893855760281 - 0.0159867798434999846 I
          0.0455288062304227970 + 0.560081529885917861 I
          0.195941661983977777 - 0.0398927850609489126 I
           -0.157624519654755629 - 16.1790776245031793 I
          -0.222962959913888870 - 0.0580942632378217971 I
          0.301558060769613556 - 0.117082487496512427 I
          0.112359863998866174 + 0.174206205416435583 I
          0.0463086433380387186 - 0.0641531350063713757 I
           -0.182460143802803093 + 0.190710685257308637 I
            11.3700653706289039 - 0.253898411568083071 I
```

(1.7)

```
> abs(F[9]);
```

$$7.991028710 \tag{1.8}$$

```
> abs(F[20]);
```

$$11.37289985 \tag{1.9}$$

```
> abs(F[14]);
```

$$16.17984543 \tag{1.10}$$

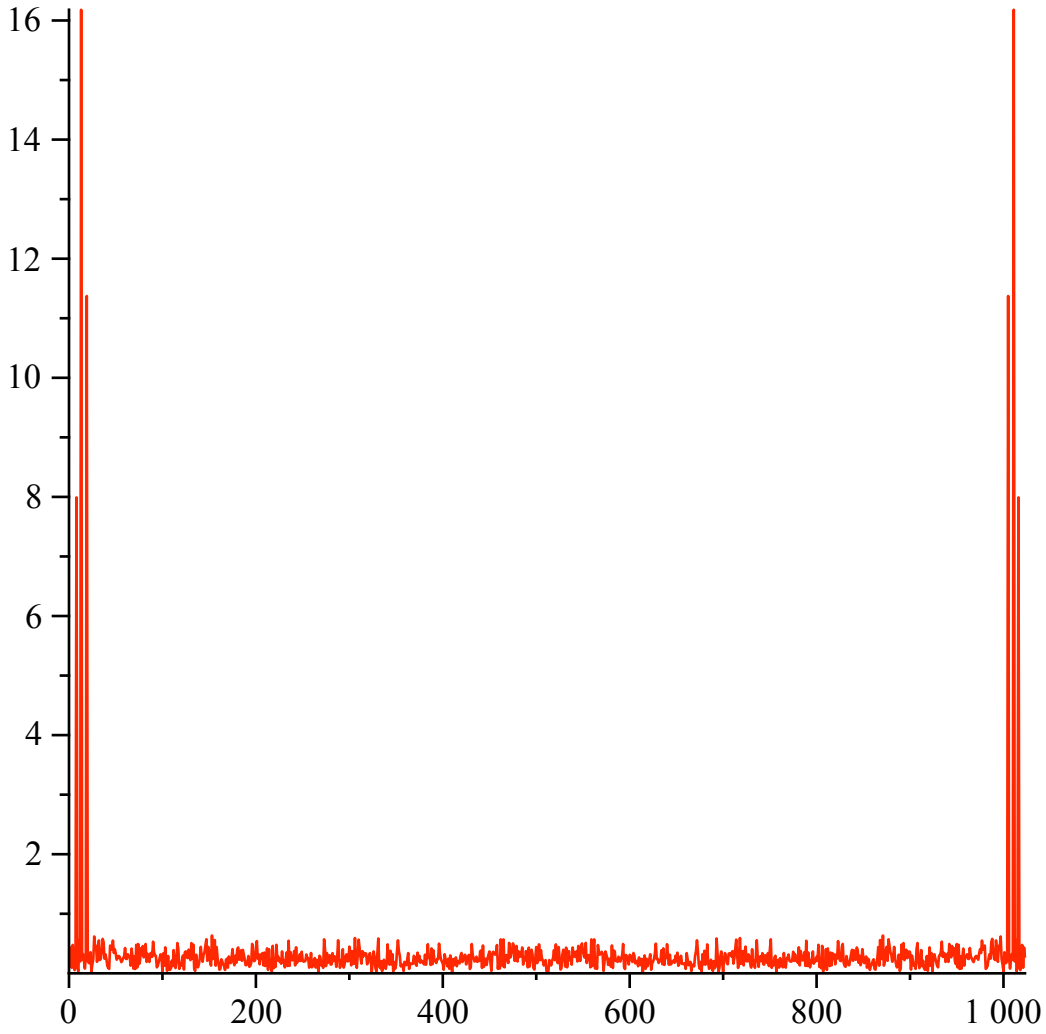
```
>
> abs(F[1017]);
```

$$7.991028710 \tag{1.11}$$

```
> abs(F[1012]);
```

$$16.17984543 \tag{1.12}$$

```
> FF:=[seq([i-1,abs(F[i])],i=1..1024)]:
> plot(FF);
```



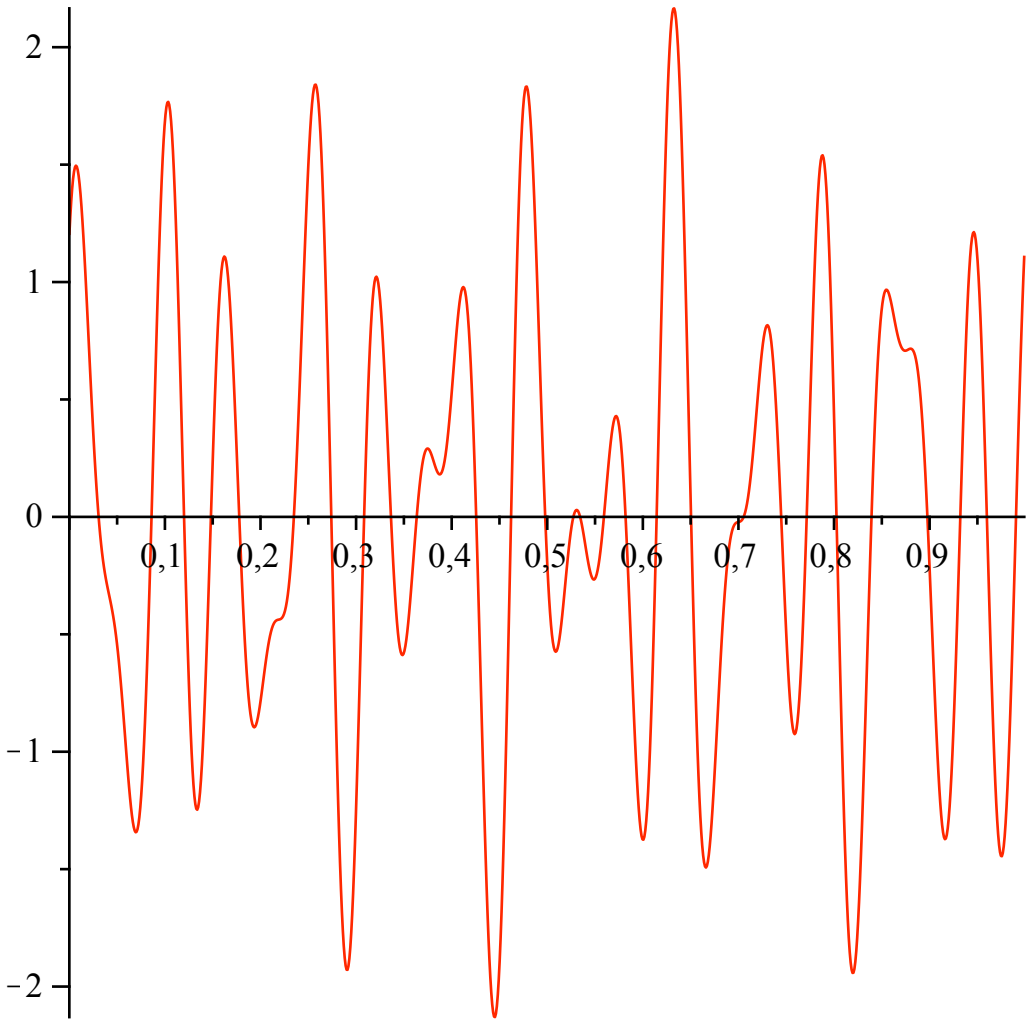

```
> FS:=Vector(1024);
```

$$FS := \begin{bmatrix} 1 .. 1024\ Vector_{column} \\ Data\ Type: anything \\ Storage: rectangular \\ Order: Fortran_order \end{bmatrix} \quad (1.13)$$

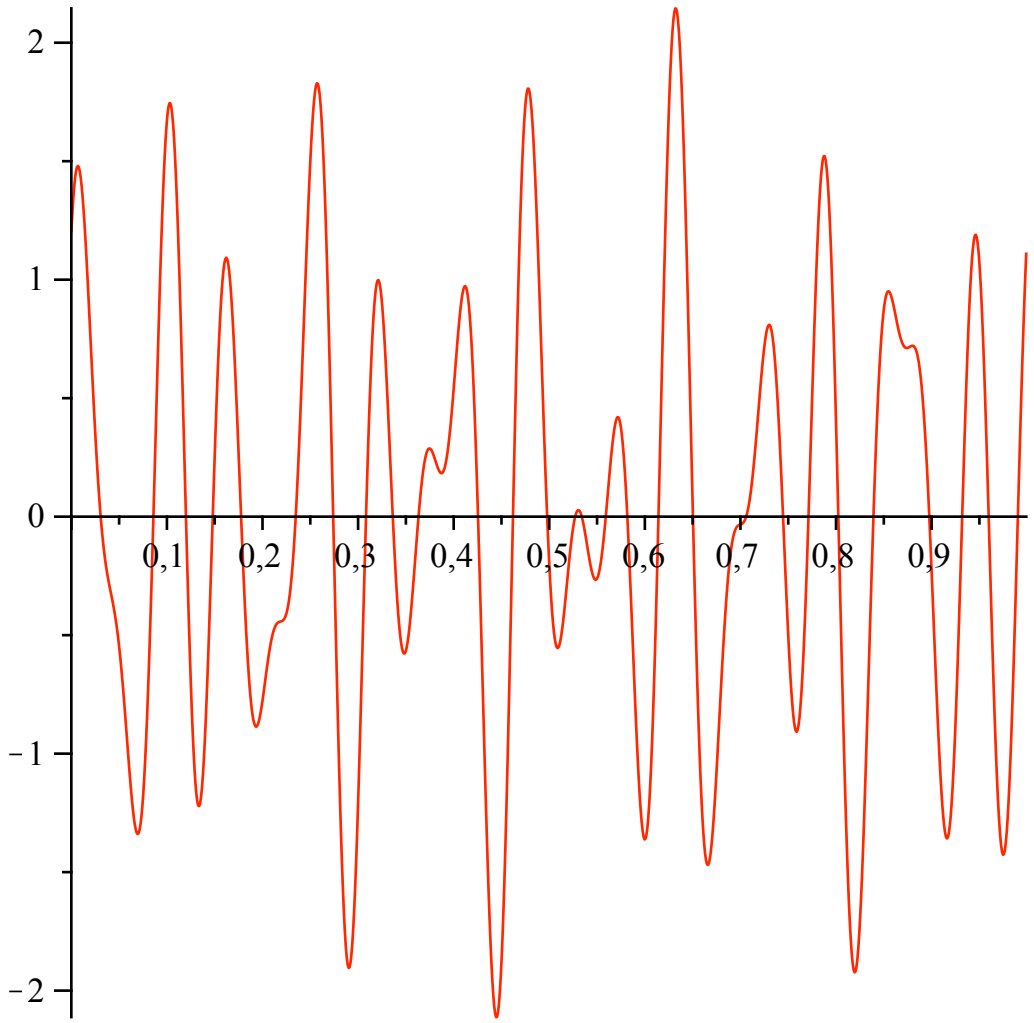
```
>
> for i from 1 to 1024 do if abs(F[i])>1 then FS[i]:=F[i] else
  FS[i]=0 fi od;
> FS[1];
```

$$0 \quad (1.14)$$

```
> S:=InverseFourierTransform(FS):
> SS:=[seq([t[i],Re(S[i])],i=1..1024)]:
> plot(SS);
```




```
> plot(X);
```



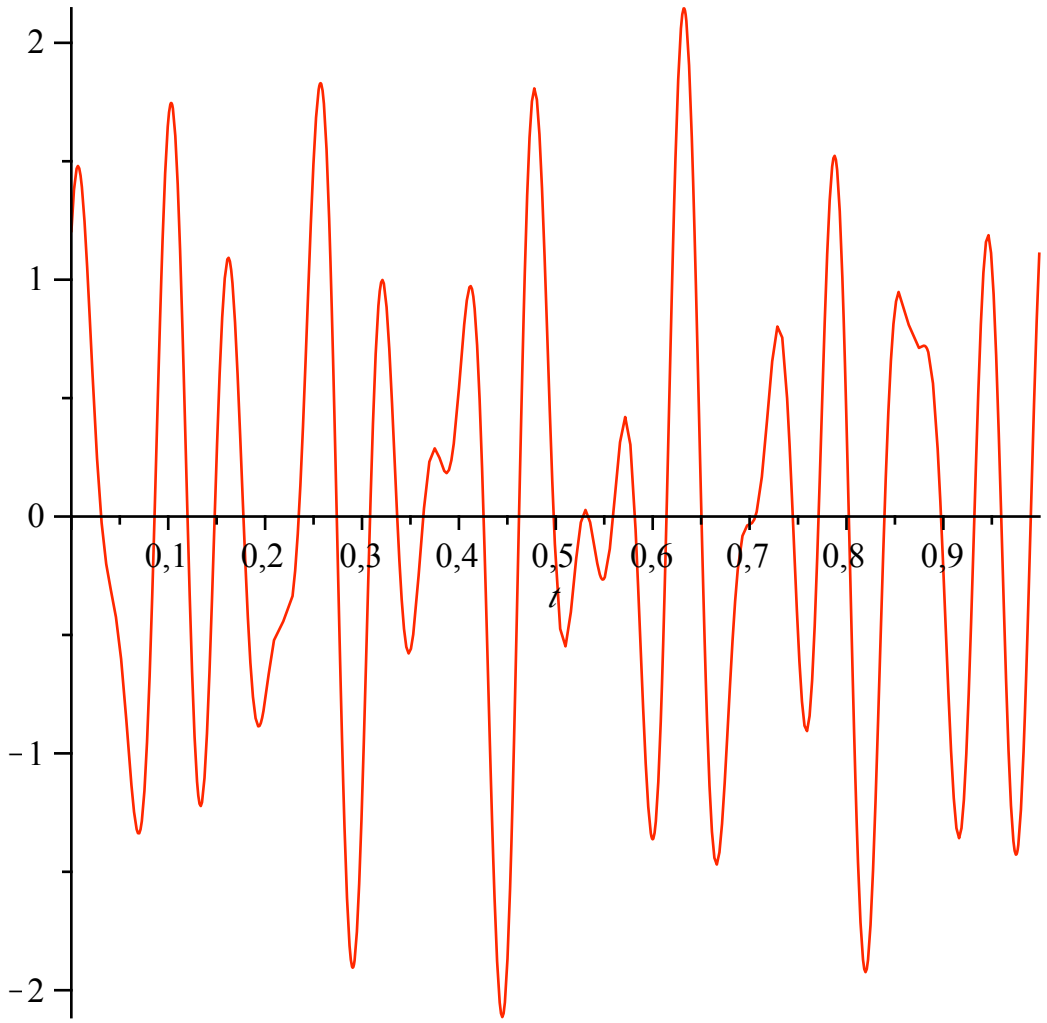
>

Exemple 1 (autre méthode)

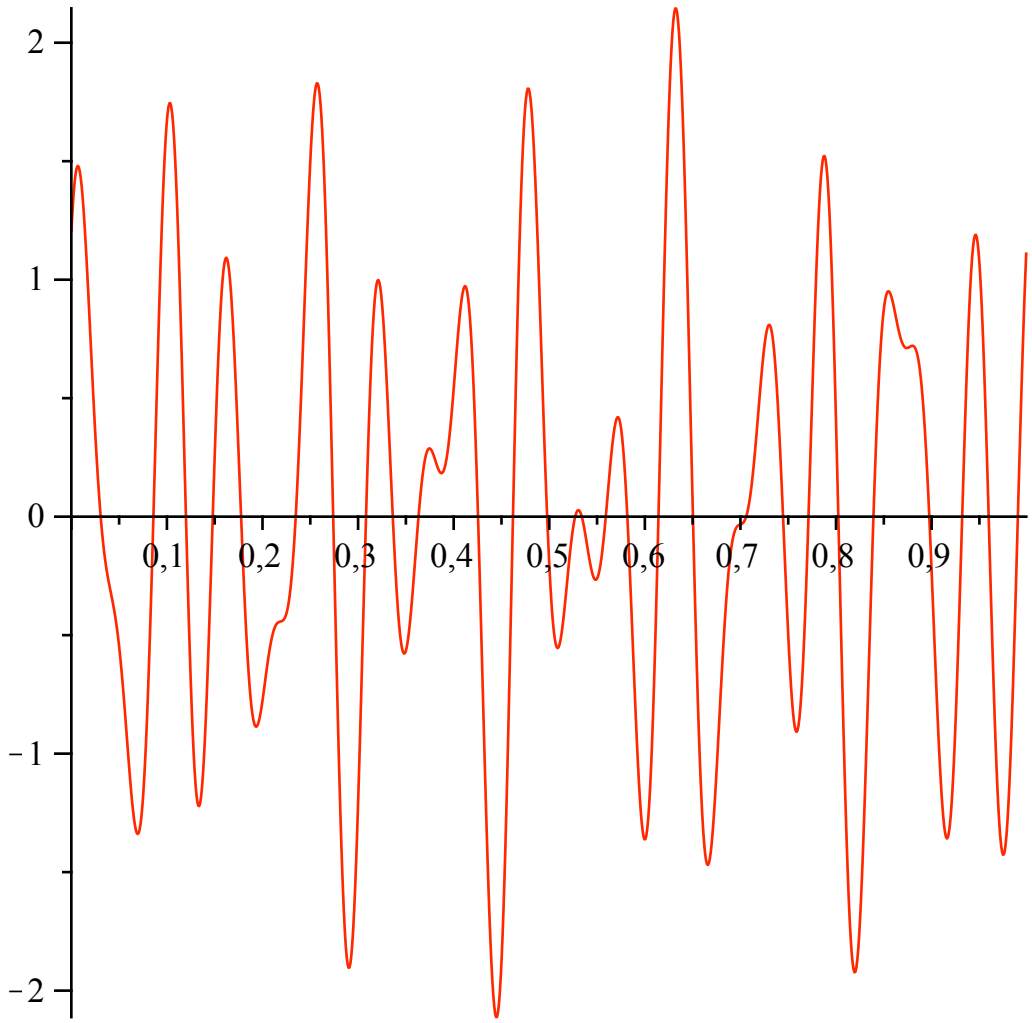
```
> restart;
> X:=sin(13*2*Pi*t)+0.5*cos(8*2*Pi*t)+0.7*cos(19*2*Pi*t);
```

$$X := \sin(26\,\pi\,t) + 0.5\cos(16\,\pi\,t) + 0.7\cos(38\,\pi\,t) \qquad \textbf{(2.1)}$$

