Beispiel zur Brownschen Brücke

AKVFM Numerische Methoden der Finanz− und Versicherungsmathematik

TU Wien, SS 2006, Reinhold Kainhofer

Definition der BB

```
In[1]:= << Statistics`NormalDistribution`
        << Graphics`

In[3]:= $TextStyle = {FontFamily -> "Helvetica"};

In[4]:= BBridgeInter[{{s_, xs_}, {u_, xu_}}] := BBridgeInter[{s, xs}, {u, xu}];
        BBridgeInter[{s_, xs_}, {u_, xu_}] := Module[{t = (u + s)/2},
          {t, xs + (t - s)/(u - s) (xu - xs) + Sqrt[(u - t) (t - s)/(u - s)] Random[NormalDistribution[0, 1]]}
         ]

In[6]:= ClearAll[vals];
```

Anfangs− und Endpunkt

```
In[62]:= vals[i_Integer] /; i == 0 :=
           vals[0] = {{0., 0.}, {10., Sqrt[10] Random[NormalDistribution[0, 1]]}};
```

Rekursive Definition durch Intervallhalbierung

```
In[8]:= vals[i_Integer] /; i > 0 := vals[i] = Module[{SSS, res, vvv},
            res = SSS[#[[1]], BBridgeInter[#]] & /@ Partition[vals[i - 1], 2, 1];
            AppendTo[res, Last@vals[i - 1]] /. SSS[vvv___] :> Sequence[vvv]
           ];
```

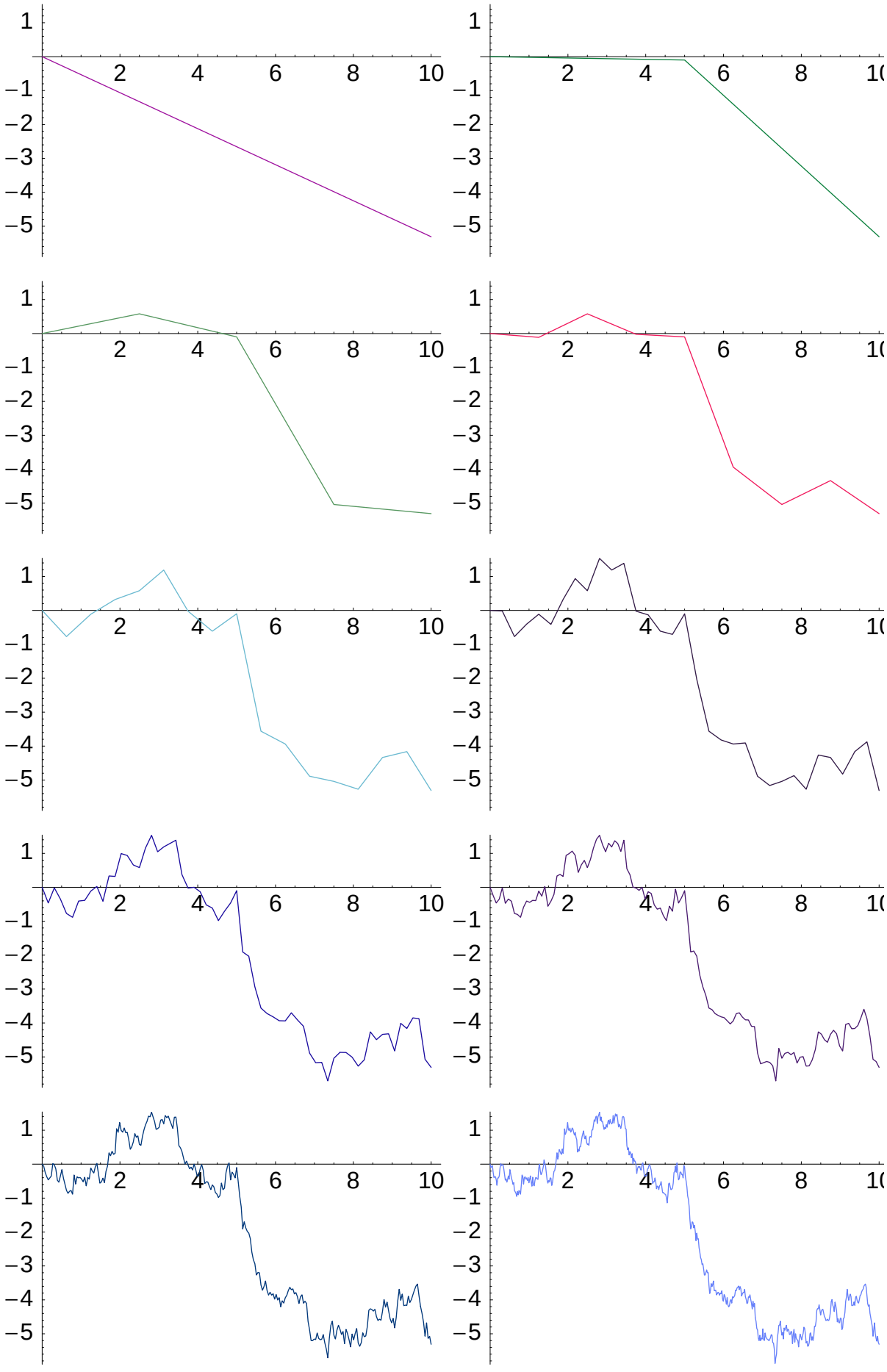
Werte zuruecksetzen, um neuen Pfad zu erzeugen (die Werte werden ja zwischengespeichert)

```
In[44]:= resetVals[i_Integer] := Module[{},
            (vals[#] =.) & /@ Range[0, i];
           ];

In[63]:= resetVals[10]

Unset::norep : Assignment on vals for vals[4] not found. Mehr...

Unset::norep : Assignment on vals for vals[5] not found. Mehr...

Unset::norep : Assignment on vals for vals[6] not found. Mehr...

General::stop : Further output of Unset::norep will be suppressed during this calculation. Mehr...
```

