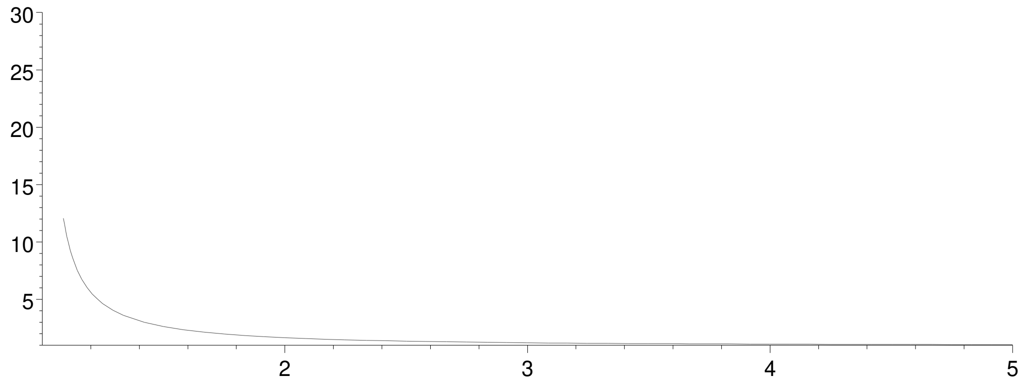
<u>TD 4 : Séries numériques, Algorithmique</u>

Autour de la fonction Zéta de Riemann

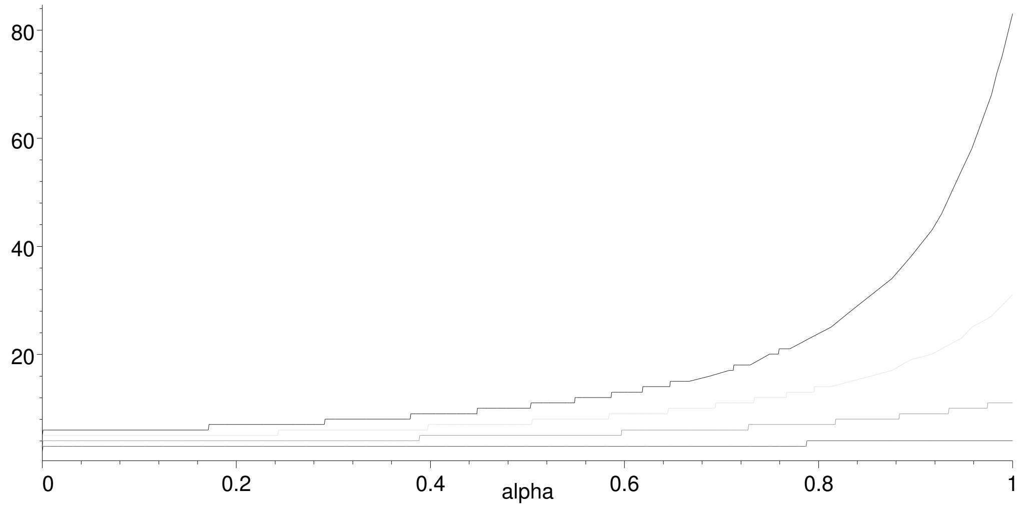
```
> Sum(1/k^2, k=1..infinity)=sum(1/k^2, k=1..infinity);
```

