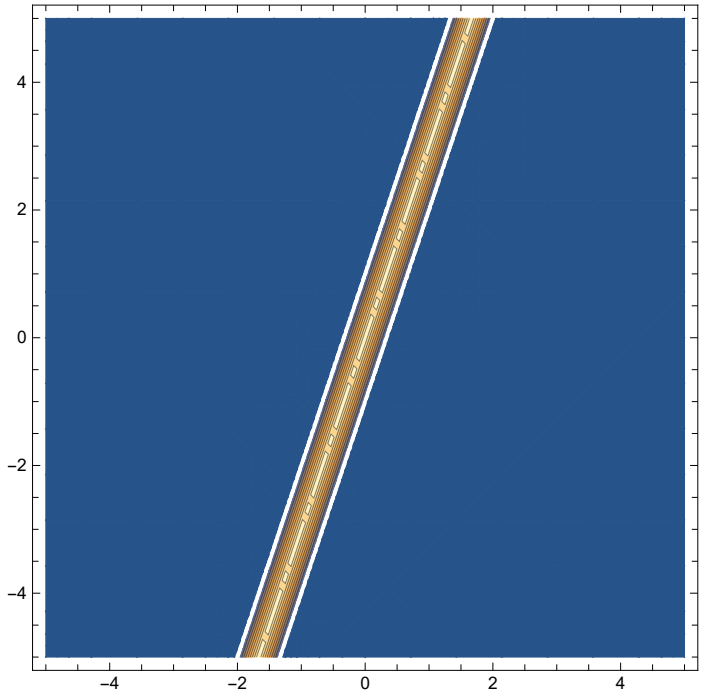
6(3)

In[ ]:=
```
eqn = D[u[t, x], t] + 3 * D[u[t, x], x] == 0;
ic = {u[0, x] == tri[x]};
```

In[ ]:=
```
u63a = DSolve[{ D[u[t, x], t] + 3 D[u[t, x], x] == 0, u[0, x] == tri[x]}, u, {t, x}];
```

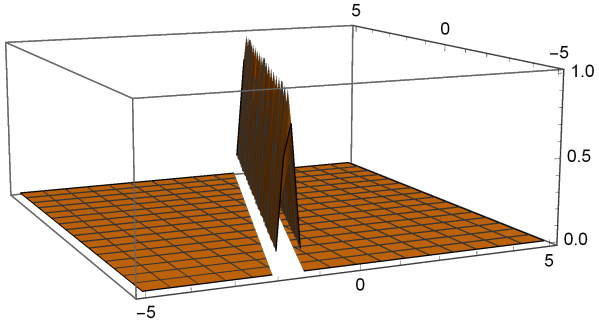
In[ ]:=
```
ContourPlot[Evaluate[u[t, x] /. u63a], {t, -5, 5}, {x, -5, 5}, PerformanceGoal → "Quality"]
```

Out[ ]=



In[ ]:=
```
Plot3D[Evaluate[u[t, x] /. u63a], {t, -5, 5}, {x, -5, 5}, PerformanceGoal → "Quality"]
```

Out[ ]=



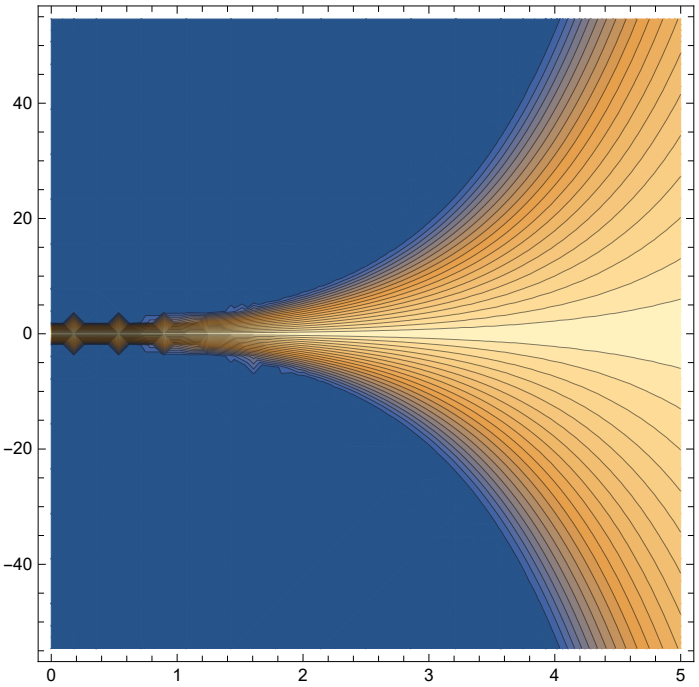
In[ ]:=
```
u63b = NDSolve[{D[u[t, x], t] + x * D[u[t, x], x] == 0, u[0, x] == tri[x]},
    u, {t, 0, 5}, {x, -Exp[5], Exp[5]}];
```