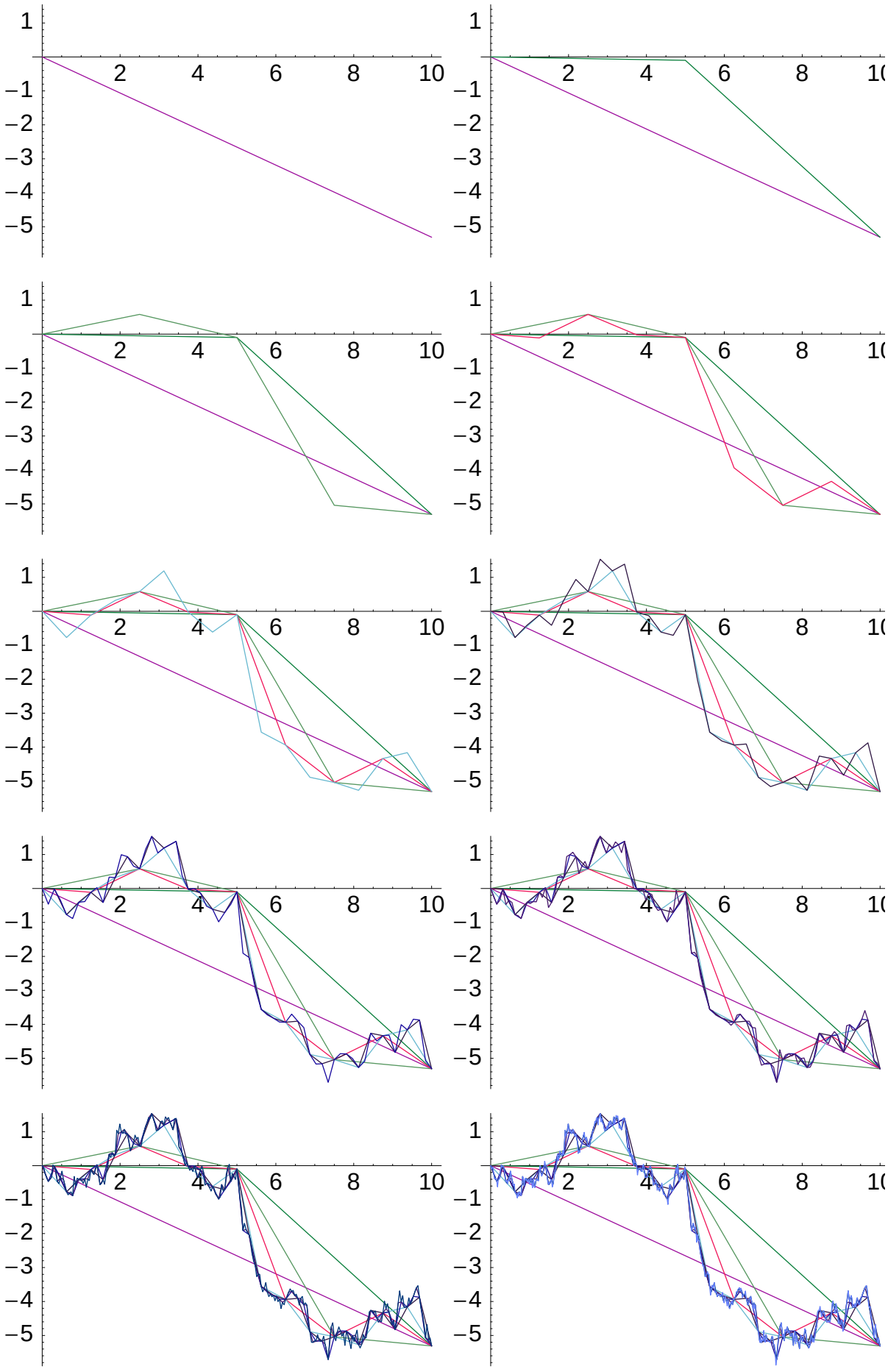
Die Werte, rekursiv

```
In[77]:= vals[1]
         vals[2]
         vals[3]

Out[77]= {{0., 0.}, {5., −0.0992343}, {10., −5.30934}}

Out[78]= {{0., 0.}, {2.5, 0.581085}, {5., −0.0992343}, {7.5, −5.04033}, {10., −5.30934}}

Out[79]= {{0., 0.}, {1.25, −0.112159}, {2.5, 0.581085}, {3.75, −0.0219451}, {5., −0.0992343},
          {6.25, −3.93832}, {7.5, −5.04033}, {8.75, −4.33567}, {10., −5.30934}}

In[80]:= recursionDepth = 10;
         range = {Min[#], Max[#]} &[Last /@ (Join @@ (vals /@ Range[0, recursionDepth]))];

In[82]:= plots = ListPlot[vals[#], PlotRange → range,
              PlotJoined → True, DisplayFunction :→ Identity, PlotStyle →
               {RGBColor[Random[], Random[], Random[]]}] & /@ Range[0, recursionDepth];
```

