Instability of the Kolmogorov flow with modified viscosity Numerical solution of the perturbation at orders 1 and 2 for f1 and f2b, and determination of the threshold Δ c

version using NDSolve

Various initializations

- **Suppression of ennoying messages**

```
Off[General::spell]
Off[General::spell1]
Off[Union::heads]
Off[NDSolve::ndinn]
Off[NDSolve::berr]
Off[MessageOptions]
```

```
Flow2 = θ → Evaluate[# + 1 / 2 Sin[2 #] &];
Flow1 = θ → Evaluate[# &];
Flow3 = θ → Evaluate[# + 1 / 4 Sin[2 #] &];
```

Solution of the 1st order perturbation

Numerical solution is now performed using NDSolve

Standard problem with presence of μ term

The two pieces of the linear operator

$$\text{Tra}[\mu\_] := \frac{\left(\mu\, \text{Sin}[\theta[y]]^2 + 2\, \text{Cos}[\theta[y]]^2\, \theta'[y]^2\right)}{\sqrt{\mu\, \text{Sin}[\theta[y]]^2 + \text{Cos}[\theta[y]]^2\, \theta'[y]^2}}$$

$$\text{Trb}[\mu\_] := \frac{\text{Sin}[2\, \theta[y]]\, \theta'[y]}{2\sqrt{\mu\, \text{Sin}[\theta[y]]^2 + \text{Cos}[\theta[y]]^2\, \theta'[y]^2}}$$

The forms of the linear operator

L0 : standard linear operator

L1 : linear operator integrated once outside

L1d : linear operator integrated once outside and inside

L2 : linear operator integrated twice outside

L2d : linear operator integrated twice outside and onc inside

```
L0[f_, μ_] := ∂y,y ( Tra[μ] ∂y,y f ) + μ ∂y,y (Trb[μ] ∂y f)
L1[f_, μ_] := ∂y ( Tra[μ] ∂y,y f ) + μ ∂y (Trb[μ] ∂y f)
L1d[f_, μ_] := ∂y ( Tra[μ] ∂y f ) + μ ∂y (Trb[μ] f)
L2[f_, μ_] := (Tra[μ] ∂y,y f ) + μ (Trb[μ] ∂y f)
L2d[f_, μ_] := (Tra[μ] ∂y f ) + μ (Trb[μ] f)
```

Solution for h1 and g1

Find solution for h1 where $\frac{1}{C}\, h1 = \partial_y f1$

Since f1 has Sin[y] symmetry, h1 has Cos[y] symmetry

h1p is an interpolating function obtained for a value of μ and a given flow.

Define also g1 such that $\frac{1}{C}\, g1 = f1$

```
h1p =
  Part[NDSolve[{L2d[h1[y], #1] == -Sin[θ[y]] /. #2, h1[π/2] == 0},
     h1, {y, 0, π}], 1, 1] &;
```

Obsolete versions to obtain g1 as a second step after getting h1

```
mg1[s_?NumberQ, μ_, Flow_] :=
 NIntegrate[h1[y] /. h1p[μ, Flow], {y, 0, s}]
mgg1 =
  Part[NDSolve[{g1'[y] == h1[y] /. h1p[#1, #2], g1[0] == 0}, g1,
     {y, 0, π/2}], 1, 1] &;
```

New version, obtaining g1 directly since NDSolve is now able to handle boundary value problems

Notice that no integration constant appear in this problem

```
mgg1 =
  Part[NDSolve[{L2d[g1'[y], #1] == -Sin[θ[y]] /. #2, g1'[π/2] == 0,
      g1[0] == 0}, g1, {y, 0, π/2}], 1, 1] &;
```

Plot of the solution for FloW1, Flow2 and Flow3

```
Ploth1[μ_] := Module[{R1, R2, R3},
  R1 = h1p[μ, Flow1];
  R2 = h1p[μ, Flow2];
  R3 = h1p[μ, Flow3];
  {Plot[h1[x] /. R1, {x, 0, π}], Plot[h1[x] /. R2, {x, 0, π}],
   Plot[h1[x] /. R3, {x, 0, π}]}]
```

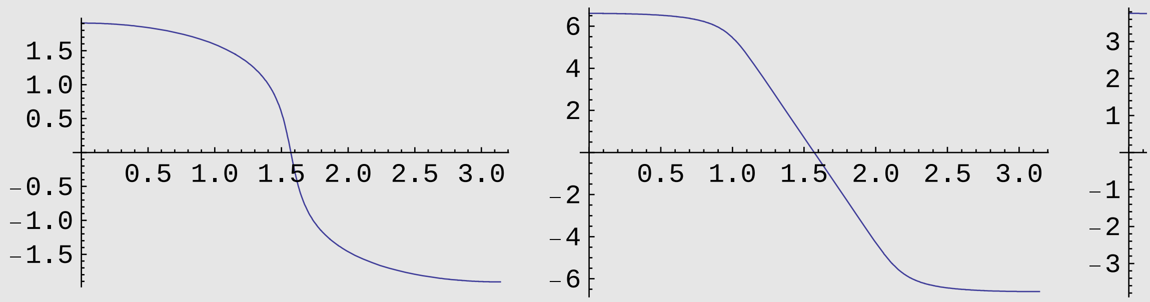
Following line produces dynamic plot of h1[y] (slow)

```
DynamicModule[{z = 0.},
 {Slider[Dynamic[z], {-2, 2}], Dynamic[10^z], Dynamic[Ploth1[10^z]]}]
```

- **plot cell for printing**
- **Plots the functions h1 and g1 for the three selected flows and as a function of μ**

