```
> restart:

  ordre := 6;

  ## HYPOTHESE DE BRANCHE ET NORMALISATIONS SUPPLEMENTAIRES

  rho := 0:

  assume(ReF30210,real):
  ImF30210 := 0:
  #assume(ImF30210,real):
  F[3,0,2,1,0] := ReF30210 + I*ImF30210:

  assume(ReG30210,real):
  ImG30210 := 0:
  #assume(ImG30210,real):
  G[3,0,2,1,0] := ReG30210 + I*ImG30210:

  alpha := 0:
  G[4,0,0,2,0] := 0:  F[4,0,0,2,0] := 0:

  lambda := 1:
  G[3,0,0,2,0] := 1: F[3,0,0,2,0] := 1:

  ## PRENORMALISATIONS TOUJOURS POSSIBLES, QUELS QUE SOIENT LES
  BRANCHEMENTS ENSUITE

  ## ORDRE 6

  ## 6 STABILISER SPORADIQUES

  g[1,0,1] := (8*I)*lambda^2*conjugate(alpha)*conjugate(G[4, 0, 0,
  2, 0])+12*lambda^2*conjugate(alpha)^2*conjugate(G[3, 0, 0, 2, 0]
  )/conjugate(lambda)-(4*I)*lambda*conjugate(alpha)^2*
  alpha/conjugate(lambda)-4*lambda*conjugate(alpha)*alpha*G[3, 0,
  0, 2, 0]-2*lambda*conjugate(alpha)*rho/conjugate(lambda):
  f[0,0,2] := -conjugate(lambda)*alpha*lambda*G[3, 0, 0, 2, 0]*
  conjugate(alpha)+2*conjugate(alpha)^2*lambda^2*conjugate(G[3, 0,
  0, 2, 0])+(1/2)*lambda*rho*conjugate(alpha)-(1/4)*conjugate
  (lambda)*g[1, 0, 1]:

  ## 6 FIN RESOLUTION PREVUE AVEC LA SPORADIQUE Eq6[3,0,3,0,0]

  g[2,0,1] := (52/3)*conjugate(lambda)*lambda*alpha^2*G[3, 0, 0,
  2, 0]^2-(14/3)*lambda*conjugate(alpha)^2*alpha^2/conjugate
  (lambda)-(8/3)*f[1, 0, 2]/conjugate(lambda)+4*conjugate(G[3, 0,
  1, 2, 0])*lambda^3*conjugate(alpha)^2/conjugate(lambda)-(104/3)*
  lambda^2*conjugate(alpha)*conjugate(G[3, 0, 0, 2, 0])*alpha*G[3,
  0, 0, 2, 0]+(100/3)*lambda^3*conjugate(alpha)^2*conjugate(G[3,
  0, 0, 2, 0])^2/conjugate(lambda)+4*conjugate(lambda)*G[3, 0, 1,
  2, 0]*lambda*alpha^2-(8*I)*conjugate(lambda)*G[3, 0, 0, 3, 0]*
  alpha^3-(8*I)*f[0, 0, 2]*alpha/conjugate(lambda)+(2/3*I)*alpha*g
  [1, 0, 1]+(8*I)*conjugate(G[3, 0, 0, 3, 0])*lambda^3*conjugate
```

```
(alpha)^3/conjugate(lambda)^2-(2*I)*conjugate(G[3, 0, 2, 1, 0])*
lambda^3*conjugate(alpha)+(2*I)*conjugate(lambda)*G[3, 0, 2, 1,
0]*lambda^2*alpha-(2*I)*conjugate(alpha)*alpha*lambda*
rho/conjugate(lambda)-(1/3)*lambda*rho^2/conjugate(lambda)+
(16/3*I)*G[3, 0, 0, 2, 0]*lambda*conjugate(alpha)*alpha^2+(7/3*
I)*lambda^2*conjugate(alpha)*conjugate(g[1, 0, 1])/conjugate
(lambda)^2+(4/3)*lambda*conjugate(f[1, 0, 2])/conjugate(lambda)
