TradingStrategiesII

Empirical Market Microstructure

(2006, Oxford University Press)

Companion *Mathematica* notebook

Joel Hasbrouck

This notebook covers order submission strategies based on diffusion barrier models

(Chapter 15, section 15.2).

```
Text @ Style["Notebook evaluated " <> DateString["DateTime"], "Subtitle"]
```

Notebook evaluated Friday 1 June 2007 21:07:10

Preliminaries

Other initializations

```
<< PlotLegends`
```

```
baseStyle = {FontFamily → "Times", FontSize → 12};
```

```
<< Notation`
```

The following commands define symbolizations that are convenient for labeling things.

```
Symbolize[Anything_Rule]; Symbolize[Anything_Rules];
```

The following command allows script capital E to be entered as ESC E ESC.

```
AddInputAlias[ℰ, "E"]
```

```
AddInputAlias[ℓ, "ll"]
```

```
Symbolize[ɣ_];
```

Additional Notations

```
Symbolize[Pr_];
Symbolize[f_];
Symbolize[Eδ_];
```

ϕ[x], Φ[X], etc.

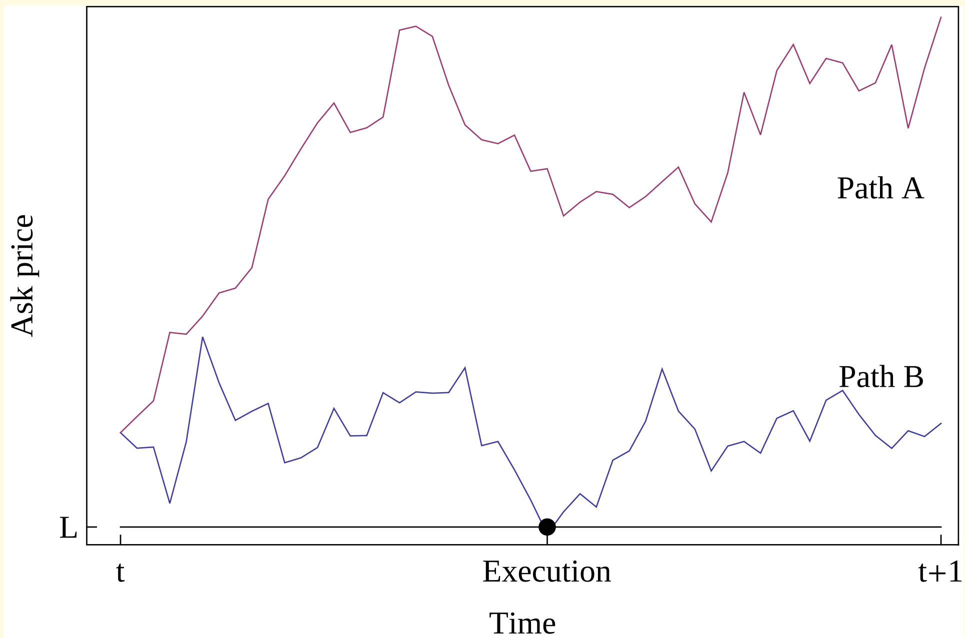
ϕ[X] and Φ[X] are often used to denote the density and distribution functions for the standard normal distribution.

```
NormalRules = {
   ϕ[x_] :→ PDF[NormalDistribution[0, 1], x],
   Φ[x_] :→ CDF[NormalDistribution[0, 1], x],
   ϕ[x_, μ_, var_] :→ PDF[NormalDistribution[μ, √var], x],
   Φ[x_, μ_, var_] :→ CDF[NormalDistribution[μ, √var], x]};
```

Graph depicting execution mechanism

```
SeedRandom[1234];
L = 7.5;
n = 50;
p =
  FoldList[Plus, 10, #] & /@ (.1 + RandomReal[NormalDistribution[0, 1], {2, n}]);
ExecTime = First @@ Position[p[[1]], First[Select[p[[1]], # ≤ L &]]];
```

```
ListPlot[p, Joined → True, Frame → True, FrameTicks →
  {{{1, "t"}, {ExecTime, "Execution"}, {n + 1, "t+1"}}, {{L, "L"}}, None, None},
 FrameLabel → {"Time", "Ask price"}, Background → GrayLevel[1],
 Epilog → {Line[{{1, L}, {n + 1, L}}],
   Text["Path A", {n, L + 9}, {1, 0}], Text["Path B", {n, L + 4}, {1, 0}],
   PointSize[0.02`], Point[{ExecTime, L}]}, BaseStyle → baseStyle]
```




```
Clear[L, n, p, ExecTime];
```

■ Barrier-diffusion model: Results from Lancaster (1997)

Lancaster, Tony, 1997. The Econometric Analysis of Transition Data (Cambridge University Press, Cambridge).

In Lancaster's framework, the particle starts at zero; the barrier is $\alpha > 0$. The survivor function (his equation 7.9) is:

$$S_{Rule} = S[t\_, \mu\_, \sigma\_, \alpha\_] :> \Phi\left[\frac{\alpha - \mu t}{\sigma \sqrt{t}}\right] - \text{Exp}\left[\frac{2 \mu \alpha}{\sigma^2}\right] \Phi\left[\frac{-\alpha - \mu t}{\sigma \sqrt{t}}\right];$$

Distribution of δ given that barrier α has not yet been hit:

$$g_{Rule} = g[\delta\_, t\_, \mu\_, \sigma\_, \alpha\_] :> \frac{\phi\left[\frac{\delta - \mu t}{\sigma \sqrt{t}}\right] - e^{\frac{2 \mu \alpha}{\sigma^2}} \phi\left[\frac{\delta - 2 \alpha - t \mu}{\sigma \sqrt{t}}\right]}{\sigma \sqrt{t}\; S[t, \mu, \sigma, \alpha]};$$

Expectation of δ given that the barrier α has not yet been hit:

```
EδCond_Rule = EδCond[t_, μ_, σ_, α_] :→ μ t - 2 α Exp[2 μ α / σ^2] Φ[(-α - μ t)/(σ Sqrt[t])] / S[t, μ, σ, α];
```

Reworking of Lancaster results for (buy) limit order analysis in unit time interval

