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$]$.

documentation of ShapeFunction

```
? ShapeFunction
```

ShapeFunction[*t*,{*τ*,*x*},*T*,*expression*] defines a time-dependent Hamiltonian or Liouvillian which depends on an absolute time variable t and also a relative time variable τ which is taken relative to the event in which the ShapeFunction is used. The variable x is used to specify the time origin of the relative time variable τ. For example *x=0* specifies that the time origin is the start of the event; *x=1* indicates a time origin at the end of the event, and *x=1/2* indicates a time origin at the middle of the event. The period argument T determines when derived propagators may be reused. A *ShapeFunction* defined through *func=ShapeFunction*[*t*,{*τ*,*x*},*T*,*expression*] may be evaluated at any time during an event that starts at time point ta and ends at time point tb using the syntax *func*[*t*,*ta*,*tb*]. A ShapeFunction may employ multiple relative time variables by using the syntax *ShapeFunction*[*t*,{{*τ1*,*x1*},{*τ2*,*x2*}..},*T*,*expression*].

Gaussian pulse

this routine defines a Gaussian pulse with any desired flip angle and frequency bandwidth

define Gaussian pulse with defined flip angle and phase, and options for the frequency width and frequency offset

```
Options[GaussianPulse] = {TruncationLevel → 1/100,
   GaussianFrequencyWidth → 2 π 100, FlipAngle → π, FrequencyOffset → 0};
```

```
GaussianPulse[{βp_, ϕ_}, opts___Rule] :=
 Module[{ωmod, σ, ωnut0, αG, trunclevel, Thalf, ωC, Δω}, (
   σ = (GaussianFrequencyWidth /. {opts} /. Options[GaussianPulse]) / 2;
   ωC = FrequencyOffset /. {opts} /. Options[GaussianPulse];
   αG = (σ / 2) ^2;
   ωnut0 = βp × Sqrt[αG / π];
   trunclevel = TruncationLevel /. {opts} /. Options[GaussianPulse];
   Thalf = Sqrt[-Log[trunclevel] / αG];
   {ShapeFunction[t, {τ, 1 / 2}, All,
     Evaluate[
      ωnut0 (opI["x"] Cos[ϕ] + opI["y"] Sin[ϕ]) Exp[-αG τ^2] - ωC opI["z"]
     ]
    ],
    2 Thalf
   }
  )]
```

comments

```
pulse = GaussianPulse[{π, 0}]
```

$\left\{\text{ShapeFunction}[<< .. >>],\ \frac{\sqrt{\text{Log}[100]}}{25\,\pi}\right\}$

The first argument of ShapeFunction (t) is the absolute time coordinate during the pulse sequence. It is not used in this case since the shape of the pulse only depends on the time taken relative to the pulse sequence element.

The second argument of ShapeFunction $\left\{\left\{\tau,\ \frac{1}{2}\right\}\right\}$ indicates that the time variable τ is taken relative to the centre of the pulse.

The third argument All indicates that the pulse propagator may be reused with any time shift. It is not relevant unless PrecalculateEvent is used.

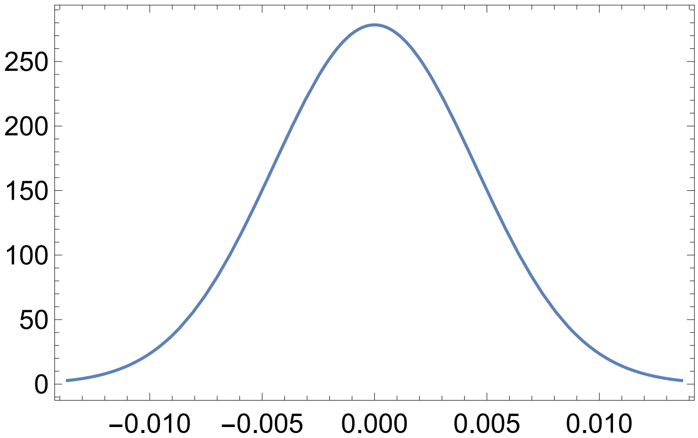
The last argument gives the form of the spin Hamiltonian during the pulse. In this case it only depends on the relative time τ, not the absolute time.

```
T = EventDuration[pulse]
```

