```
> #Maple proof of (18) for tracking critical points
  restart;
  with(linalg):

  #Taylor expansion of 2D time-dependent vectro field
  #V(x,y,z) = (uf(x,y,z) , vf(x,y,z) )^T
  #z is the time component

  uf := u + u_x*x + u_y*y + u_z*z
  + u_xx*x*x + u_xy*x*y + u_xz*x*z
  + u_yy*y*y + u_yz*y*z + u_zz*z*z:

  vf := v + v_x*x + v_y*y + v_z*z
  + v_xx*x*x + v_xy*x*y + v_xz*x*z
  + v_yy*y*y + v_yz*y*z + v_zz*z*z:

  #feature line passes through (0,0,0)
  u := 0:
  v := 0:

  #partials of V
  uf_x := diff(uf,x):
  uf_y := diff(uf,y):
  uf_z := diff(uf,z):

  vf_x := diff(vf,x):
  vf_y := diff(vf,y):
  vf_z := diff(vf,z):


  #the feature flow field F = (uff,vff,wff)^T
  uff := uf_y*vf_z - vf_y*uf_z:
  vff := uf_z*vf_x - vf_z*uf_x:
  wff := uf_x*vf_y - vf_x*uf_y:

  #gradient field G

  #HG := (det(V,V_x) , det(V,V_y), det(V,V_z))^T
  hugf := vf*uf_x -uf*vf_x:
  hvgf := vf*uf_y -uf*vf_y:
  hwgf := vf*uf_z -uf*vf_z:

  #Correction Field G = (ugf,vgf,wgf)^T = -F/|F| x HG
  ugf := (hvgf*wff - hwgf*vff)/(uff^2 + vff^2 + wff^2)^(1/2):
  vgf := (hwgf*uff - hugf*wff)/(uff^2 + vff^2 + wff^2)^(1/2):
  wgf := (hugf*vff - hvgf*uff)/(uff^2 + vff^2 + wff^2)^(1/2):

  #partials of G
  ugf_x := diff(ugf,x):
  vgf_x := diff(vgf,x):
  wgf_x := diff(wgf,x):

  ugf_y := diff(ugf,y):
  vgf_y := diff(vgf,y):
  wgf_y := diff(wgf,y):

  ugf_z := diff(ugf,z):
  vgf_z := diff(vgf,z):
  wgf_z := diff(wgf,z):

  #renaming: F = (fffu,fffv,fffw)^T
```

```
fffu := (uff):
fffv := (vff):
fffw := (wff):

#FN = F/|F| = (fffun , fffvn , fffwn)^T
fffun := fffu / sqrt(fffu^2 + fffv^2 + fffw^2):
fffvn := fffv / sqrt(fffu^2 + fffv^2 + fffw^2):
fffwn := fffw / sqrt(fffu^2 + fffv^2 + fffw^2):

#renaming: G = (gggu,gggv,gggw)^T
gggu := ugf:
gggv := vgf:
gggw := wgf:


#partials of F
fffu_x := diff(fffu,x):
fffv_x := diff(fffv,x):
fffw_x := diff(fffw,x):

fffu_y := diff(fffu,y):
fffv_y := diff(fffv,y):
fffw_y := diff(fffw,y):

fffu_z := diff(fffu,z):
fffv_z := diff(fffv,z):
fffw_z := diff(fffw,z):

#partials of FN
fffun_x := diff(fffun,x):
fffvn_x := diff(fffvn,x):
fffwn_x := diff(fffwn,x):

fffun_y := diff(fffun,y):
fffvn_y := diff(fffvn,y):
fffwn_y := diff(fffwn,y):

fffun_z := diff(fffun,z):
fffvn_z := diff(fffvn,z):
fffwn_z := diff(fffwn,z):


#partials of G
gggu_x := diff(gggu,x):
gggv_x := diff(gggv,x):
gggw_x := diff(gggw,x):

gggu_y := diff(gggu,y):
gggv_y := diff(gggv,y):
gggw_y := diff(gggw,y):

gggu_z := diff(gggu,z):
gggv_z := diff(gggv,z):
gggw_z := diff(gggw,z):


#we are interested in the behavior at (0,0,0),
#all necessary derivatives are computed
x := 0:
y := 0:
z := 0:
```

```
#hf1 = det(F,F_y,F_z)
hf1 :=
+ fffu*fffv_y*fffw_z
+ fffv*fffw_y*fffu_z
+ fffw*fffu_y*fffv_z
- fffw*fffv_y*fffu_z
- fffu*fffw_y*fffv_z
- fffv*fffu_y*fffw_z:

#hf2 = det(F_x,F,F_z)
hf2 :=