```
> plot(X,t=0..1023/1024);
```



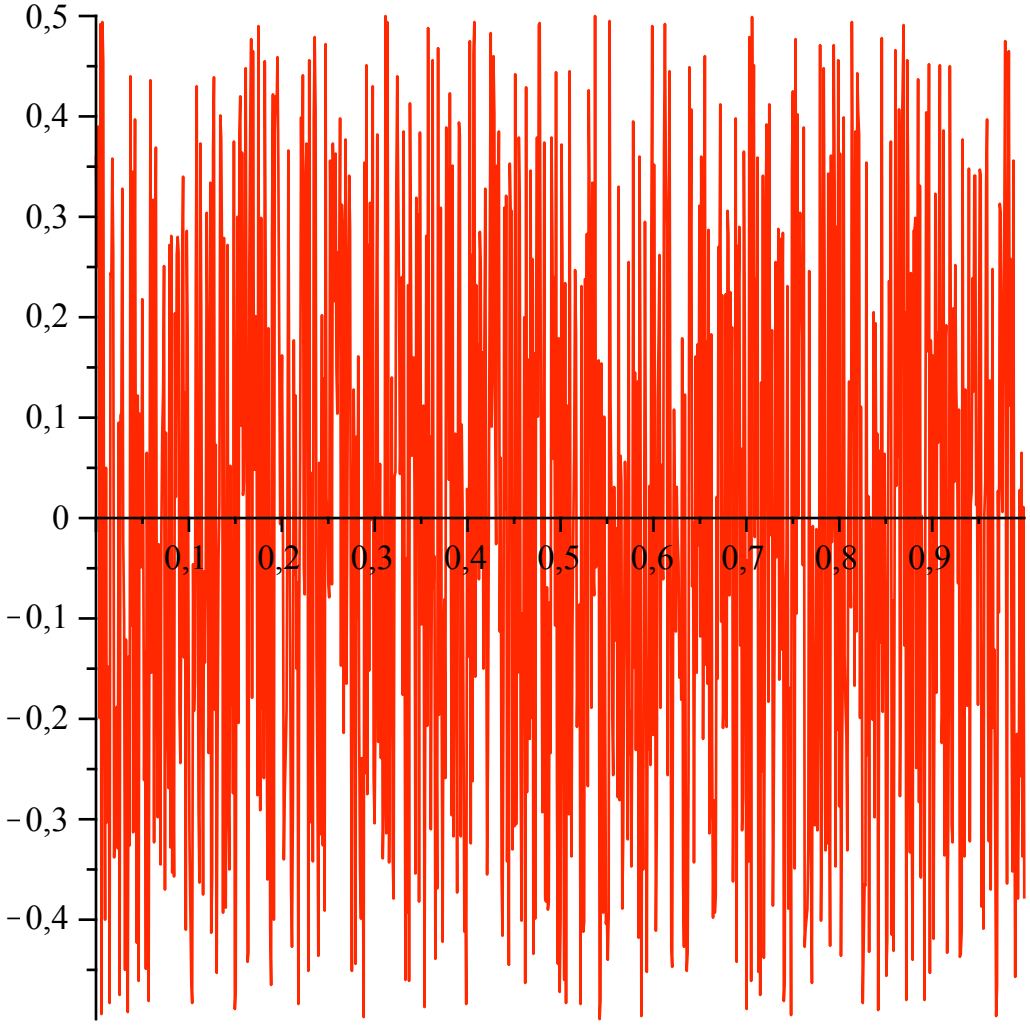

```
> t:=[evalf(($ 0..1023)/1024)]:
> X:=[seq([t[i],evalf(sin(13*2*Pi*t[i])+0.5*cos(8*2*Pi*t[i])
  +.7*cos(19*2*Pi*t[i]))],i=1..1024)]:
> plot(X);
```



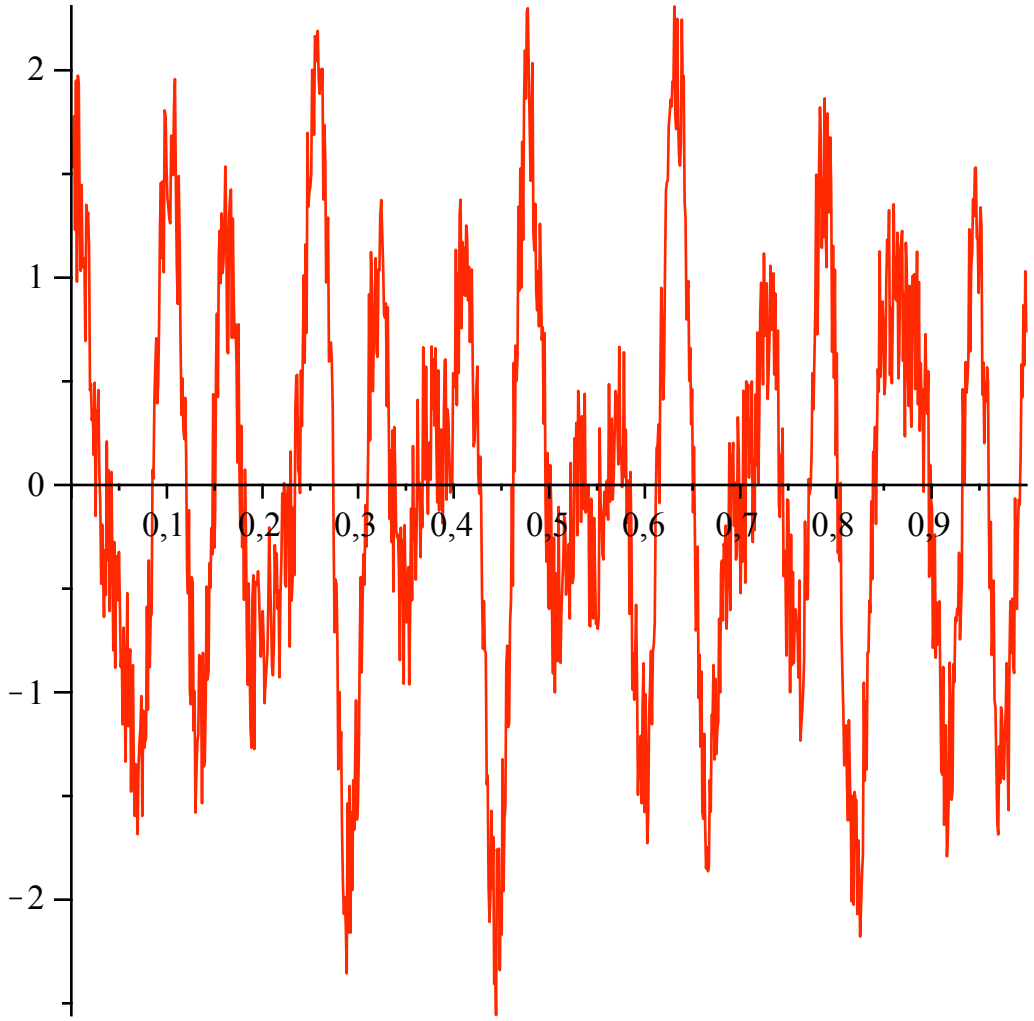

```
> h:=evalf(rand(-499..500)/1000);
```

$h := 0.001000000000$ **proc**( ) **proc**( ) **option** *builtin* = *RandNumberInterface*; **end proc**(6, 1000, 10) − 499 **end proc** **(2.2)**

```
> Y:=[seq([t[i],h()],i=1..1024)]:
> Z:=[seq([t[i],X[i][2]+Y[i][2]],i=1..1024)]:
> plot(Y);
```




```
> plot(Z);
```




```
> ZZ:=[seq(Z[i][2],i=1..1024)]:
> with(DiscreteTransforms);
```

[*FourierTransform*, *InverseFourierTransform*] **(2.3)**

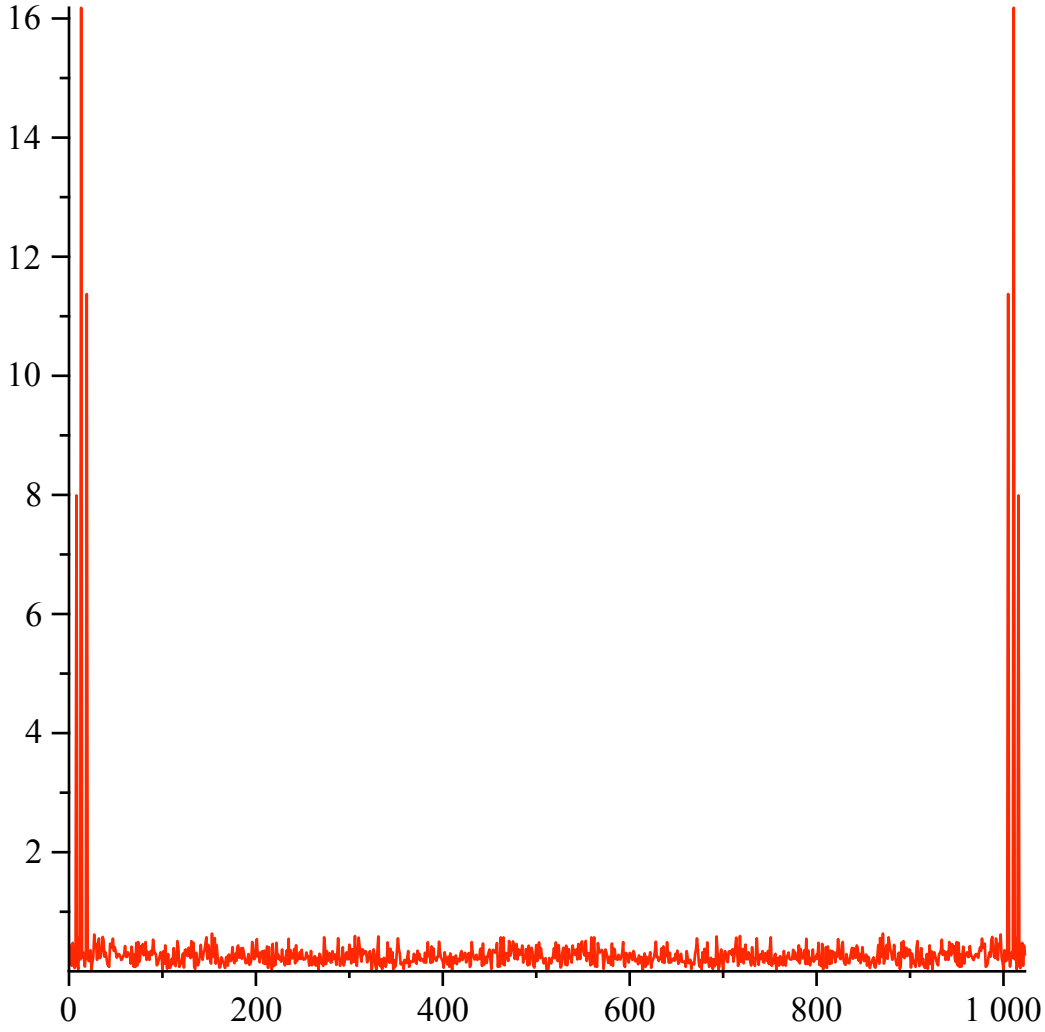
```
> ZZ:=convert(ZZ,Vector):
> for i from 1 to 10 do print(ZZ[i]) od;
```

1.561000000
1.533328530
1.588508473
1.779233318
1.230331391
1.949764686
0.9806259650
1.974134224
1.929628584
1.258560707 **(2.4)**

```
> F:=FourierTransform(ZZ);
```

$$F := \begin{bmatrix} 1 .. 1024\ Vector_{column} \\ Data\ Type: anything \\ Storage: rectangular \\ Order: Fortran_order \end{bmatrix} \quad (2.5)$$

```
> FF:=[seq([i-1,abs(F[i])],i=1..1024)]:
> plot(FF);
```




```
> FS:=Vector(1024);
```

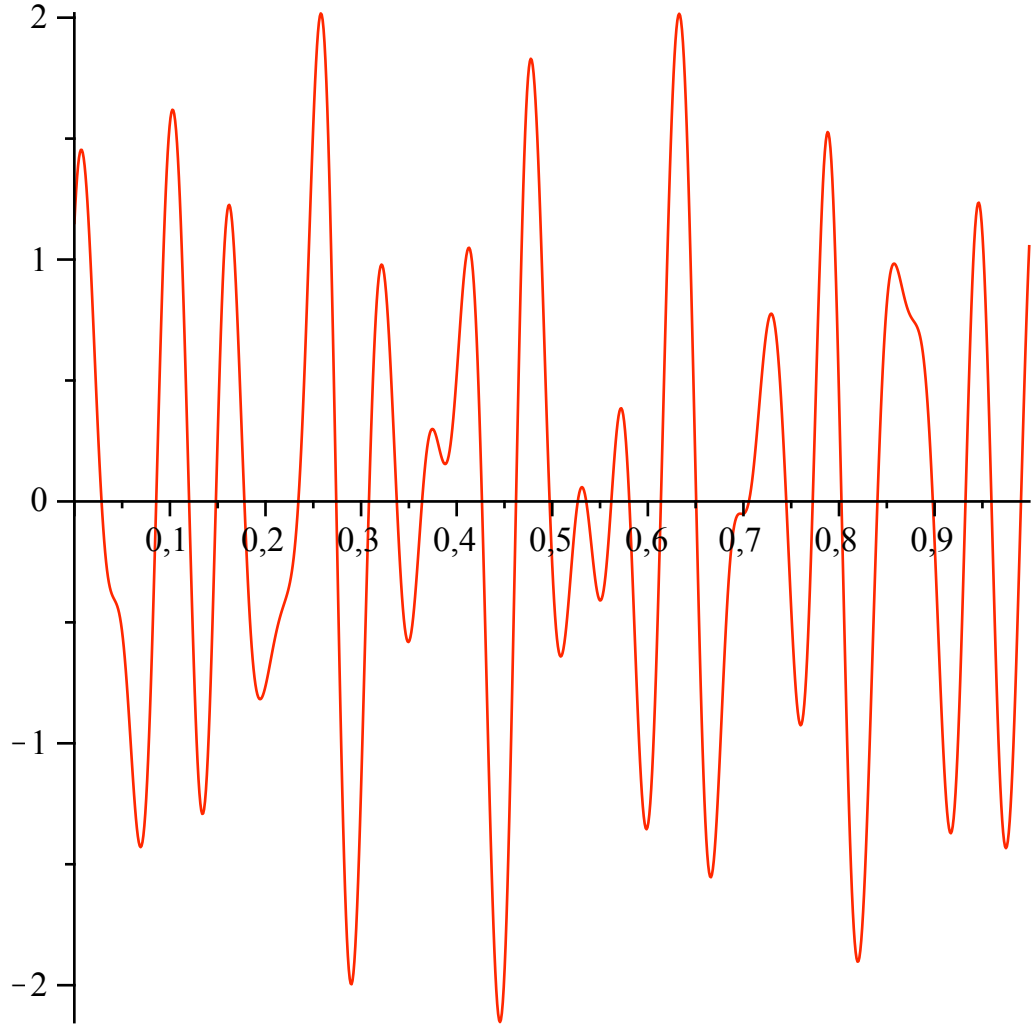
$$FS := \begin{bmatrix} 1 .. 1024\ Vector_{column} \\ Data\ Type: anything \\ Storage: rectangular \\ Order: Fortran_order \end{bmatrix} \quad (2.6)$$