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{1}{6}\pi^2$$

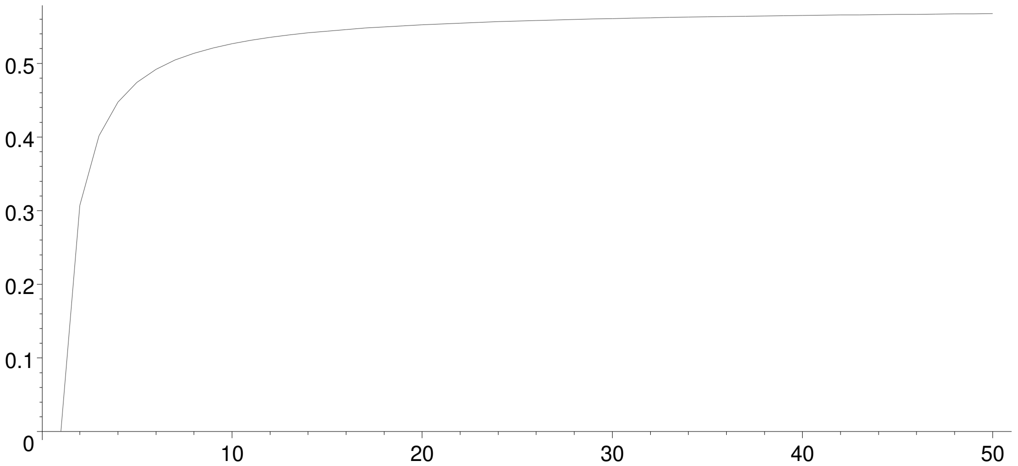
```
> plot(Zeta, 1..5, 1..30);
```



```
> nombre:=proc(A:: realcons, alpha:: realcons)
    local n, s;
    if (evalf(alpha<0)) or (evalf(alpha)>1) then ERROR(`alpha
  doit être compris entre 0 et 1 !`); fi;
    n:=1; s:=0;
    while s<evalf(A) do
      s:=s+evalf(1/n^alpha);
      n:=n+1;
    od;
    n-1;
  end:
> plot([seq('nombre'(A, alpha),A=2..5)], alpha=0..1);
```



```
> plot([[n, sum(1/k, k=1..n-1)-ln(n)] $n=1..50]);
  asympt(sum(1/k, k=1..n-1)-ln(n), n, 3);
```



$$\gamma - \frac{1}{2}\frac{1}{n} - \frac{1}{12}\frac{1}{n^2} + O\left(\frac{1}{n^4}\right)$$

Différentes formes de séries

```
> S:=sum(ln(k+n), k=1..n);
  'S'=expand(asympt(S, n, 3));
```

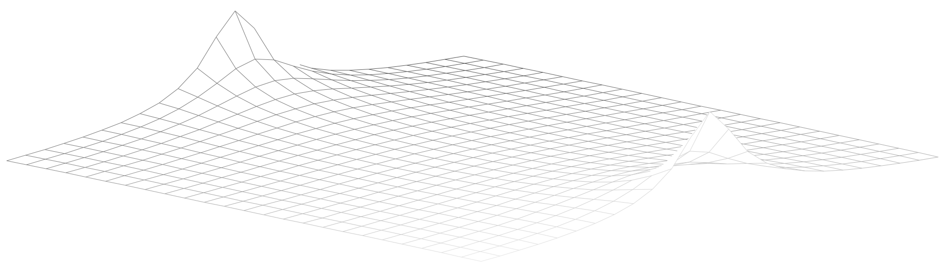
$$S := \ln(\Gamma(2\,n+1)) - \ln(\Gamma(n+1))$$

$$S = n\ln(n) - n + n\ln(4) + \frac{1}{2}\ln(2) + O\left(\frac{1}{n}\right)$$

```
> S1:=sum(z^k, k=0..infinity);
  S2:=sum((-1)^k*z^(2*k), k=0..infinity);
  plot3d((abs(1/(1+(x+I*y)^2))), x=-0.9..0.9, y=-0.9..0.9);
```

$$S1 := -\frac{1}{z-1}$$

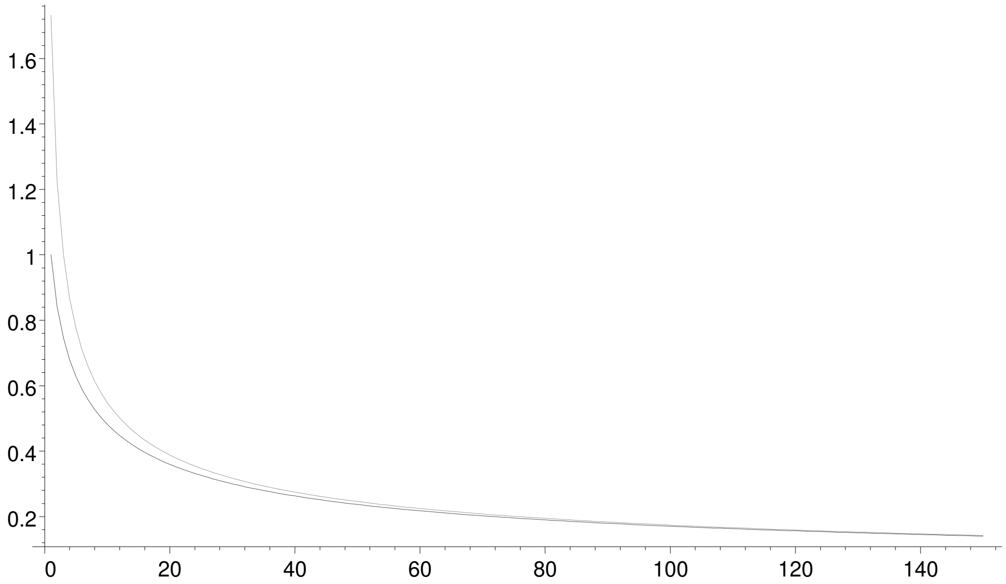
$$S2 := \frac{1}{z^2+1}$$



```
> u0:=Pi/2;
  u:=proc(n::integer)
  option remember;
  if(n=0) then u0; else sin(u(n-1)); fi;
  end:
```