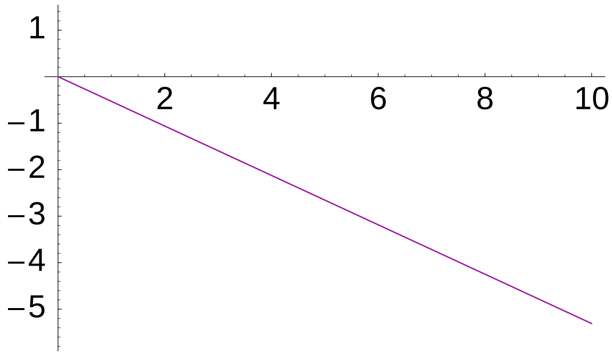
```
In[85]:= Show@GraphicsArray[Partition[plots, 2], GraphicsSpacing → 0, ImageSize → 600];
```

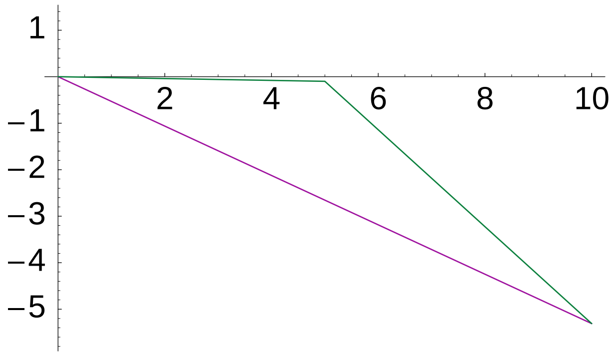


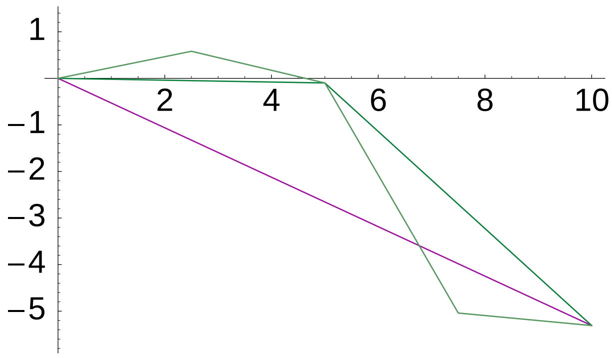
```
In[86]:= Show@GraphicsArray[Partition[Show[Take[plots, #]] & /@ Range[recursionDepth], 2],
           GraphicsSpacing → 0, ImageSize → 600];
```

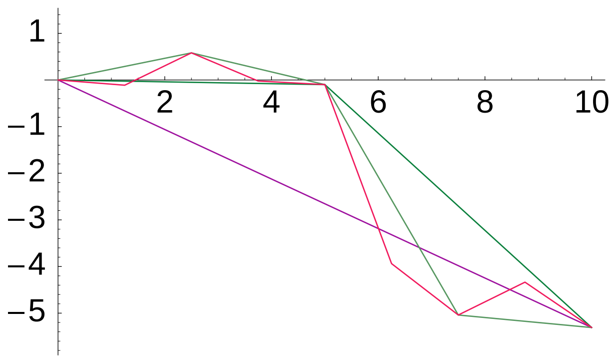


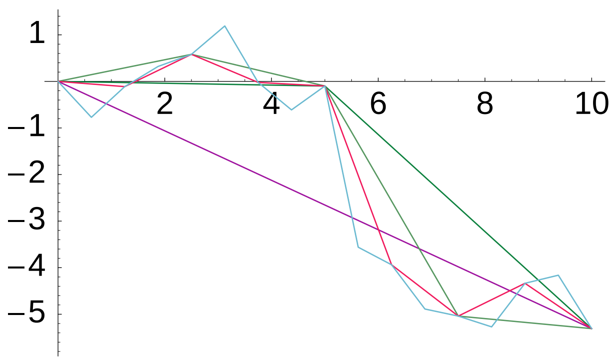
In[87]:=
```
Show[Take[plots, #], DisplayFunction :> $DisplayFunction] & /@ Range[recursionDepth];
```