^2-8*rho*lambda*alpha*G[3, 0, 0, 2, 0]-(112/3*I)*conjugate(G[3,
0, 0, 2, 0])*lambda^2*conjugate(alpha)^2*alpha/conjugate(lambda)
:

## 6 STABILISER 0 = F[h,i,2,0,l] = G[h,i,2,0,l]

g[0,2,1] := lambda*alpha^2/conjugate(lambda)+(6*I)*lambda^2*
alpha*conjugate(G[3, 0, 0, 2, 0])/conjugate(lambda):
g[1,1,1] := 4*lambda*alpha^2*G[3, 0, 0, 2, 0]-28*lambda^2*
conjugate(G[3, 0, 0, 2, 0])*conjugate(alpha)*alpha/conjugate
(lambda)-6*lambda*rho*alpha/conjugate(lambda)+(12*I)*lambda*
conjugate(alpha)*alpha^2/conjugate(lambda)+2*lambda^2*conjugate
(g[1, 0, 1])/conjugate(lambda)^2+12*lambda*conjugate(f[0, 0, 2])
/conjugate(lambda)^2:
g[0,4,0] := 0:
g[1,3,0] := 0:
g[2,2,0] := 0:
g[3,1,0] := (4*I)*alpha^3*lambda/conjugate(lambda):
g[4,0,0] := -(2*I)*lambda^2*conjugate(G[3,0,0,2,0])*conjugate
(alpha)*alpha^2/conjugate(lambda)-(18*I)*lambda*alpha^3*G[3,0,0,
2,0]+8*lambda^2*alpha^2*G[4,0,0,2,0]-I*lambda*rho*
alpha^2/conjugate(lambda)-lambda*conjugate(alpha)*
alpha^3/conjugate(lambda)+(4*I)*lambda^2*conjugate(g[1,0,1])*
alpha/conjugate(lambda)^2+(8*I)*lambda*conjugate(f[0,0,2])*
alpha/conjugate(lambda)^2-4*lambda^2*conjugate(g[0,0,2])
/conjugate(lambda)^2:

## 6 STABILISER 0 = F[h,i,1,0,l] = G[h,i,1,0,l]

h[0,0,3] := -(1/2*I)*lambda*alpha*conjugate(alpha*lambda*rho)-
(1/3)*lambda*conjugate(rho^2*lambda)-(13/3*I)*alpha*lambda^2*
conjugate(lambda*alpha^2*G[3,0,0,2,0])+(1/3*I)*G[3,0,0,2,0]*
alpha^2*lambda*conjugate(lambda^2*alpha)+(1/3*I)*lambda^2*
conjugate(alpha*g[1,0,1])+(1/4*I)*lambda^2*alpha*conjugate(g[3,
0,0])+(1/6*I)*alpha*g[1,0,1]*conjugate(lambda)^2-(2/3)*lambda*
alpha^2*conjugate(lambda*alpha^2)+(1/3)*lambda*conjugate(f[1,0,
2])+(1/3)*f[1,0,2]*conjugate(lambda)-(8/3)*G[3,0,0,2,0]*alpha*
lambda^2*conjugate(G[3,0,0,2,0]*alpha*lambda^2)+(4/3)*G[3,0,0,2,
0]^2*lambda*alpha^2*conjugate(lambda)^3+(4/3)*lambda^3*conjugate
(G[3,0,0,2,0]^2*alpha^2*lambda):

f[0,1,2] := -(5/2*I)*alpha^2*lambda*conjugate(alpha)-(11/2)*
conjugate(lambda)*G[3,0,0,2,0]*alpha^2*lambda+(7/2)*alpha*
lambda^2*conjugate(G[3,0,0,2,0])*conjugate(alpha)-(1/4)*lambda*
alpha*rho+(1/4)*lambda^2*conjugate(g[1,0,1])/conjugate(lambda)+
(1/8)*conjugate(lambda)*g[3,0,0]:
f[0,3,1] := 0:
f[1,2,1] := 0:
f[2,1,1] := -(6*I)*alpha^3*lambda:
```

```
f[3,0,1] := -(14*I)*conjugate(lambda)*lambda*alpha^3*G[3,0,0,2,
0]+(3/2*I)*conjugate(lambda)*alpha*g[3,0,0]+(6*I)*alpha^2*
lambda^2*conjugate(G[3,0,0,2,0])*conjugate(alpha)-I*alpha^2*
lambda*rho+2*lambda^2*conjugate(g[0,0,2])/conjugate(lambda)+6*
alpha^3*lambda*conjugate(alpha):
f[0,5,0] := 0:
f[1,4,0] := 0:
f[2,3,0] := 0:
f[3,2,0] := 0:
f[4,1,0] := 0:
f[5,0,0] := alpha^4*lambda+(2*I)*G[5,0,0,1,0]*lambda^4*alpha:

## 6 STABILISER 0 = F[h,i,0,0,l] = G[h,i,0,0,l]

h[0,2,2] := 0:
h[1,1,2] := -(4*I)*alpha^3*lambda*conjugate(lambda):
h[2,0,2] := (9*I)*alpha^2*conjugate(lambda)*lambda^2*conjugate(G
[3,0,0,2,0])*conjugate(alpha)-(25*I)*lambda*alpha^3*G[3,0,0,2,0]
*conjugate(lambda)^2+12*lambda*conjugate(alpha)*conjugate
(lambda)*alpha^3+(3/2*I)*alpha*lambda^2*conjugate(g[1,0,1])-
(7/2*I)*alpha^2*conjugate(lambda)*lambda*rho+(5/4*I)*alpha*
conjugate(lambda)^2*g[3,0,0]+lambda^2*conjugate(g[0,0,2]):
h[0,4,1] := 0:
h[1,3,1] := 0:
h[2,2,1] := 0:
h[3,1,1] := 0:
h[4,0,1] := 5*alpha^4*lambda*conjugate(lambda):
h[0,6,0] := 0:
h[1,5,0] := 0:
h[2,4,0] := 0:
h[3,3,0] := 0:
h[4,2,0] := 0:
h[5,1,0] := 0:
h[6,0,0] := 0:

## ORDRE 5

## 5 SPORADIQUES

g[2,0,0] := (6*I)*lambda^2*conjugate(alpha)*conjugate(G[3,0,0,2,
0])/conjugate(lambda)-(2*I)*lambda*alpha*G[3,0,0,2,0]-I*lambda*
rho/conjugate(lambda)+3*lambda*conjugate(alpha)*alpha/conjugate
(lambda):
g[0,0,1] := lambda*conjugate(alpha)^2/conjugate(lambda):

## 5 STABILISER 0 = F[h,i,2,0,l] = G[h,i,2,0,l]

g[0,1,1] := conjugate(alpha)*lambda*alpha/conjugate(lambda)-
(1/2)*lambda^2*conjugate(g[2,0,0])/conjugate(lambda)^2-(1/2)*g
[2,0,0]:
g[0,3,0] := 0:
g[1,2,0] := 0:
g[2,1,0] := 2*lambda*alpha^2/conjugate(lambda)-5*lambda^2*
conjugate(g[0,0,1])/conjugate(lambda)^2:
g[3,0,0] := (2*I)*lambda*conjugate(alpha)*alpha^2/conjugate
(lambda)+8*G[3,0,0,2,0]*lambda*alpha^2-(6*I)*lambda^2*conjugate
```

```
(g[0,0,1])*conjugate(alpha)/conjugate(lambda)^2-I*g[2,0,0]*
alpha+I*lambda^2*alpha*conjugate(g[2,0,0])/conjugate(lambda)
^2+4*lambda*alpha*rho/conjugate(lambda)-2*lambda^2*conjugate(g
[1,0,1])/conjugate(lambda)^2-8*conjugate(f[0,0,2])*
lambda/conjugate(lambda)^2:

## 5 STABILISER 0 = F[h,i,1,0,l] = G[h,i,1,0,l]

f[0,2,1] := 0:
f[1,1,1] := 2*lambda*alpha^2+lambda^2*conjugate(g[0,0,1])
