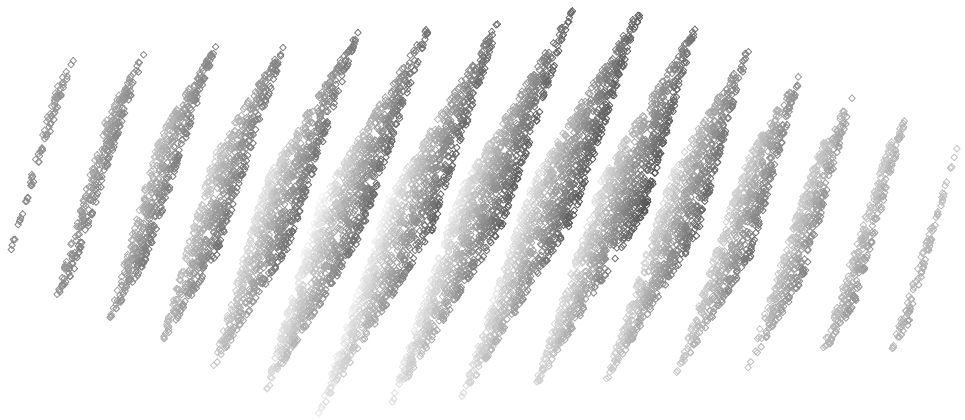
Générateurs pseudo-aléatoires et tests statistiques

```
> restart;
  gen_a:=n->[evalf(i*exp(1)-floor(i*exp(1))) $ i=1..n]:
  gen_b:=proc(n) local phi; phi:=(1+sqrt(5))/2;
  [evalf(i*phi-floor(i*phi)) $ i=1..n];end:
  gen_c:=n->evalf([log(i)-floor(log(i)) $ i=1..n]):
  gen_l:=proc(n) local i,L; L:=[342636321]; for i from 2 to n do
  L:=[op(L), (65539*L[i-1]) mod 2^31]; od; evalf(L/2^31);end:
  gen_m:=proc(n) local i,M; M:=[342636321]; for i from 2 to n do
  M:=[op(M), (16807*M[i-1]) mod (2^31-1)]; od;
  evalf(M/(2^31-1));end:
  gen_p:=proc(n) local i,M; M:=2^30; [seq(evalf(rand(M)()/M),
  i=1..n)];end:
  gen_q:=n->map(x->evalf(sin(Pi/2*x)), gen_p(n)):
```

```
> with(plots): N:=10000: L:=l(N): pts:={seq([L[i], L[i+1],
  L[i+2]], i=1..N-2)}: pointplot3d(pts, symbol=diamond);
Warning, the name changecoords has been redefined
```



```
> Phi:=r-> int(1/sqrt(2*Pi)*exp(-t^2/2), t=-r..r):
  quant:=proc(alpha) option remember; fsolve(Phi('r')-(1-alpha),
  'r'=1): end:
  n:=500: p:=0.2: eps:=0.03: u:=eps*sqrt(n/(p*(1-p))):
  P(abs(F-p)>eps) = evalf(sum(binomial(n, k) * p^k*(1-p)^(n-k),
  k=0..ceil(n*(p-eps)-1)) + sum(binomial(n, k) * p^k*(1-p)^(n-k),
  k=floor(n*(p+eps)+1)..n) ); # calcul exact
  P(abs(F-p)>eps) = 1-evalf(Phi(u)); # avec le TCL
```

$$P(.03 < |F - .2|) = .08282688717$$
$$P(.03 < |F - .2|) = .0935325127$$

```
> P(abs(F-p)>eps) <= p*(1-p)/(n*eps^2); # inégalité de
  Bienaymé-Chebyshev
  P(abs(F-p)>eps) <= 2*exp(-2*n*eps^2); # Hoeffding
```