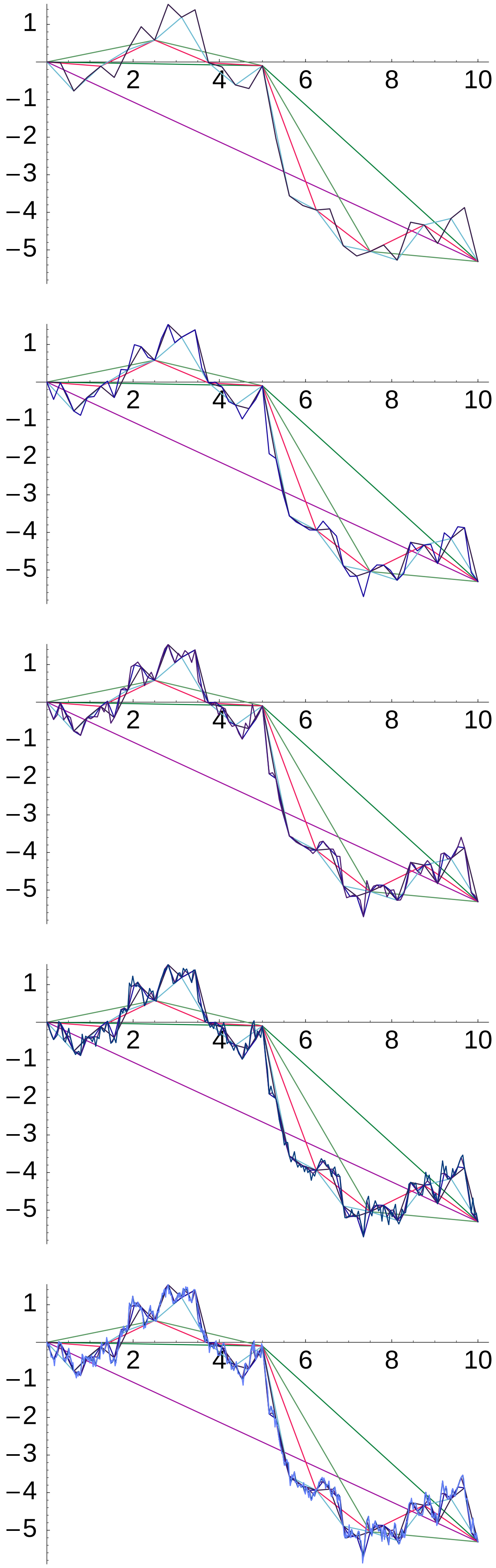

1
2
4
6
8
10
−1
−2
−3
−4
−5


1
2
4
6
8
10
−1
−2
−3
−4
−5

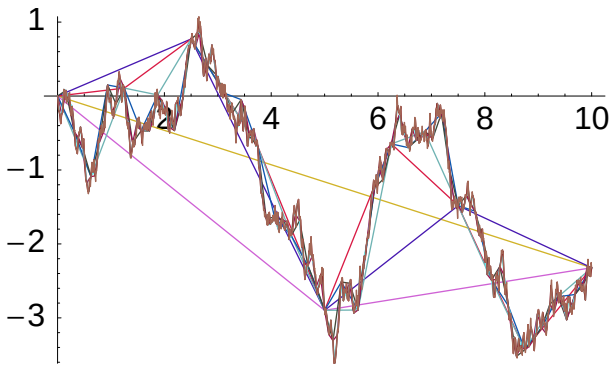

1
2
4
6
8
10
−1
−2
−3
−4
−5


1
2
4
6
8
10
−1
−2
−3
−4
−5

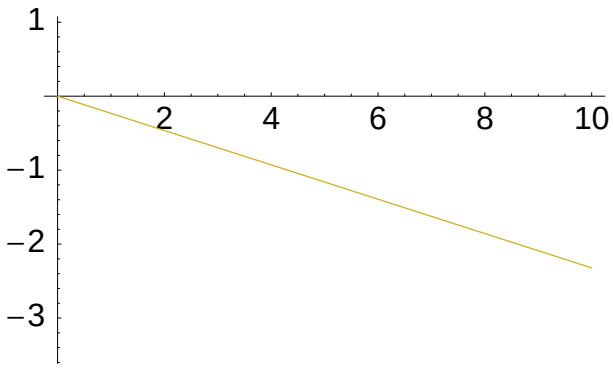

1
2
4
6
8
10
−1
−2
−3
−4
−5

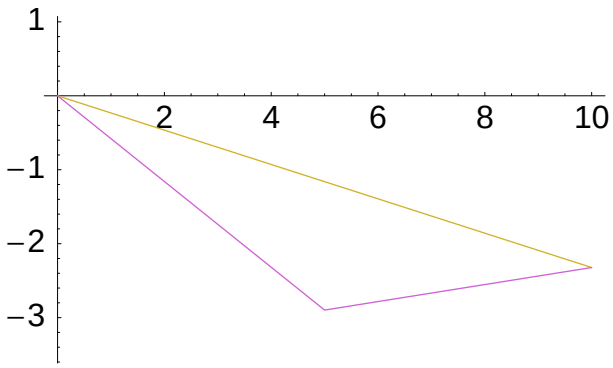

1
−1
−2
−3
−4
−5
