tested 190817 using *SpinDynamica* 3.0.1 under *Mathematica* 11.0

```
Needs["SpinDynamica`"]
```

```
SpinDynamica version 3.0.1 loaded
```

ModifyBuiltIn: The following built-in routines have been modified in SpinDynamica:

{Chop, Dot, Duration, Exp, Expand, ExpandAll, NumericQ, Plus, Power, Simplify, Times, WignerD}.

Evaluate ??*symbol* to generate the additional definitions for *symbol*.

```
SetSpinSystem[1]
```

SetSpinSystem: the spin system has been set to $\{\{1, \frac{1}{2}\}\}$

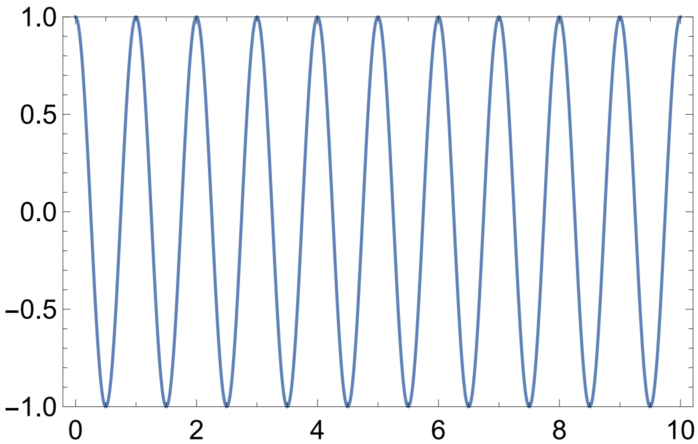
SetBasis: the state basis has been set to ZeemanBasis$[\{\{1, \frac{1}{2}\}\}$, BasisLabels → Automatic$]$.

```
SetOptions[Plot, PlotRange → {-1, 1}, Frame → True];
```

time-dependent pulse shapes

A simple trajectory of z-magnetization under a rf field along the (rotating-frame) x-axis:

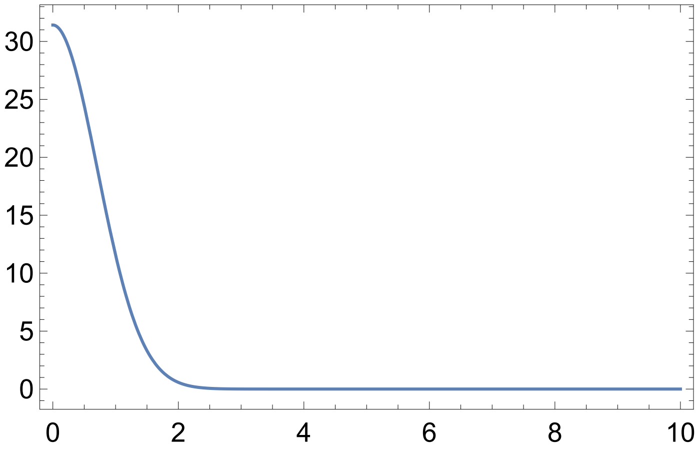
```
Plot[
 Evaluate[Trajectory[opI["z"] → opI["z"], {2 π opI["x"], 10}][t]],
 {t, 0, 10}
]
```



Modulate the amplitude of the rf field using a time-dependent function

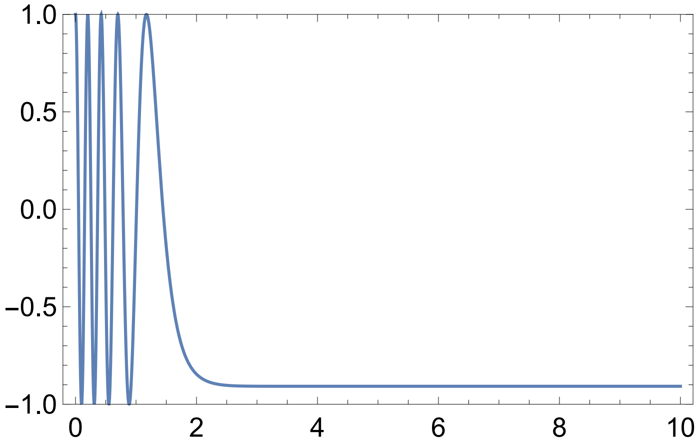
```
RfAmplitude[t_] := 2 π 5 Exp[-t^2];
```

```
Plot[RfAmplitude[t], {t, 0, 10}, PlotRange → All]
```



note that the amplitude is only at the beginning of the evolution. Now simulate:

```
Plot[
 Evaluate[
  Trajectory[opI["z"] → opI["z"],
    {Function[t, RfAmplitude[t] opI["x"]], 10}
   ][t]
 ],
 {t, 0, 10}
]
```