NDSolve: Using maximum number of grid points 10000 allowed by the MaxPoints or MinStepSize options for independent variable x.

NDSolve: Warning: an insufficient number of boundary conditions have been specified for the direction of independent variable x. Artificial boundary effects may be present in the solution.

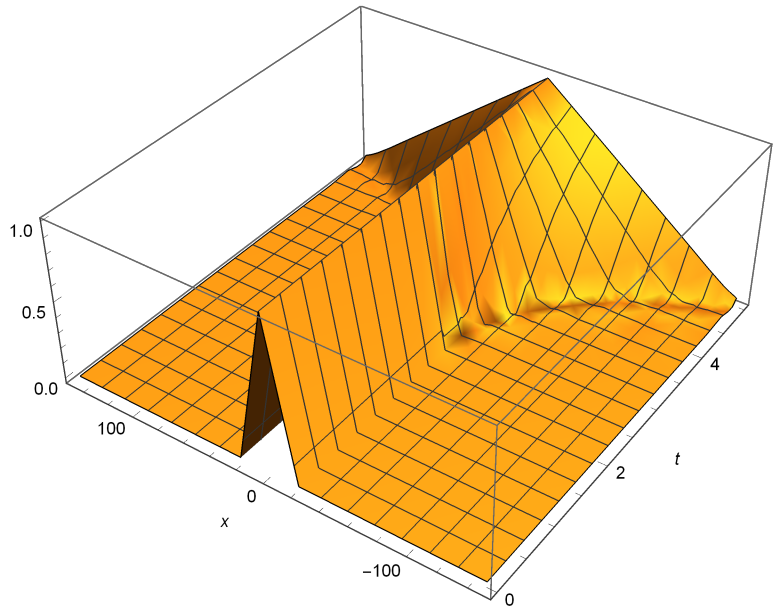
In[•]:=
```
ContourPlot[Evaluate[u[t, x] /. u63b], {t, 0, 5}, {x, -Exp[4], Exp[4]},
 PerformanceGoal → "Quality", AxesLabel → {t, x}, Contours → 20]
```

Out[•]=



In[•]:=
```
Plot3D[Evaluate[u[t, x] /. u63b], {t, 0, 5}, {x, -Exp[5], Exp[5]},
 PerformanceGoal → "Quality", AxesLabel → {t, x}, PlotRange → Full]
```

