Maple Script

Set up

```
> restart;
  with(LinearAlgebra):
  with(PDEtools):
  with(plots):
  with(CodeGeneration):
  declare(h1(t,x),eta1(t,x),h2(t,x),eta2(t,x)):
  constants:=constants,H1,H2,m2,m3,sigma1,sigma2:
```

$h1(t,x)$ *will now be displayed as* $h1$

$\eta 1(t,x)$ *will now be displayed as* $\eta 1$

$h2(t,x)$ *will now be displayed as* $h2$

$\eta 2(t,x)$ *will now be displayed as* $\eta 2$ **(1.1)**

```
> A:=Matrix(7,7,[[h1(t,x)^2/2,-h1(t,x)^2/2/m2,0,h1(t,x),-h1(t,x)
  /m2,-1/m2,0],
  [0,h2(t,x)^2/2/m2,-(h2(t,x)-1)^2/2/m3,0,h2(t,x)/m2,1/m2,-(h2(t,
  x)-1)/m3],
  [h1(t,x),-h1(t,x),0,1,-1,0,0],
  [0,h2(t,x),-(h2(t,x)-1),0,1,0,-1],
  [1,-1,0,0,0,0,0],
  [0,1,-1,0,0,0,0],
  [h1(t,x)^3/6,(h2(t,x)^3-h1(t,x)^3)/6/m2,(-(h2(t,x)-1)^3)/6/m3,
  h1(t,x)^2/2,(h2(t,x)^2-h1(t,x)^2)/2/m2,(h2(t,x)-h1(t,x))/m2,(-
  (h2(t,x)-1)^2)/2/m3]]);
  a:=Vector(7,[0,0,0,0,-diff(h1(t,x),x$3)*sigma1,-diff(h2(t,x),
  x$3)*sigma2,Q]);
```

$$A := \left[\left[\frac{1}{2}\, h1^2,\ -\frac{1}{2}\, \frac{h1^2}{m2},\ 0,\ h1,\ -\frac{h1}{m2},\ -\frac{1}{m2},\ 0\right],\right.$$

$$\left[0,\ \frac{1}{2}\, \frac{h2^2}{m2},\ -\frac{1}{2}\, \frac{(h2-1)^2}{m3},\ 0,\ \frac{h2}{m2},\ \frac{1}{m2},\ -\frac{h2-1}{m3}\right],$$

$$\left[h1,\ -h1,\ 0,\ 1,\ -1,\ 0,\ 0\right],$$

$$\left[0,\ h2,\ -h2+1,\ 0,\ 1,\ 0,\ -1\right],$$

$$\left[1,\ -1,\ 0,\ 0,\ 0,\ 0,\ 0\right],$$

$$\left[0,\ 1,\ -1,\ 0,\ 0,\ 0,\ 0\right],$$

$$\left[\frac{1}{6}\, h1^3,\ \frac{1}{6}\, \frac{h2^3-h1^3}{m2},\ -\frac{1}{6}\, \frac{(h2-1)^3}{m3},\ \frac{1}{2}\, h1^2,\ \frac{1}{2}\, \frac{h2^2-h1^2}{m2},\ \frac{h2-h1}{m2},\right.$$

$$-\frac{1}{2}\,\frac{(h2-1)^2}{m3}\Big]\Big]$$

$$a := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -h1_{x,x,x}\,\sigma 1 \\ -h2_{x,x,x}\,\sigma 2 \\ Q \end{bmatrix} \qquad (1.2)$$

u1 = p1x/2*y^2 + c1*y
u2 = p2x/(2*m2)*y^2 + c2/m2*y + c3/m2
u3 = p3x/(2*m3)*(y-1)^2 + c4/m3*(y-1)
b = [p1x,p2x,p3x,c1,c2,c3,c4]

Solving and building solutions

```
> b:=factor(MatrixInverse(A).a):
  u1:=collect(b[1]*y^2/2+b[4]*y,y,factor):
  u2:=collect(b[2]*y^2/2/m2+b[5]*y/m2+b[6]/m2,y,factor):
  u3:=collect(b[3]*(y-1)^2/2/m3+b[7]*(y-1)/m3,y,factor):
```

```
> u1_steady := simplify(eval(u1, [sigma1 = 0, sigma2 = 0])) :
  u2_steady := simplify(eval(u2, [sigma1 = 0, sigma2 = 0])) :
  u3_steady := simplify(eval(u3, [sigma1 = 0, sigma2 = 0])) :
```

```
> evolution_eqn_1:=collect(int(u1,y=0..h1(t,x)),{diff(h1(t,x),
  x$3),diff(h2(t,x),x$3)},factor):#NOTE: These equations still
  need to be differentiated by x.
  evolution_eqn_2:=collect(int(u3,y=h2(t,x)..1),{diff(h1(t,x),
  x$3),diff(h2(t,x),x$3)},factor):#NOTE: These equations still
  need to be differentiated by x.
  evolution_eqn_1_diff:=diff(evolution_eqn_1, x):
  evolution_eqn_2_diff:=diff(evolution_eqn_2, x):
  evolution_eqn_1_delta:=eval(evolution_eqn_1,{h1(t,x)=H1+delta*
  eta1(t,x),h2(t,x)=H2+delta*eta2(t,x)}):
  evolution_eqn_2_delta:=eval(evolution_eqn_2,{h1(t,x)=H1+delta*
  eta1(t,x),h2(t,x)=H2+delta*eta2(t,x)}):
  evolution_eqn_1_linear:=diff(eta1(t,x),t)+diff(coeff(taylor
  (evolution_eqn_1_delta,delta,2),delta,1),x):
  evolution_eqn_2_linear:=diff(eta2(t,x),t)-diff(coeff(taylor
  (evolution_eqn_2_delta,delta,2),delta,1),x):
```