$\frac{\sqrt{\text{Log}[100]}}{25\,\pi}$

plot Gaussian pulse

```
Plot[
 Evaluate[OperatorAmplitude[opI["x"], First[pulse][t, -T/2, T/2]]],
 {t, -T/2, T/2}, PlotRange → All, Frame → True
]
```



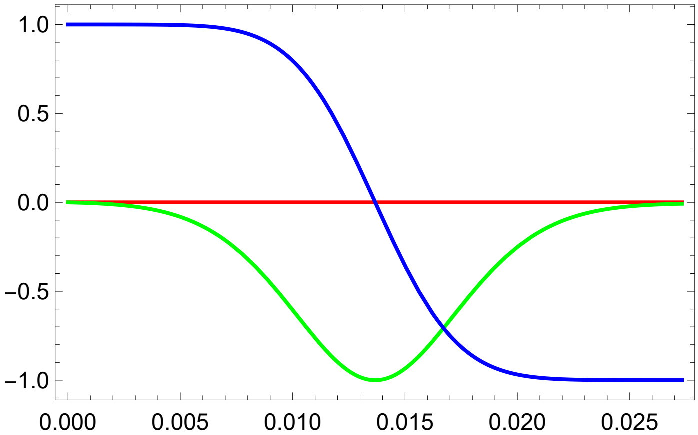
simulate magnetization trajectories

on-resonance

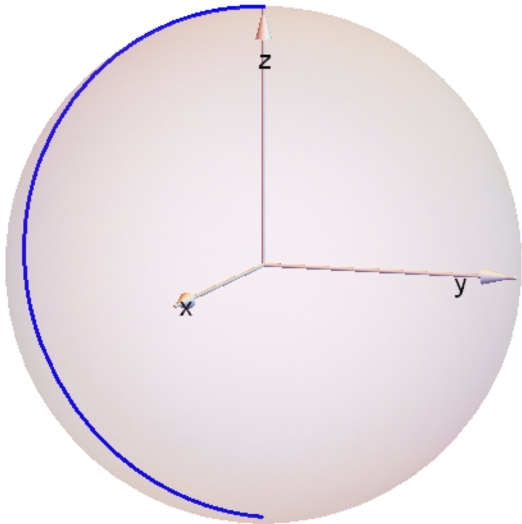
```
{xtraj, ytraj, ztraj} =
 Trajectory[opI["z"] → {opI["x"], opI["y"], opI["z"]}, GaussianPulse[{π, 0}]]
```

```
{TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>]}
```

```
Plot[{xtraj[t], ytraj[t], ztraj[t]}, {t, 0, T}, Frame → True,
 PlotRange → All, PlotStyle → {{Thick, Red}, {Thick, Green}, {Thick, Blue}},
 LabelStyle → Directive[Medium, FontFamily → "Helvetica"]]
```



```
Show[
 Graphics3D@{Opacity[0.2], Sphere[{0, 0, 0}, 1]},
 ParametricPlot3D[
  Re@{xtraj[t], ytraj[t], ztraj[t]}, {t, 0, T},
  Boxed → True, Axes → None, PlotStyle → {{Thick, Blue}}
 ],
 Axes3D[], Boxed → False, ViewPoint → {6, 2, 1}, ViewVertical → ez
]
```