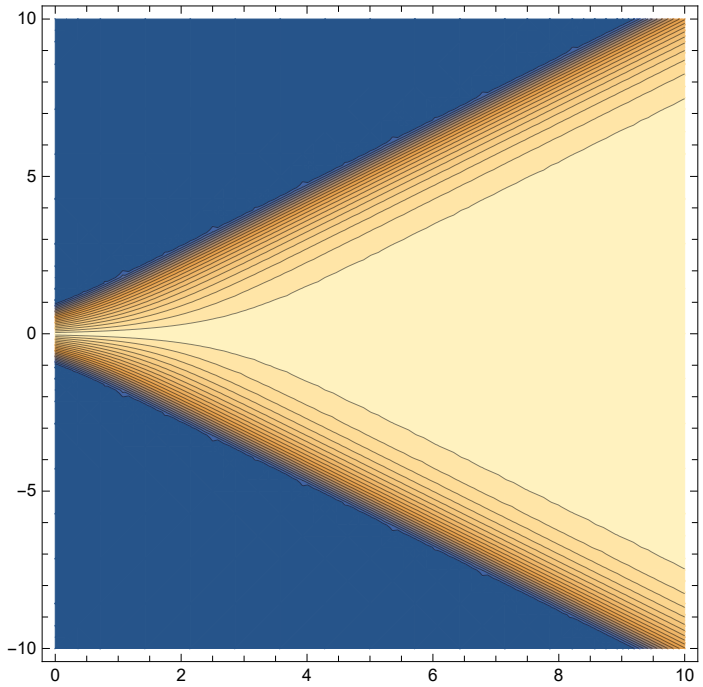
Out[•]=



In[•]:=
```
u63c = NDSolve[{D[u[t, x], t] + Tanh[x] * D[u[t, x], x] == 0, u[0, x] == tri[x]},
   u, {t, 0, 10}, {x, -10, 10}];
```

In[ ]:=
```
ContourPlot[Evaluate[u[t, x] /. u63c], {t, 0, 10}, {x, -10, 10},
 PerformanceGoal → "Quality", AxesLabel → {t, x}, Contours → 20]
```

Out[ ]=



In[ ]:=
```
u63d = NDSolve[{D[u[t, x], t] - Tanh[x] * D[u[t, x], x] == 0, u[0, x] == tri[x]},
   u, {t, -10, 0}, {x, -10, 10}];
```

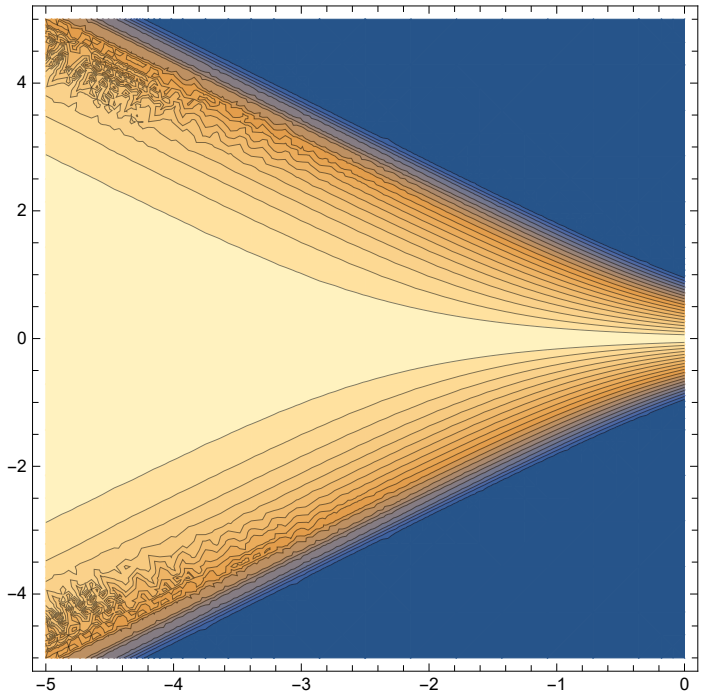
... NDSolve: Using maximum number of grid points 10000 allowed by the MaxPoints or MinStepSize options for independent variable x.

... NDSolve: Warning: an insufficient number of boundary conditions have been specified for the direction of independent variable x. Artificial boundary effects may be present in the solution.

... NDSolve: Warning: scaled local spatial error estimate of 1068.7322596928088` at t = −10. in the direction of independent variable x is much greater than the prescribed error tolerance. Grid spacing with 10001 points may be too large to achieve the desired accuracy or precision. A singularity may have formed or a smaller grid spacing can be specified using the MaxStepSize or MinPoints method options.