```
> for i from 1 to 30 do FS[i]:=F[i] od:
  for i from 31 to 994 do FS[i]:=0 od:
  for i from 995 to 1024 do FS[i]:=F[i] od:
>
> FS[1];
```

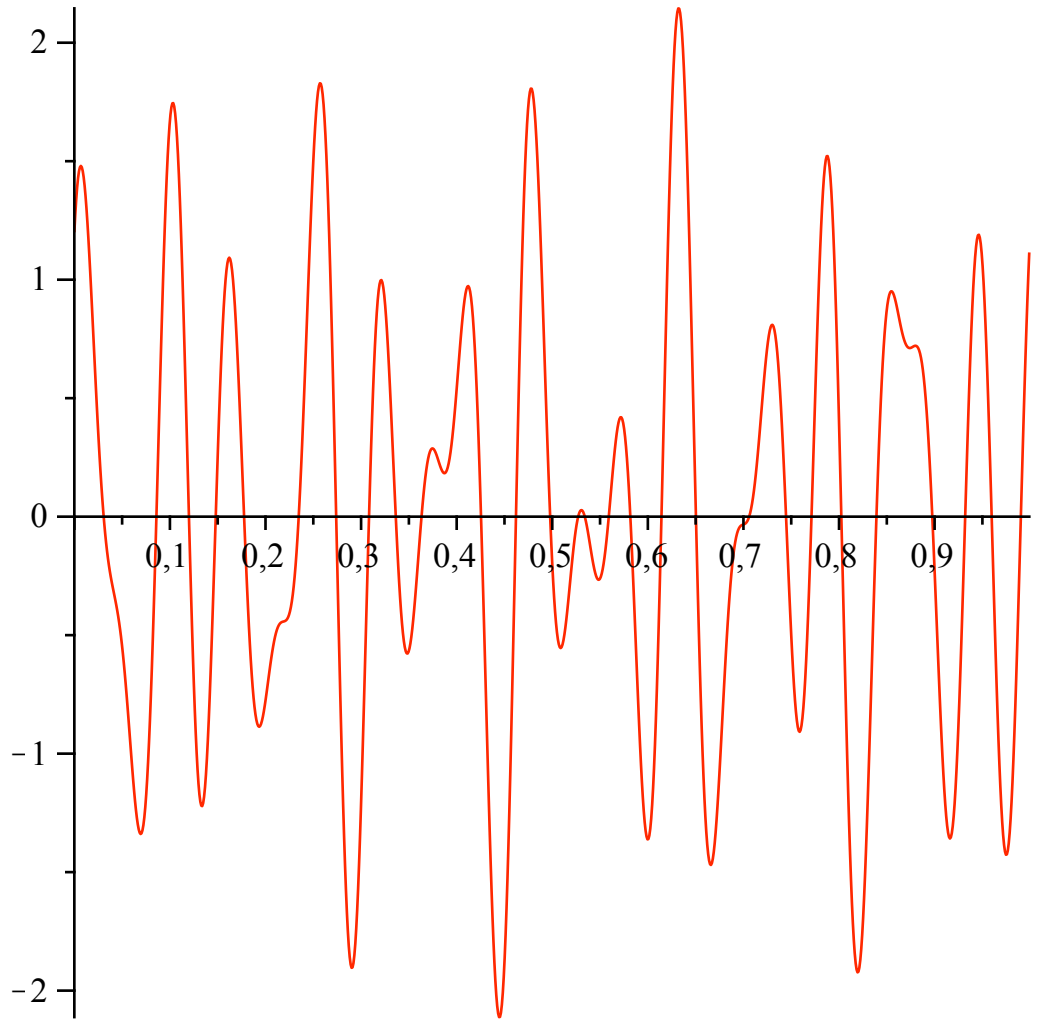
(2.7)

-0.279249995909375158 + 0. I (2.7)

```
> S:=InverseFourierTransform(FS):
> SS:=[seq([t[i],Re(S[i])],i=1..1024)]:
> plot(SS);
```




```
> plot(X);
```



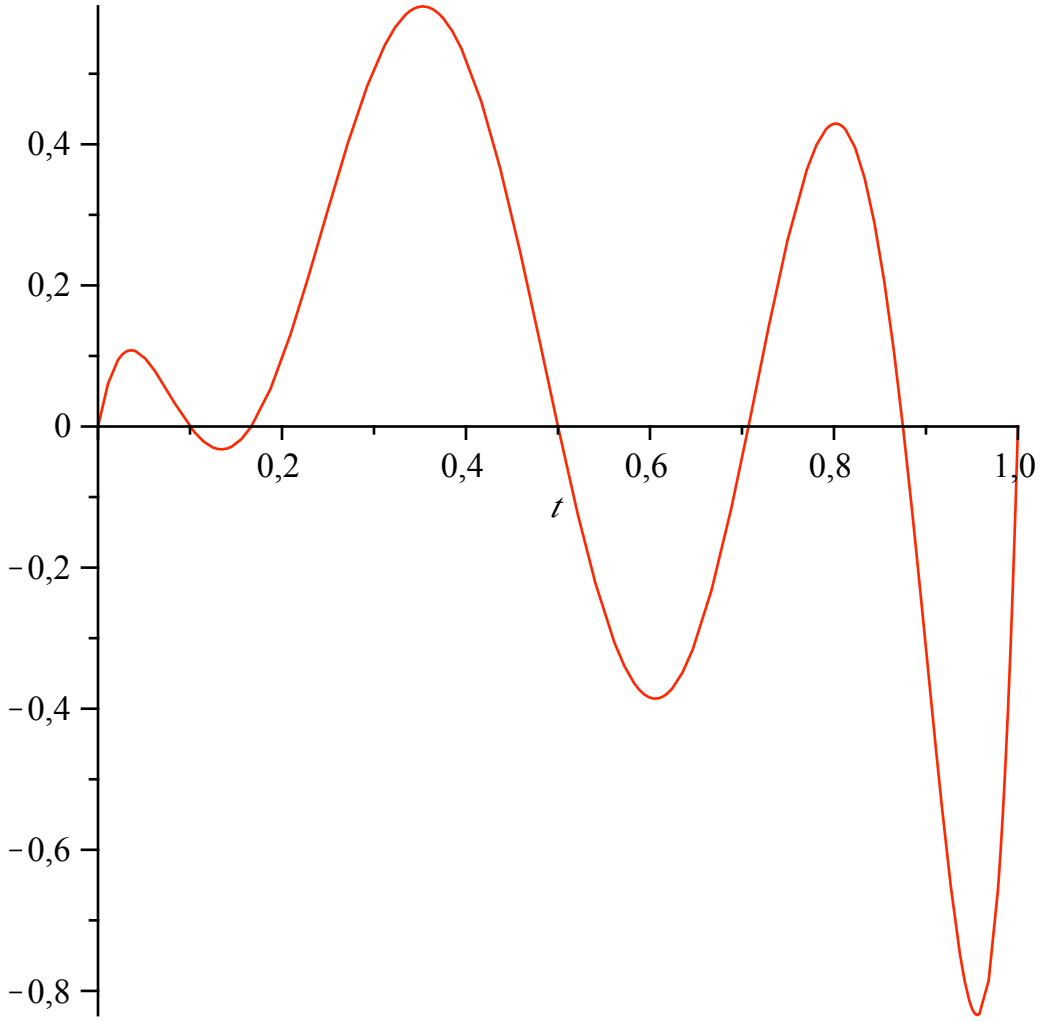
>

Exemple 2 (polynôme)

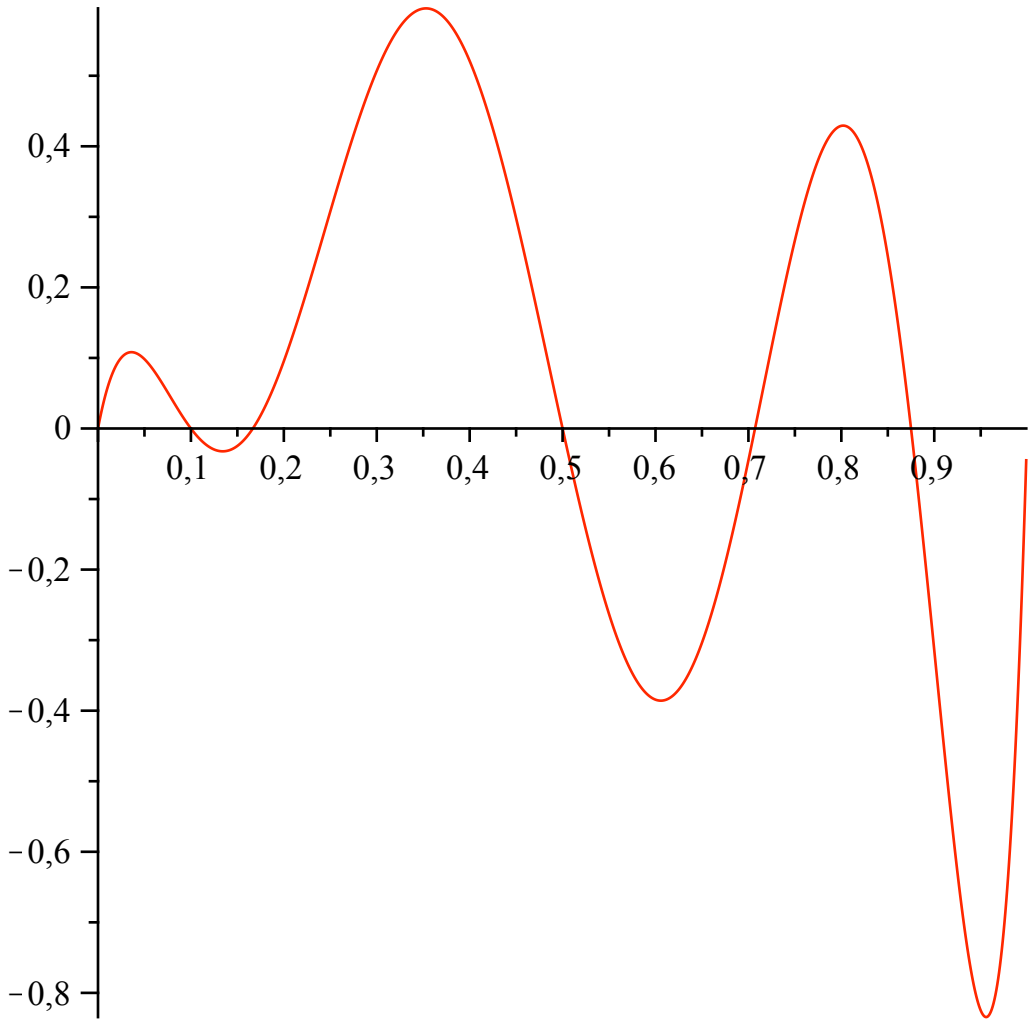
```
> restart;
> X:=expand(1920*(t-1/6)*(t^2-1/2)*(t-1/2)*(t-7/8)*(t-1/10)*
  (t-1)*t);
```

$$X := 1920\,t^8 - 5072\,t^7 + 3768\,t^6 + 692\,t^5 - 2082\,t^4 + 908\,t^3 - 141\,t^2 + 7\,t \qquad (3.1)$$

```
> plot(X,t=0..1);
```



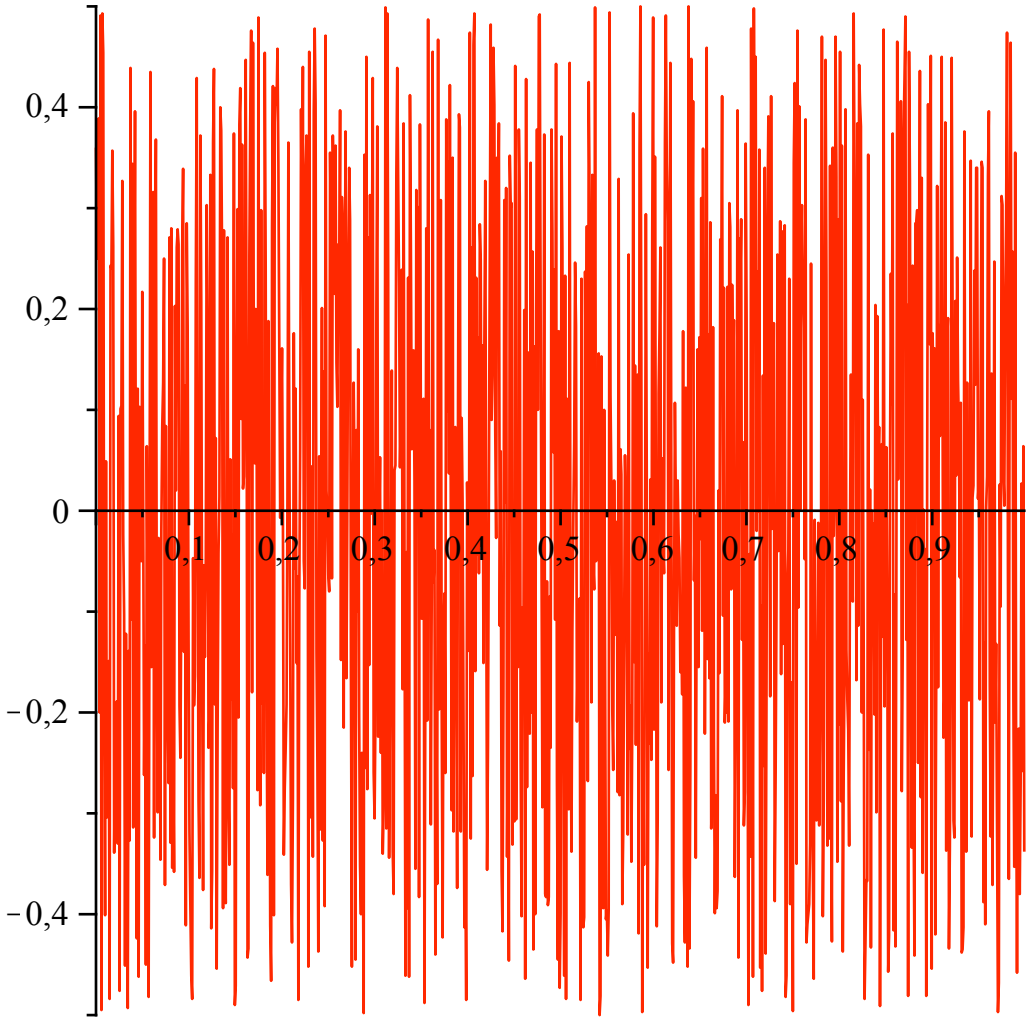

```
> t:=[evalf(($ 0..1023)/1024)]:
> X:=[seq([t[i],evalf(1920*(t[i]-1/6)*(t[i]^2-1/2)*(t[i]-1/2)*
  (t[i]-7/8)*(t[i]-1/10)*(t[i]-1)*t[i])],i=1..1024)]:
> plot(X);
```



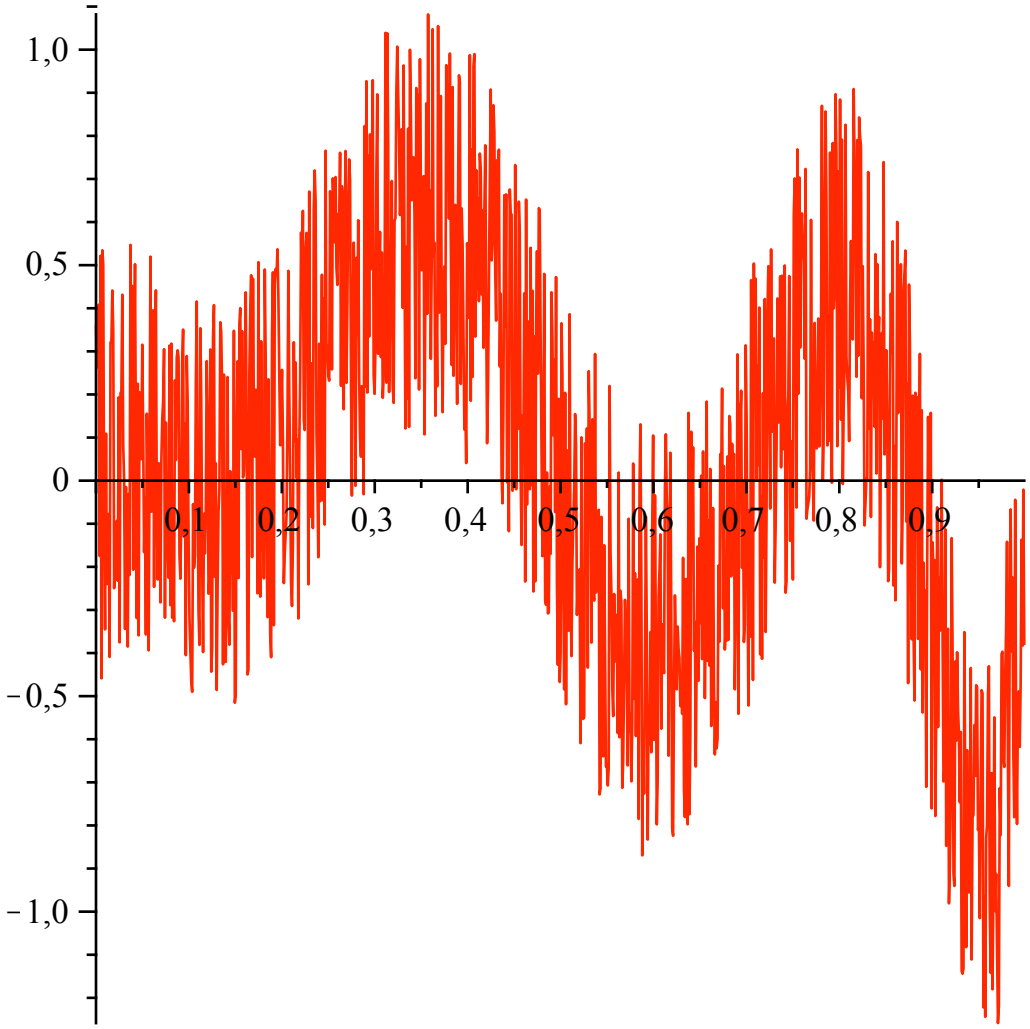

```
> h:=evalf(rand(-500..500)/1000);
```

$h := 0.001000000000$ **proc**( ) **proc**( ) **option** *builtin* = *RandNumberInterface*; **end proc**(6, 1001, 10) − 500 **end proc** **(3.2)**

```
> Y:=[seq([t[i],h()],i=1..1024)]:
> Z:=[seq([t[i],X[i][2]+Y[i][2]],i=1..1024)]:
> plot(Y);
```