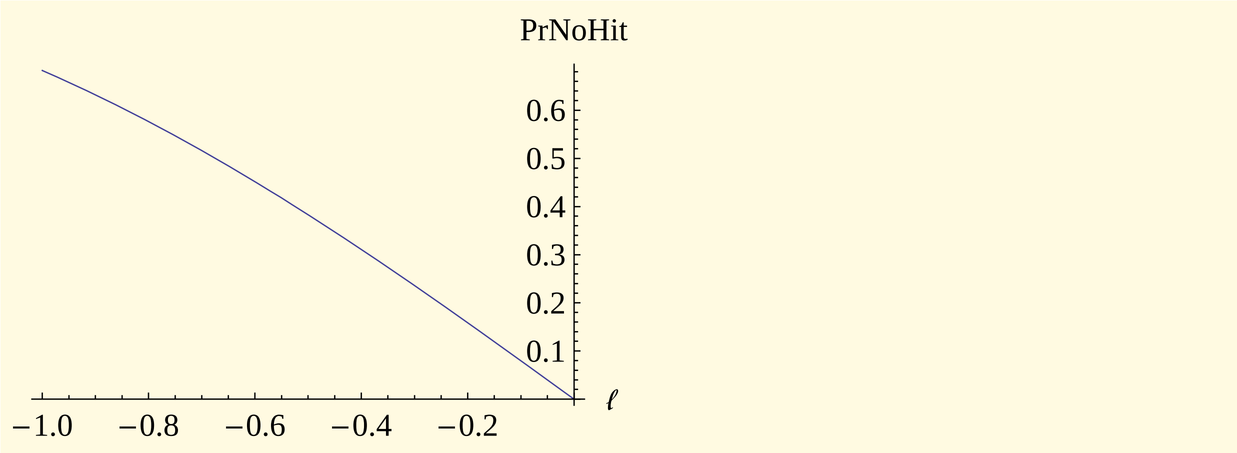
ℓ is the limit price position relative to the current price: $\ell = \mathtt{L} - \mathtt{p}$. (Note $\ell < 0$. Execution probabilities for buy and sell orders are:

```
Pr_Rules = {
   PrNoHit[ℓ_, μ_, σ_] :→ S[1, -μ, σ, -ℓ],
   PrHit[ℓ_, μ_, σ_] :→ 1 - S[1, -μ, σ, -ℓ]};
```

Plots

Probability of no hit

```
Plot[PrNoHit[ℓ, 0, 1] /. Pr_Rules /. S_Rule /. Normal_Rules,
 {ℓ, -1, 0}, AxesLabel → {ℓ, PrNoHit}, BaseStyle → baseStyle]
```




```
lineStyles =
  Table[{GrayLevel[0], Dashing[.01 r], AbsoluteThickness[.5 r]}, {r, 0, 2}];
```

```
legendLines = Table[Graphics[{GrayLevel[0], Dashing[.1 r],
     AbsoluteThickness[.5 r], Line[{{0, 0}, {2, 0}}]}], {r, 0, 2}];
```

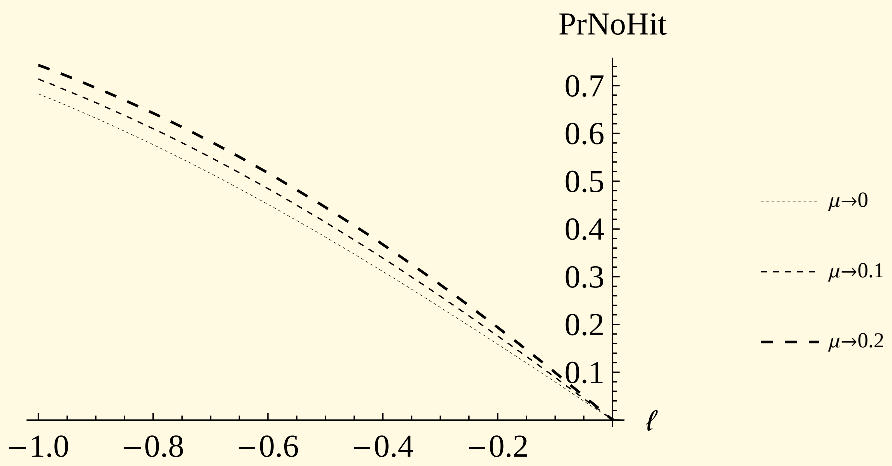
```
legendText =
  Style["μ→" <> ToString[#], Black, FontFamily → "Times"] & /@ {0, 0.1, 0.2};
```

```
legendEntries = Transpose[{legendLines, legendText}];
```

When μ increases, the probability of no hit increases:

```
p1 = Plot[Evaluate[
    (PrNoHit[ℓ, μ, 1] /. PrRules /. SRule /. NormalRules /. μ → #1 &) /@ {0, 0.1`, 0.2`}],
   {ℓ, -1, 0}, PlotStyle → lineStyles, AxesLabel → {ℓ, PrNoHit},
   BaseStyle → baseStyle];
```

```
ShowLegend[p1,
 {legendEntries, LegendPosition → {1, -.4}, LegendSize → 0.6, LegendShadow → None}]
```



□ *Dependence on* σ

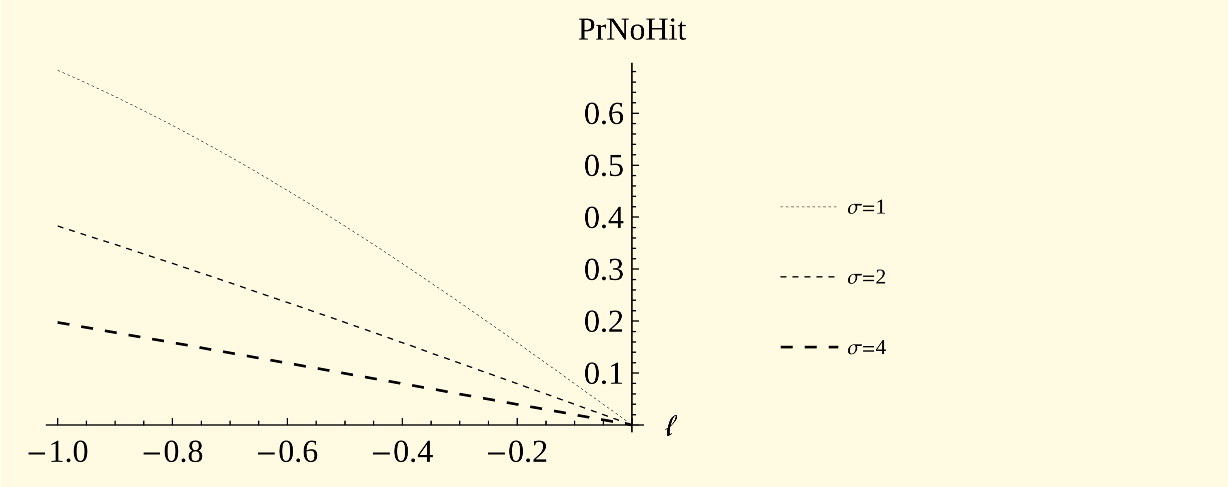
When σ increases, the probability of no hit decreases:

```
p2 = Plot[Evaluate[
    (PrNoHit[ℓ, 0, σ] /. PrRules /. SRule /. NormalRules /. σ → #1 &) /@ {1, 2, 4}],
   {ℓ, -1, 0}, PlotStyle → lineStyles, AxesLabel → {ℓ, PrNoHit},
   BaseStyle → baseStyle];
```

```
legendText = Style["σ=" <> ToString[#], Black, FontFamily → "Times"] & /@ {1, 2, 4};
```

```
legendEntries = Transpose[{legendLines, legendText}];
```

```
ShowLegend[p2,
 {legendEntries, LegendPosition → {1, -.4}, LegendSize → 0.6, LegendShadow → None}]
```