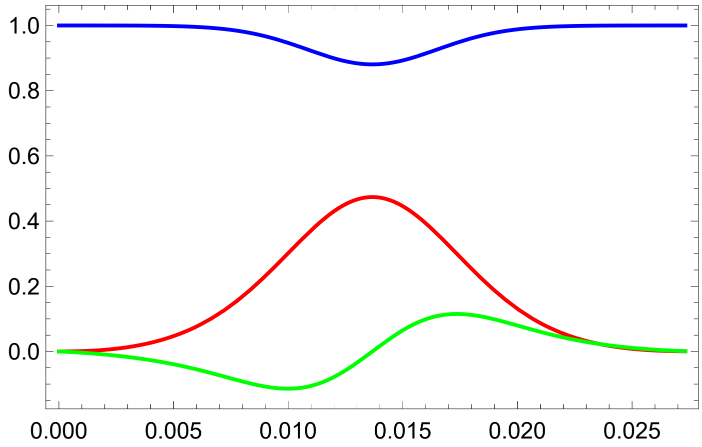


off-resonance

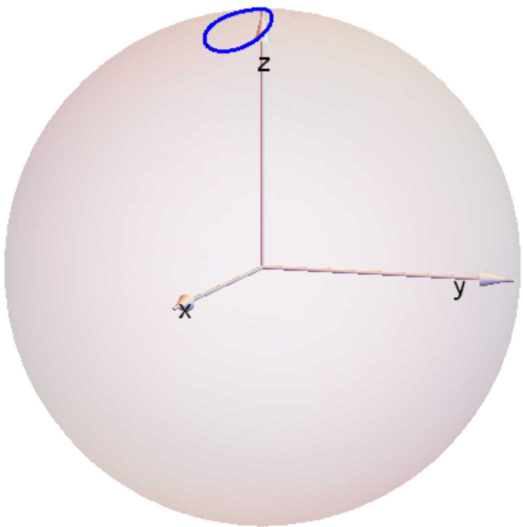
```
{xtraj, ytraj, ztraj} =
 Trajectory[
  opI["z"] → {opI["x"], opI["y"], opI["z"]}, GaussianPulse[{π, 0}],
  BackgroundGenerator → 2 π 100 opI["z"]
 ]
```

```
{TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>]}
```

```
Plot[{xtraj[t], ytraj[t], ztraj[t]}, {t, 0, T}, Frame → True,
 PlotRange → All, PlotStyle → {{Thick, Red}, {Thick, Green}, {Thick, Blue}},
 LabelStyle → Directive[Medium, FontFamily → "Helvetica"]]
```




```
Show[
 Graphics3D@{Opacity[0.2], Sphere[{0, 0, 0}, 1]},
 ParametricPlot3D[
  Re@{xtraj[t], ytraj[t], ztraj[t]}, {t, 0, T},
  Boxed → True, Axes → None, PlotStyle → {{Thick, Blue}}
 ],
 Axes3D[], Boxed → False, ViewPoint → {6, 2, 1}, ViewVertical → ez
]
```

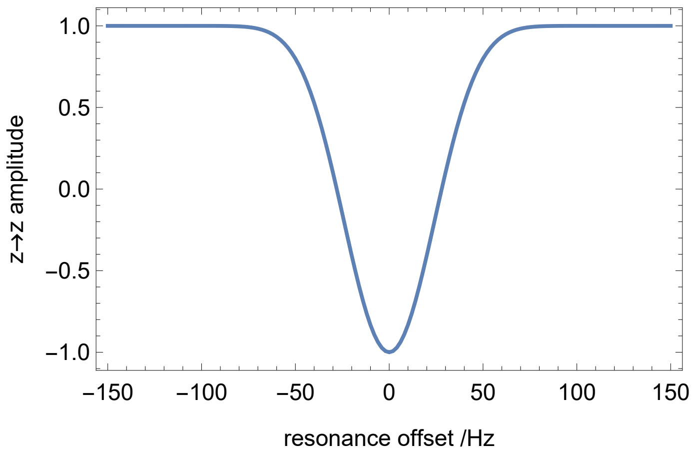


simulate magnetization inversion as a function of offset

```
TransformationAmplitude[
 opI["z"] → opI["z"],
 GaussianPulse[{π, 0}]
]
```

```
-0.999971
```

```
ListPlot[
 Table[
  {νoff,
   TransformationAmplitude[
    opI["z"] → opI["z"],
    GaussianPulse[{π, 0}],
    BackgroundGenerator → 2 π νoff opI["z"]
   ]
  },
  {νoff, -150, 150, 2}
 ],
 Frame → True, PlotRange → All, PlotStyle → Thick, Joined → True,
 LabelStyle → Directive[Medium, FontFamily → "Helvetica"],
 FrameLabel → {"resonance offset /Hz", "z→z amplitude"}
]
```