```
> plot(Z);
```



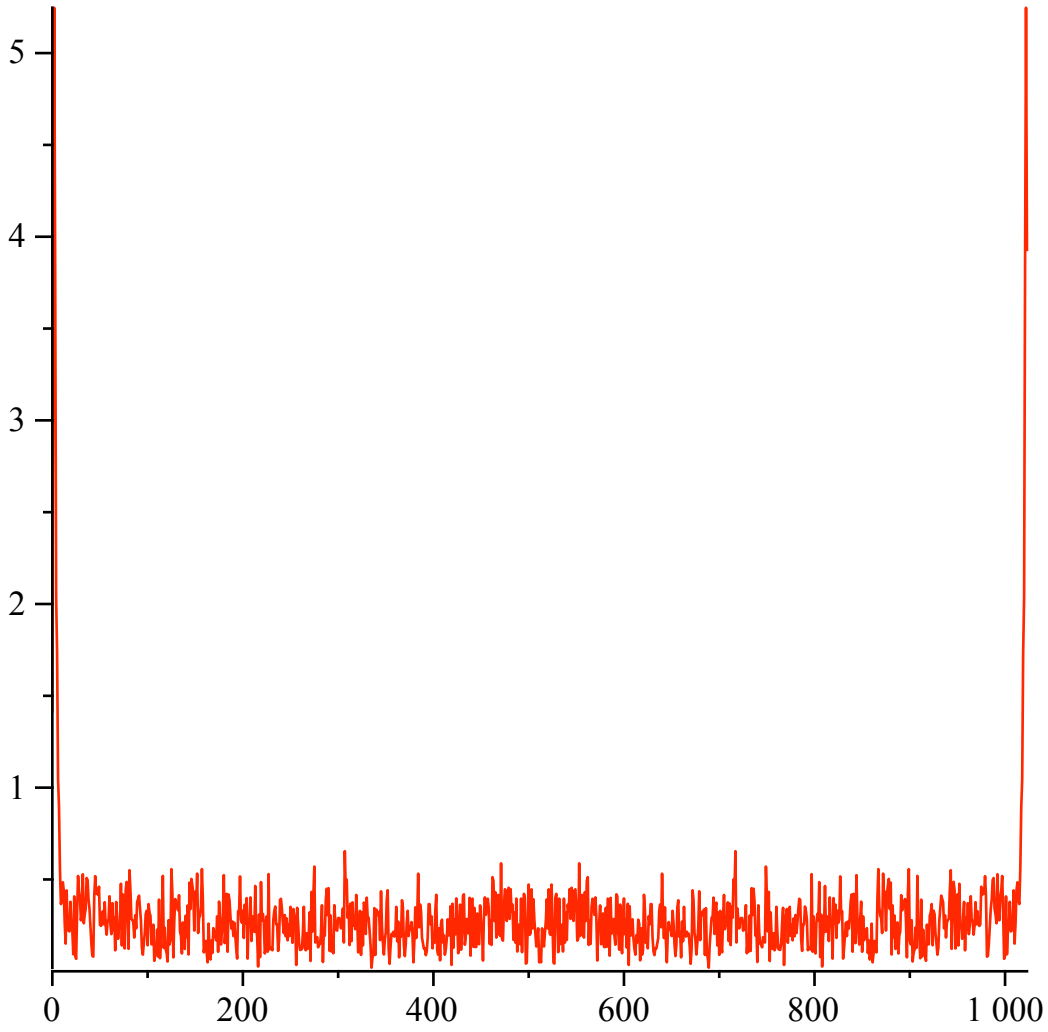

```
> ZZ:=[seq(Z[i][2],i=1..1024)]:
> with(DiscreteTransforms);
```

$$[FourierTransform, InverseFourierTransform] \tag{3.3}$$

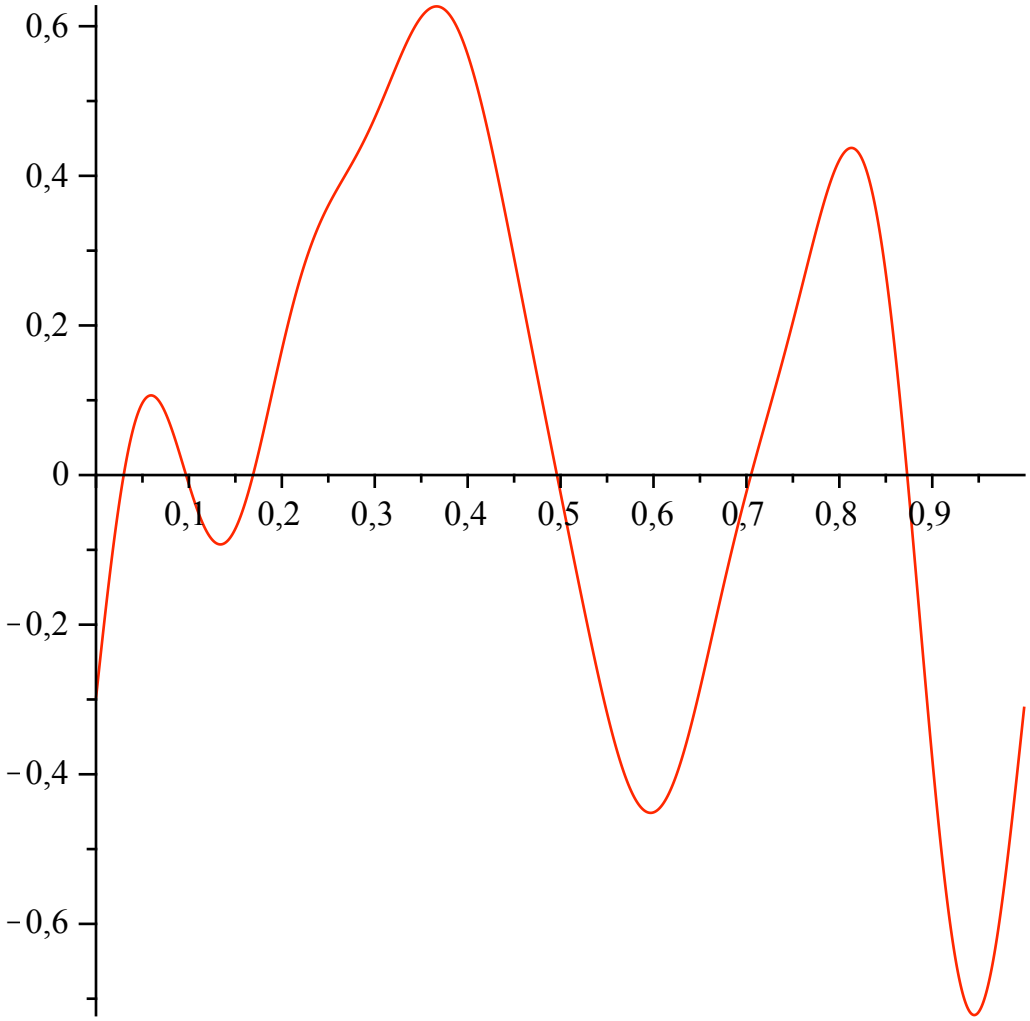
```
> ZZ:=convert(ZZ,Vector):
> F:=FourierTransform(ZZ);
```

$$F := \begin{bmatrix} 1 .. 1024\ Vector_{column} \\ Data\ Type: anything \\ Storage: rectangular \\ Order: Fortran_order \end{bmatrix} \tag{3.4}$$

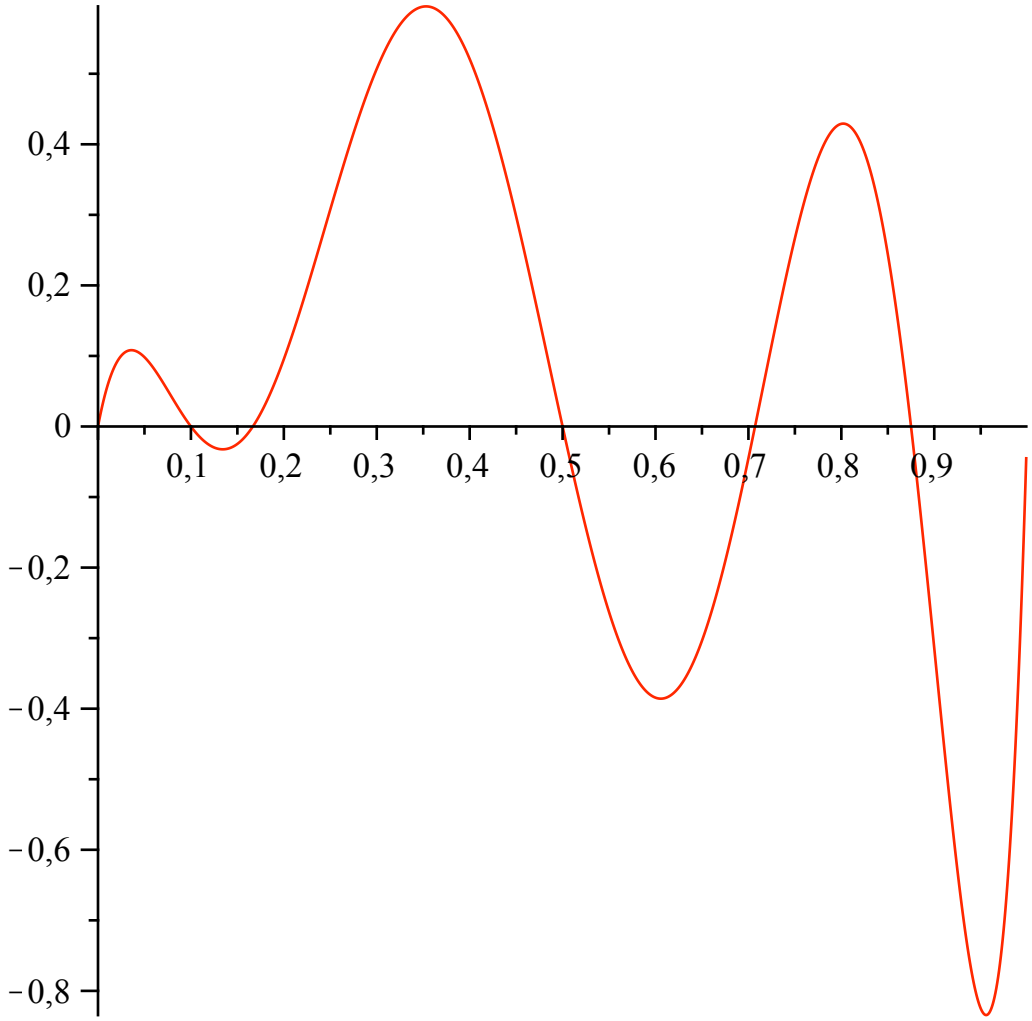
```
> FF:=[seq([i-1,abs(F[i])],i=1..1024)]:
> plot(FF);
```




```
> FS:=Vector(1024):
> for i from 1 to 1024 do if abs(F[i])>1 then FS[i]:=F[i] else
  FS[i]=0 fi od;
> S:=InverseFourierTransform(FS):
> SS:=[seq([t[i],Re(S[i])],i=1..1024)]:
> plot(SS);
```




```
> plot(X);
```



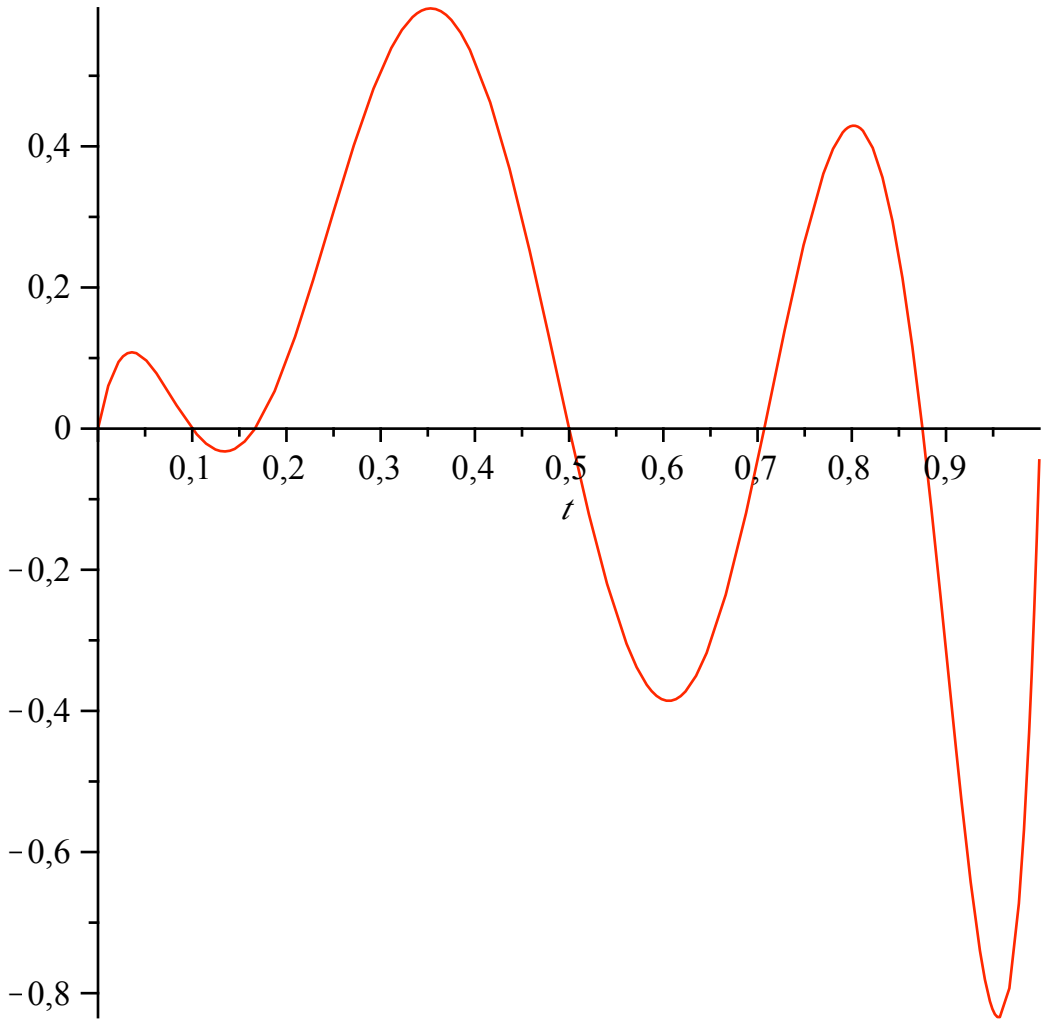
>

Exemple 2 (autre méthode)

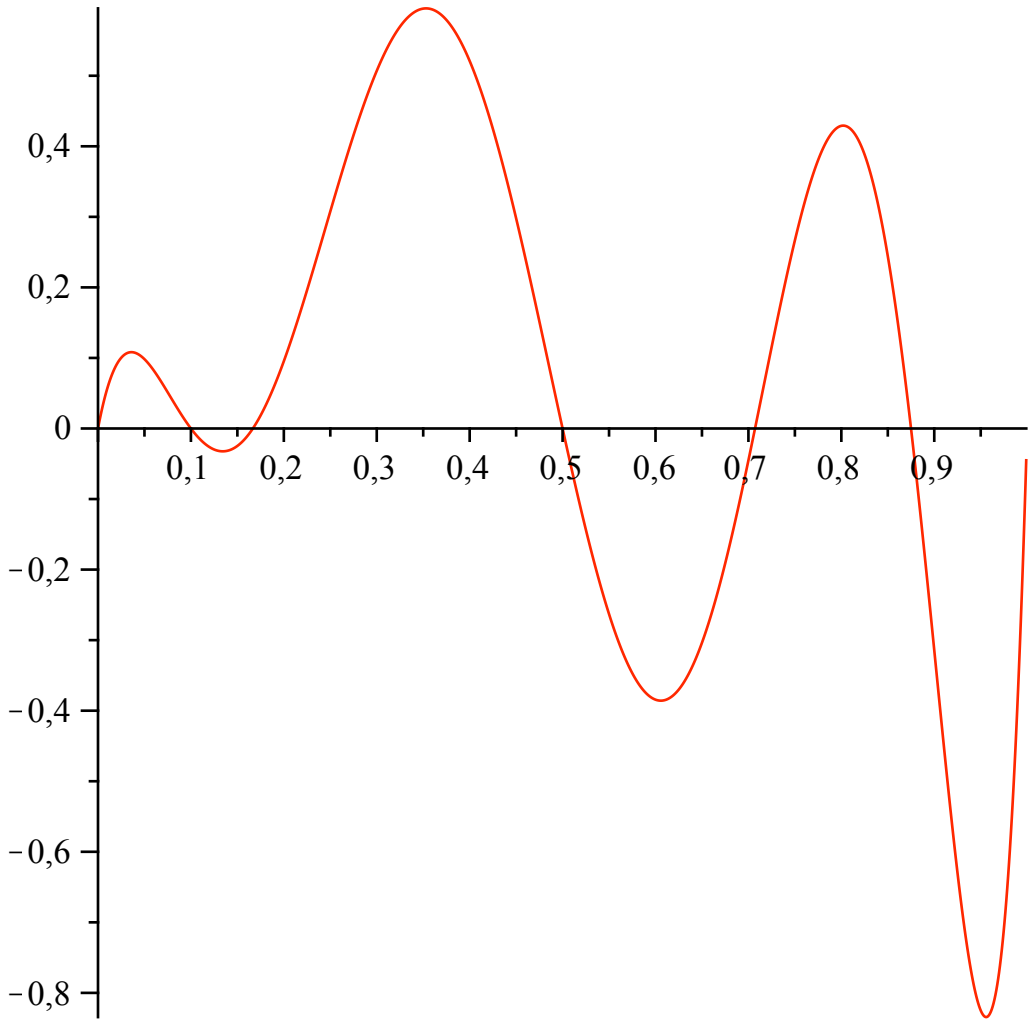
```
> restart;
> X:=expand(1920*(t-1/6)*(t^2-1/2)*(t-1/2)*(t-7/8)*(t-1/10)*
  (t-1)*t);
```

$$X := 1920\,t^8 - 5072\,t^7 + 3768\,t^6 + 692\,t^5 - 2082\,t^4 + 908\,t^3 - 141\,t^2 + 7\,t \qquad (4.1)$$