■ Distribution of terminal price given no execution: $p_{t+\triangle t} = p_t + \delta$:

```
fδRule = fδ[δ_, ℓ_, μ_, σ_] :→ g[-δ, 1, -μ, σ, -ℓ];
```

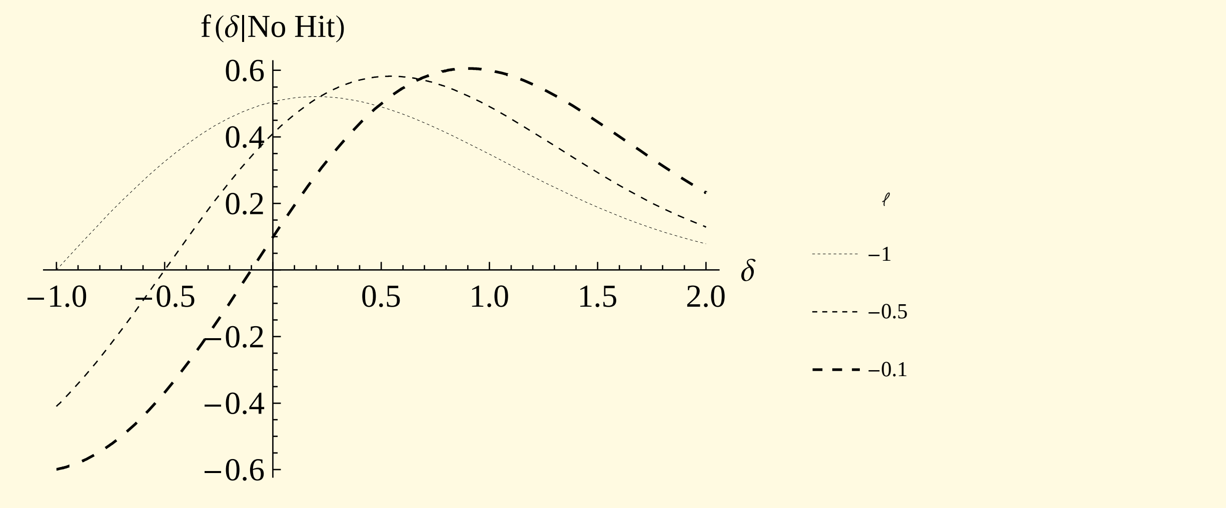
□ *Plots for various limit orders ...*

```
p3 = Plot[Evaluate[(fδ[δ, ℓ, 0, 1] /. fδRule /. gRule /. SRule /. NormalRules /. ℓ → #1 &) /@
     {-1, -0.5`, -0.1`}], {δ, -1, 2}, PlotStyle → lineStyles,
   AxesLabel → {δ, "f(δ|No Hit)"}, BaseStyle → baseStyle];
```

```
legendText =
  Style[ToString[#], Black, FontFamily → "Times"] & /@ {-1, -0.5`, -0.1`};
```

```
legendEntries = Transpose[{legendLines, legendText}];
```

```
ShowLegend[p3, {legendEntries, LegendPosition → {1, -.4},
  LegendSize → 0.6, LegendShadow → None, LegendLabel → "ℓ"}]
```



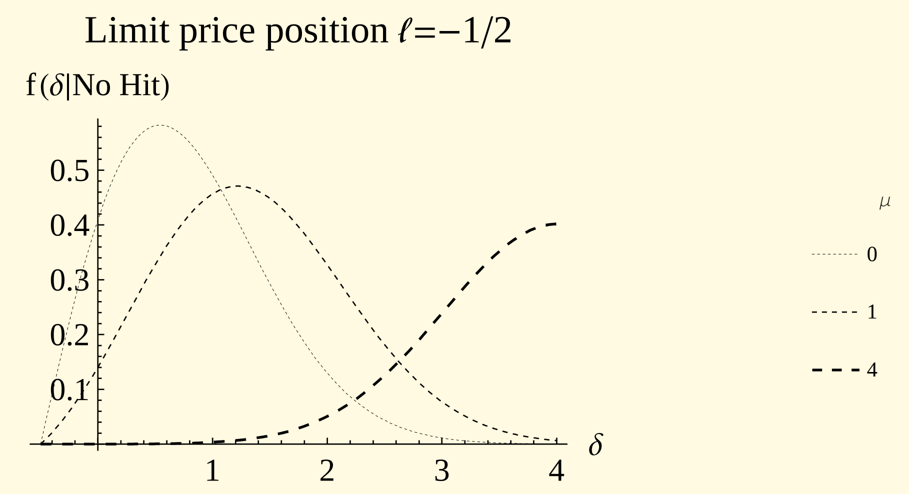
When μ increases, the distribution shifts to the right

```
p4 = Plot[
   Evaluate[(fδ[δ, -0.5`, μ, 1] /. fδRule /. gRule /. SRule /. NormalRules /. μ → #1 &) /@
     {0, 1, 4}], {δ, -1/2, 4}, PlotStyle → lineStyles,
   AxesLabel → {δ, "f(δ|No Hit)"}, PlotLabel → "Limit price position ℓ=-1/2",
   BaseStyle → baseStyle];
```

```
legendText = Style[ToString[#], Black, FontFamily → "Times"] & /@ {0, 1, 4};
```

```
legendEntries = Transpose[{legendLines, legendText}];
```

```
ShowLegend[p4, {legendEntries, LegendPosition → {1, -.4},
  LegendSize → 0.6, LegendShadow → None, LegendLabel → "μ"}]
```