Gaussian pulse with additional cosine modulation (version 1)

in this case the Gaussian pulse is given an additional cosine modulation, which has phase 0 at the centre of the pulse.

```
Options[CosineModulatedGaussianPulse1] =
  {TruncationLevel → 1 / 100, GaussianFrequencyWidth → 2 π 100,
   FlipAngle → π, FrequencyOffset → 0, ModulationFrequency → 2 π 100};
```

Note the factor of 2 in the expression for the central nutation frequency. The effective field at the sideband positions is one-half that for an unmodulated pulse.

```
CosineModulatedGaussianPulse1[{βp_, ϕ_}, opts___Rule] :=
 Module[{ωmod, σ, ωnut0, αG, trunclevel, Thalf, ωC, ωm}, (
   σ = (GaussianFrequencyWidth /. {opts} /. Options[CosineModulatedGaussianPulse1]) / 2;
   ωC = FrequencyOffset /. {opts} /. Options[CosineModulatedGaussianPulse1];
   ωm = ModulationFrequency /. {opts} /. Options[CosineModulatedGaussianPulse1];
   αG = (σ / 2) ^2;
   ωnut0 = 2 βp × Sqrt[αG / π];
   trunclevel = TruncationLevel /. {opts} /. Options[CosineModulatedGaussianPulse1];
   Thalf = Sqrt[-Log[trunclevel] / αG];
   {ShapeFunction[t,
     {τ, 1 / 2},
     All,
     Evaluate[
      ωnut0 (opI["x"] Cos[ϕ] + opI["y"] Sin[ϕ]) Exp[-αG τ^2] Cos[ωm τ] - ωC opI["z"]
     ]
    ],
    2 Thalf
   }
  )]
```

comments

```
pulse = CosineModulatedGaussianPulse1[{π, 0}]
```

$\left\{\text{ShapeFunction}[\text{<< .. >>}],\ \dfrac{\sqrt{\text{Log}[100]}}{25\,\pi}\right\}$

The first argument of ShapeFunction (t) is the absolute time coordinate during the pulse sequence. This coordinate is not used in this case.

The second argument of ShapeFunction $\left\{\left\{\tau,\ \frac{1}{2}\right\}\right\}$ indicates that the relative time τ is taken relative to the centre of the pulse.

The third argument All indicates that the pulse propagator is invariant with respect to time shifts.

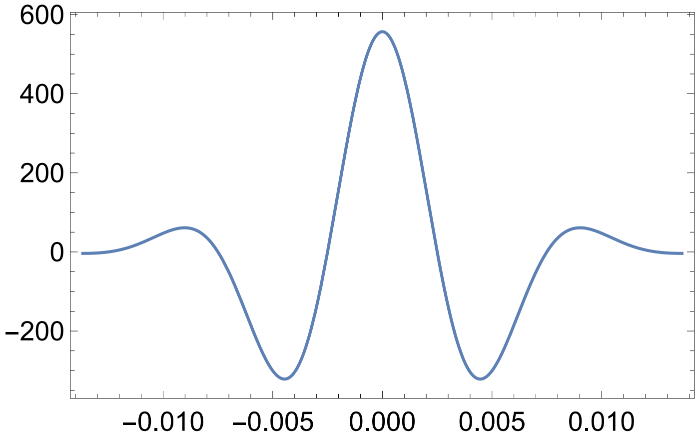
The last argument gives the form of the spin Hamiltonian during the pulse. In this case the modulation depends on the absolute time coordinate in the pulse sequence, not only the relative time

```
T = EventDuration[CosineModulatedGaussianPulse1[{π, 0}]]
```