Plot h1

```
GraphicsGrid[{Ploth1[0.01]}]
```

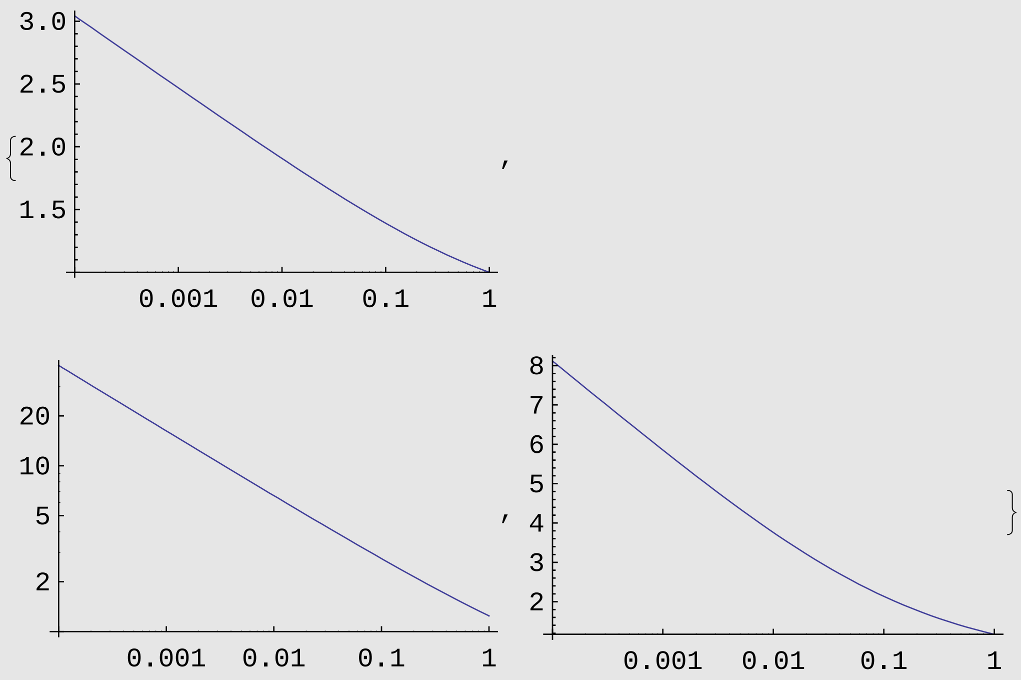



```
h10[μ_, Flow_] := h1[0] /. h1p[μ, Flow];
```

Notice that h1[0] increases as - Log[μ]pour Flow1 et Flow3 et en puissance de μ pour Flow2

Plot now h1[0] as a function of μ

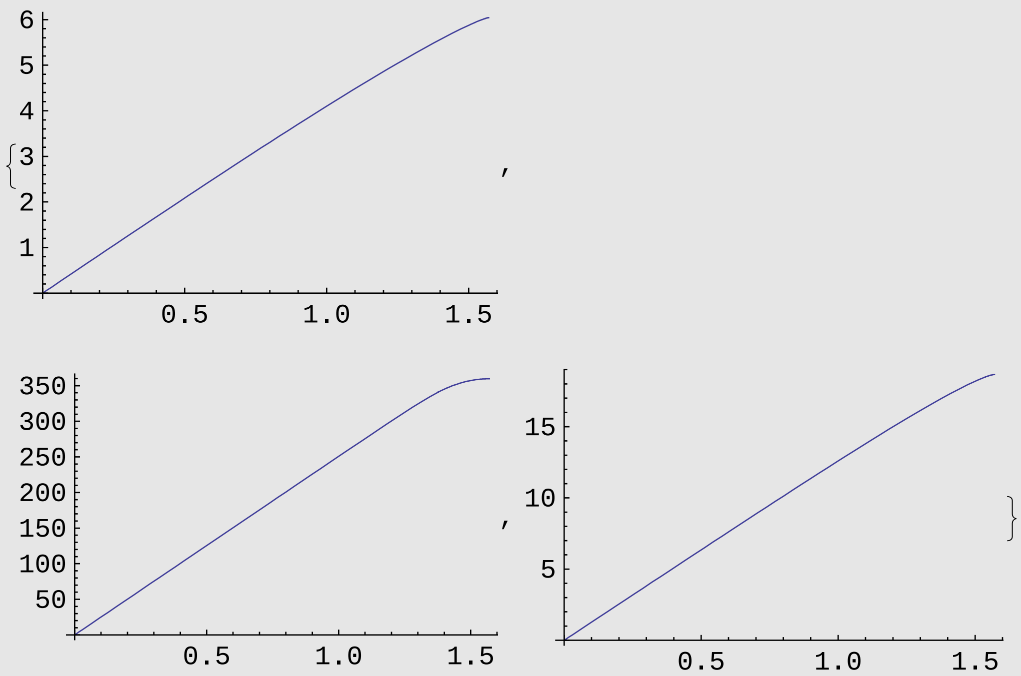
```
{LogLinearPlot[h10[m, Flow1], {m, 0.0001, 1.}],
 LogLogPlot[h10[m, Flow2], {m, 0.0001, 1.}],
 LogLinearPlot[h10[m, Flow3], {m, 0.0001, 1.}]}
```



Following line produces dynamic plot of g1[y] (slow)

```
DynamicModule[{z = 0.},
 {Slider[Dynamic[z], {-2, 2}], Dynamic[10^z], Dynamic[Plotg1[10^z]]}]
```

```
Plotg1[0.000001]
```



■ **Produce printing plots**

Solution to the second order perturbation problem for f2b

Generation of the source functions

$$\texttt{Src2b1} = \frac{\left(\mu\, \texttt{Sin}[\theta[y]]^2 + 2\, \texttt{Cos}[\theta[y]]^2\, \theta'[y]^2\right)}{\sqrt{\mu\, \texttt{Sin}[\theta[y]]^2 + \texttt{Cos}[\theta[y]]^2\, \theta'[y]^2}};$$

```
Src2b2 = Sin[θ[y]] g1'[y] - Cos[θ[y]] θ'[y] g1[y] ;
Src2b3 = Sin[θ[y]] g1'[y] θ'[y]^2 +
   Cos[θ[y]] (θ'[y] g1''[y] - θ''[y] g1'[y]);
```

The problem to be solved at this order is

$$\texttt{L0 (f2b)} = \partial_{y^2}\, \texttt{Src2b1} + \frac{1}{C}\, \partial_y\, \texttt{Src2b2} + \frac{D}{C}\, \partial_y\, \texttt{Src2b3}$$

Notice that Src2b2 and Src2b3 vanish for Flow1 and $\mu = 1$, when θ [y]=y and g1[y]=Sin[y]