```
> plot(X,t=0..1023/1024);
```



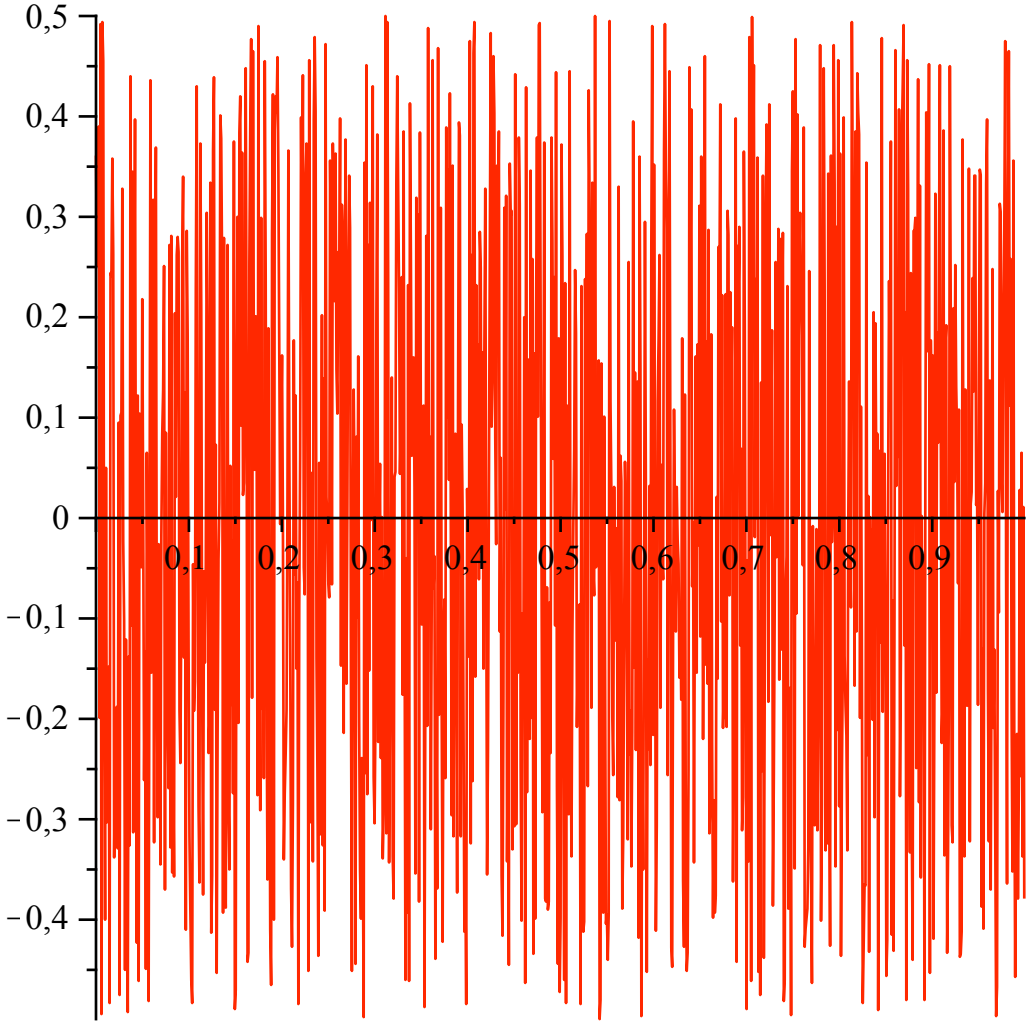

```
> t:=[evalf(($ 0..1023)/1024)]:
> X:=[seq([t[i],evalf(1920*(t[i]-1/6)*(t[i]^2-1/2)*(t[i]-1/2)*
  (t[i]-7/8)*(t[i]-1/10)*(t[i]-1)*t[i])],i=1..1024)]:
> plot(X);
```



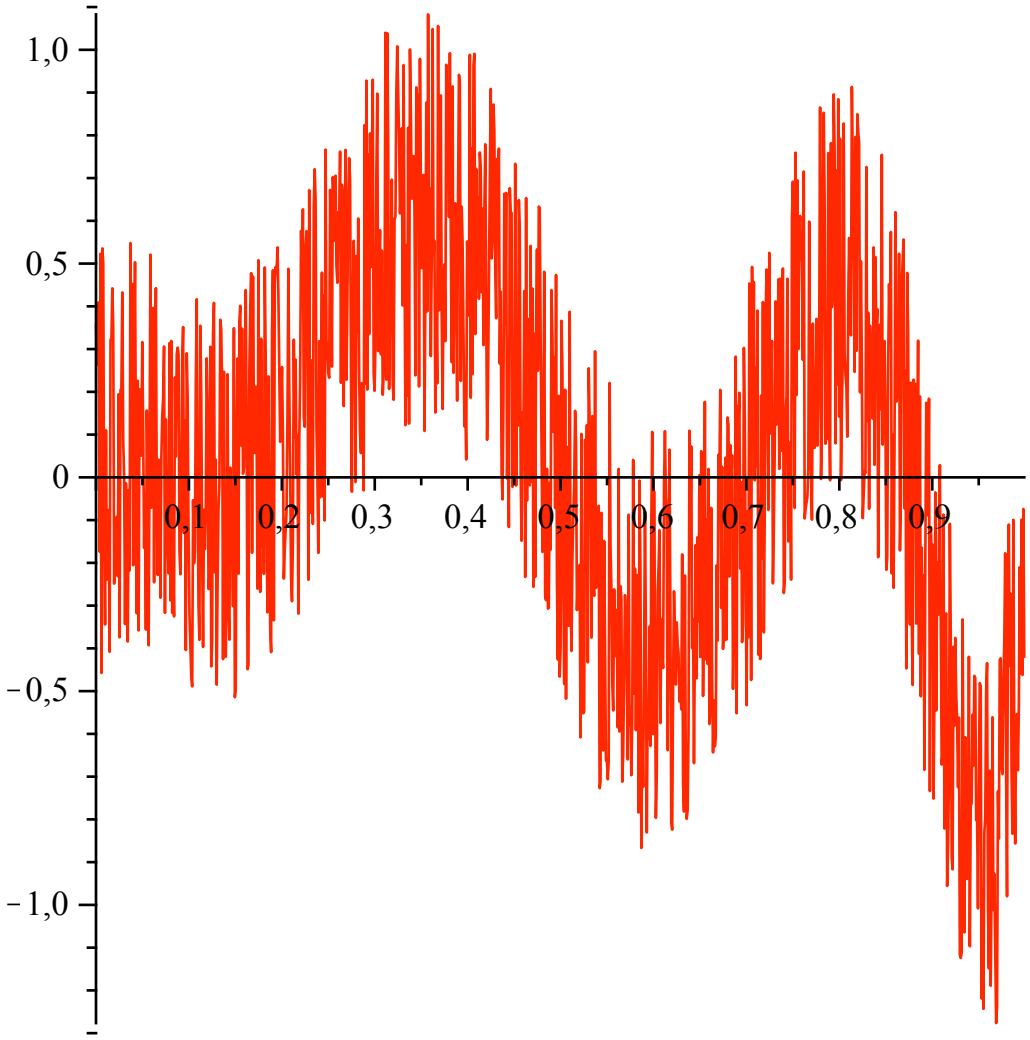

```
> h:=evalf(rand(-499..500)/1000);
```

$h := 0.001000000000$ **proc**( ) **proc**( ) **option** *builtin* = *RandNumberInterface*; **end proc**(6, 1000, 10) − 499 **end proc** **(4.2)**

```
> Y:=[seq([t[i],h()],i=1..1024)]:
> Z:=[seq([t[i],X[i][2]+Y[i][2]],i=1..1024)]:
> plot(Y);
```



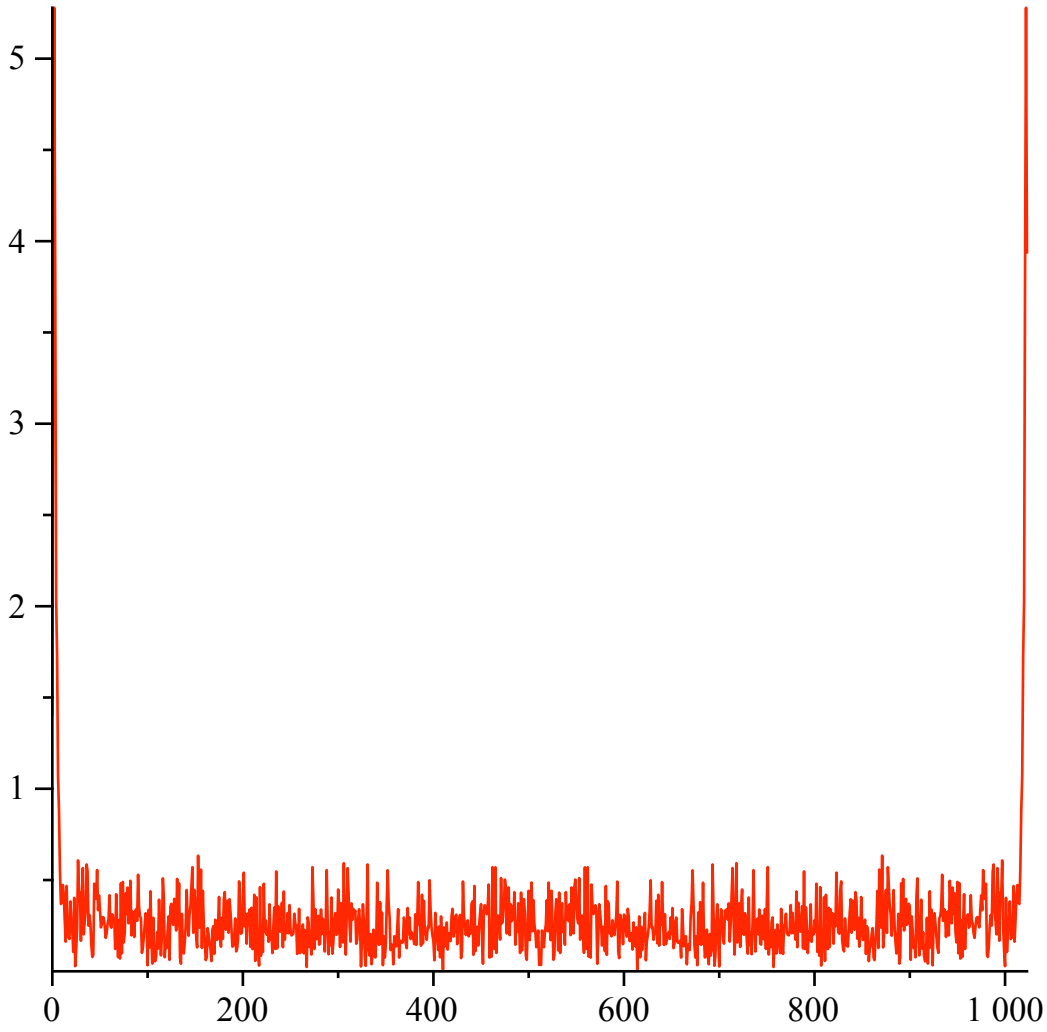

```
> plot(Z);
```



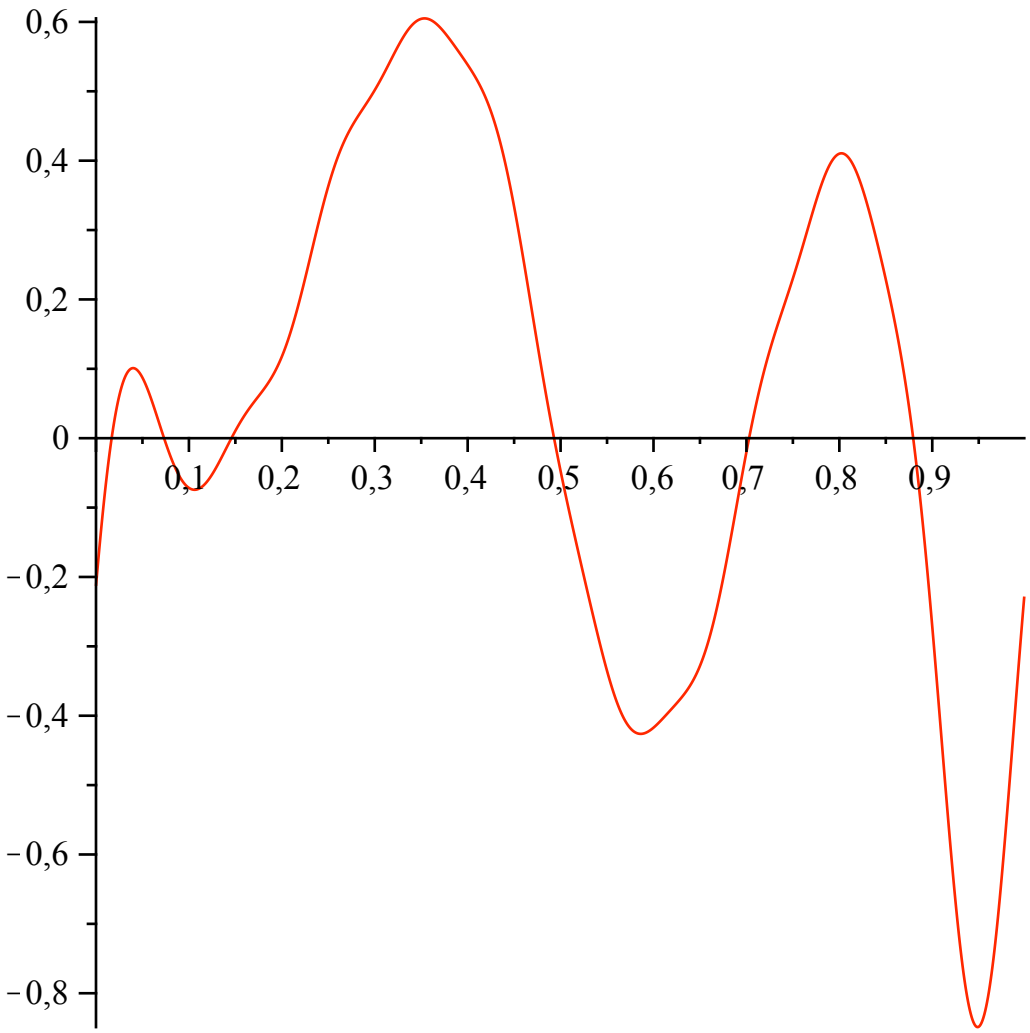

```
> ZZ:=[seq(Z[i][2],i=1..1024)]:
> with(DiscreteTransforms);
```

[*FourierTransform*, *InverseFourierTransform*] **(4.3)**

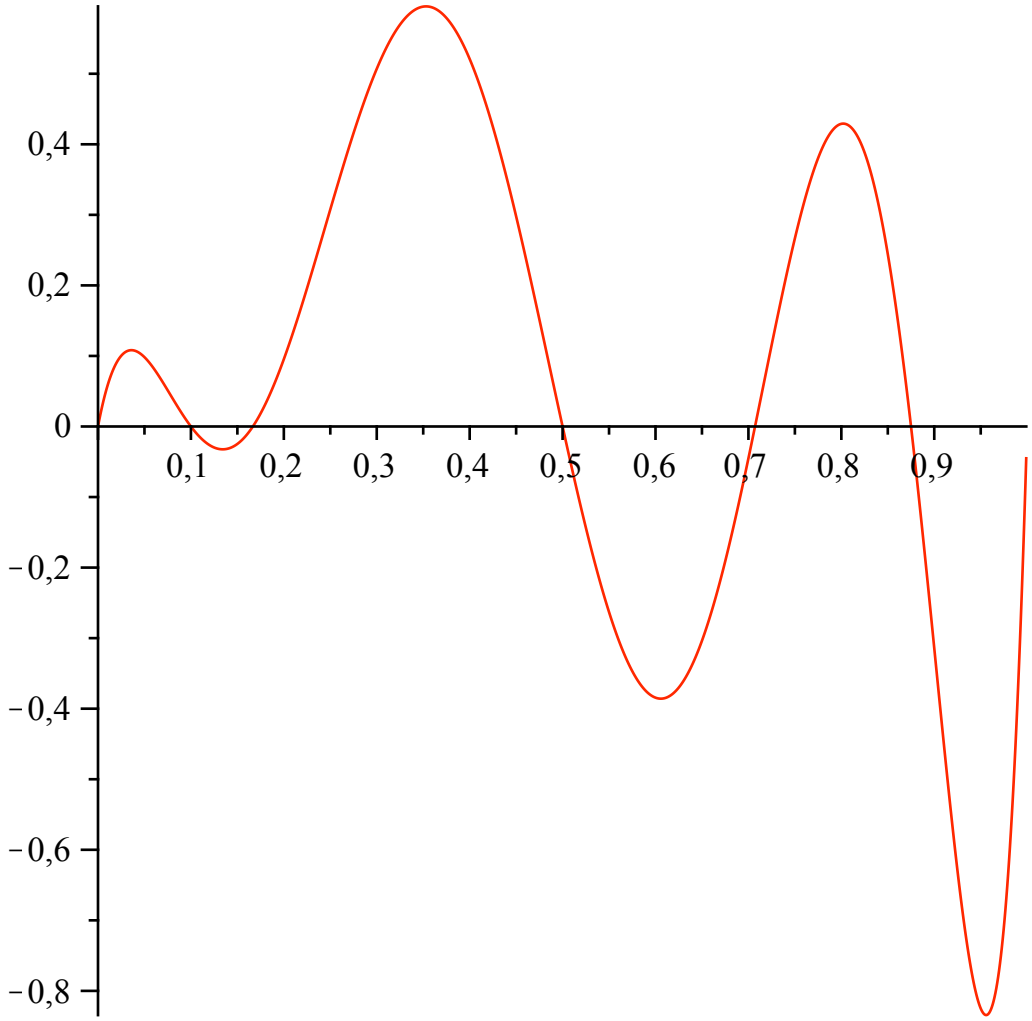
```
> ZZ:=convert(ZZ,Vector):
> F:=FourierTransform(ZZ):
> FF:=[seq([i-1,abs(F[i])],i=1..1024)]:
> plot(FF);
```