/conjugate(lambda):
f[2,0,1] := -lambda*alpha*rho-(1/2*I)*alpha^2*lambda*conjugate
(alpha)+(4*I)*lambda^2*conjugate(alpha)*conjugate(g[0,0,1])
/conjugate(lambda)+(1/2)*lambda^2*conjugate(g[1,0,1])/conjugate
(lambda)+(1/4*I)*lambda^2*alpha*conjugate(g[2,0,0])/conjugate
(lambda)+(5/4*I)*conjugate(lambda)*g[2,0,0]*alpha+4*lambda*
conjugate(f[0,0,2])/conjugate(lambda):
f[0,4,0] := 0:
f[1,3,0] := 0:
f[2,2,0] := 0:
f[3,1,0] := 0:
f[4,0,0] := -I*alpha^3*lambda+(2*I)*alpha*lambda^2*conjugate(g
[0,0,1])/conjugate(lambda):

## 5 STABILISER 0 = F[h,i,0,0,l] = G[h,i,0,0,l]

h[0,1,2] := alpha^2*conjugate(lambda)*lambda:
h[1,0,2] := (2*I)*lambda^2*conjugate(alpha)*conjugate(g[0,0,1])+
I*conjugate(lambda)*alpha^2*lambda*conjugate(alpha)+2*conjugate
(f[0,0,2])*lambda+(1/2*I)*conjugate(lambda)^2*g[2,0,0]*alpha+
(1/2*I)*lambda^2*alpha*conjugate(g[2,0,0])+2*lambda*alpha*rho*
conjugate(lambda):
h[0,3,1] := 0:
h[1,2,1] := 0:
h[2,1,1] := 0:
h[3,0,1] := (4*I)*lambda^2*conjugate(g[0,0,1])*alpha:
h[0,5,0] := 0:
h[1,4,0] := 0:
h[2,3,0] := 0:
h[3,2,0] := 0:
h[4,1,0] := 0:
h[5,0,0] := 0:

## ORDRE 4

g[1,0,0] := (2*I)*lambda*conjugate(alpha)/conjugate(lambda):

f[1,0,1] := -3*lambda*conjugate(alpha)*alpha-(1/2*I)*conjugate
(lambda)*alpha*g[1,0,0]+(3/2*I)*lambda^2*conjugate(alpha)*
conjugate(g[1,0,0])/conjugate(lambda)-(5/8)*lambda*g[1,0,0]*
conjugate(g[1,0,0])-(1/4)*conjugate(lambda)*g[2,0,0]-(1/4)*
lambda^2*conjugate(g[2,0,0])/conjugate(lambda)+I*lambda*rho:

g[0,2,0] := 0:
g[1,1,0] := -(2*I)*lambda*alpha/conjugate(lambda):
```

```
f[0,1,1] := I*lambda*alpha:
f[0,3,0] := 0:
f[1,2,0] := 0:
f[2,1,0] := 0:
f[3,0,0] := (3*I)*lambda^2*alpha*conjugate(g[1,0,0])/conjugate
(lambda)-4*lambda*alpha^2-lambda^2*conjugate(g[0,0,1])/conjugate
(lambda)+(1/2)*lambda^3*conjugate(g[1,0,0])^2/conjugate(lambda)
^2:

h[0,2,1] := 0:
h[1,1,1] := 0:
h[2,0,1] := I*lambda^2*alpha*conjugate(g[1,0,0])-4*alpha^2*
conjugate(lambda)*lambda-lambda^2*conjugate(g[0,0,1]):
h[0,4,0] := 0:
h[1,3,0] := 0:
h[2,2,0] := 0:
h[3,1,0] := 0:
h[4,0,0] := 0:

## ORDRE 3

g[0,1,0] := lambda/conjugate(lambda):

f[0,2,0] := 0:
f[1,1,0] := 0:
f[2,0,0] := -(1/2)*lambda^2*conjugate(g[1,0,0])/conjugate
(lambda)-(2*I)*lambda*alpha:

h[0,1,1] := 0:
h[1,0,1] := -(2*I)*lambda*alpha*conjugate(lambda):
h[0,3,0] := 0:
h[1,2,0] := 0:
h[2,1,0] := 0:
h[3,0,0] := 0:

## ORDRE 2

f[0,1,0] := 0:

h[0,0,1] := lambda*conjugate(lambda):
h[0,2,0] := 0:
h[1,1,0] := 0:
h[2,0,0] := 0:

## ORDRE 1

h[1,0,0] := 0:
h[0,1,0] := 0:

## CALCULS

FF :=

z*zb

+ (1/2)*zb^2*s+(1/2)*z^2*sb
```

```
+ z*zb*s*sb

+ (1/2)*s*sb^2*z^2+(1/2)*zb^2*s^2*sb
+ conjugate(F[3,0,0,2,0])*s^2*zb^3 + F[3,0,0,2,0]*z^3*sb^2

+ z*zb*s^2*sb^2
+ F[5,0,0,1,0]*z^5*sb
+ conjugate(F[5,0,0,1,0])*s*zb^5
+ F[4,0,0,2,0]*z^4*sb^2
+ conjugate(F[4,0,0,2,0])*s^2*zb^4
+ F[3,0,2,1,0]*z^3*zb^2*sb
+ conjugate(F[3,0,2,1,0])*z^2*s*zb^3
+ F[3,0,1,2,0]*z^3*zb*sb^2
+ conjugate(F[3,0,1,2,0])*z*s^2*zb^3
+ F[3,0,0,3,0]*z^3*sb^3
+ conjugate(F[3,0,0,3,0])*s^3*zb^3
+ 3*conjugate(F[3,0,0,2,0])*z*s^2*zb^2*sb
+ 3*F[3,0,0,2,0]*z^2*s*zb*sb^2:

GG :=

z1*z1b

+ (1/2)*z1b^2*s1+(1/2)*z1^2*s1b

+ z1*z1b*s1*s1b

+ (1/2)*s1*s1b^2*z1^2+(1/2)*z1b^2*s1^2*s1b
+ conjugate(G[3,0,0,2,0])*s1^2*z1b^3 + G[3,0,0,2,0]*z1^3*s1b^2

+ z1*z1b*s1^2*s1b^2
+ G[5,0,0,1,0]*z1^5*s1b
+ conjugate(G[5,0,0,1,0])*s1*z1b^5
+ G[4,0,0,2,0]*z1^4*s1b^2
+ conjugate(G[4,0,0,2,0])*s1^2*z1b^4
+ G[3,0,2,1,0]*z1^3*z1b^2*s1b
+ conjugate(G[3,0,2,1,0])*z1^2*s1*z1b^3
+ G[3,0,1,2,0]*z1^3*z1b*s1b^2
+ conjugate(G[3,0,1,2,0])*z1*s1^2*z1b^3
+ G[3,0,0,3,0]*z1^3*s1b^3
+ conjugate(G[3,0,0,3,0])*s1^3*z1b^3
+ 3*conjugate(G[3,0,0,2,0])*z1*s1^2*z1b^2*s1b
+ 3*G[3,0,0,2,0]*z1^2*s1*z1b*s1b^2:



f[0,0,0] := 0:  g[0,0,0] := 0:  h[0,0,0] := 0:

f[1,0,0] := lambda:  f[0,0,1] := - I*lambda*conjugate(alpha):
#assume(rho,real):
h[0,0,2] := lambda*conjugate(lambda)*(-alpha*conjugate(alpha)+I*
rho):

z1 := add(add(add(f[hh,i,l]*z^hh*s^i*w^l, hh=0..ordre-1-i-2*l),
i=0..ordre-1-2*l), l=0..(ordre-1)/2):
```

```
z1b := add(add(add(conjugate(f[hh,i,l])*zb^hh*sb^i*wb^l, hh=0..
ordre-1-i-2*l), i=0..ordre-1-2*l), l=0..(ordre-1)/2):


s1 := add(add(add(g[hh,i,l]*z^hh*s^i*w^l, hh=0..ordre-2-i-2*l),
i=0..ordre-2-2*l), l=0..(ordre-2)/2):

s1b := add(add(add(conjugate(g[hh,i,l])*zb^hh*sb^i*wb^l, hh=0..
ordre-2-i-2*l), i=0..ordre-2-2*l), l=0..(ordre-2)/2):


w1 := add(add(add(h[hh,i,l]*z^hh*s^i*w^l, hh=0..ordre-i-2*l), i=
0..ordre-2*l), l=0..ordre/2):

w1b := add(add(add(conjugate(h[hh,i,l])*zb^hh*sb^i*wb^l, hh=0..