Plot of the source term

Solution using NDSolve

```
The solution is
f2b = g2b1 + (1/C) g2b2 + (D/C) g2b3
Since only the derivative of f2b is
 needed to get the amplitude equation, we define
and calculate
h2b1 = g2b1', h2b2 = g2b2', h2b3 = g2b3'
```

Solution of the problem for h2b1 and g2b1

Solution for h2b1 with an auxilliary integration constant K such that the solvability condition and the boundary conditions are satisfied

The integration constant appears here because h2b1 has Sin 2 y symmetry resulting in a constant contribution by the product with Trb

```
h2b1p =
  Part[NDSolve[{L2d[h2b1[y], #1] == Src2b1 - K[y] /. {#2, μ → #1},
      K'[y] == 0, h2b1[0] == 0, h2b1[π/2] == 0}, {h2b1, K}, {y, 0, π/2}],
    1, 1] &;
```

This provides directly g2b1 with the trick that g2b1 is the derivative of a function with same symmetries as h2b1 in order to impose the solvability condition in a way that NDSolve understands

```
mgg2b1 =
  Part[NDSolve[{L2d[g2b1'[y], #1] == Src2b1 - K[y] /. {#2, μ → #1},
      K'[y] == 0, g2b1'[0] == 0, g2b1'[π/2] == 0, IG'[y] == g2b1[y],
      IG[0] == 0, IG[π/2] == 0}, {g2b1, K, IG}, {y, 0, π/2}], 1, 1] &;
```

Plot of the solution for h2b1 and g2b1

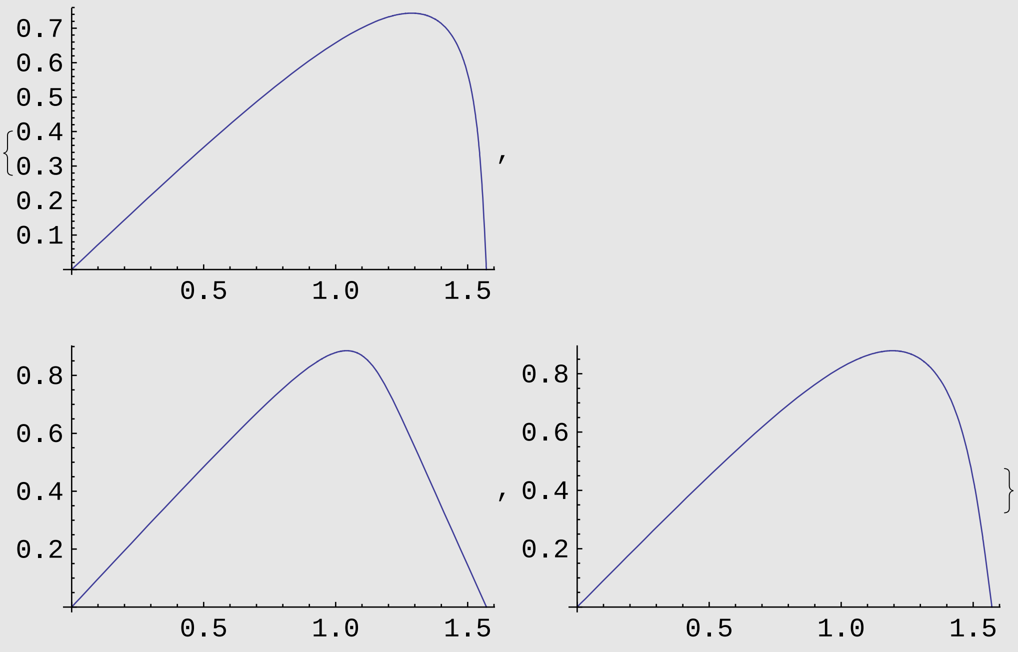
```
Ploth2b1[μ0_] := Module[{R1, R2, R3},
  R1 = h2b1p[μ0, Flow1];
  R2 = h2b1p[μ0, Flow2];
  R3 = h2b1p[μ0, Flow3];
  {Plot[h2b1[y] /. R1 , {y, 0, π/2}], Plot[h2b1[y] /. R2 , {y, 0, π/2}],
   Plot[h2b1[y] /. R3 , {y, 0, π/2}]}
 ]
```

```
Plotg2b1[μ0_] := Module[{R1, R2, R3},
  R1 = mgg2b1[μ0, Flow1];
  R2 = mgg2b1[μ0, Flow2];
  R3 = mgg2b1[μ0, Flow3];
  {Plot[g2b1[y] /. R1 , {y, 0, π/2}], Plot[g2b1[y] /. R2 , {y, 0, π/2}],
   Plot[g2b1[y] /. R3 , {y, 0, π/2}]}
 ]
```

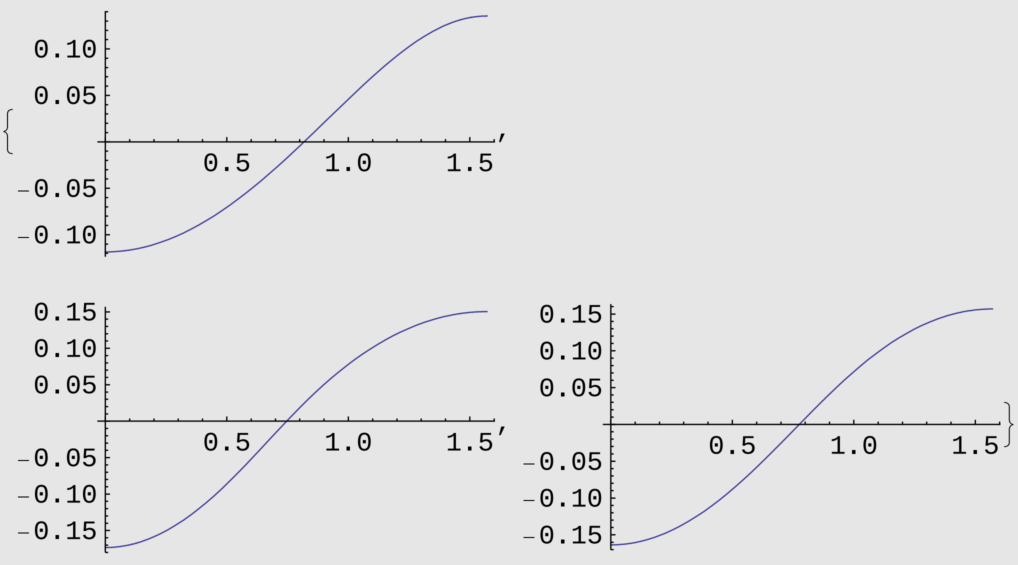
Following line produces dynamic plot of the solution (very slow)

```
DynamicModule[{z = 0.},
 {Slider[Dynamic[z], {-2, 2}], Dynamic[10^z], Dynamic[Plotg2b1[10^z]]}]
```

```
Ploth2b1[0.001]
```



```
Plotg2b1[0.5]
```