2
4
6
8
10

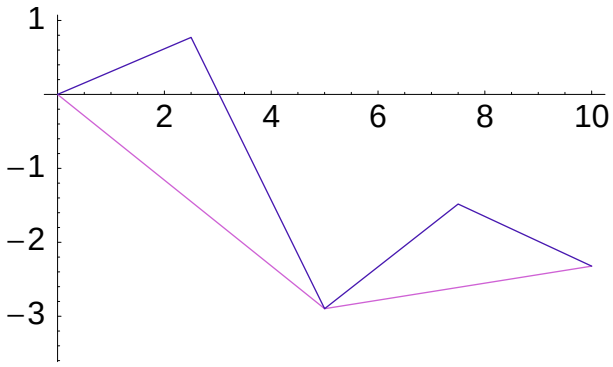
```
In[73]:= Show[plots, DisplayFunction :> $DisplayFunction];
```

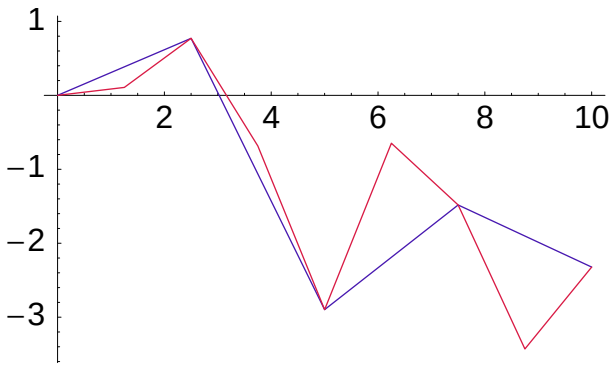


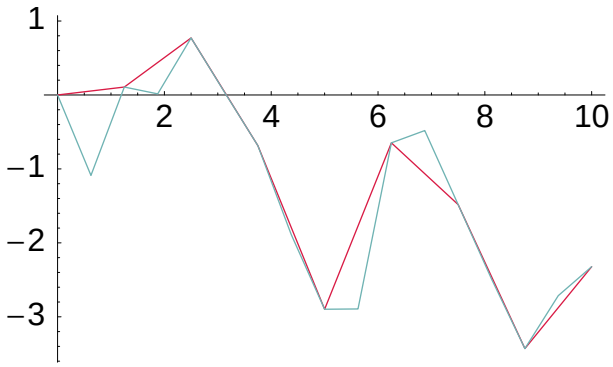

```
In[74]:= Show[#, DisplayFunction :> $DisplayFunction] & /@
           Prepend[Partition[plots, 2, 1], plots[[1]]];
```

