

poly

Outils divers sur les polynômes

Calculatrices : 89 92 92+ V200

Niveaux : Tale S Sup

Descriptif : Groupe de programmes - Calculs sur des polynômes : relation de Bezout, degré, division euclidienne, liste des coefficients par ordre croissant/décroissant, zéros et leur ordre de multiplicité, décomposition en éléments simples sur **C** d'une fraction à coefficients réels

Auteur : Claude Morin

Mots-clefs : polynômes, division euclidienne, Bezout, zéros, éléments simples

Date de dernière révision : Septembre 2002

Présentation :

Il s'agit d'un ensemble de fonctions et programmes consacrés aux polynômes :

- *bezout* : renvoie u , v et d tels que pour deux polynômes a et b la relation de Bezout $a.u + b.v = d = \text{PGCD}(a, b)$ soit satisfaite. Nécessite *div* et *pcoefd*
- *deg* : renvoie le degré d'un polynôme, -1 s'il est nul
- *desc3* : décompose en éléments simples sur **C** une fraction rationnelle à coefficients réels. Nécessite *mczeros*, *div*, *pcoefd*
- *div* : effectue la division euclidienne d'un polynôme a par un polynôme b . Nécessite *pcoefd*
- *mczeros* : renvoie une matrice dont les colonnes sont les zéros d'un polynôme avec leur ordre de multiplicité
- *pcoefcn* : renvoie la liste des coefficients d'un polynôme par ordre croissant
- *pcoefd* : renvoie la liste des coefficients d'un polynôme par ordre décroissant

Mode d'emploi :

Pour utiliser ces fonctions, placez-vous d'abord dans le répertoire dans lequel vous les avez copiés.

Puis les fonctions s'utilisent très simplement :

- *bezout* : il suffit de taper *bezout* (a , b , x), où a et b sont les deux polynômes et x leur variable
- *deg* : il suffit de taper *deg* (a , x) où a est le polynôme et x sa variable
- *desc3* : il suffit de taper *desc3* (f , x) où f est la fraction rationnelle et x sa variable
- *div* : il suffit de taper *div* (a , b , x) où a et b sont les polynômes et x leur variable
- *mczeros* : il suffit de taper *mczeros* (a , x) où a est le polynôme et x sa variable
- *pcoefcn* : il suffit de taper *pcoefcn* (a , x) où a est le polynôme et x sa variable
- *pcoefd* : il suffit de taper *pcoefd* (a , x) où a est le polynôme et x sa variable

Sources :

Bezout (a,b,x)

Func

© entrer 2 polynômes a et b et leur variable;renvoie {u,v,d} tel que $a*u+b*v=d=\text{pgcd}(a,b)$;utilise div,pcoefd

Local t

{1,0,a}→a:{0,1,b}→b

While string(b[3])≠"0"

b→t

expand(a-b*div(a[3],b[3],x)[1])→b

t→a

EndWhile

expand(a)

EndFunc

Deg (p,x)

Func

© degré d'un polynôme;entrer p et la variable;renvoie -1 si p=0

Local k

-1→k

While string(p)≠"0"

k+1→k

d(p,x)→p

EndWhile

k

EndFunc

Desc3 (f,x)

Prgm

© décomposition en éléments simples de f sur C; entrer f et x;utilise

mczeros,div,pcoefd

Local nu,de,l,l1,l2,k,z,a,b,m,g,lz

DelVar i:ClrIO

getNum(f)→nu:getDenom(f)→de

div(nu,de,x)→l

Pause l[1]:l[2]/de→f

mczeros(de,x)→l

{ }→lz

For k,1,dim(l)[2]

l[1,k]→z:imag(z)→b

real(z)→a:l[2,k]→m

If string(b)≠"0" Then

taylor(f*(x-a)^m,x,m-1,a)→g

g|x=a+tt→g

g/tt^m→g

Pause g|tt=x-a

Else

augment(lz,{z})→lz

If string(product(lz-conj(z)))≠"0" Then

f*(x^2-2*a*x+a^2+b^2)^m→g

taylor(g,x,m-1,z)→g

g|x=z+tt→g

```

        taylor(g*(tt+2*b*i)^(-m),tt,m-1)→g
        g/tt^m→g
        Pause  expand(g|i=i and tt=x-a-b*i):Pause  expand(g|i=-i and
tt=x-a+b*i)
        EndIf
        EndIf
    EndFor
EndPrgm

Div (a,b,x)
Func
© division euclidienne; entrer a,b,la variable;renvoie {q,r};utilise
pcoefd
Local  da,db,dq,i,k,q
pcoefd(a,x)→a:pcoefd(b,x)→b
dim(a)-1→da:dim(b)-1→db
If da<db Then
{0,polyEval(a,x)}
Else
da-db→dq:newList(dq+1)→q
For  k,1,dq+1
    a[k]/(b[1])→q[k]
    If string(q[k])≠"0" Then
        For  i,0,db
            a[k+i]-q[k]*b[i+1]→a[k+i]
        EndFor
    EndIf
EndFor
{polyEval(q,x),polyEval(a,x)}
EndIf
EndFunc

Mczeros (p,x)
Func
© renvoie une matrice avec racines et ordre de multiplicité;entrer pol
et var
Local  l1,l2,l3,n,k,m,d
cZeros(p,x)→l1
dim(l1)→n:{ }→l2
{d(expand(p),x)}→l3:1→d
For  k,1,n
    1→m
    While  string(expand(l3[m]|x=l1[k]))="0"
        m+1→m
        If m>d Then
            d(l3[d],x)→l3[m]
            d+1→d
        EndIf
    EndWhile
    augment(l2,{m})→l2
EndFor
list→mat(augment(l1,l2),n)
EndFunc

Pcoefcn (p,x,n)
Func

```

```

© liste des coefficients d'un polynôme par ordre croissant, tronquée à
l'ordre n; entrer p,x et n
Local k,l
expand(p)→p:{ }→1
For k,0,n
  augment(1,{p|x=0}/(k!))→1
  d(p,x)→p
EndFor
1
EndFunc

Pcoefd (p,x)
Func
© liste des coefficients d'un polynôme p par ordre
décroissant;(réciproque de PolyEval) :donne { } pour p=0; entrer p et la
variable
Local k,l
expand(p)→p:{ }→1:0→k
While string(p)≠"0"
  augment({p|x=0}/(k!),1)→1
  d(p,x)→p
  k+1→k
EndWhile
1
EndFunc

```