- **Verification against the case $\mu = 1, \theta[y] = y$**

- **Solution for h2b2 and g2b2**

Because there is only only derivative in front of Scr2b2, we solve the linear operator L1d and we do not need to introduce an integration constant here

```
h2b2p =
  Part[NDSolve[{L1d[h2b2[y], #1] == Src2b2 /. mgg1[#1, #2] /. #2,
      h2b2[0] == 0, h2b2[π/2] == 0}, h2b2, {y, 0, π/2}], 1, 1] &;
```

```
mgg2b2 =
  Part[NDSolve[{L1d[g2b2'[y], #1] == Src2b2 /. mgg1[#1, #2] /. #2,
      g2b2'[0] == 0, g2b2'[π/2] == 0, IG'[y] == g2b2[y], IG[0] == 0,
      IG[π/2] == 0}, {g2b2, IG}, {y, 0, π/2}], 1, 1] &;
```

- **Plot of the solution for h2b2 and g2b2**
- **Solution for h2b3 and g2b3**

Again, because there is only only derivative in front of Scr2b2, we solve the linear operator L1d and we do not need to introduce an integration constant here

```
h2b3p =
  Part[NDSolve[{L1d[h2b3[y], #1] == Src2b3 /. mgg1[#1, #2] /. #2,
      h2b3[0] == 0, h2b3[π/2] == 0}, h2b3, {y, 0, π/2}], 1, 1] &;
```

```
mgg2b3 =
  Part[NDSolve[{L1d[g2b3'[y], #1] == Src2b3 /. mgg1[#1, #2] /. #2,
      g2b3'[0] == 0, g2b3'[π/2] == 0, IG'[y] == g2b3[y], IG[0] == 0,
      IG[π/2] == 0}, {g2b3, IG}, {y, 0, π/2}], 1, 1] &;
```

- **Plot of the solution for h2b3 and g2b3**

Contributions to the turbulent viscosity

Contribution of the zero and first order terms

- Term in < $\frac{\left(\mu \,\text{Sin}[\theta[y]]^2 + 2\,\text{Cos}[\theta[y]]^2\, \theta'[y]^2\right)}{\sqrt{\mu \,\text{Sin}[\theta[y]]^2 + \text{Cos}[\theta[y]]^2\, \theta'[y]^2}}$ >, in factor of A C

```
I1[μ_, Flow_] := 2/π NIntegrate[Tra[μ] /. Flow, {y, 0, π/2}]
```

- How does it look and example
- Term in < g1[y] Sin[θ [y]]>, in factor of - $\frac{A}{C}$

```
I2[μ_, Flow_] :=
 2/π NIntegrate[ g1[y] Sin[θ[y]] /. mgg1[μ, Flow] /. Flow, {y, 0, π/2}]
```

- How does it look and example
- Term in < g1'[y] Cos[θ [y]]θ '[y]>, in factor of - $\frac{A\,D}{C}$

```
I3[μ_, Flow_] :=
 2/π NIntegrate[ g1'[y] Cos[θ[y]] θ'[y] /. mgg1[μ, Flow] /. Flow,
   {y, 0, π/2}]
```

- How does it look and example

Definition : the famous function G

```
G[y_, μ_, Flow_] :=
 2 θ'[y] Cos[θ[y]]
   ( -Sin[θ[y]] θ'[y]^2 / Sqrt[μ Sin[θ[y]]^2 + Cos[θ[y]]^2 θ'[y]^2] +
     1/(2 (μ Sin[θ[y]]^2 + Cos[θ[y]]^2 θ'[y]^2)^(3/2))
      (3 μ θ''[y] Sin[θ[y]]^2 Cos[θ[y]] - μ θ'[y]^2 Sin[θ[y]] +
        2 θ'[y]^2 θ''[y] Cos[θ[y]]^3) ) /. Flow
```

One can easily check that this expression is identical to the following one,

```
1/(μ Sin[θ[y]]^2 + Cos[θ[y]]^2 θ'[y]^2)^(3/2) (Cos[θ[y]] θ'[y]
   (-Sin[θ[y]] θ'[y]^2 (μ + 2 μ Sin[θ[y]]^2 + 2 Cos[θ[y]]^2 θ'[y]^2) +
     Cos[θ[y]] (3 μ Sin[θ[y]]^2 + 2 Cos[θ[y]]^2 θ'[y]^2) θ''[y]))
```

or this one taken from CH - eff1 - nu - 8. m6.nb

```
(Cos[θ[y]] θ'[y]
   (-2 Cos[θ[y]]^2 Sin[θ[y]] θ'[y]^4 + 3 μ Cos[θ[y]] Sin[θ[y]]^2 θ''[y] -
     θ'[y]^2 (μ (Sin[θ[y]] + 2 Sin[θ[y]]^3) - 2 Cos[θ[y]]^3 θ''[y]))) /
 (μ Sin[θ[y]]^2 + Cos[θ[y]]^2 θ'[y]^2)^(3/2)
```

or this one in the paper

```
1/(2 (μ Sin[θ[y]]^2 + Cos[θ[y]]^2 θ'[y]^2)^(3/2))
 (D[D[Sin[θ[y]], y]^4, y] + μ D[Sin[θ[y]]^2, y]
   (3 Sin[θ[y]] D[Sin[θ[y]], {y, 2}] - D[Sin[θ[y]], y]^2))
```