Dispersion Relation

```
> evolution_eqn_1_normal_modes:=simplify(eval
  (evolution_eqn_1_linear,{eta1(t,x)=A_1*exp(omega*t+I*k*x),eta2
  (t,x)=A_2*exp(omega*t+I*k*x)})/exp(omega*t+I*k*x)):
```

```
evolution_eqn_2_normal_modes:=simplify(eval
(evolution_eqn_2_linear,{eta1(t,x)=A_1*exp(omega*t+I*k*x),eta2
(t,x)=A_2*exp(omega*t+I*k*x)})/exp(omega*t+I*k*x)):
M:=Matrix(2,2,[coeff(evolution_eqn_1_normal_modes,A_1),coeff
(evolution_eqn_1_normal_modes,A_2),coeff
(evolution_eqn_2_normal_modes,A_1),coeff
(evolution_eqn_2_normal_modes,A_2)]):
dispersionRelation:=collect(Determinant(M),omega) = 0:
#eigenvalues_linear:=solve(dispersionRelation,omega):
#eigenvalues_linear_real:=[Re(eigenvalues_linear[1]), Re
(eigenvalues_linear[2])]:
```

Plotting the Dispersion Relation

```
> #plot(eval(eigenvalues_linear_real,[H1=5/12,H2=7/12,Q=1,m2=0.4,
m3=0.4,sigma1=1,sigma2=1]),k=0..5);
> #plot(max(eval(eigenvalues_linear_real,[H1=5/12,H2=7/12,Q=1,m2=
0.4,m3=0.4,sigma1=1,sigma2=1])),k=0..5);
> #plot3d(eval(eigenvalues_linear_real,[H1=5/12,H2=7/12,sigma1=1,
sigma2=1,Q=1,m3=0.4]),k=0..5,m2=1..0.1,title="Dispersion
Relation Plot");
> #contourplot(max(eval(eigenvalues_linear_real,[H1=5/12,H2=7/12,
sigma1=1,sigma2=1,Q=1,k=1])),m2=0..3,m3=0..3,title="Dispersion
Relation Plot",filledregions = true, coloring = ["White",
"DarkViolet"],contours=[0]);
> #interactiveparams(plot,[eigenvalues_linear_real,k=0..1],H1=0.
.1,H2=0..1,sigma1=0..2,sigma2=0..2,Q=0..10,m2=0..2,m3=0..2);
> #interactiveparams(contourplot,[max(eigenvalues_linear_real),
m2=0..1,m3=0..1,filledregions = true, coloring = ["White",
"DarkViolet"],contours=[0]],H1=0..1,H2=0..1,sigma1=0..2,sigma2=
0..2,Q=0..50,k=0..10);
```

Determining the Eigenvalues of the Matrices

Linear

$$\eta_t + Q\,G\cdot\begin{pmatrix}\eta_{1x}\\ \eta_{2x}\end{pmatrix} + F\cdot\begin{pmatrix}\eta_{1xxxx}\\ \eta_{2xxxx}\end{pmatrix} = 0$$

Surface Tension

```
> F_linear:=simplify(Matrix(2, 2, [coeff
(evolution_eqn_1_linear, diff(eta1(t, x), x$4)), coeff
(evolution_eqn_1_linear, diff(eta2(t, x), x$4)), coeff
(evolution_eqn_2_linear, diff(eta1(t, x), x$4)), coeff
(evolution_eqn_2_linear, diff(eta2(t, x), x$4))])):
simplify(Eigenvalues(F_linear)):
```

Flux

```
> g_linear:=simplify([coeff(evolution_eqn_1_linear, Q),coeff
(evolution_eqn_2_linear, Q)]):
G_linear:=Matrix(2,2,[coeff(g_linear[1], diff(eta1(t, x), x)
),coeff(g_linear[1], diff(eta2(t, x), x)),coeff(g_linear[2],
diff(eta1(t, x), x)),coeff(g_linear[2], diff(eta2(t, x), x))]
):
#simplify((G_linear[1,1]-G_linear[2,2])^2+4*G_linear[1,2]*
```

```
G_linear[2,1]):
eigenvalues_linear:=simplify(Eigenvalues(G_linear)):
```

Non-linear

h_t + (g*Q + f_1*h_1xxx + f_2*h_2xxx)_x = 0

```
> g_nonlinear:=[coeff(evolution_eqn_1, Q),-coeff
  (evolution_eqn_2, Q)]:
  G_nonlinear:=Matrix(2,2,[Physics[diff](g_nonlinear,h1(t,x)),
  Physics[diff](g_nonlinear,h2(t,x))]):
  simplify(G_nonlinear[1,2]*G_nonlinear[2,1]):
  eigenvalues_nonlinear := Eigenvalues(G_nonlinear):
  interface1:=0.4+(2^(-5))*cos(x):
  interface2:=0.6+(2^(-5))*cos(x+theta):
> plot(eval([0,interface1,interface2,1],theta = 0),x=0..2*Pi,
  title="Interfaces"):
> plot(eval(eigenvalues_nonlinear,eval([h1(t,x)=interface1,h2
  (t,x)=interface2,m2=0.8,m3=0.6],theta = 0)),x=0..2*Pi,title=
  "Eigenvalues"):
> interactiveparams(plot,[eval(eigenvalues_nonlinear,[h1(t,x)=
  interface1,h2(t,x)=interface2,m2=0.8,m3=0.6]),x=0..2*Pi,
  title="Eigenvalues"],theta=0..Pi):
>
```