$$u0 := \frac{1}{2}\pi$$

```
> plot([[[n, 'u(n)']$ n=1..150], [[n, sqrt(3/n)] $ n=1..150]]);
```




```
> with(linalg):
  P:=matrix(3, 3, [1, 1, 0, 1, -1, 0, 0, 0, 1]):
  Di:=matrix(3, 3, [1, 0, 0, 0, 2, 0, 0, 0, 3]):
  A:=evalm(P &* Di &* inverse(P));
  Sum((A &^ k)/k!, k=0..infinity)=exponential(A);
  'exponential(A)'='P' &* 'exponential(Di)' &* 'inverse'(P);
Warning, new definition for norm
Warning, new definition for trace
```

$$A := \begin{bmatrix} \frac{3}{2} & \frac{-1}{2} & 0 \\ \frac{-1}{2} & \frac{3}{2} & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

$$\sum_{k=0}^{\infty} \frac{A \mathbin{\&\hat{}} k}{k!} = \begin{bmatrix} \frac{1}{2}\mathbf{e} + \frac{1}{2}\mathbf{e}^2 & -\frac{1}{2}\mathbf{e}^2 + \frac{1}{2}\mathbf{e} & 0 \\ -\frac{1}{2}\mathbf{e}^2 + \frac{1}{2}\mathbf{e} & \frac{1}{2}\mathbf{e} + \frac{1}{2}\mathbf{e}^2 & 0 \\ 0 & 0 & \mathbf{e}^3 \end{bmatrix}$$

$$\mathrm{exponential}(A) = (P \mathbin{\&*} \mathrm{exponential}(Di)) \mathbin{\&*} \mathrm{inverse}(P)$$

Algorithmique : premiers algorithmes de tri

```
> maxi:=proc(l)
    local k, kmax, lmax;
    kmax:=1; lmax:=l[1];
    for k from 2 to nops(l) do
      if l[k]>lmax then
        kmax:=k; lmax:=l[k];
      fi;
    od;
    lmax, [seq(l[k], k=1..kmax-1), seq(l[k],
  k=kmax+1..nops(l))];
  end:
> tri_insertion:=proc(l)
    local k, ltemp, res, X;
    res:=[]; ltemp:=l;
    for k from 1 to nops(l) do
      X:=maxi(ltemp);
      ltemp:=X[2];
      res:=[X[1], op(res)];
    od;
    res;
  end:
> fusion:=proc(l1, l2)
    local la, lb, res;
    res:=[]; la:=l1; lb:=l2;
    while nops(la)>0 and nops(lb)>0 do
      if la[1]<lb[1] then
        res:=[op(res), la[1]]; la:=[la[k] $ k=2..nops(la)];
      else
        res:=[op(res), lb[1]]; lb:=[lb[k] $ k=2..nops(lb)];
      fi
    od;
    [op(res), op(la), op(lb)];
  end:
> tri_fusion:=proc(l)
    local m, l1, l2;
    if nops(l)<=50 then tri_insertion(l) else
      m:=floor(nops(l)/2);
      l1:=[l[k] $ k=1..m];
      l2:=[l[k] $ k=m+1..nops(l)];
      l1:=tri_fusion(l1); l2:=tri_fusion(l2);
      fusion(l1, l2);
    fi;
  end:
```