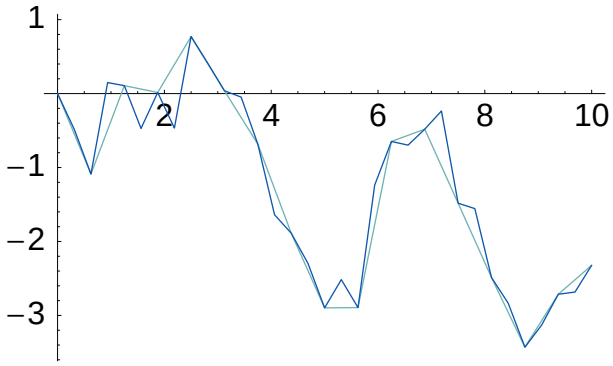


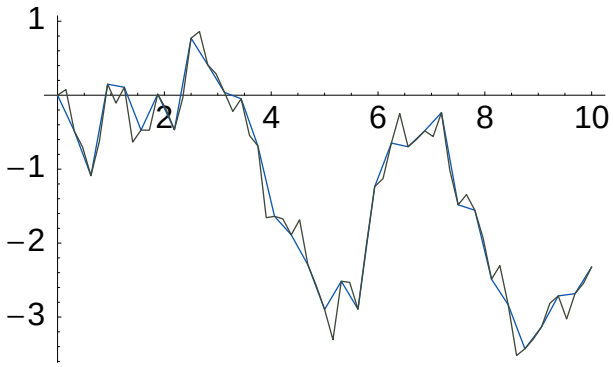


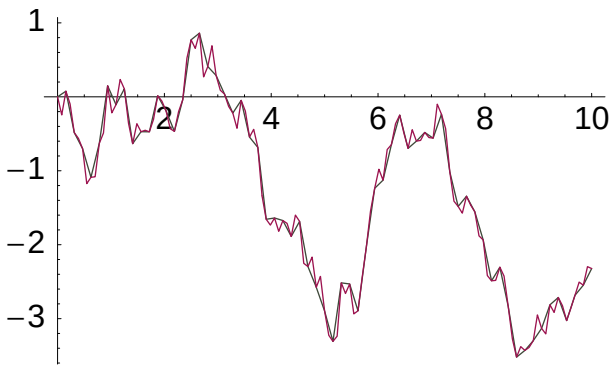


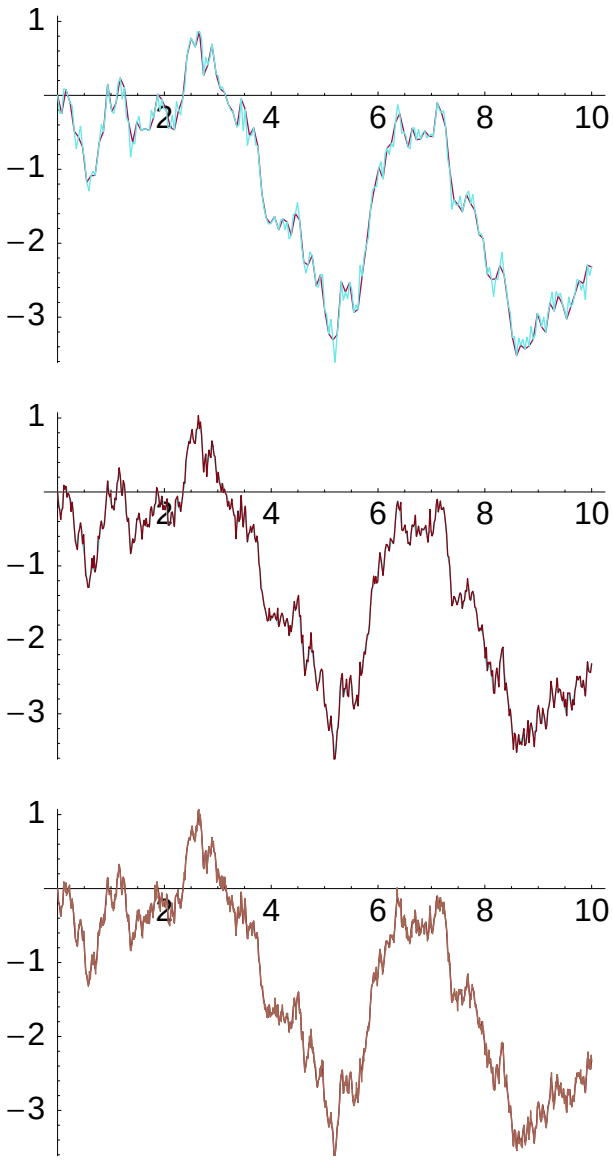

1
2
4
6
8
10
−1
−2
−3


1
2
4
6
8
10
−1
−2
−3

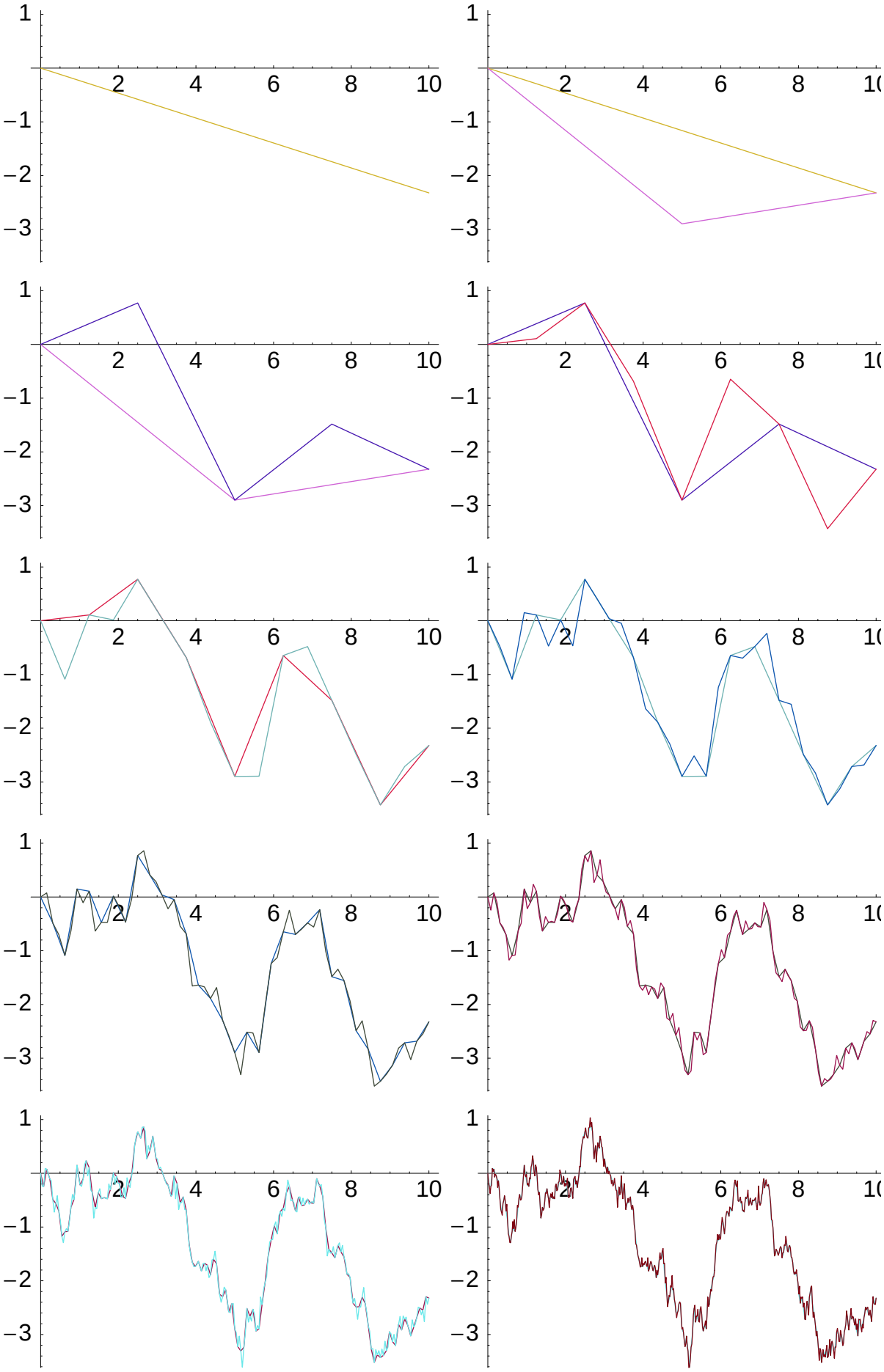

1
2
4
6
8
10
−1
−2
−3


1
2
4
6
8
10
−1
−2
−3


1
2
4
6
8
10
−1
−2
−3



In[75]:=
```
Show@
  GraphicsArray[Partition[Show /@ Prepend[Partition[plots, 2, 1], plots[[1]]], 2],
   GraphicsSpacing → 0, ImageSize → 600];
```


1
2
4
6
8
10
−1
−2
−3

Alle Werte löschen, damit neuer Pfad erstellt wird:

In[76]:=
```
resetVals[recursionDepth]
```