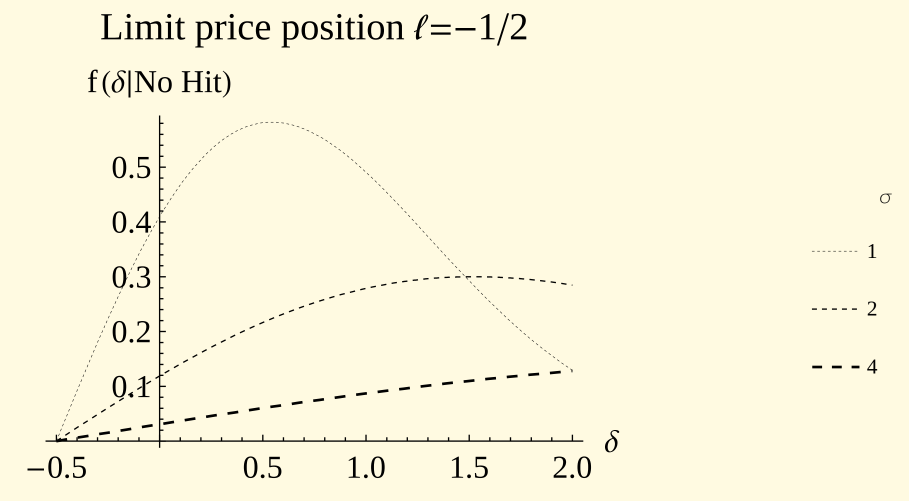
When σ increases, the distribution also shifts upwards

```
p5 = Plot[
   Evaluate[(fδ[δ, -0.5`, 0, σ] /. fδRule /. gRule /. SRule /. NormalRules /. σ → #1 &) /@
     {1, 2, 4}], {δ, -1/2, 2}, PlotStyle → lineStyles,
   AxesLabel → {δ, "f(δ|No Hit)"}, PlotLabel → "Limit price position ℓ=-1/2",
   BaseStyle → baseStyle];
```

```
legendText = Style[ToString[#], Black, FontFamily → "Times"] & /@ {1, 2, 4};
```

```
legendEntries = Transpose[{legendLines, legendText}];
```

```
ShowLegend[p5, {legendEntries, LegendPosition → {1, -.4},
  LegendSize → 0.6, LegendShadow → None, LegendLabel → "σ"}]
```



Expectation of δ conditional on not hitting barrier

```
EδRule = EδNoHit[ℓ_, μ_, σ_] :→ -EδCond[1, -μ, σ, -ℓ];
```

```
EδNoHit[ℓ, μ, σ] /. EδRule /. EδCondRule /. ℓ → L - pt
```

$$\mu - \frac{2\, e^{\frac{2\mu (L-p_t)}{\sigma^2}} (L - p_t)\, \Phi\left[\frac{L+\mu-p_t}{\sigma}\right]}{S[1, -\mu, \sigma, -L + p_t]}$$

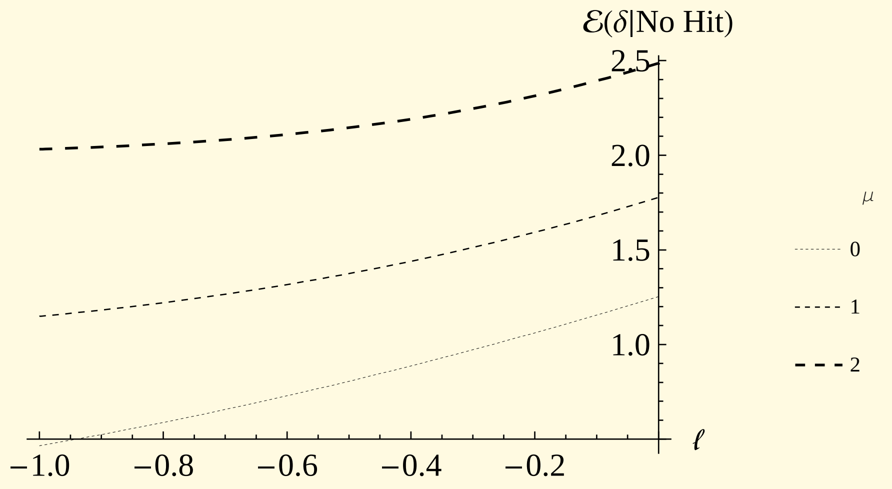
When μ increases, so does Eδ:

```
p6 = Plot[Evaluate[
    (EδNoHit[ℓ, μ, 1] /. EδRule /. EδCondRule /. SRule /. NormalRules /. μ → #1 &) /@
     {0, 1, 2}], {ℓ, -1, 0}, PlotStyle → lineStyles,
   AxesLabel → {ℓ, "ℰ(δ|No Hit)"}, BaseStyle → baseStyle];
```

```
legendText = Style[ToString[#], Black, FontFamily → "Times"] & /@ {0, 1, 2};
```

```
legendEntries = Transpose[{legendLines, legendText}];
```

```
ShowLegend[p6, {legendEntries, LegendPosition → {1, -.4},
  LegendSize → 0.6, LegendShadow → None, LegendLabel → "μ"}]
```



Analysis of pure barrier model (i.e., without c)

```
AllRules = {S_Rule, g_Rule, Eδ_Rule, Pr_Rules, fδ_Rule, EδCond_Rule, Normal_Rules} // Flatten;
```

```
va[ℓ_, μ_: 0, σ_: 1] := ℓ PrHit[ℓ, μ, σ];
```

```
vb[ℓ_, μ_: 0, σ_: 1] := EδNoHit[ℓ, μ, σ] PrNoHit[ℓ, μ, σ]
```

```
v[ℓ_, μ_: 0, σ_: 1] := va[ℓ, μ, σ] + vb[ℓ, μ, σ]
```