■ Plots

Contribution of the terms in f2b

- **Contribution < h2b1[y] G[y] >, in factor of $2\ \frac{A}{C}$**

```
Ib1[μ_, Flow_] :=
 2/π NIntegrate[h2b1[y] G[y, μ, Flow] /. Evaluate[h2b1p[μ, Flow]],
   {y, 0, π/2}]
```

- **How does it look and example**
- **Contribution < h2b2[y] G[y] >, in factor of $2\ A$ C**

```
Ib2[μ_, Flow_] :=
 2/π NIntegrate[h2b2[y] G[y, μ, Flow] /. Evaluate[h2b2p[μ, Flow]],
   {y, 0, π/2}]
```

- **How does it look and example**
- **Contribution < h2b3[y] G[y] >, in factor of $2\ A$ D**

```
Ib3[μ_, Flow_] :=
 2/π NIntegrate[h2b3[y] G[y, μ, Flow] /. Evaluate[h2b3p[μ, Flow]],
   {y, 0, π/2}]
```

- **How dos it look and example**

Large-scale instability

The turbulent viscosity ν T is such that

$$\frac{1}{A}\nu T = \frac{1}{C}(-I2 + Ib2) + C\,(I1 + Ib1) + \frac{D}{C}(-I3 + Ib3)$$

$$= \Delta^2\,Cs\,(Cs\,(I1 + Ib1)) + \frac{1}{12\,Cs}(Ib3 - I3) + \frac{1}{Cs\,\Delta^2}(-I2 + Ib2)$$

Verifications with respect to the analytical case for Flow1 and μ =1

Definitions

```
B1[μ_, Flow_] := I1[μ, Flow] + Ib1[μ, Flow]
B2[μ_, Flow_] := -I2[μ, Flow] + Ib2[μ, Flow]
B3[μ_, Flow_] := (-I3[μ, Flow] + Ib3[μ, Flow]) / 12
```

```
DD[μ_, Flow_] := B3[μ, Flow]^2 - 4 Cs^2 B1[μ, Flow] B2[μ, Flow] /. Cs → 0.008;
```

```
rc[μ_, Flow_] := Sqrt[(-B3[μ, Flow] + Sqrt[DD[μ, Flow]]) / (2 Cs^2 B1[μ, Flow])] /. Cs → 0.008;
```

Plots

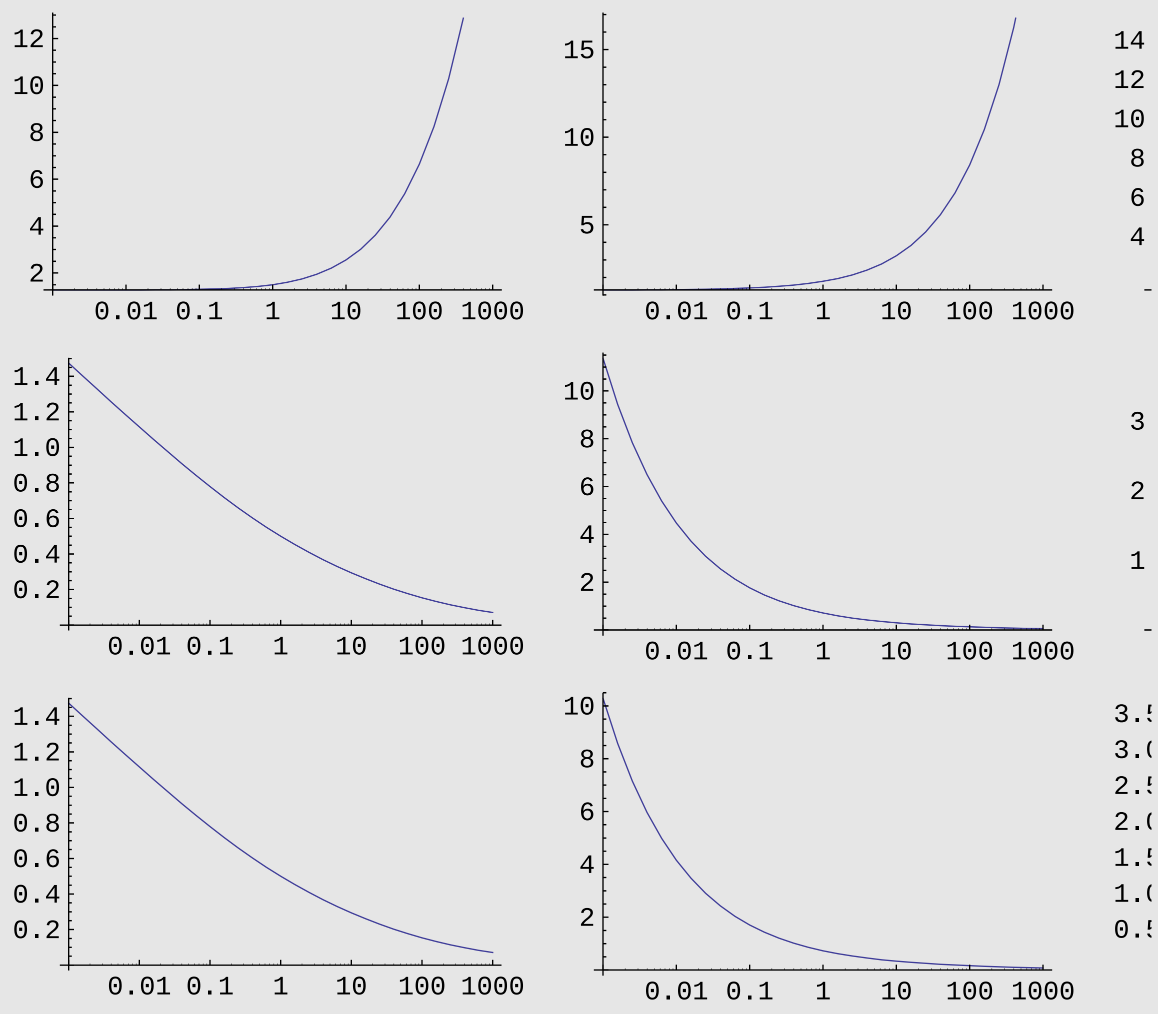
■ Plots with lists

Make a logatithmic sample of μ values

```
fac = 10^(1/5);
μramp = Table[0.001 fac^(i-1), {i, 1, 31}];
```