```
> FS:=Vector(1024):
> for i from 1 to 10 do FS[i]:=F[i] od:
  for i from 11 to 1014 do FS[i]:=0 od:
  for i from 1014 to 1024 do FS[i]:=F[i] od:
> S:=InverseFourierTransform(FS):
> SS:=[seq([t[i],Re(S[i])],i=1..1024)]:
> plot(SS);
```




```
> plot(X);
```



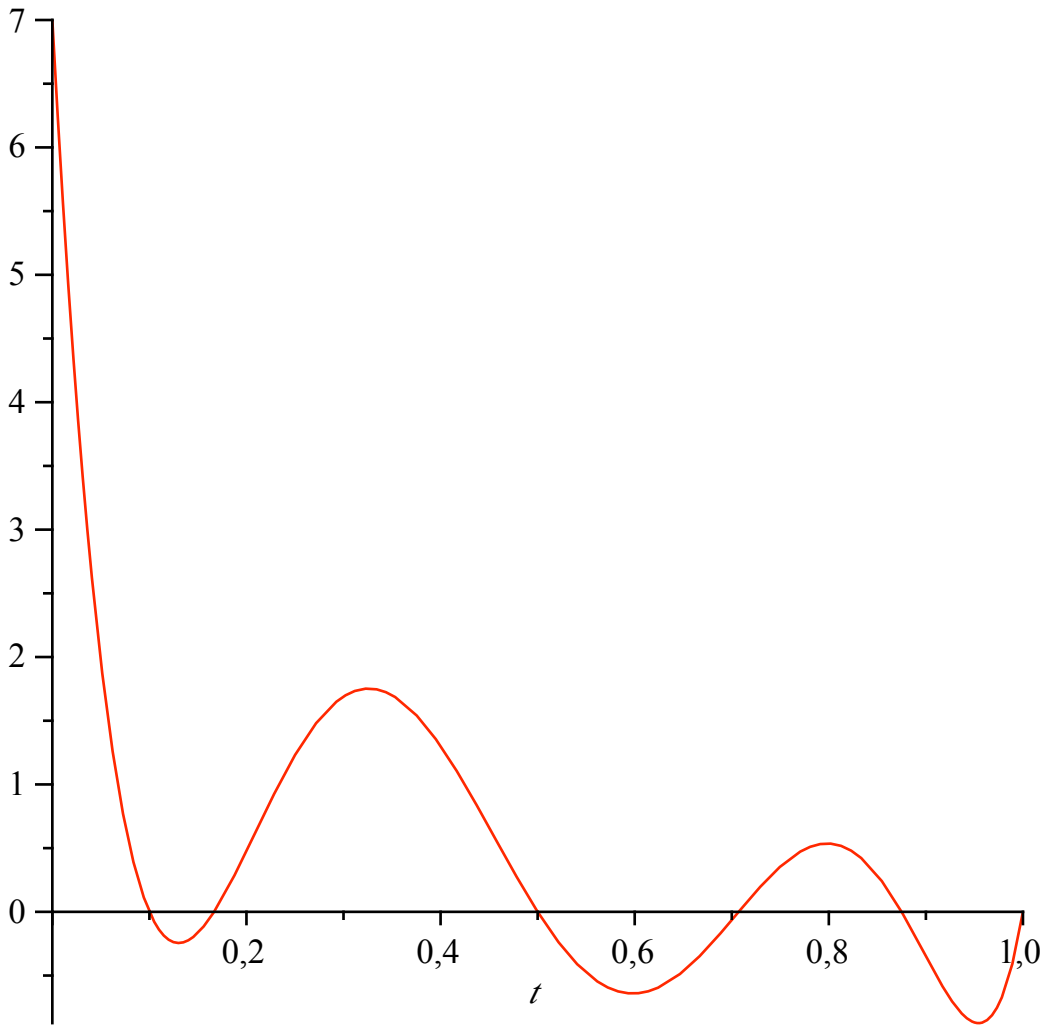
>

Exemple 3 (polynôme, avec "effet de bord")

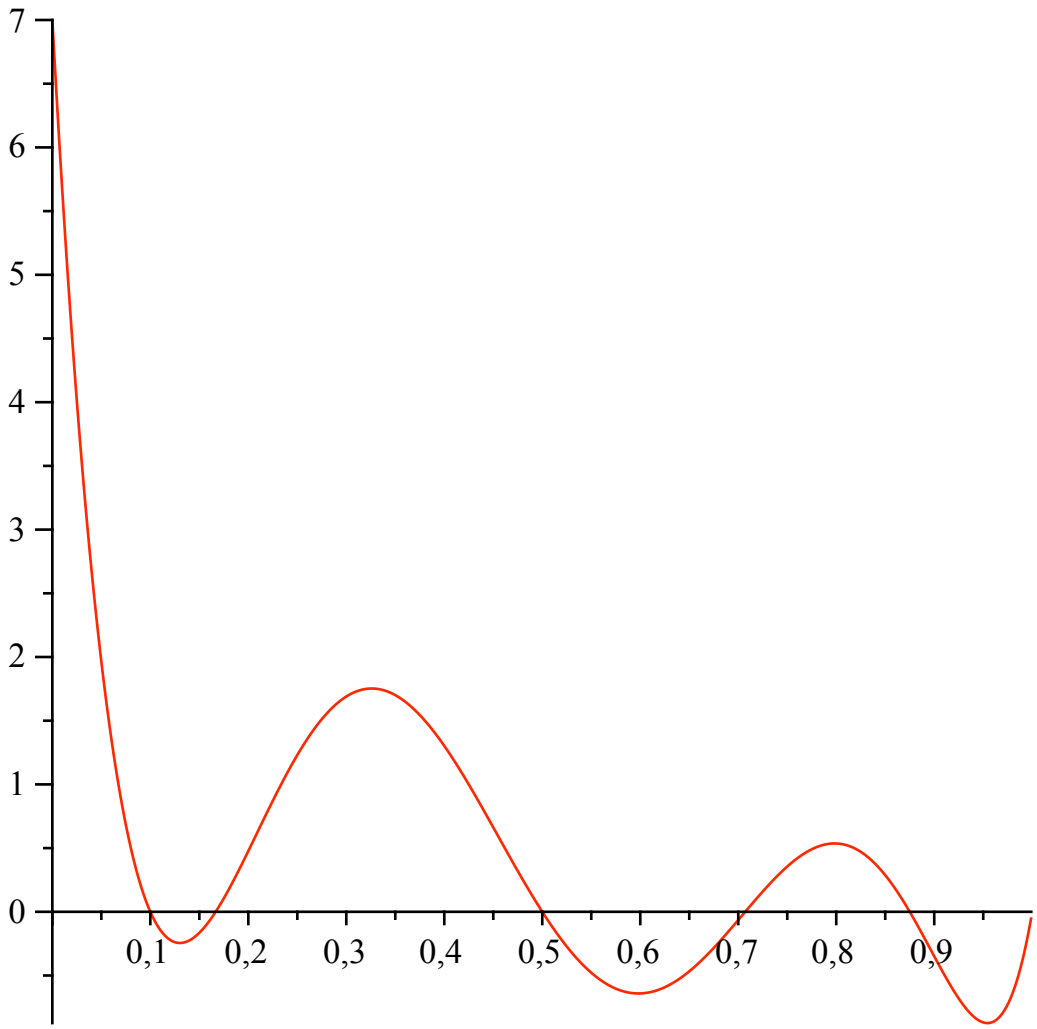
```
> restart;
> X:=expand(1920*(t-1/6)*(t^2-1/2)*(t-1/2)*(t-7/8)*(t-1/10)*
  (t-1));
```

$$X := 1920\,t^7 - 5072\,t^6 + 3768\,t^5 + 692\,t^4 - 2082\,t^3 + 908\,t^2 - 141\,t + 7 \qquad \textbf{(5.1)}$$

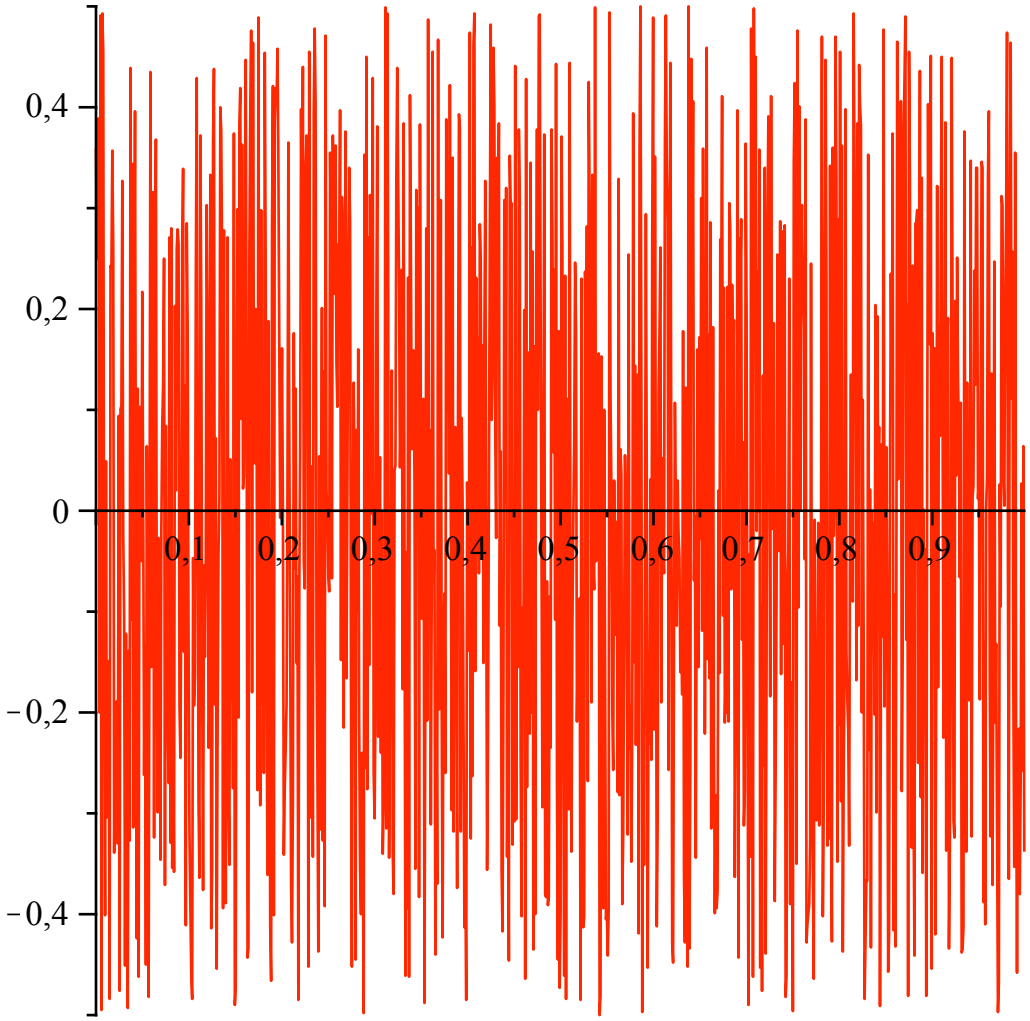
```
> plot(X,t=0..1);
```



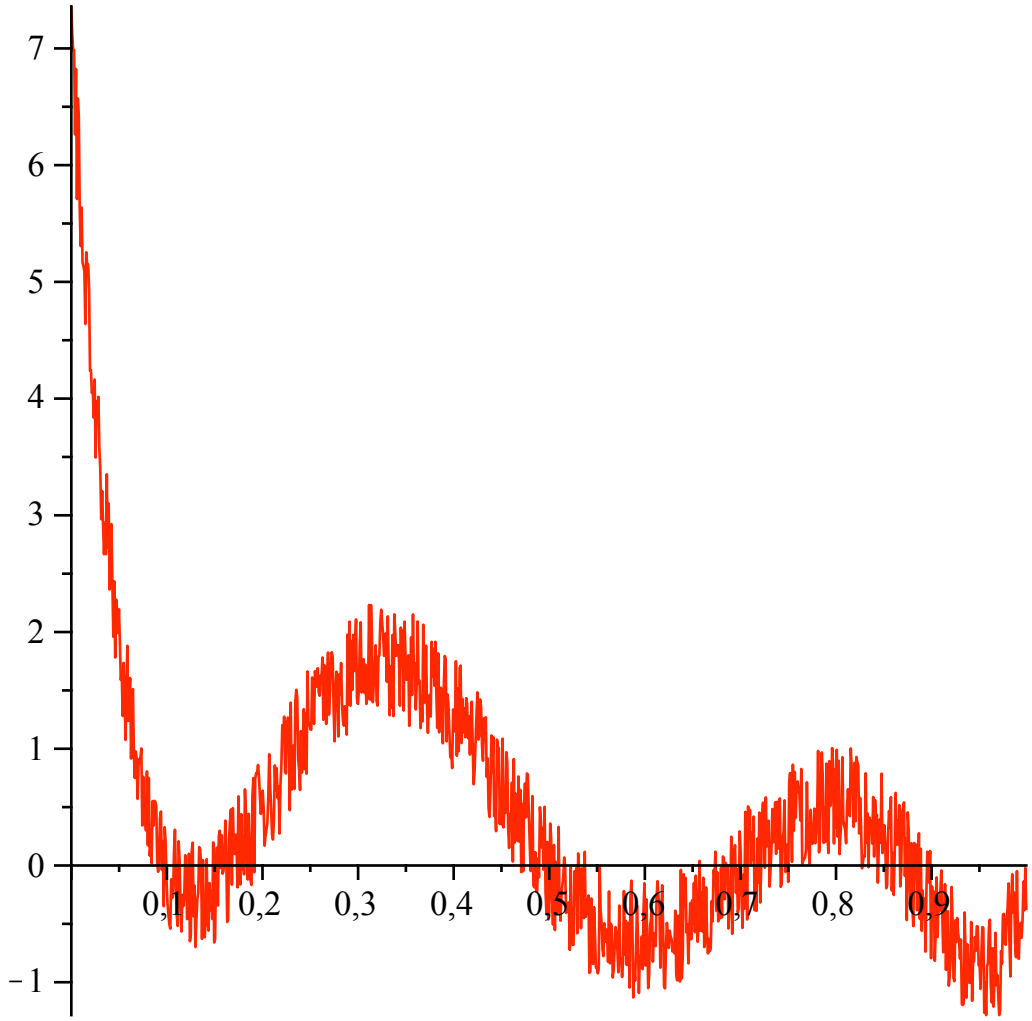

```
> t:=[evalf(($ 0..1023)/1024)]:
> X:=[seq([t[i],evalf(1920*(t[i]-1/6)*(t[i]^2-1/2)*(t[i]-1/2)*
  (t[i]-7/8)*(t[i]-1/10)*(t[i]-1))],i=1..1024)]:
> plot(X);
```



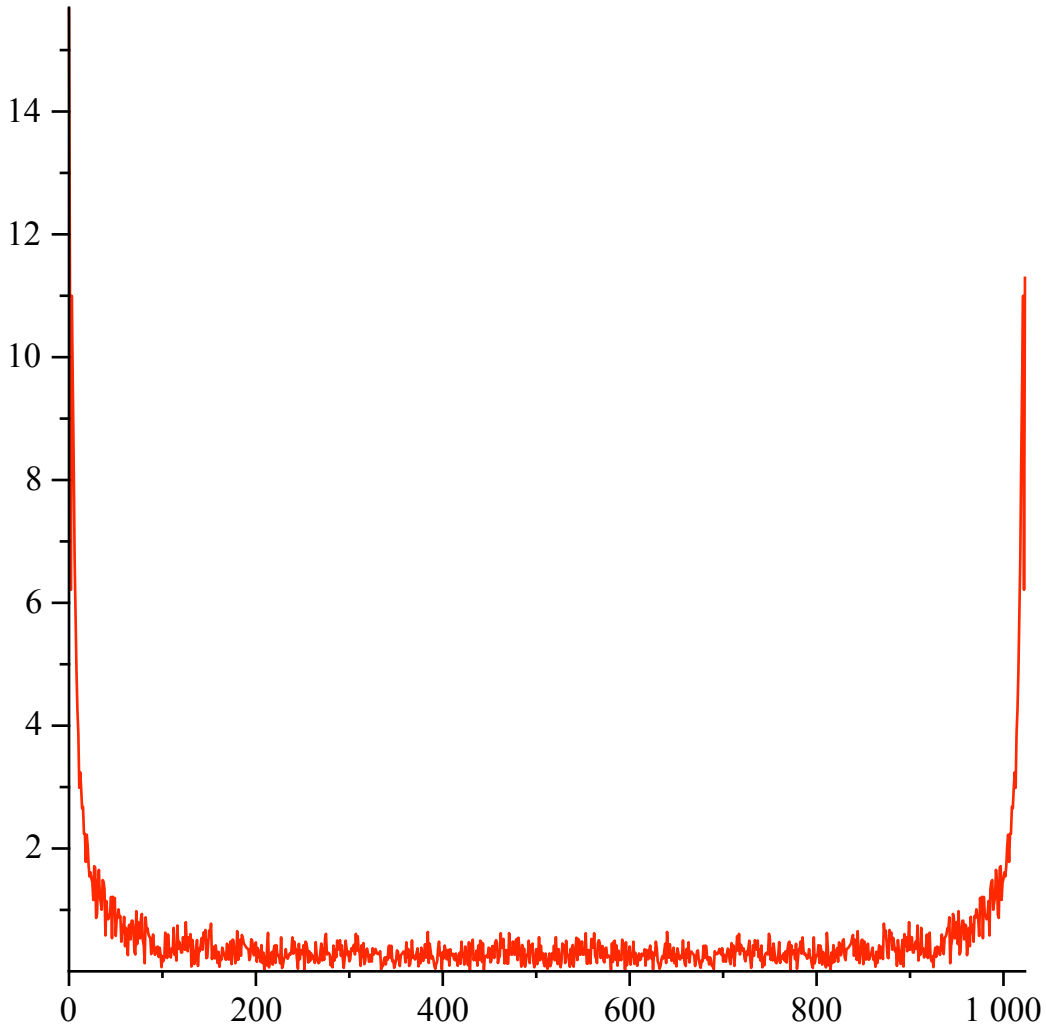