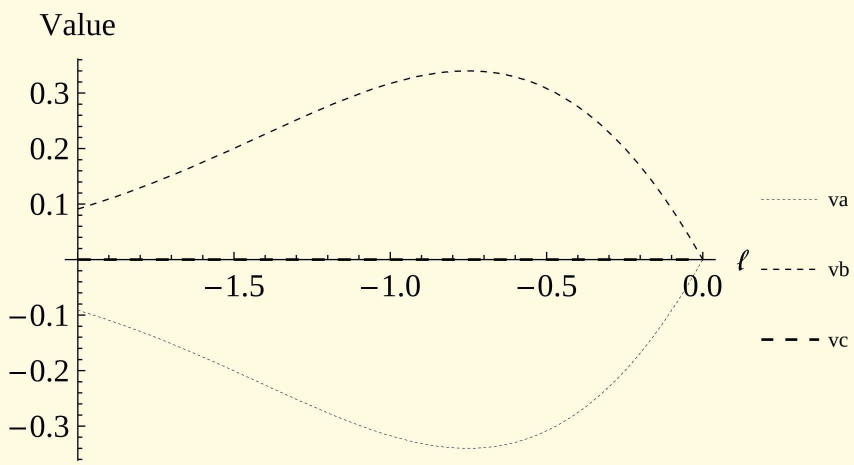
```
GraphStyle = Table[Thickness[.003 x], {x, 1, 3}];
xyLabels = {"ℓ", "Ev)"};
```

```
p7 = Plot[Evaluate[{va[ℓ, 0, 1], vb[ℓ, 0, 1], v[ℓ, 0, 1]} //. AllRules],
   {ℓ, -2, 0}, PlotRange → All, PlotStyle → lineStyles,
   AxesLabel → {ℓ, "Value"}, AxesOrigin → {-2, 0}, BaseStyle → baseStyle];
```

```
legendText =
  Style[ToString[#], Black, FontFamily → "Times"] & /@ {"va", "vb", "vc"};
```

```
legendEntries = Transpose[{legendLines, legendText}];
```

```
ShowLegend[p7,
 {legendEntries, LegendPosition → {1, -.4}, LegendSize → 0.6, LegendShadow → None}]
```

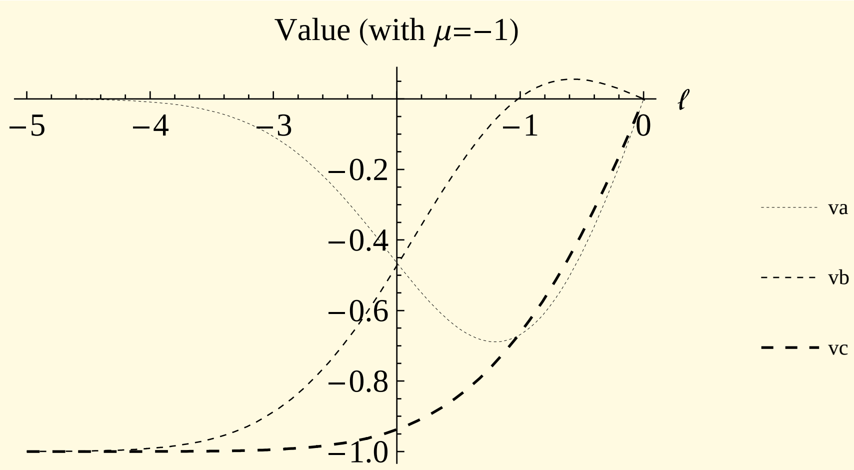


With non-zero drift:

```
f = Evaluate[{va[ℓ, μ, 1], vb[ℓ, μ, 1], v[ℓ, μ, 1]} //. AllRules /. μ → -1];
```

```
p7 = Plot[Evaluate[f], {ℓ, -5, 0}, PlotRange → All,
   PlotStyle → lineStyles, AxesLabel → {ℓ, "Value (with μ=-1)"},
   AxesOrigin → {-2, 0}, BaseStyle → baseStyle];
```

```
ShowLegend[p7,
 {legendEntries, LegendPosition → {1, -.4}, LegendSize → 0.6, LegendShadow → None}]
```




```
Clear[v, f]
```

Analysis with exponential c

Distribution

```
fcRule = fc[c_, λ_] :→ PDF[ExponentialDistribution[λ], c];
```

```
AllRules = {SRule, gRule, EδRule, PrRules, EδCondRule, NormalRules, fcRule} // Flatten;
```

```
ModelAssumptions =
  {ℓ ∈ Reals, ℓ < 0, λ ∈ Reals, λ > 0, σ ∈ Reals, σ > 0, c ∈ Reals, c ≥ 0};
```

Objective functions *(This may take a minute or two to evaluate, due to the complexity of the integrals.)*

```
va[ℓ_, λ_, μ_, σ_] := Evaluate[
   Integrate[ℓ fc[c, λ] /. AllRules, {c, -ℓ, ∞}, Assumptions → ModelAssumptions]];
```

```
vb[ℓ_, λ_, μ_, σ_] := Evaluate[Integrate[ℓ fc[c, λ] PrHit[ℓ + c, μ, σ] //. AllRules,
    {c, 0, -ℓ}, Assumptions → ModelAssumptions]];
```

```
vc[ℓ_?NumberQ, ω_, λ_, μ_, σ_] := NIntegrate[
   (EδNoHit[ℓ + c, μ, σ] + ω) PrNoHit[ℓ + c, μ, σ] fc[c, λ] //. AllRules, {c, 0, -ℓ}];
```

```
v[ℓ_?NumberQ, ω_: 0, λ_: 5, μ_: 0, σ_: 1] :=
  va[ℓ, λ, μ, σ] + vb[ℓ, λ, μ, σ] + vc[ℓ, ω, λ, μ, σ];
```

The following are not necessary to solve the optimization, but are useful descriptive constructs:

```
PrNoHit[ℓ_?NumberQ, λ_, μ_: 0, σ_: 1] := NIntegrate[
  Evaluate[PrNoHit[ℓ + c, μ, σ] fc[c, λ] //. AllRules // Simplify], {c, 0, -ℓ}]
```

```
PrNoHit[-.1, 5, 0, 1]
```

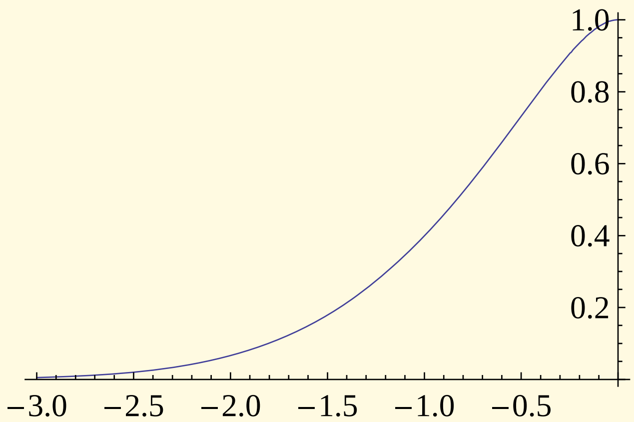
```
0.0169848
```

```
PrHit[ℓ_?NumberQ, λ_, μ_: 0, σ_: 1] := 1 - PrNoHit[ℓ, λ, μ, σ]
```

```
PrHit[-.1, 5, 0, 1]
```

```
0.983015
```

```
Plot[PrHit[ℓ, 5, 0.1`, 1], {ℓ, -3, 0}, BaseStyle → baseStyle]
```