$\dfrac{\sqrt{\text{Log}[100]}}{25\,\pi}$

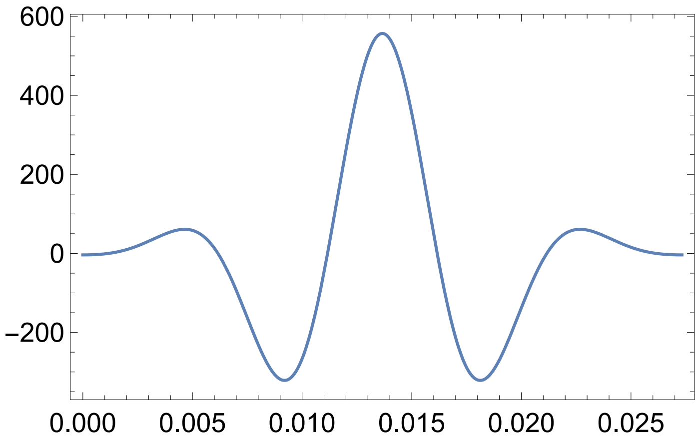
plot CosineModulatedGaussianPulse1

```
Plot[
 Evaluate[
  OperatorAmplitude[opI["x"],
   First[CosineModulatedGaussianPulse1[{π, 0}]][t, -T/2, T/2]
  ]
 ],
 {t, -T/2, T/2}, PlotRange → All, Frame → True
]
```



If the pulse is shifted, the shape is the same.

```
Plot[
 Evaluate[
  OperatorAmplitude[opI["x"],
   First[CosineModulatedGaussianPulse1[{π, 0}]][t, 0, T]
  ]
 ],
 {t, 0, T}, PlotRange → All, Frame → True
]
```



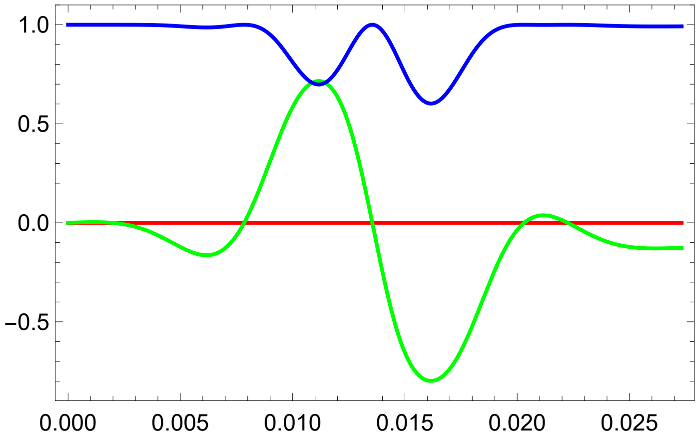
simulate magnetization trajectories

on-resonance

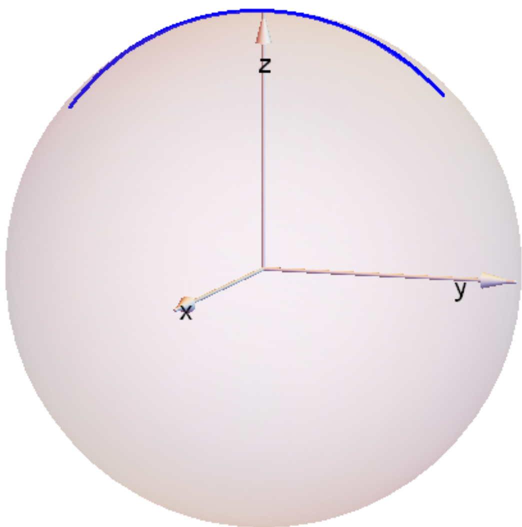
```
{xtraj, ytraj, ztraj} = Trajectory[
  opI["z"] → {opI["x"], opI["y"], opI["z"]}, CosineModulatedGaussianPulse1[{π, 0}]]
```

```
{TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>]}
```

```
Plot[{xtraj[t], ytraj[t], ztraj[t]}, {t, 0, T}, Frame → True,
 PlotRange → All, PlotStyle → {{Thick, Red}, {Thick, Green}, {Thick, Blue}},
 LabelStyle → Directive[Medium, FontFamily → "Helvetica"]]
```



```
Show[
 Graphics3D@{Opacity[0.2], Sphere[{0, 0, 0}, 1]},
 ParametricPlot3D[
  Re@{xtraj[t], ytraj[t], ztraj[t]}, {t, 0, T},
  Boxed → True, Axes → None, PlotStyle → {{Thick, Blue}}
 ],
 Axes3D[], Boxed → False, ViewPoint → {6, 2, 1}, ViewVertical → ez
]
```