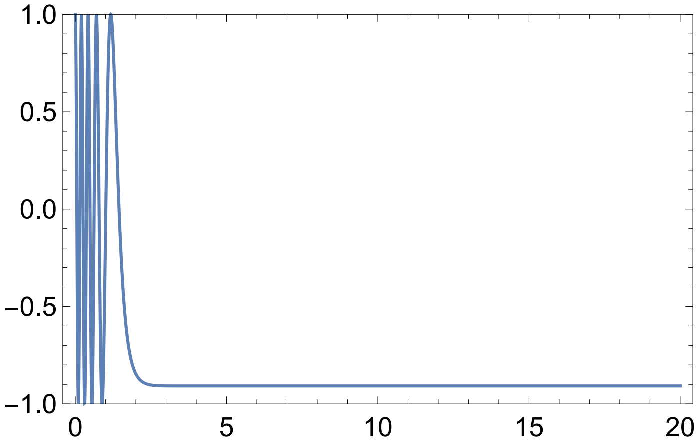


the pulse only acts at short times where its amplitude is large

Two consecutive pulses of the same form

```
RfAmplitude[t_] := 2 π 5 Exp[-t^2];
```

```
Plot[
 Evaluate[
  Trajectory[opI["z"] → opI["z"],
    {{Function[t, RfAmplitude[t] opI["x"]], 10},
     {Function[t, RfAmplitude[t] opI["x"]], 10}}
   ][t]
 ],
 {t, 0, 20}
]
```



note that the second event does not influence the spins since the pulse shape is referenced to the *global* time coordinate t, so the second pulse has negligible amplitude.

Two consecutive pulses of the same form, but using a local time coordinate τ

```
RfAmplitude[t_] := 2 π 5 Exp[-t^2];
```

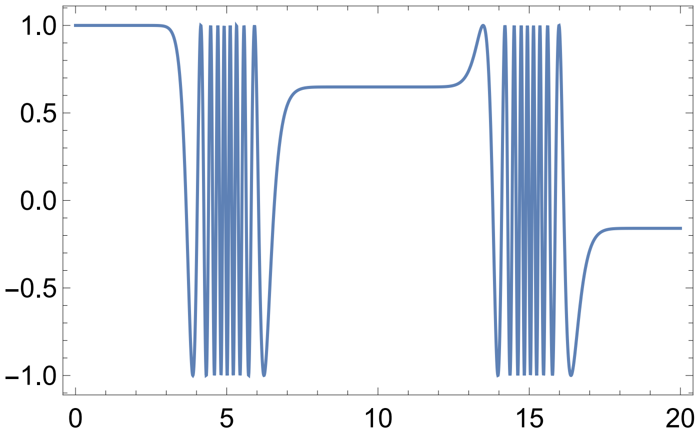
The ShapeFunction[{τ,1/2},function] syntax uses a local time variable τ which has a time origin at the *centre* of the event in which it occurs.

```
ShapedPulse[1, 10, {τ, RfAmplitude[τ]}]
```

ShapedPulse$\left[\{1\}, 10, \left\{\left\{\tau, \frac{1}{2}\right\}, \left\{10\, e^{-\tau^2}\pi, 0, 0\right\}\right\}\right]$

```
shape = {ShapeFunction[{τ, 1/2}, RfAmplitude[τ] opI["x"]], 10};
```

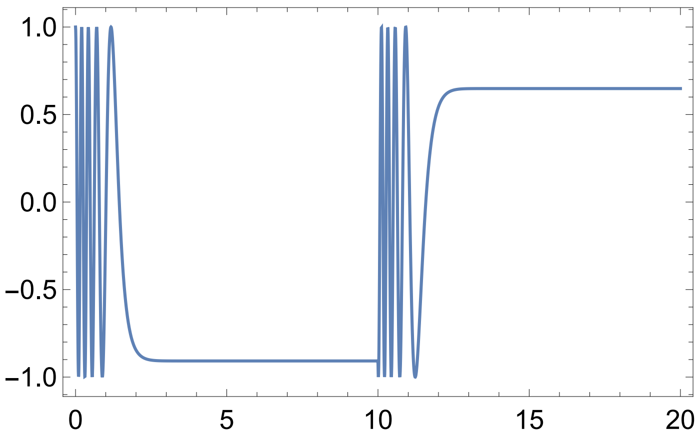
```
Plot[
 Evaluate[
  Trajectory[opI["z"] → opI["z"],
    {shape, shape}
   ][t]
 ],
 {t, 0, 20}, PlotRange → Automatic
]
```