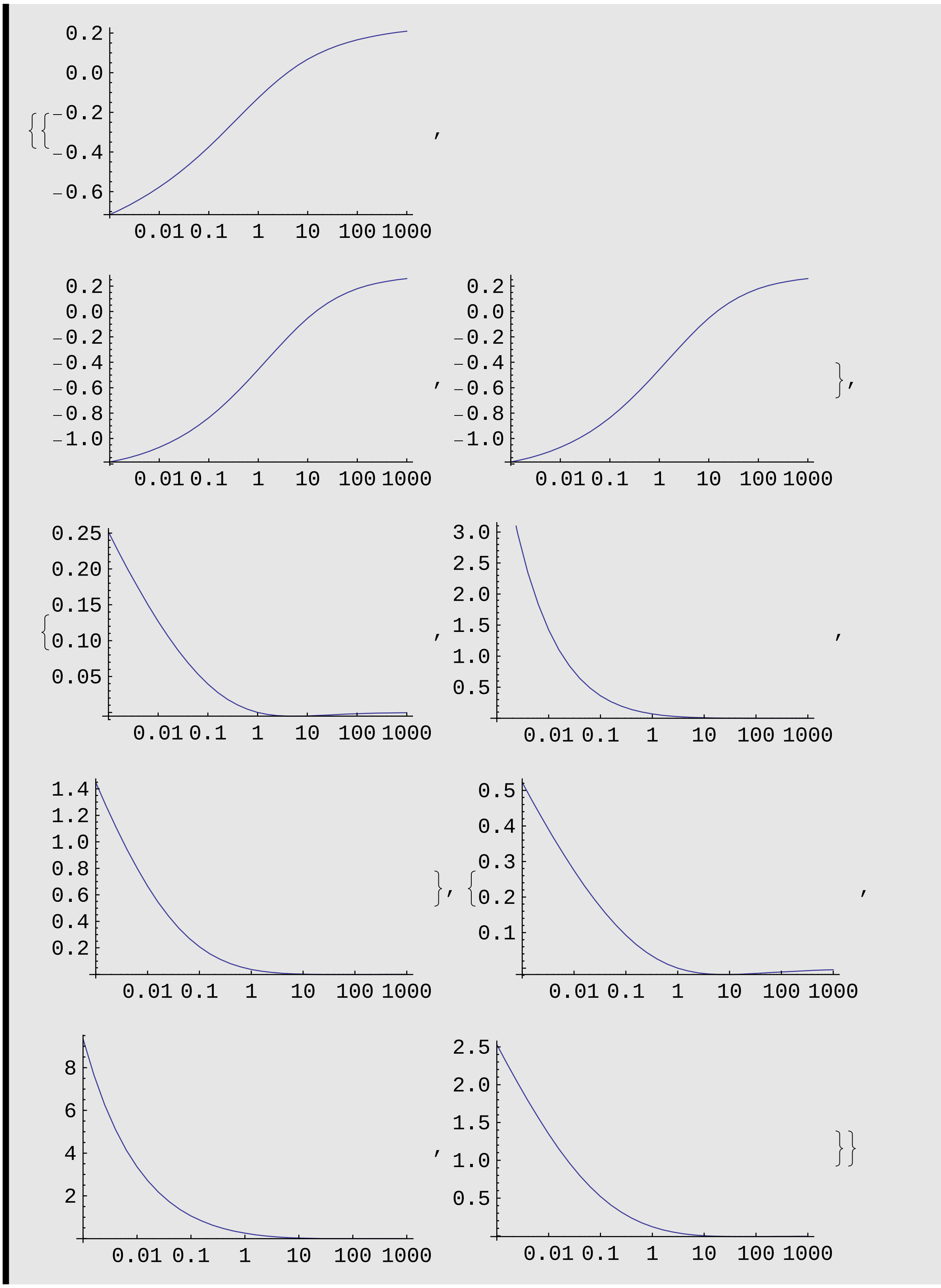
Plot integrals I1, I2 and I3 as a function of μ and Flow

```
GraphicsGrid[
 {{ListLogLinearPlot[Transpose[{μramp, Map[I1[#, Flow1] &, μramp]}],
    Joined → True],
   ListLogLinearPlot[Transpose[{μramp, Map[I1[#, Flow2] &, μramp]}],
    Joined → True],
   ListLogLinearPlot[Transpose[{μramp, Map[I1[#, Flow3] &, μramp]}],
    Joined → True]},
  {ListLogLinearPlot[Transpose[{μramp, Map[I2[#, Flow1] &, μramp]}],
    Joined → True],
   ListLogLinearPlot[Transpose[{μramp, Map[I2[#, Flow2] &, μramp]}],
    Joined → True],
   ListLogLinearPlot[Transpose[{μramp, Map[I2[#, Flow3] &, μramp]}],
    Joined → True]},
  {ListLogLinearPlot[Transpose[{μramp, Map[I3[#, Flow1] &, μramp]}],
    Joined → True],
   ListLogLinearPlot[Transpose[{μramp, Map[I3[#, Flow2] &, μramp]}],
    Joined → True],
   ListLogLinearPlot[Transpose[{μramp, Map[I3[#, Flow3] &, μramp]}],
    Joined → True]}}]
```