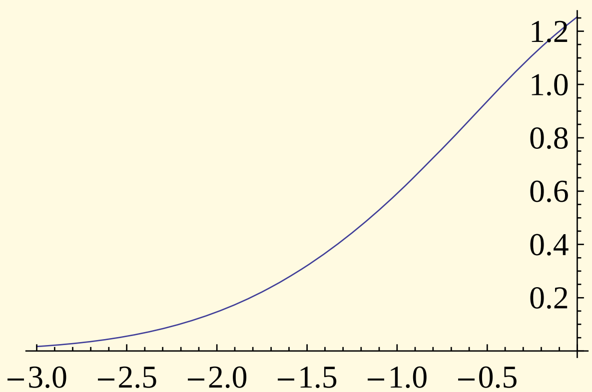



```
EδNoHit[ℓ_?NumberQ, λ_, μ_, σ_] :=
  NIntegrate[Evaluate[EδNoHit[c + ℓ, μ, σ] fc[c, λ] //. AllRules], {c, 0, -ℓ}]
  ----------------------------------------------------------------------------
             NIntegrate[Evaluate[fc[c, λ] //. AllRules], {c, 0, -ℓ}]
```

```
EδNoHit[-.001, 5, 0, 1]
```

```
1.25281
```

```
Plot[EδNoHit[ℓ, 5, 0, 1], {ℓ, -3, 0}, BaseStyle → baseStyle]
```



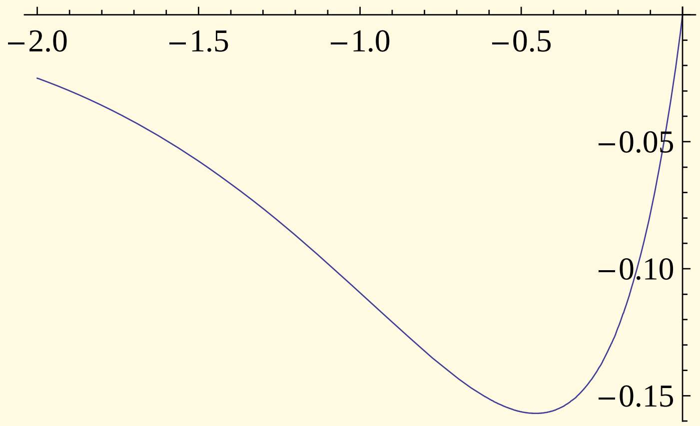
Explore the value function

```
v[-0.1]
```

```
-0.0781731
```

The value function (with default parameter values);

```
Plot[v[ℓ], {ℓ, -2, 0}, BaseStyle → baseStyle]
```




```
FindMinimum[v[ℓ], {ℓ, -.5}, Method → Gradient]
```

```
{-0.156961, {ℓ → -0.45505}}
```

General backwards recursive solution

```
LimitSolve[T_, λ_: 5, μ_: 0, σ_: 1] :=
 Module[{ω = 0, r = {}, s, vOpt, ℓOpt, ProbExec, EPriceNoExec},
  Do[
   s = FindMinimum[v[ℓ, ω, λ, μ, σ], {ℓ, -0.01}, Method -> "Gradient"];
   (*Print["t:",t," Solution:", s];*)
   vOpt = s[[1]];
   ℓOpt = ℓ /. s[[2, 1]];
   ProbExec = PrHit[ℓOpt, λ, μ, σ];
   EPriceNoExec = EδNoHit[ℓOpt, λ, μ, σ];
   r = Append[r, {t, vOpt, ℓOpt, ProbExec, EPriceNoExec}];
   ω = vOpt;,
   {t, T, 1, -1}
  ];
  r
 ]
```

```
(r = LimitSolve[5]) // TableForm
```

5	−0.156961	−0.454949	0.784865	0.968808
4	−0.202964	−0.743495	0.588816	0.763124
3	−0.224957	−0.984499	0.439646	0.600758
2	−0.238444	−1.19301	0.329362	0.474908
1	−0.248001	−1.36388	0.253566	0.384019

Graphs

Plots of value functions by ℓ

```
(r = LimitSolve[5]) // TableForm
```

5	−0.156961	−0.454949	0.784865	0.968808
4	−0.202964	−0.743495	0.588816	0.763124
3	−0.224957	−0.984499	0.439646	0.600758
2	−0.238444	−1.19301	0.329362	0.474908
1	−0.248001	−1.36388	0.253566	0.384019

```
optCoordinates = Table[{r[[i, 3]], r[[i, 2]]}, {i, Length[r]}];
```

```
graphNotes = Prepend[(Point[#] & /@ optCoordinates), PointSize[0.02]];
```

```
ω = Prepend[Drop[r[[All, 2]], -1], 0];
```