$$P(.03 < |F - .2|) \le .3555555556$$
$$P(.03 < |F - .2|) \le .8131393194$$

```
> tests_proba:=proc(gen, a, b, n, alpha)
  local i, X, L, F, p, eps1, eps2, eps3;
    X:=gen(n); L:=map(x-> if (x>=a) and (x<=b) then 1 else 0 fi,
  X);
    F:=sum(L[i], i=1..n)/n; #print('F'=F);
    p:=b-a;
    eps1 := sqrt(p*(1-p)/(n*alpha)); # avec l'inégalité de
  Bienaymé-Chebyshev
    eps2 := sqrt(-log(alpha)/(2*n)); # avec l'inégalité de
  Hoeffding
    eps3 := quant(alpha)*sqrt(p*(1-p)/n); # avec le TCL
    ['BC', 'Hoef', 'TCL']=map(evalb, [abs(F-p) <= eps1, abs(F-p)
  <= eps2, abs(F-p)<= eps3])
  end:
```

```
> tests_proba(gen_q, 0.4, 0.6, 100, 0.10);
```

$$[BC, Hoef, TCL] = [true, true, false]$$

```
> estimerisque_proba:=proc(N)
  local res, i, r, j;
  res:=[0, 0, 0]:
  for i from 1 to N do
    r:=rhs(tests_proba(gen_p, 0.4, 0.6, 100, 0.05)):
    for j from 1 to 3 do if not(r[j]) then res[j]:=res[j]+1: fi;
  od;
  od;
  ['risque_BC', 'risque_Hoef', 'risque_TCL']=map(x->evalf(x/N),
  res);
  end:
  estimerisque_proba(100);
```

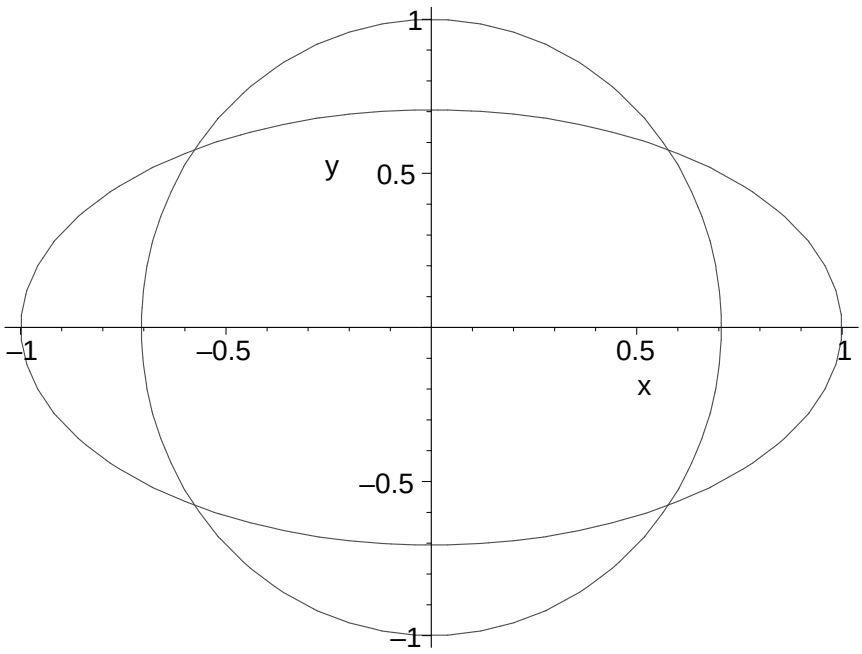
$$[risque_BC, risque_Hoef, risque_TCL] = [0., 0., .05000000000]$$

```
> IC_proba:=proc(gen, a, b, n, alpha)
  local i, X, L, F, p, eps1, eps2, eps3;
    X:=gen(n); L:=map(x-> if (x>=a) and (x<=b) then 1 else 0 fi,
  X);
    F:=sum(L[i], i=1..n)/n; # print('F'=F);
    p:=b-a;
    eps1 := sqrt(1/(4*n*alpha)); # avec l'inégalité de
  Bienaymé-Chebyshev
    eps2 := sqrt(-log(alpha)/(2*n)); # avec l'inégalité de
  Hoeffding
    eps3 := quant(alpha)*sqrt(F*(1.-F)/n); # avec le TCL
    'I_C'= [F-eps1, F+eps1], 'I_H'= [F-eps2, F+eps2],'I_a'=
  [F-eps3, F+eps3];
   end:
  IC_proba(gen_q, 0.4, 0.6, 1000, 0.05);
```

