TD 1 : interpolation de Lagrange et phénomène de Runge

```
> restart;
  produit:=proc(l)
  # calcule le produit des éléments de la liste l

    local i, total;
    total:=1;
    for i from 1 to nops(l)
    do
      total:=total*l[i];
    od;
    total;
    end:
  #remarque : si l'on ne voulait pas programmer, on pouvait faire :
  produit2:=l -> product(l[k], k=1..nops(l)):
> interpole:=proc(lx, ly)
  # calcule le polynôme d'interpolation des points (lx(k), ly(k))

    local K, L, res, i, Q;
    res:=0; Q:=produit(map(x->X-x, lx));
    for i from 1 to nops(lx)
    do
      K:=Q/(X-lx[i]);
      L:=ly[i]*K/subs(X=lx[i], K);
      res:=res+L;
    od;
    expand(res);
  end:
> x_uniforme:=proc(x_min, x_max, N)
  # retourne N+1 points uniforméments répartis entre x_min et x_max
    local k, step;
    step:=(x_max-x_min)/(N);
    [seq(x_min+k*step,  k=0..N)];
  end:
> x_chebishev:=proc(x_min, x_max, N)
  # retourne N+1 points entre x_min et x_max selon la distribution de
  Chebishev
    local k, step;
    [seq(evalf((x_min+x_max)/2+(x_max-x_min)/2*cos(k*Pi/N)), k=0..N)];
  end:

> alpha:=1/4:
  runge:=x->1/(alpha^2+x^2):
> N:=14:
  lx:=x_uniforme(-1,1,N):
  ly:=map(runge, lx):
  Q:=interpole(lx, ly):
  plot([Q, runge(X)], X=-1 .. 1);

> alpha:=1/4;
  lx:=x_chebishev(-1,1,N):
  ly:=map(runge, lx):
  Q:=interpole(lx, ly):
  plot([Q, runge(X)], X=-1 .. 1);
  alpha:=1/10;
```

```
ly:=map(runge, lx):
Q:=interpole(lx, ly):
plot([Q, runge(X)], X=-1 .. 1);
```

$$\alpha := \frac{1}{4}$$

$$\alpha := \frac{1}{10}$$

```
> alpha:=1/4:
legendre:=n->orthopoly[P](n, x)/sqrt(int(orthopoly[P](n, x)^2, x=-1..1)):
Q:=sum('int(runge(x)*legendre(n), x=-1..1)*legendre(n)', n=0..20):
plot([Q, runge(x)], x=-1..1);
```

```
> w:=x->(1-x^2)^(-1/2);
cheb:=n->orthopoly[T](n, x)/sqrt(evalf(int(orthopoly[T](n, x)^2*w(x),
x=-1..1)));
Q:=sum('evalf(int(runge(x)*cheb(n), x=-1..1))*cheb(n)', n=0..10);
plot([Q, runge(x)], x=-1..1);
```

$Q :=$

$$1015.984969\, x^8 - 350.7240509\, x^{10} + 559.6987014\, x^4 - 135.6029214\, x^2 - 1104.511038\, x^6 + 14.95292091$$

```
> with(numapprox): with(orthopoly):
alpha:=1/4: Q:=chebyshev(runge(x), x=-1..1, .1):degree(Q, x);
alpha:=1/10: Q:=chebyshev(runge(x), x=-1..1, .1): degree(Q, x);
```

$$14$$

$$38$$

```
> nevill_rat:=proc(x, y)
local n, R, m, i;
n:=nops(x);
R:=array(-1..n-1, 1..n);
for i from 1 to n do R[-1, i]:=0; R[0,i]:=y[i]; od;
for m from 1 to n-1 do
  for i from 1 to n-m do
    R[m,i]:=R[m-1, i+1]+(R[m-1, i+1]-R[m-1,
i])/((X-x[i])/(X-x[i+m])*(1-(R[m-1, i+1]-R[m-1, i])/(R[m-1, i+1]-R[m-2,
i+1]))-1);
  od;
od;
R[n-1, 1];
end:
```

```
> N:=4:
x:=x_uniforme(-1,1,N):
y:=map(runge, x):
Q:=nevill_rat(x, y):
'Q'=simplify(Q);
```

$$Q = \frac{16}{16\, X^2 + 1}$$

```
>
```