```
> h:=evalf(rand(-500..500)/1000):
> Y:=[seq([t[i],h()],i=1..1024)]:
> Z:=[seq([t[i],X[i][2]+Y[i][2]],i=1..1024)]:
> plot(Y);
```



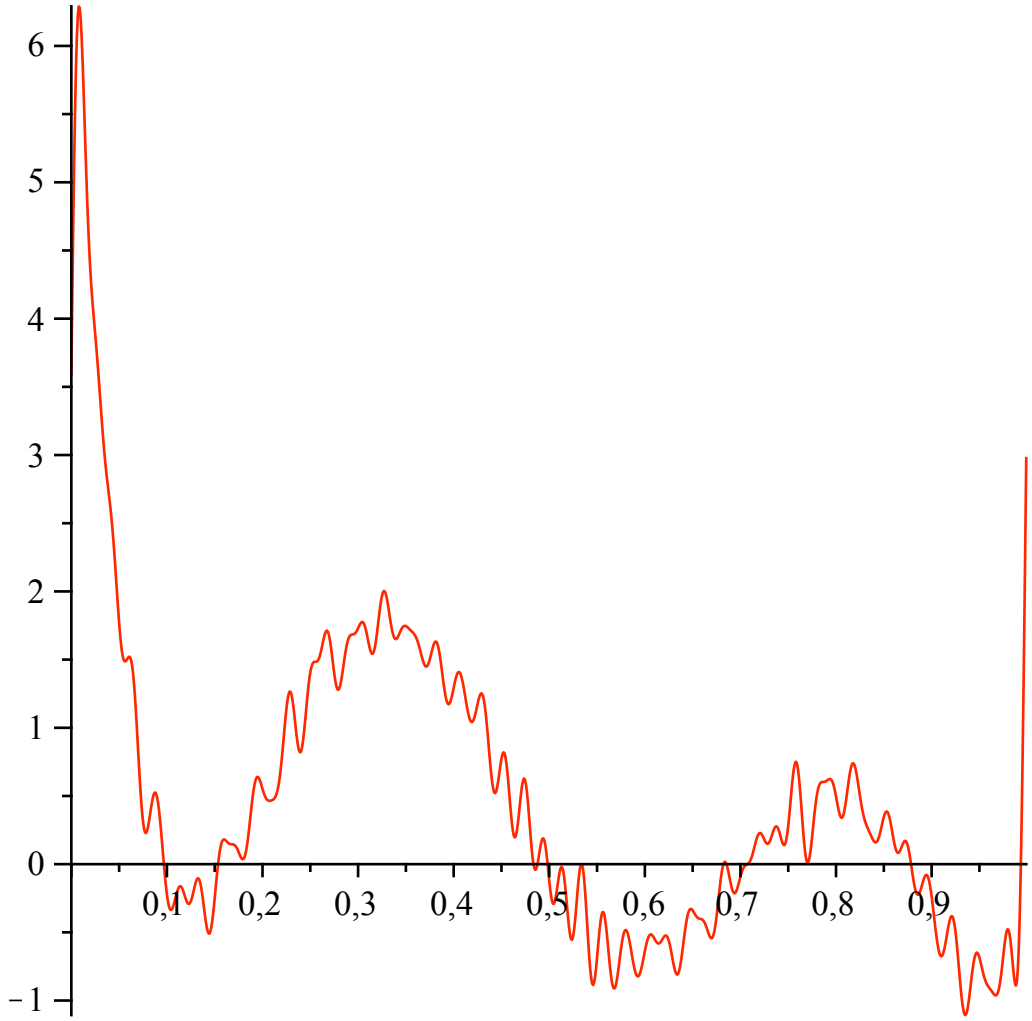

```
> plot(Z);
```



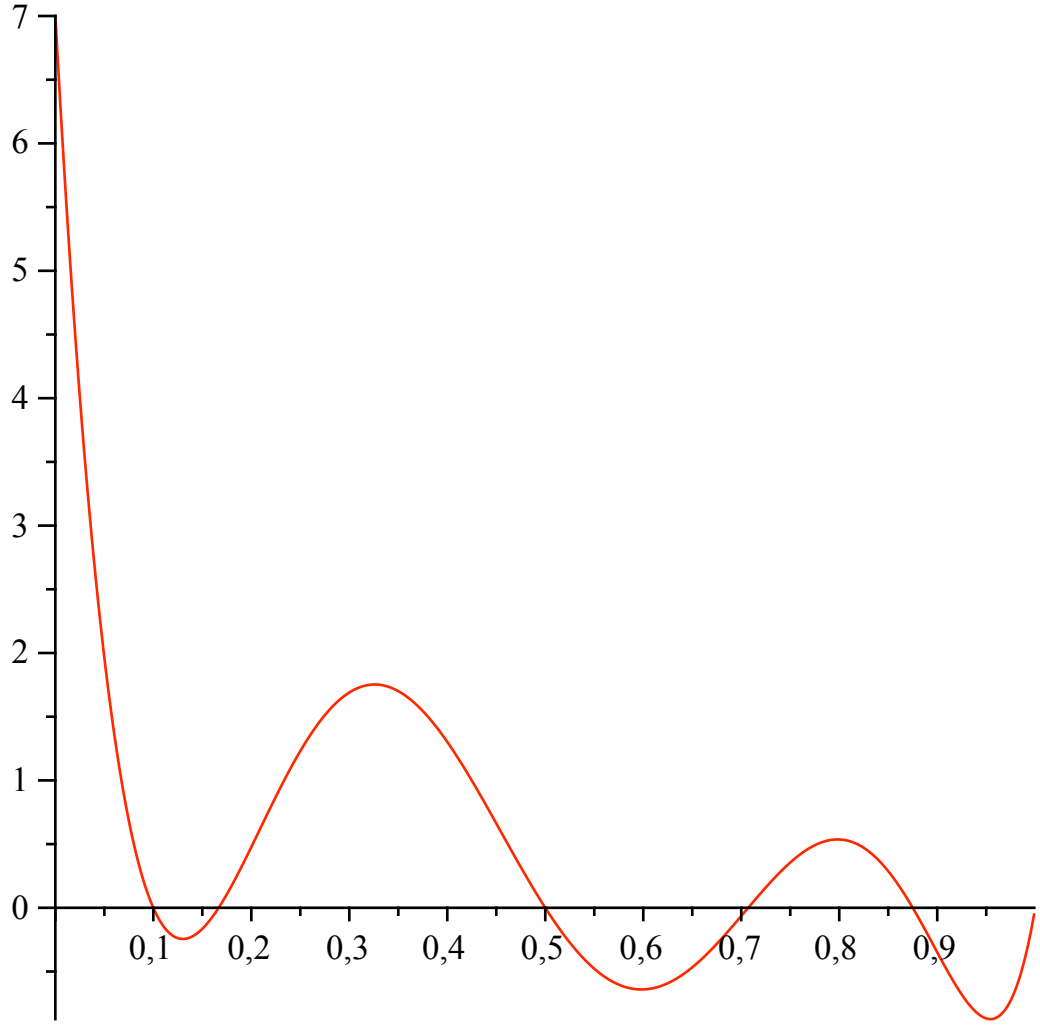

```
> ZZ:=[seq(Z[i][2],i=1..1024)]:
> with(DiscreteTransforms):
> ZZ:=convert(ZZ,Vector):
> F:=FourierTransform(ZZ):
> FF:=[seq([i-1,abs(F[i])],i=1..1024)]:
> plot(FF);
```




```
> FS:=Vector(1024):
> for i from 1 to 1024 do if abs(F[i])>1 then FS[i]:=F[i] else
  FS[i]=0 fi od;
> S:=InverseFourierTransform(FS):
> SS:=[seq([t[i],Re(S[i])],i=1..1024)]:
> plot(SS);
```




```
> plot(X);
```



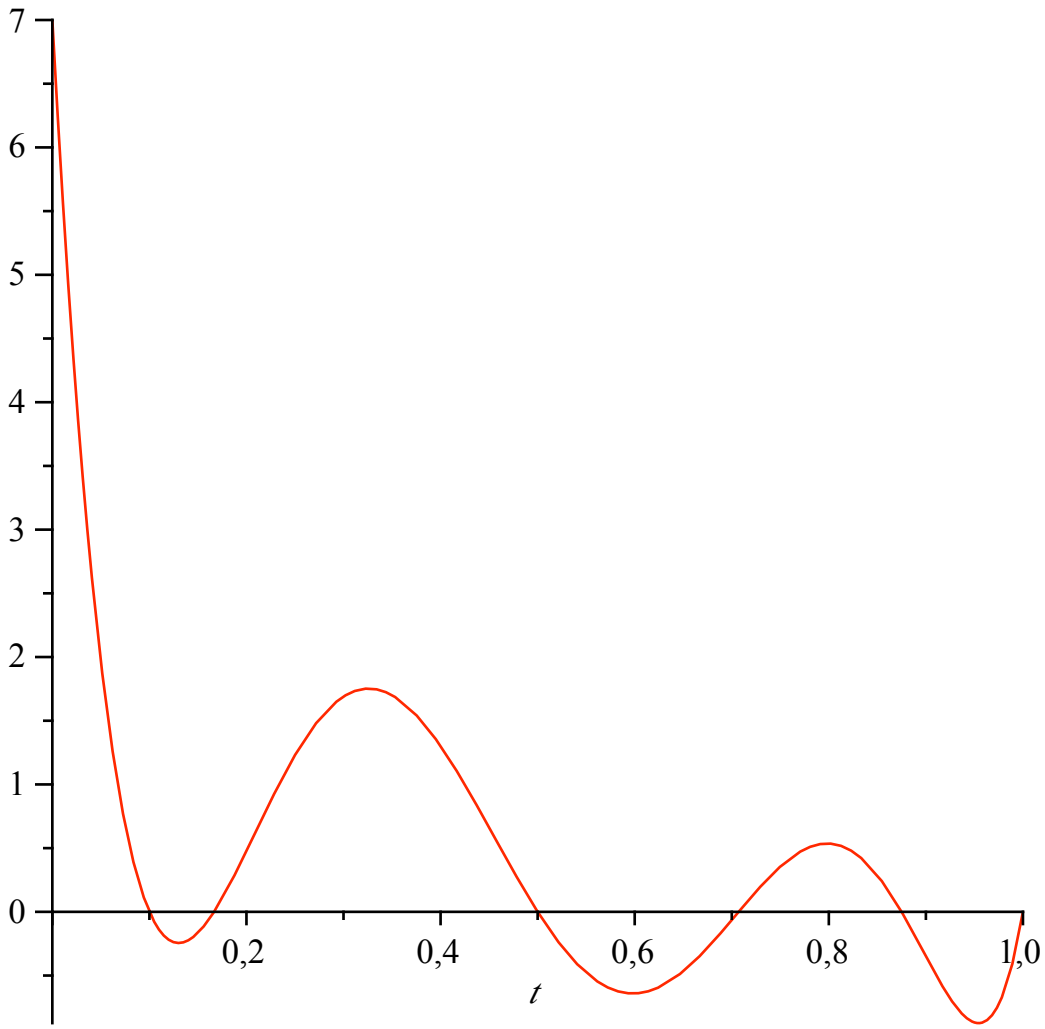
>

Exemple 3 (autre méthode)

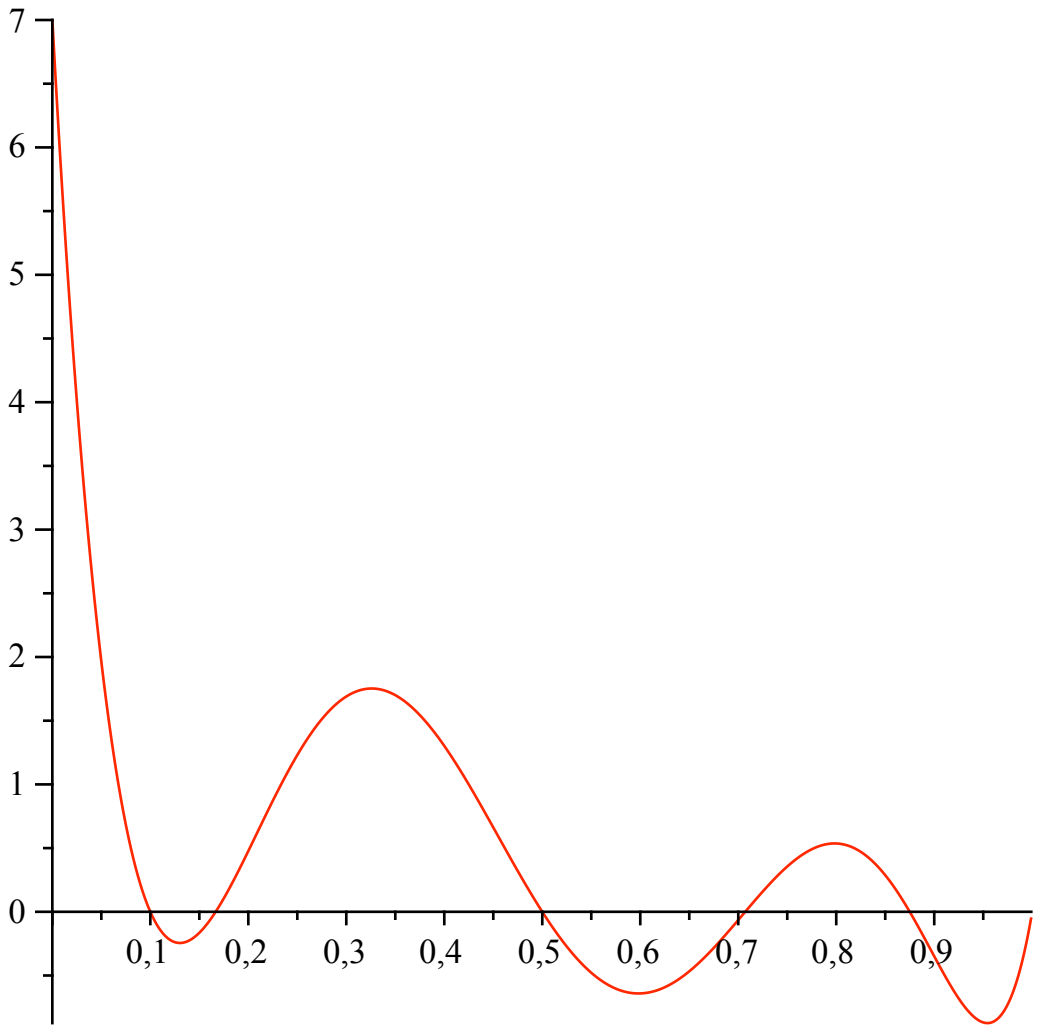
```
> restart;
> X:=expand(1920*(t-1/6)*(t^2-1/2)*(t-1/2)*(t-7/8)*(t-1/10)*
  (t-1));
```

$$X := 1920\,t^7 - 5072\,t^6 + 3768\,t^5 + 692\,t^4 - 2082\,t^3 + 908\,t^2 - 141\,t + 7 \qquad \textbf{(6.1)}$$

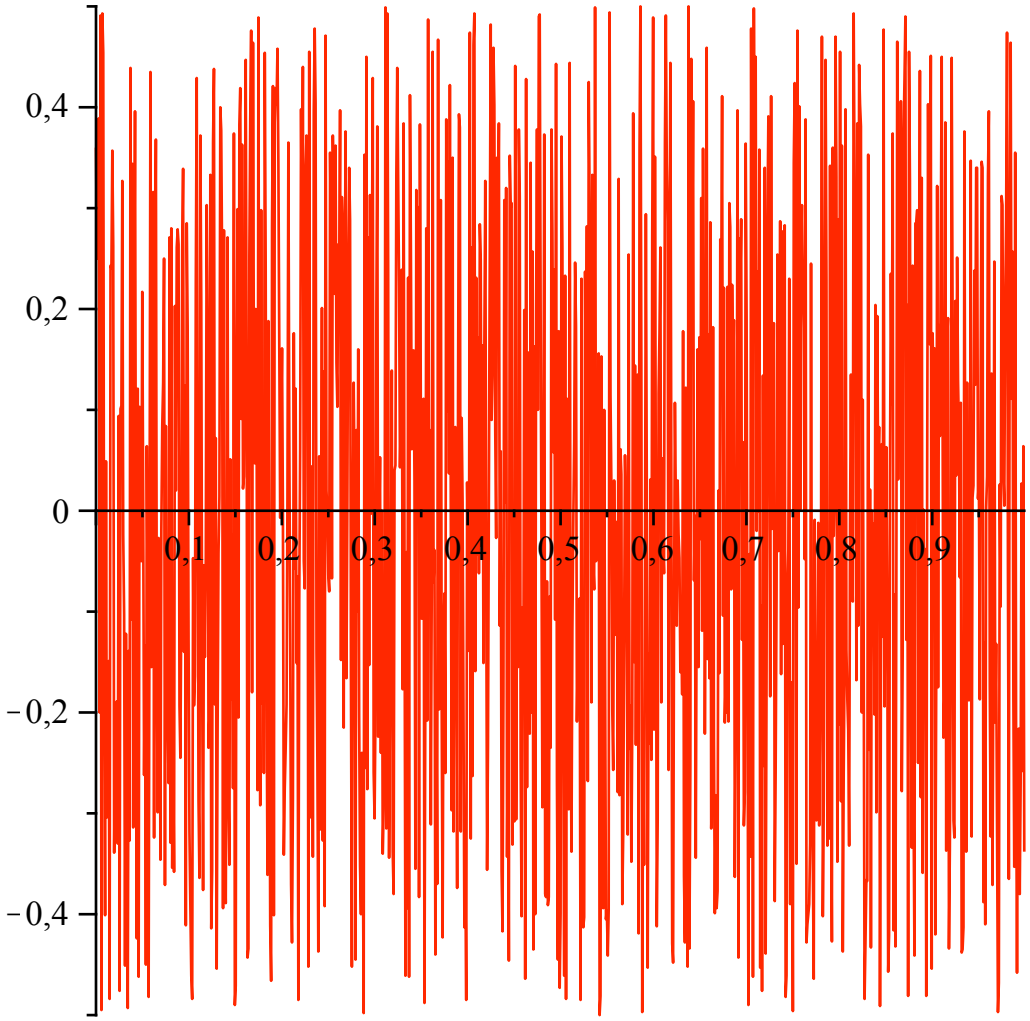
```
> plot(X,t=0..1);
```



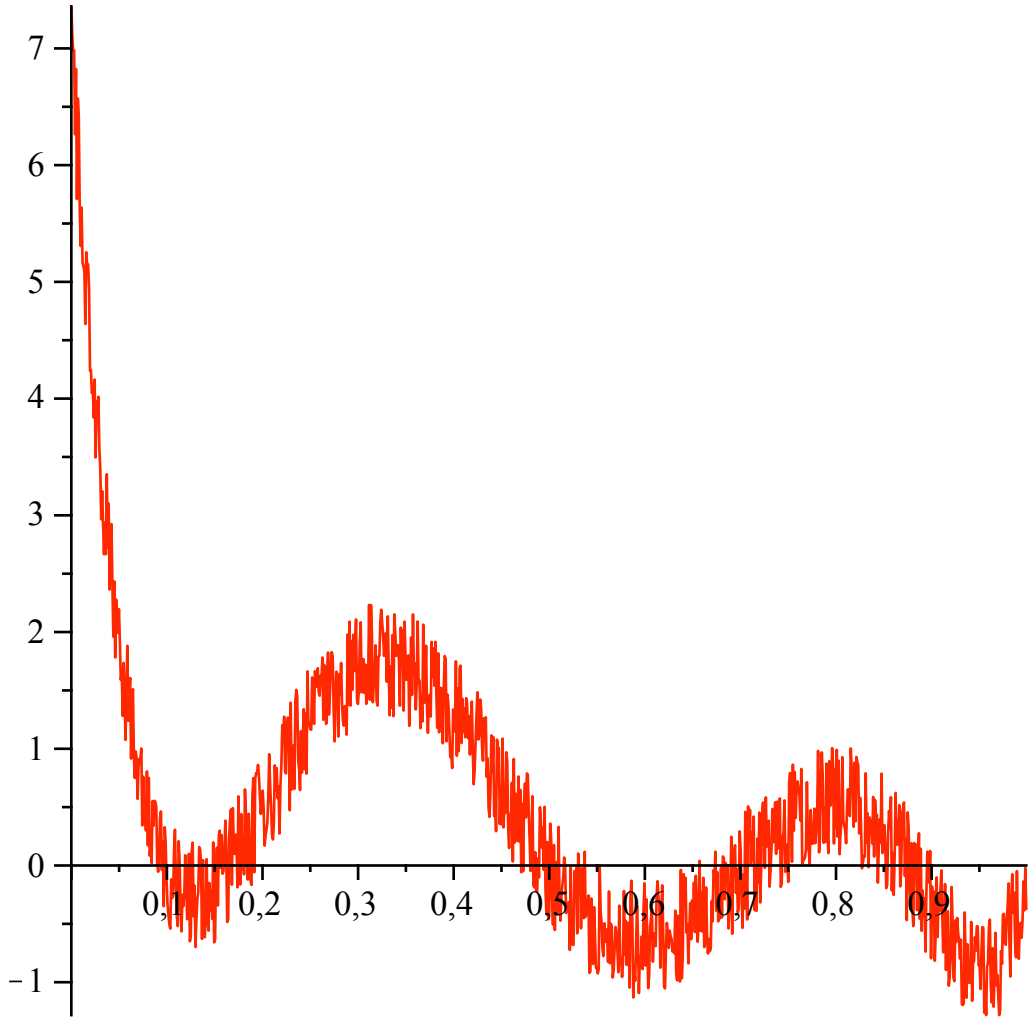