+ fffu_x*fffv*fffw_z
+ fffv_x*fffw*fffu_z
+ fffw_x*fffu*fffv_z
- fffw_x*fffv*fffu_z
- fffu_x*fffw*fffv_z
- fffv_x*fffu*fffw_z:

#hf3 := det(F_x,F_y,F)
hf3 :=
+ fffu_x*fffv_y*fffw
+ fffv_x*fffw_y*fffu
+ fffw_x*fffu_y*fffv
- fffw_x*fffv_y*fffu
- fffu_x*fffw_y*fffv
- fffv_x*fffu_y*fffw:

hhhf := factor(hf1*fffu + hf2*fffv + hf3*fffw):

####

hg1 :=
+ fffu*gggv_y*gggw_z
+ fffv*gggw_y*gggu_z
+ fffw*gggu_y*gggv_z
- fffw*gggv_y*gggu_z
- fffu*gggw_y*gggv_z
- fffv*gggu_y*gggw_z:

hg2 :=
+ gggu_x*fffv*gggw_z
+ gggv_x*fffw*gggu_z
+ gggw_x*fffu*gggv_z
- gggw_x*fffv*gggu_z
- gggu_x*fffw*gggv_z
- gggv_x*fffu*gggw_z:

hg3 :=
+ gggu_x*gggv_y*fffw
+ gggv_x*gggw_y*fffu
+ gggw_x*gggu_y*fffv
- gggw_x*gggv_y*fffu
- gggu_x*gggw_y*fffv
- gggv_x*gggu_y*fffw:

hhhg := factor(hg1*fffu + hg2*fffv + hg3*fffw):

####################

hh1 :=
+ hhhu*hhhv_y*hhhw_z
+ hhhv*hhhw_y*hhhu_z
+ hhhw*hhhu_y*hhhv_z
- hhhw*hhhv_y*hhhu_z
- hhhu*hhhw_y*hhhv_z
- hhhv*hhhu_y*hhhw_z:

hh2 :=
+ hhhu_x*hhhv*hhhw_z
+ hhhv_x*hhhw*hhhu_z
+ hhhw_x*hhhu*hhhv_z
- hhhw_x*hhhv*hhhu_z
- hhhu_x*hhhw*hhhv_z
- hhhv_x*hhhu*hhhw_z:

hh3 :=
+ hhhu_x*hhhv_y*hhhw
+ hhhv_x*hhhw_y*hhhu
+ hhhw_x*hhhu_y*hhhv
- hhhw_x*hhhv_y*hhhu
- hhhu_x*hhhw_y*hhhv
- hhhv_x*hhhu_y*hhhw:

hhhh := factor(hh1*hhhu + hh2*hhhv + hh3*hhhw):

###########

hk1 :=
+ fffu*fffv_y*gggw_z
+ fffv*fffw_y*gggu_z
+ fffw*fffu_y*gggv_z
- fffw*fffv_y*gggu_z
- fffu*fffw_y*gggv_z
- fffv*fffu_y*gggw_z

+ fffu*gggv_y*fffw_z
+ fffv*gggw_y*fffu_z
+ fffw*gggu_y*fffv_z
- fffw*gggv_y*fffu_z
- fffu*gggw_y*fffv_z
- fffv*gggu_y*fffw_z:


hk2 :=
+ fffu_x*fffv*gggw_z
+ fffv_x*fffw*gggu_z
+ fffw_x*fffu*gggv_z
- fffw_x*fffv*gggu_z
- fffu_x*fffw*gggv_z
- fffv_x*fffu*gggw_z

+ gggu_x*fffv*fffw_z
+ gggv_x*fffw*fffu_z
+ gggw_x*fffu*fffv_z
- gggw_x*fffv*fffu_z
- gggu_x*fffw*fffv_z
- gggv_x*fffu*fffw_z:

hk3 :=
+ fffu_x*gggv_y*fffw
+ fffv_x*gggw_y*fffu
+ fffw_x*gggu_y*fffv
- fffw_x*gggv_y*fffu
```

```
    - fffu_x*gggw_y*fffv
    - fffv_x*gggu_y*fffw

    + gggu_x*fffv_y*fffw
    + gggv_x*fffw_y*fffu
    + gggw_x*fffu_y*fffv
    - gggw_x*fffv_y*fffu
    - gggu_x*fffw_y*fffv
    - gggv_x*fffu_y*fffw:

    hhhk := factor(hk1*fffu + hk2*fffv + hk3*fffw):

    ###########

    s0 := factor(fffun_x+ fffvn_y+ fffwn_z):
    s1 := (+(gggu_x + gggv_y + gggw_z)) / (fffu^2 + fffv^2 + fffw^2)^(1/2):

    sh := s0 + al*s1:

    p0 := hhhf / (fffu^2 + fffv^2 + fffw^2)^2:
    p1 := hhhk / (fffu^2 + fffv^2 + fffw^2)^2:
    p2 := hhhg / (fffu^2 + fffv^2 + fffw^2)^2:

    ph := p0 + al*p1 + al^2*p2:
Warning, the protected names norm and trace have been redefined and unprotected


> #now the proof of (18):
  factor(s1);
  factor(s1^2 - 4*p2);
  factor(2*s0*s1 - 4*p1);

                                   -2

                                   0

                                   0

>
```