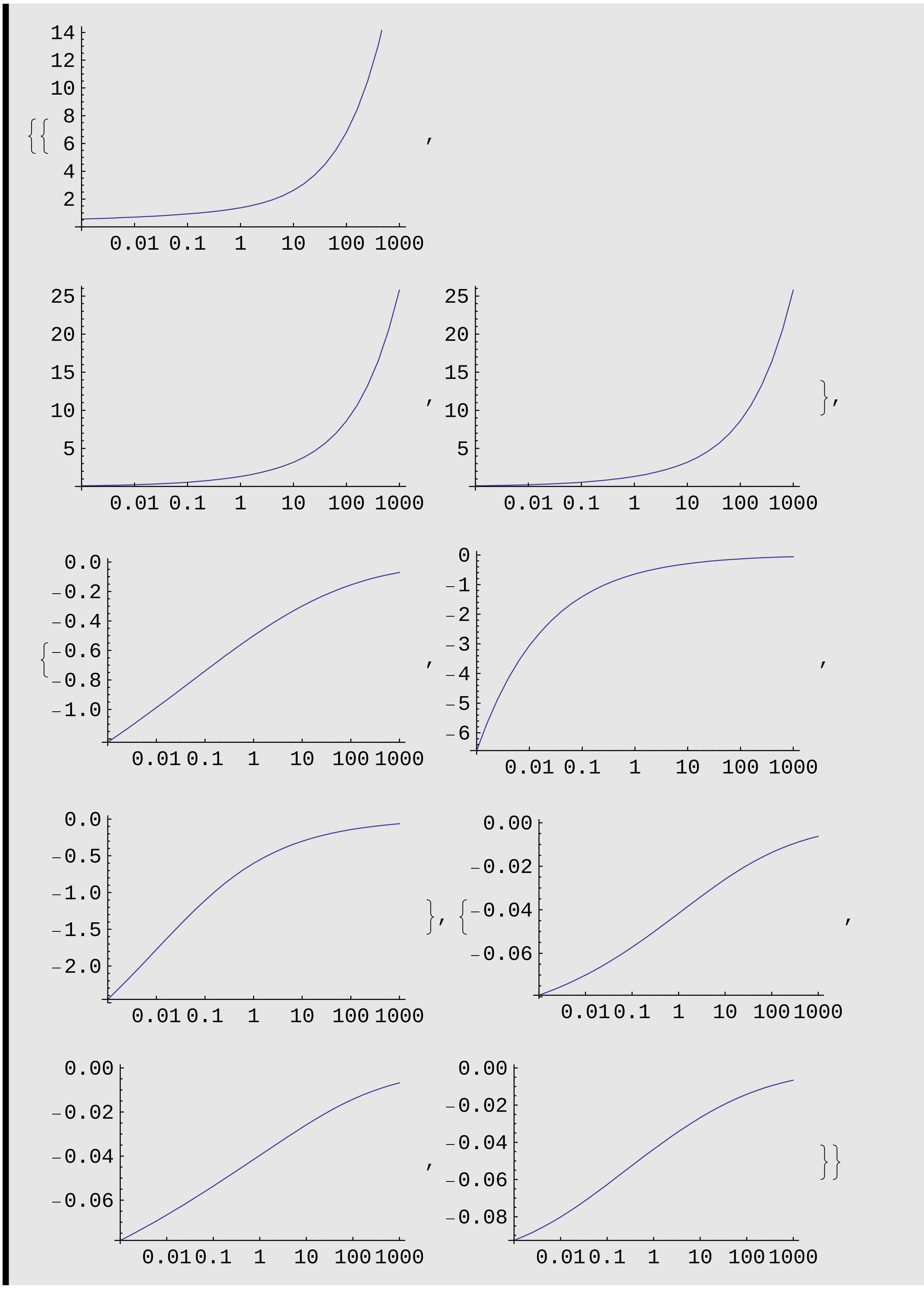
Plot integrals Ib1, Ib2 and Ib3 as a function of μ and Flow

```
{{ListLogLinearPlot[Transpose[{μramp, Map[Ib1[#, Flow1] &, μramp]}],
   Joined → True],
  ListLogLinearPlot[Transpose[{μramp, Map[Ib1[#, Flow2] &, μramp]}],
   Joined → True],
  ListLogLinearPlot[Transpose[{μramp, Map[Ib1[#, Flow2] &, μramp]}],
   Joined → True]},
 {ListLogLinearPlot[Transpose[{μramp, Map[Ib2[#, Flow1] &, μramp]}],
   Joined → True],
  ListLogLinearPlot[Transpose[{μramp, Map[Ib2[#, Flow2] &, μramp]}],
   Joined → True],
  ListLogLinearPlot[Transpose[{μramp, Map[Ib2[#, Flow3] &, μramp]}],
   Joined → True]},
 {ListLogLinearPlot[Transpose[{μramp, Map[Ib3[#, Flow1] &, μramp]}],
   Joined → True],
  ListLogLinearPlot[Transpose[{μramp, Map[Ib3[#, Flow2] &, μramp]}],
   Joined → True],
  ListLogLinearPlot[Transpose[{μramp, Map[Ib3[#, Flow3] &, μramp]}],
   Joined → True]}}
```



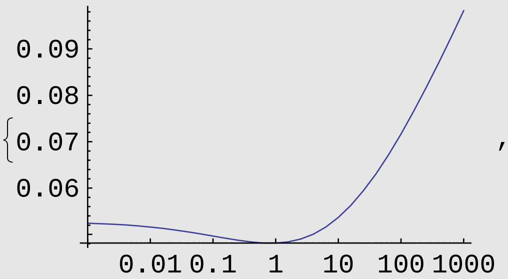
Plot integrals B1, B2 and B3 as a function of μ and Flow

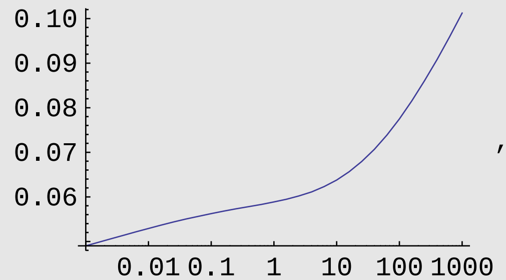
```
{{ListLogLinearPlot[Transpose[{μramp, Map[B1[#, Flow1] &, μramp]}],
   Joined → True],
  ListLogLinearPlot[Transpose[{μramp, Map[B1[#, Flow2] &, μramp]}],
   Joined → True],
  ListLogLinearPlot[Transpose[{μramp, Map[B1[#, Flow2] &, μramp]}],
   Joined → True]},
 {ListLogLinearPlot[Transpose[{μramp, Map[B2[#, Flow1] &, μramp]}],
   Joined → True],
  ListLogLinearPlot[Transpose[{μramp, Map[B2[#, Flow2] &, μramp]}],
   Joined → True],
  ListLogLinearPlot[Transpose[{μramp, Map[B2[#, Flow3] &, μramp]}],
   Joined → True]},
 {ListLogLinearPlot[Transpose[{μramp, Map[B3[#, Flow1] &, μramp]}],
   Joined → True],
  ListLogLinearPlot[Transpose[{μramp, Map[B3[#, Flow2] &, μramp]}],
   Joined → True],
  ListLogLinearPlot[Transpose[{μramp, Map[B3[#, Flow3] &, μramp]}],
   Joined → True]}}
```