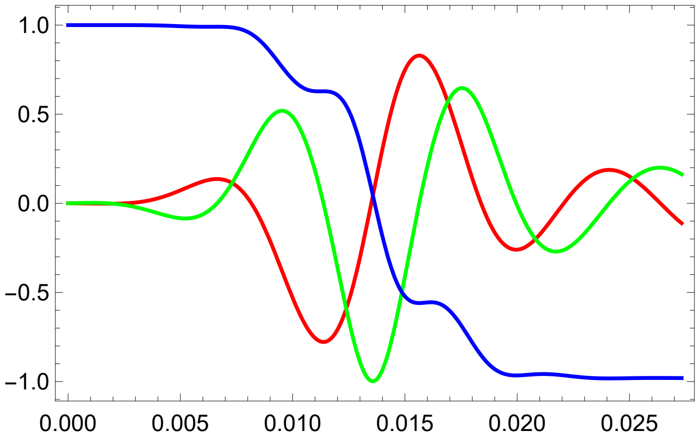


The modulated pulse is ineffective, on-resonance

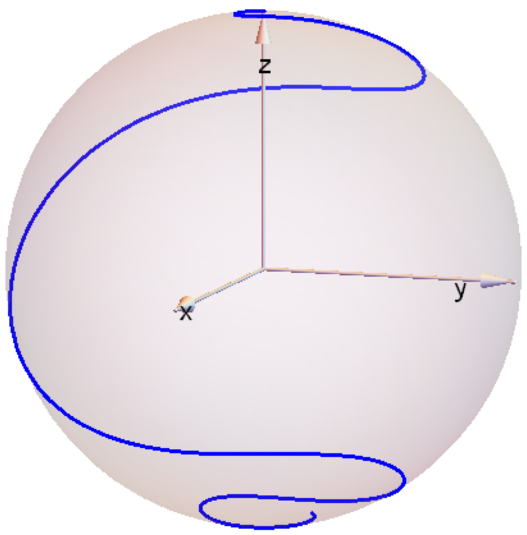
off-resonance

```
{xtraj, ytraj, ztraj} =
 Trajectory[
  opI["z"] → {opI["x"], opI["y"], opI["z"]}, CosineModulatedGaussianPulse1[{π, 0}],
  BackgroundGenerator → 2 π 100 opI["z"]
 ]
{TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>]}
```

```
Plot[{xtraj[t], ytraj[t], ztraj[t]}, {t, 0, T}, Frame → True,
 PlotRange → All, PlotStyle → {{Thick, Red}, {Thick, Green}, {Thick, Blue}},
 LabelStyle → Directive[Medium, FontFamily → "Helvetica"]]
```