```
> t:=[evalf(($ 0..1023)/1024)]:
> X:=[seq([t[i],evalf(1920*(t[i]-1/6)*(t[i]^2-1/2)*(t[i]-1/2)*
  (t[i]-7/8)*(t[i]-1/10)*(t[i]-1))],i=1..1024)]:
> plot(X);
```



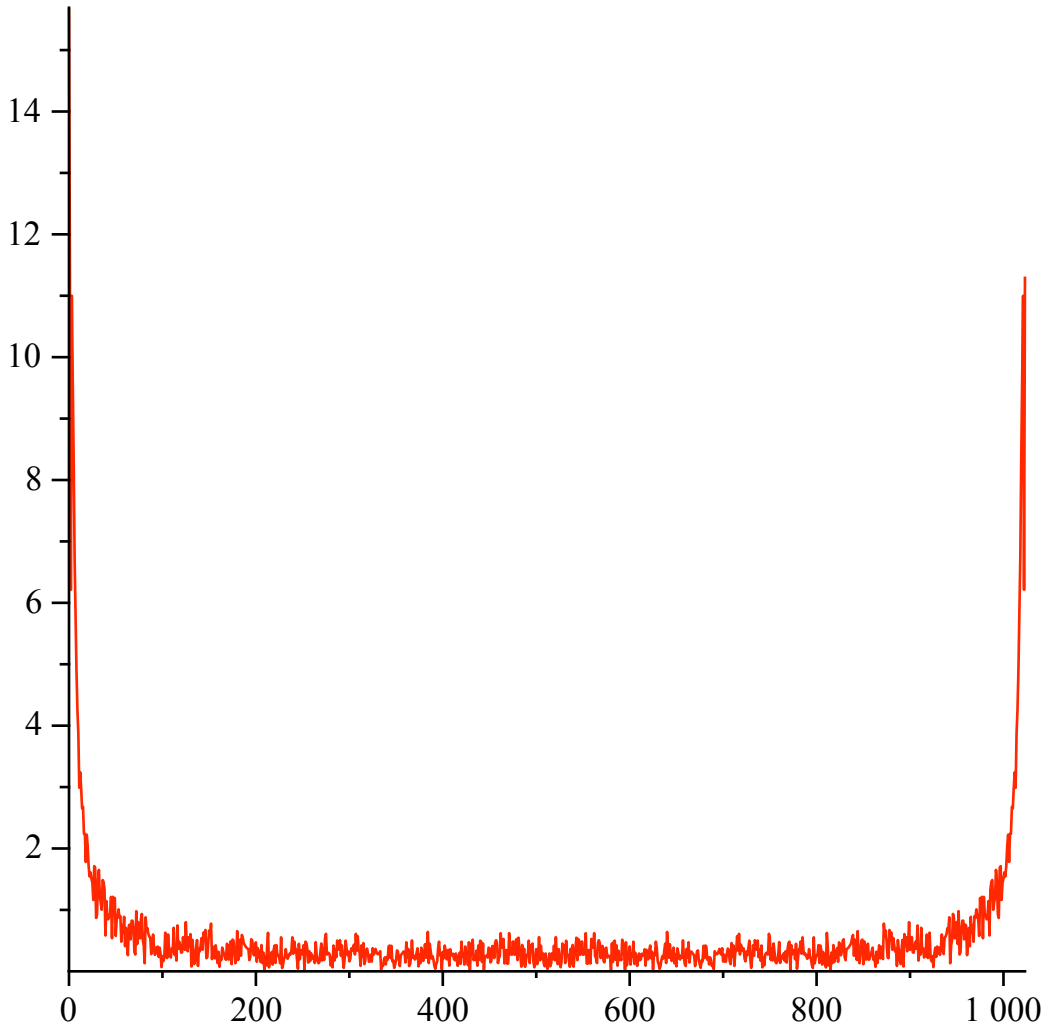

```
> h:=evalf(rand(-500..500)/1000):
> Y:=[seq([t[i],h()],i=1..1024)]:
> Z:=[seq([t[i],X[i][2]+Y[i][2]],i=1..1024)]:
> plot(Y);
```



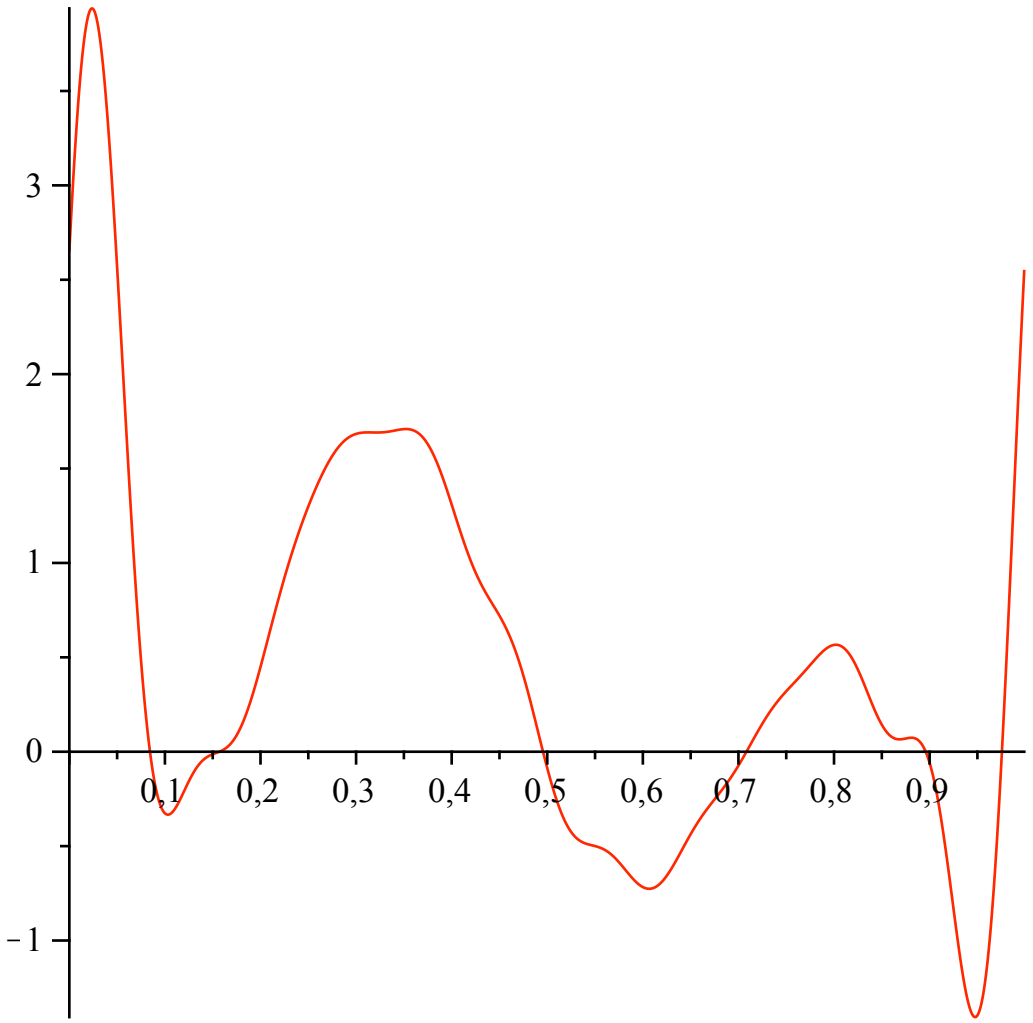

```
> plot(Z);
```



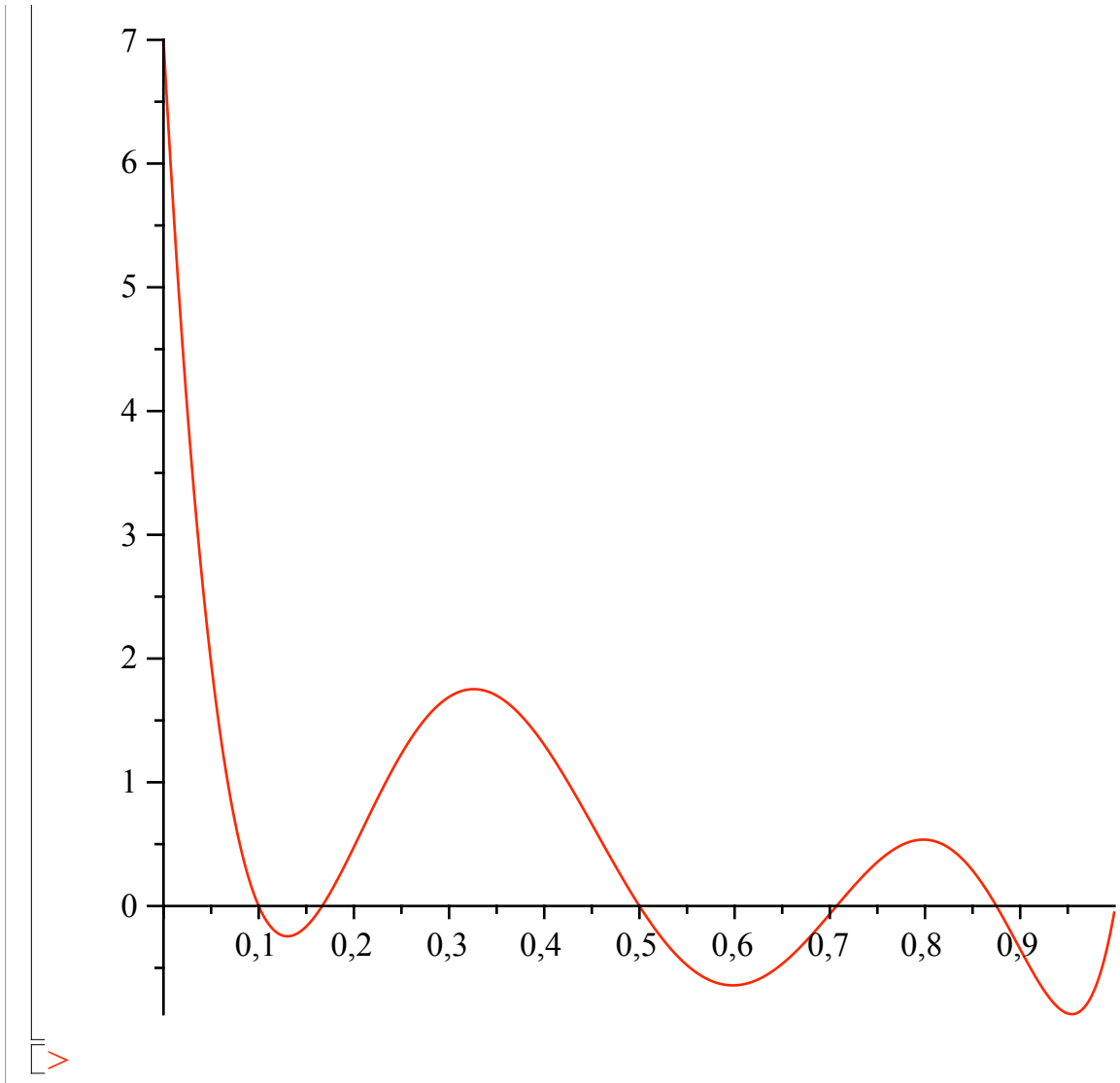

```
> ZZ:=[seq(Z[i][2],i=1..1024)]:
> with(DiscreteTransforms):
> ZZ:=convert(ZZ,Vector):
> F:=FourierTransform(ZZ):
> FF:=[seq([i-1,abs(F[i])],i=1..1024)]:
> plot(FF);
```




```
> FS:=Vector(1024):
> for i from 1 to 10 do FS[i]:=F[i] od:
  for i from 11 to 1014 do FS[i]:=0 od:
  for i from 1014 to 1024 do FS[i]:=F[i] od:
> S:=InverseFourierTransform(FS):
> SS:=[seq([t[i],Re(S[i])],i=1..1024)]:
> plot(SS);
```




```
> plot(X);
```


7
6
5
4
3
2
1
0
0,1
0,2
0,3
0,4
0,5
0,6
0,7
0,8
0,9