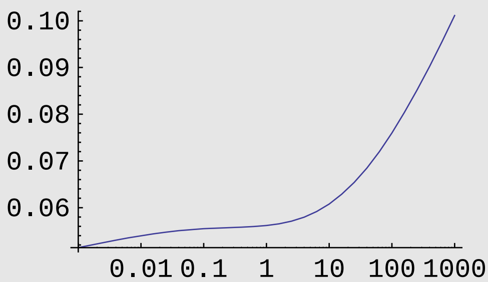


Plot the discriminant as a function of μ and Flow

```
{ListLogLinearPlot[Transpose[{μramp, Map[DD[#, Flow1] &, μramp]}],
  Joined → True],
 ListLogLinearPlot[Transpose[{μramp, Map[DD[#, Flow2] &, μramp]}],
  Joined → True],
 ListLogLinearPlot[Transpose[{μramp, Map[DD[#, Flow3] &, μramp]}],
  Joined → True]}
```

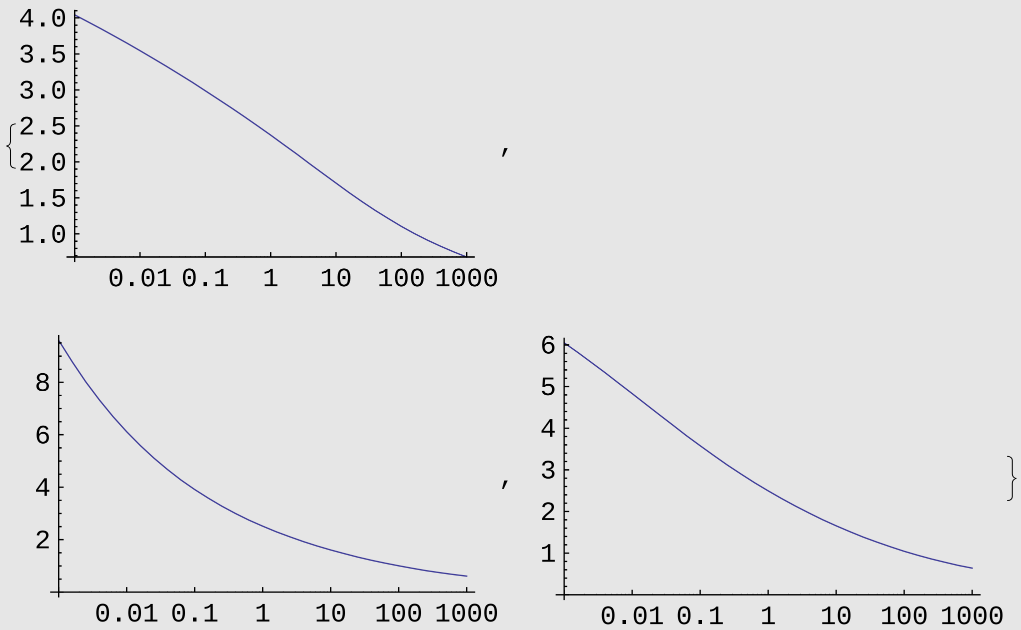






Plot the critical length as a function of μ and Flow

```
{ListLogLinearPlot[Transpose[{μramp, Map[Γc[#, Flow1] &, μramp]}],
  Joined → True],
 ListLogLinearPlot[Transpose[{μramp, Map[Γc[#, Flow2] &, μramp]}],
  Joined → True],
 ListLogLinearPlot[Transpose[{μramp, Map[Γc[#, Flow3] &, μramp]}],
  Joined → True]}
```



■ Produce printing plots

```
q1 = Map[Γc[#, Flow1] &, μramp];
q2 = Map[Γc[#, Flow2] &, μramp];
q3 = Map[Γc[#, Flow3] &, μramp];
```

Let us print to make plots at a later stage

```
q1
q2
q3
```

```
{47.2826, 45.8484, 44.3773, 42.8678, 41.3195, 39.7324, 38.1069, 36.4441,
 34.7459, 33.0145, 31.2519, 29.4605, 27.6419, 25.7978, 23.9306,
 22.0449, 20.1497, 18.2601, 16.3985, 14.593, 12.8743, 11.2715, 9.80784,
 8.49761, 7.3455, 6.34741, 5.4924, 4.7653, 4.14908, 3.62683, 3.18299}
```

```
{113.202, 102.06, 91.9566, 82.7999, 74.5048, 66.9947, 60.1986, 54.0523,
 48.4967, 43.4771, 38.9432, 34.8483, 31.1493, 27.8059, 24.7811, 22.0413,
 19.5566, 17.3013, 15.2543, 13.3991, 11.7234, 10.2183, 8.87684, 7.69208,
 6.65604, 5.75865, 4.98769, 4.32932, 3.76902, 3.29253, 2.88669}
```

```
{75.0683, 71.4809, 67.806, 64.0572, 60.2541, 56.4218, 52.5903, 48.7927,
 45.0629, 41.4328, 37.9308, 34.5799, 31.3964, 28.3906, 25.5667, 22.9248,
 20.4624, 18.1758, 16.0622, 14.1201, 12.3494, 10.7504, 9.32244, 8.06231,
 6.96328, 6.01488, 5.20341, 4.51315, 3.92768, 3.43108, 3.00886}
```

```
ListLogLinearPlot[{Transpose[{μramp, q1}], Transpose[{μramp, q2}],
  Transpose[{μramp, q3}]}, Joined → True,
 PlotStyle → {{Black, Thick}, {Black, Thick, Dashed},
   {Black, Thick, Dotted}},
 AxesLabel → {ToExpression["\mu", TeXForm],
   ToExpression["\Gamma_c", TeXForm]}]
```