In[ ]:=
```
ContourPlot[Evaluate[u[t, x] /. u63d], {t, -5, 0}, {x, -5, 5},
 PerformanceGoal → "Quality", AxesLabel → {t, x}, Contours → 20]
```

Out[ ]=



In[ ]:=
```
u63e = NDSolve[
    {D[u[t, x], t] + u[t, x] * D[u[t, x], x] == 0, u[0, x] == tri[x]}, u, {t, -5, 5}, {x, -5, 5}];
```

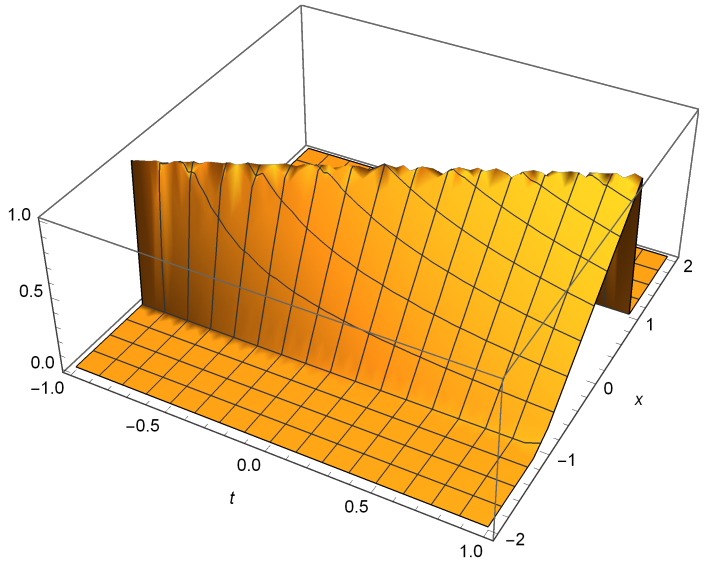
NDSolve: At t == 2.4011503558198757`, step size is effectively zero; singularity or stiff system suspected.

NDSolve: Warning: scaled local spatial error estimate of 4.2885779030897796`*^8 at t = −1.02179 in the direction of independent variable x is much greater than the prescribed error tolerance. Grid spacing with 10001 points may be too large to achieve the desired accuracy or precision. A singularity may have formed or a smaller grid spacing can be specified using the MaxStepSize or MinPoints method options.

NDSolve: Warning: scaled local spatial error estimate of 1.1384713954639986`*^10 at t = 2.4011503558198757` in the direction of independent variable x is much greater than the prescribed error tolerance. Grid spacing with 10001 points may be too large to achieve the desired accuracy or precision. A singularity may have formed or a smaller grid spacing can be specified using the MaxStepSize or MinPoints method options.

In[ ]:=
```
Plot3D[Evaluate[u[t, x] /. u63e], {t, -1, 1}, {x, -2, 2},
 PerformanceGoal → "Quality", AxesLabel → {t, x}, Contours → 20]
```

Out[ ]=

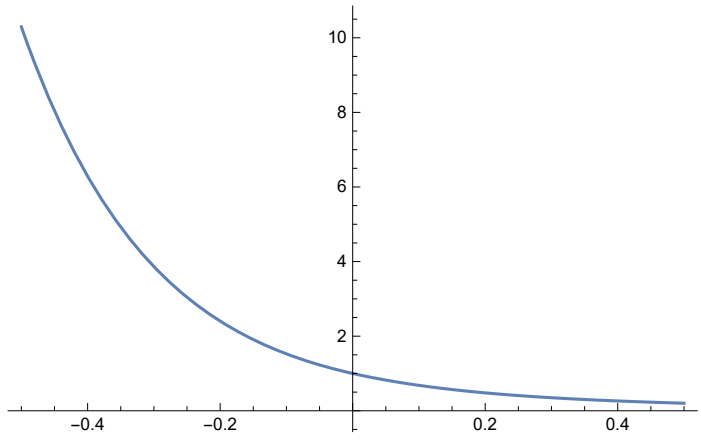


7(2) Visualization

In[ ]:=
```
u[t_] := Exp[-5 t] + Exp[-5 t] * Integrate[tri[x] * Exp[5 x], {x, 0, t}]
```

In[ ]:=
```
Plot[u[t], {t, -.5, .5}]
```

Out[ ]=



In[ ]:=
```
xi[x_, t_] := x + 5 t; tau[x_, t_] := x - 5 t;
u7c[xi1_, tau1_] := 1/10 tri[xi1] * tau1;
```

In[ ]:=
```
Plot3D[u7c[xi[x, t], tau[x, t]], {t, -1, 1}, {x, -1, 1}]
```

Out[ ]=