The ShapeFunction[{τ,0},function] syntax uses a local time variable τ which has a time origin at the *centre* of the event in which it occurs.

```
shape = {ShapeFunction[{τ, 0}, RfAmplitude[τ] opI["x"]], 10};
```

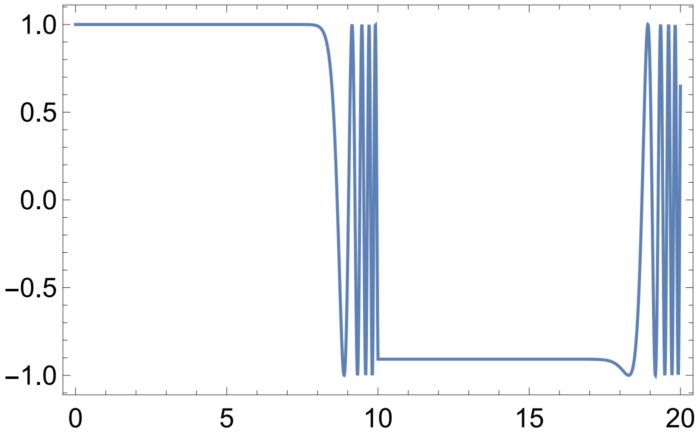
```
Plot[
 Evaluate[
  Trajectory[opI["z"] → opI["z"],
    {shape, shape}
   ][t]
 ],
 {t, 0, 20}, PlotRange → Automatic
]
```



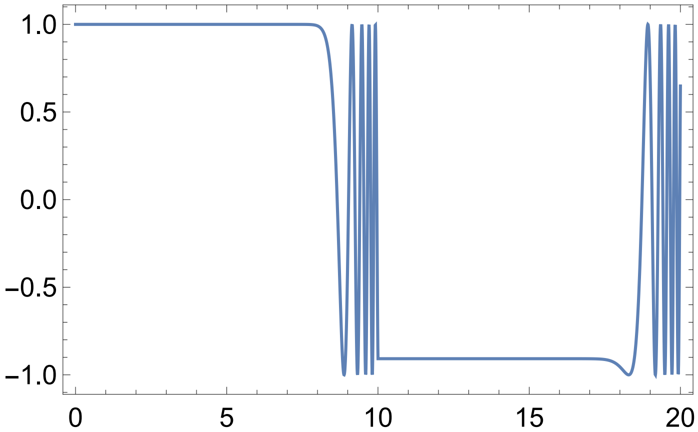
The ShapeFunction[{τ,1},function] syntax uses a local time variable τ which has a time origin at the *end* of the event in which it occurs.

```
shape = {ShapeFunction[{τ, 1}, RfAmplitude[τ] opI["x"]], 10};
```

```
Plot[
 Evaluate[
  Trajectory[opI["z"] → opI["z"],
    {shape, shape}
   ][t]
 ],
 {t, 0, 20}, PlotRange → Automatic
]
```



```
Plot[
 Evaluate[
  Trajectory[opI["z"] → opI["z"],
    {ShapedPulse[1, 10, {{τ, 1}, RfAmplitude[τ]}],
     ShapedPulse[1, 10, {{τ, 1}, RfAmplitude[τ]}]}
   ][t]
 ],
 {t, 0, 20}, PlotRange → Automatic
]
```



Two consecutive pulses using a local time variable τ as well as a global time variable t

This syntax is usually needed if two pulses must be phase coherent with each other but may also have local defined amplitude or phase shapes.

```
shape = {ShapeFunction[t, {τ, 1/2}, RfAmplitude[τ] opI["x"] Cos[2 π t]], 10};
```

```
Plot[
 Evaluate[
  Trajectory[opI["z"] → opI["z"],
    {shape, shape}
   ][t]
 ],
 {t, 0, 20}
]
```