ordre-i-2*l), i=0..ordre-2*l), l=0..ordre/2):




w  := FF + I*v:
wb := FF - I*v:

u1 := (w1+w1b)/2:
v1 := (w1-w1b)/(2*I):




eqdef := mtaylor( - u1 + GG, [z,s,zb,sb,v], ordre+1, [1,1,1,1,2]
) :


## TOUTES LES EQUATIONS COMPLEXES

printlevel := 4:
for m from 0 to ordre do
for l from 0 to m/2 do
for k from 0 to m-2*l do
for j from 0 to m-k-2*l do
for i from 0 to m-j-k-2*l do

EEq||m[m-i-j-k-2*l,i,j,k,l] :=
expand(expand(coeftayl(eqdef, [z,s,zb,sb,v]=[0,0,0,0,0], [m-i-j-
k-2*l,i,j,k,l])));

od; od; od; od; od;

## SELECTIONNER CELLES QUI SONT NON NULLES

printlevel := 5:
for m from 0 to ordre do
for l from 0 to m/2 do
for k from 0 to m-2*l do
for j from 0 to m-k-2*l do
for i from 0 to m-j-k-2*l do
```

```
if EEq||m[m-i-j-k-2*l,i,j,k,l] <> 0 then
   Eq||m[m-i-j-k-2*l,i,j,k,l] := expand(expand(EEq||m[m-i-j-
k-2*l,i,j,k,l])); end if;

od; od; od; od; od;
```

$$ordre := 6 \tag{1}$$

```
>

## Eq[h,i,0,0,l]

for l from 0 to ordre/2 do
for i from 0 to ordre-2*l do

if expand(EEq||ordre[ordre-i-2*l,i,0,0,l]) <> 0 then
Eq||ordre[ordre-i-2*l,i,0,0,l] := expand(EEq||ordre[ordre-i-2*l,
i,0,0,l]); end if;

od; od;
```

```
>

## Eq[h,i,1,0,l]

for l from 0 to (ordre-1)/2 do
for i from 0 to ordre-1-2*l do

if expand(EEq||ordre[ordre-1-i-2*l,i,1,0,l]) <> 0 then
Eq||ordre[ordre-1-i-2*l,i,1,0,l] := expand(EEq||ordre[ordre-1-
i-2*l,i,1,0,l]); end if;

od; od;
```

```
>

## Eq[h,i,2,0,l]

for l from 0 to (ordre-2)/2 do
for i from 0 to ordre-2-2*l do

if expand(EEq||ordre[ordre-2-i-2*l,i,2,0,l]) <> 0 then
Eq||ordre[ordre-2-i-2*l,i,2,0,l] := expand(EEq||ordre[ordre-2-
i-2*l,i,2,0,l]); end if;
```

```
od; od;
```

```
> l1 := (ordre - (0+1+3+0))/2;
  for l from floor(l1) to l1 do
  if l1-floor(l1) = 0
  and expand(EEq||ordre[0,1,3,0,l]) <> 0 then
  Eq||ordre[0,1,3,0,l] := expand(EEq||ordre[0,1,3,0,l]); end if;
  od;
```

```
> l2 := (ordre - (0+1+4+0))/2;
  for l from floor(l2) to l2 do
  if l2-floor(l2) = 0
  and expand(EEq||ordre[0,1,4,0,l]) <> 0 then
  Eq||ordre[0,1,4,0,l] := expand(EEq||ordre[0,1,4,0,l]); end if;
  od;
```

```
> l3 := (ordre - (1+1+3+0))/2;
  for l from floor(l3) to l3 do
  if l3-floor(l3) = 0
  and expand(EEq||ordre[1,1,3,0,l3]) <> 0 then
  Eq||ordre[1,1,3,0,l3] := expand(EEq||ordre[1,1,3,0,l3]); end if;
  od;
```

```
> l4 := (ordre - (1+1+4+0))/2;
  for l from floor(l4) to l4 do
  if l4-floor(l4) = 0
  and expand(EEq||ordre[1,1,4,0,l4]) <> 0 then
  Eq||ordre[1,1,4,0,l4] := expand(EEq||ordre[1,1,4,0,l4]); end if;
  od;
```

```
> l5 := (ordre - (3+0+3+0))/2;
  for l from floor(l5) to l5 do
  if l5-floor(l5) = 0
  and expand(EEq||ordre[3,0,3,0,l5]) <> 0 then
  Eq||ordre[3,0,3,0,l5] := expand(EEq||ordre[3,0,3,0,l5]); end if;
  od;
```