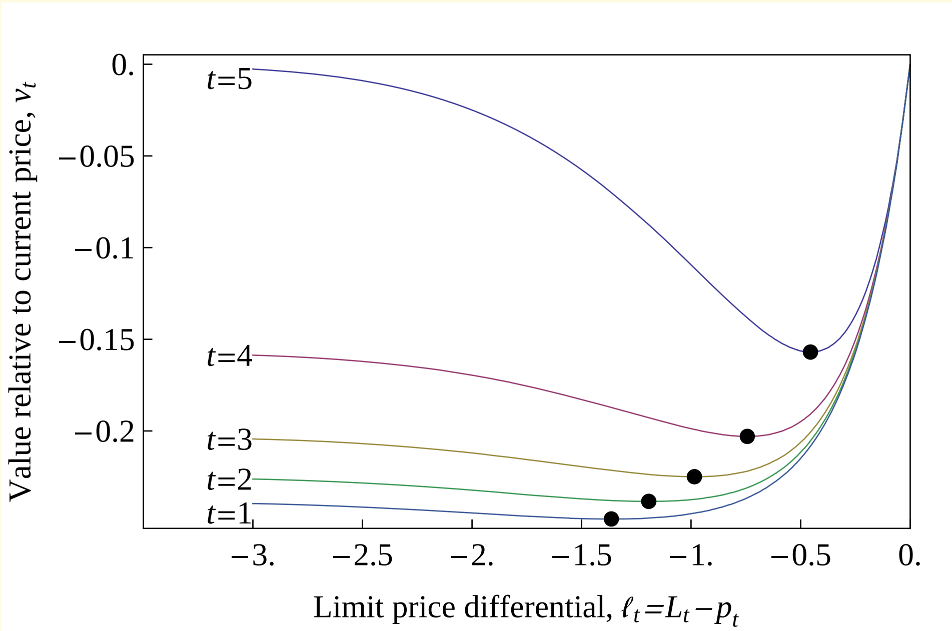
```
graphText = {
   Text["t=5", {-3, v[-3, ω[[1]] - .005]}, {1, 0}],
   Text["t=4", {-3, v[-3, ω[[2]]]}, {1, 0}],
   Text["t=3", {-3, v[-3, ω[[3]]]}, {1, 0}],
   Text["t=2", {-3, v[-3, ω[[4]]]}, {1, 0}],
   Text["t=1", {-3, v[-3, ω[[5]] - .005}, {1, 0}]};
```

```
graphAdd = Join[graphNotes, graphText];
```

```
ℓLabel = "Limit price differential, ℓt=Lt-pt";
```

```
ValueLabel = "Value relative to current price, vt";
```

```
Plot[Evaluate[(v[ℓ, #1] &) /@ Prepend[Drop[r⟦All, 2⟧, -1], 0]],
 {ℓ, -3, 0}, PlotRange → {{-3.5`, 0}, Automatic}, Frame → True,
 FrameLabel → {ℓLabel, ValueLabel, " ", " "}, Background → GrayLevel[1],
 FrameTicks → {Range[0, -3, -0.5`], Range[0, -0.2`, -0.05`], None, None},
 Epilog → graphAdd, BaseStyle → baseStyle]
```



☐ *Plots by μ*

```
μVals = Range[0, .1, .01];
```

```
r = LimitSolve[5, 5, #] & /@ μVals;
```

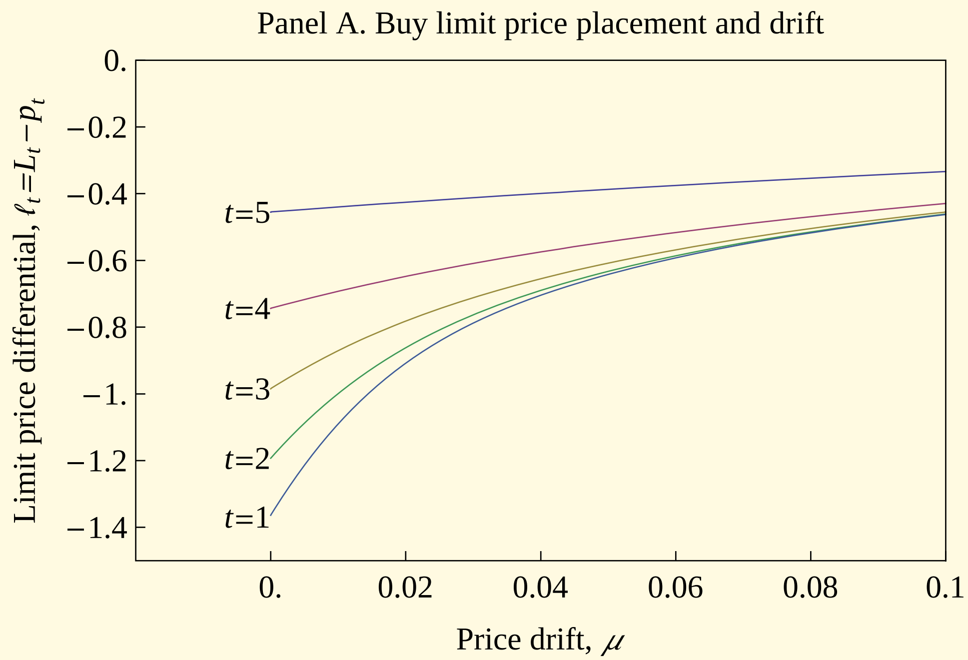
```
MatrixForm[r, TableDirections → {Column, Column, Row}];
```

```
f = Table[
   Interpolation[Table[{μVals[[i]], r[[i, j, 3]]}, {i, Length[μVals]}]], {j, 5}];
```

```
graphText = {
   Text["t=5", {0, r[[1, 1, 3]]}, {1, 0}],
   Text["t=4", {0, r[[1, 2, 3]]}, {1, 0}],
   Text["t=3", {0, r[[1, 3, 3]]}, {1, 0}],
   Text["t=2", {0, r[[1, 4, 3]]}, {1, 0}],
   Text["t=1", {0, r[[1, 5, 3]] - .005}, {1, 0}]};
```

```
μLabel = "Price drift, μ";
```

```
pA = Plot[Evaluate[Table[f[[i]][x], {i, 1, 5}]], {x, 0.`, 0.1`}, Frame → True,
  PlotRange → {{-0.02`, 0.1`}, {-1.5`, 0}}, Axes → {True, None}, FrameLabel →
   {μLabel, ℓLabel, "Panel A. Buy limit price placement and drift", None},
  FrameTicks → {Range[0, 0.1`, 0.02`], Range[0, -1.4`, -0.2`], None, None},
  Epilog → graphText, BaseStyle → baseStyle]
```