$$I_C = [.08028932188, .2217106781],\ I_H = [.1122977244, .1897022756],$$
$$I_a = [.1288083083, .1731916917]$$

```
> E1:=x^2+2*y^2-1: E2:=2*x^2+y^2-1: with(plots): implicitplot({E1,
  E2}, x=-1..1, y=-1..1);
  IC_aire:=proc(gen, prec, alpha)
  local n, X, Y, L, i, F; global E1, E2;
    n:=min(ceil(4*log(1/alpha)/(2*prec^2)),
  ceil(4*1/(4*alpha*prec^2))); # on prend la meilleure inégalité
```

```
  X:=map(x->2*x-1, gen_p(n));  Y:=map(x->2*x-1, gen_p(n));
  L:=0; for i from 1 to n do if subs({x=X[i], y=Y[i]}, E1)<0 and
subs({x=X[i], y=Y[i]}, E1)<0 then L:=L+1 fi; od;
  F:=L/n; # print('F'=F);
  'I_C'= [4*F-prec, 4*F+prec];
 end:
IC_aire(gen_q, 0.02, 0.05);
```



$$I_C = [2.207385006, 2.247385006]$$

```
> tests_moyenne:=proc(gen, n, alpha)
local i, X, L, F, p, eps1, eps2, eps3;
  X:=gen(n);
  F:=sum(X[i], i=1..n)/n; # print('F'=F);
  p:=b-a;
  eps1 := sqrt(1/(12.*n*alpha)); # avec l'inégalité de
Bienaymé-Chebyshev
  eps2 := sqrt(-log(alpha)/(2*n)); # avec l'inégalité de
Hoeffding
  eps3 := quant(alpha)*sqrt(1/(12.*n)); # avec le TCL
  ['BC', 'Hoef', 'TCL']=map(evalb, [abs(F-1/2) <= eps1,
abs(F-1/2) <= eps2, abs(F-1/2)<= eps3])
end:
> tests_moyenne(gen_q, 1000, 0.05);
```

$$[BC, Hoef, TCL] = [false, false, false]$$

```
> estimerisque_moyenne:=proc(N)
local res, i, r, j;
res:=[0, 0, 0]:
for i from 1 to N do
  r:=rhs(tests_moyenne(gen_p, 100, 0.05)):
  for j from 1 to 3 do if not(r[j]) then res[j]:=res[j]+1: fi;
```

```
  od;
  od;
  ['risque_BC', 'risque_Hoef', 'risque_TCL']=map(x->evalf(x/N),
  res);
  end:
  estimerisque_moyenne(1000);
```

$$[risque_BC, risque_Hoef, risque_TCL] = [0., 0., .05100000000]$$

```
> test_chi2:=proc(gen, K, n, alpha)
  local i, j, X, N, distchi2;
    X:=gen(n); N:=[0 $ 'i'=1..K];
    for j from 1 to n do i:=1+floor(K*X[j]); N[i]:=N[i]+1; od;
    distchi2:=evalf(sum((N['i']-n/K)^2/(n/K), 'i'=1..K));
    (distchi2<16.92) = evalb(distchi2<16.92);
    end:
> test_chi2(gen_p, 10, 100, 0.05);
```

$$(4. < 16.92) = true$$

```
> estimerisque_chi2:=proc(N)
  local res, i, r, j;
  res:=0:
  for i from 1 to N do
    r:=rhs(test_chi2(gen_p, 10, 100, 0.05)):
    if not(r) then res:=res+1: fi;
  od;
  'risque_chi2' = evalf(res/N);
  end:
  estimerisque_chi2(1000);
```

$$risque_chi2 = .05000000000$$

```
> ?plot
>
```