◻ *Plots by σ*

```
σVals = Range[.1, 1.2, .1];
```

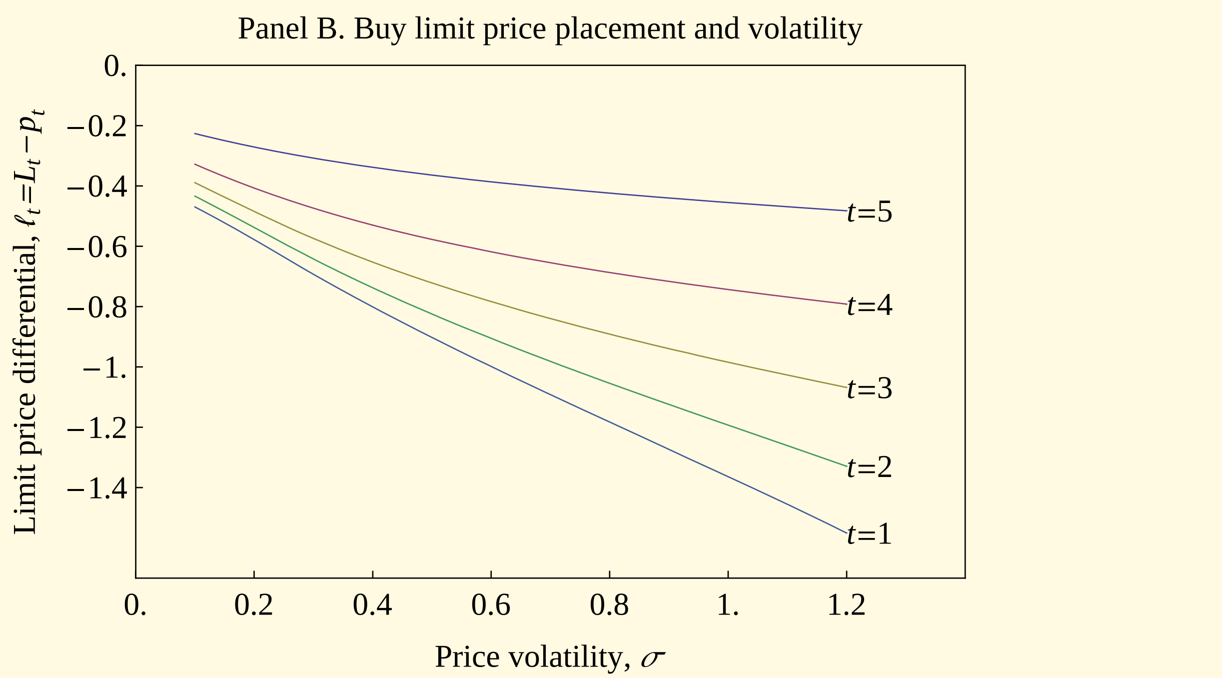
```
r = LimitSolve[5, 5, 0, #] & /@ σVals;
```

```
f = Table[
   Interpolation[Table[{σVals[[i]], r[[i, j, 3]]}, {i, Length[σVals]}]], {j, 5}];
```

```
graphText = {
   Text["t=5", {1.2, r[[Length[r], 1, 3]]}, {-1, 0}],
   Text["t=4", {1.2, r[[Length[r], 2, 3]]}, {-1, 0}],
   Text["t=3", {1.2, r[[Length[r], 3, 3]]}, {-1, 0}],
   Text["t=2", {1.2, r[[Length[r], 4, 3]]}, {-1, 0}],
   Text["t=1", {1.2, r[[Length[r], 5, 3]]}, {-1, 0}]};
```

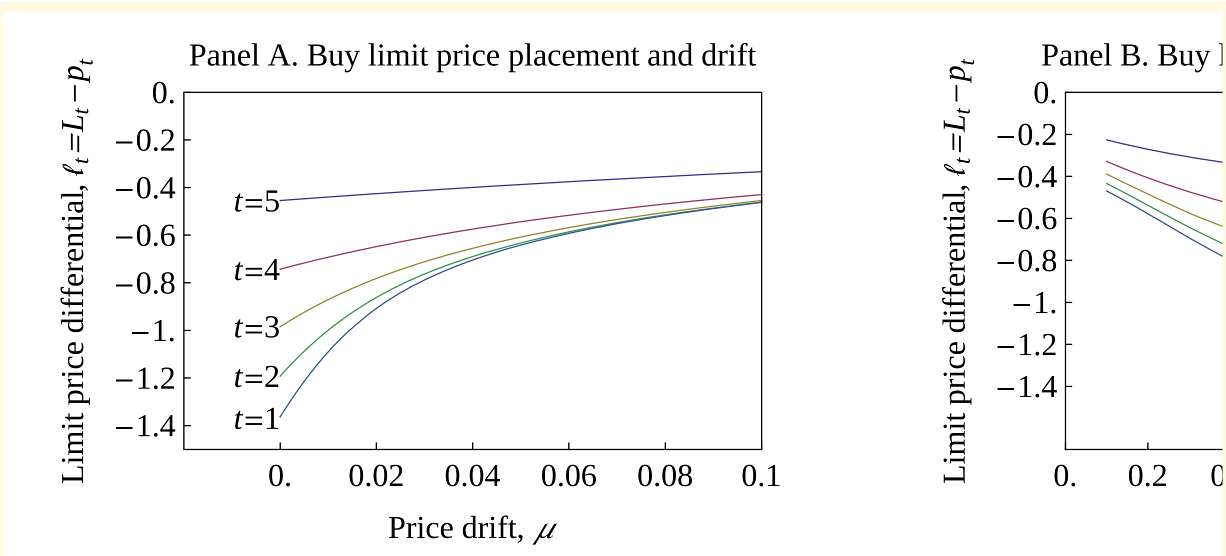
```
σLabel = "Price volatility, σ";
```

```
pB = Plot[Evaluate[Table[f[[i]][x], {i, 1, 5}]], {x, 0.1`, 1.2`},
  Frame → True, PlotRange → {{0, 1.4`}, {-1.7`, 0}}, FrameLabel →
   {σLabel, ℓLabel, "Panel B. Buy limit price placement and volatility", None},
  FrameTicks → {Range[0, 1.2`, 0.2`], Range[0, -1.4`, -0.2`], None, None},
  Epilog → graphText, BaseStyle → baseStyle]
```



◻ *Combine the graphs*

```
Show[GraphicsRow[{pA, pB}], Background → GrayLevel[1]]
```



◻ *Plots by λ (not used in book)*

```
λVals = Range[1, 6, .2];
```

```
r = LimitSolve[5, #] & /@ λVals;
```

```
f = Table[
   Interpolation[Table[{λVals[[i]], r[[i, j, 3]]}, {i, Length[λVals]}]], {j, 5}];
```

```
graphText = {
   Text["t=5", {6, r[[Length[r], 1, 3]]}, {-1, 0}],
   Text["t=4", {6, r[[Length[r], 2, 3]]}, {-1, 0}],
   Text["t=3", {6, r[[Length[r], 3, 3]]}, {-1, 0}],
   Text["t=2", {6, r[[Length[r], 4, 3]] - .05}, {-1, 0}],
   Text["t=1", {6, r[[Length[r], 5, 3]] - .1}, {-1, 0}]};
```

```
Plot[Evaluate[Table[f[[i]][x], {i, 1, 5}]],
 {x, 1, 6}, Frame → True, PlotRange → {{1, 7}, {0, -3.5`}},
 FrameLabel → {"λ, c density parameter", ℓLabel, None, None},
 FrameTicks → {Range[1, 6, 1], Range[0, -3, -1], None, None},
 Epilog → graphText, BaseStyle → baseStyle]
```