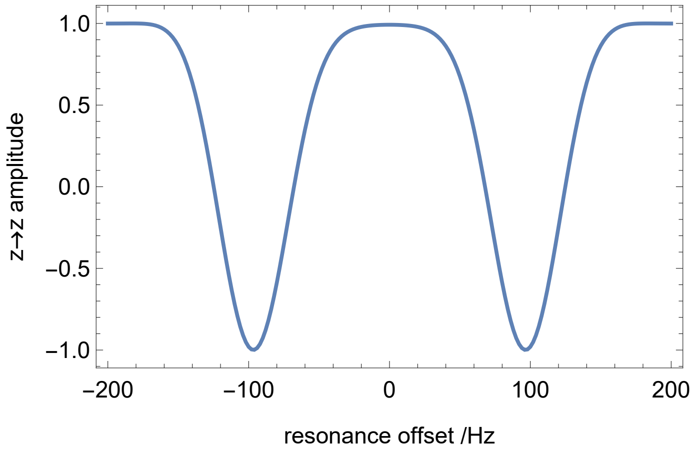



```
Show[
 Graphics3D@{Opacity[0.2], Sphere[{0, 0, 0}, 1]},
 ParametricPlot3D[
  Re@{xtraj[t], ytraj[t], ztraj[t]}, {t, 0, T},
  Boxed → True, Axes → None, PlotStyle → {{Thick, Blue}}
 ],
 Axes3D[], Boxed → False, ViewPoint → {6, 2, 1}, ViewVertical → ez
]
```



simulate magnetization inversion as a function of offset

```
ListPlot[
 Table[
  {νoff,
   TransformationAmplitude[
    opI["z"] → opI["z"],
    CosineModulatedGaussianPulse1[{π, 0}],
    BackgroundGenerator → 2 π νoff opI["z"]
   ]
  },
  {νoff, -200, 200, 2}
 ],
 Frame → True, PlotRange → All, PlotStyle → Thick, Joined → True,
 LabelStyle → Directive[Medium, FontFamily → "Helvetica"],
 FrameLabel → {"resonance offset /Hz", "z→z amplitude"}
]
```



This shows how the modulation generates sidebands in which there is good magnetization inversion

Gaussian pulse with additional cosine modulation (version 2)

in this case the Gaussian pulse is given an additional cosine modulation, which has a phase at the origin of the absolute time-coordinate

```
Options[CosineModulatedGaussianPulse2] =
  {TruncationLevel → 1 / 100, GaussianFrequencyWidth → 2 π 100,
   FlipAngle → π, FrequencyOffset → 0, ModulationFrequency → 2 π 100};
```

Note the factor of 2 in the expression for the central nutation frequency. The effective field at the sideband positions is one-half that for an unmodulated pulse.

```
CosineModulatedGaussianPulse2[{βp_, ϕ_}, opts___Rule] :=
 Module[{ωmod, σ, ωnut0, αG, trunclevel, Thalf, ωC, ωm}, (
   σ = (GaussianFrequencyWidth /. {opts} /. Options[CosineModulatedGaussianPulse2]) / 2;
   ωC = FrequencyOffset /. {opts} /. Options[CosineModulatedGaussianPulse2];
   ωm = ModulationFrequency /. {opts} /. Options[CosineModulatedGaussianPulse2];
   αG = (σ / 2) ^2;
   ωnut0 = 2 βp × Sqrt[αG / π];
   trunclevel = TruncationLevel /. {opts} /. Options[CosineModulatedGaussianPulse2];
   Thalf = Sqrt[-Log[trunclevel] / αG];
   {ShapeFunction[t, {τ, 1 / 2},
     Evaluate[2 π / ωm],
     Evaluate[
      ωnut0 (opI["x"] Cos[ϕ] + opI["y"] Sin[ϕ]) Exp[-αG τ^2] Cos[ωm t] - ωC opI["z"]
     ]
    ],
    2 Thalf
   }
  )]
```

comments

```
pulse = CosineModulatedGaussianPulse2[{π, 0}]
```

$$\left\{\text{ShapeFunction}[<< .. >>],\ \frac{\sqrt{\text{Log}[100]}}{25\,\pi}\right\}$$

The first argument of ShapeFunction (t) is the absolute time coordinate during the pulse sequence. This coordinate is now used for the cosine modulation. This ensures that the cosine modulation is phase coherent between shaped pulses applied at different times.

The second argument of ShapeFunction $\{\{\tau, \frac{1}{2}\}\}$ indicates that the relative time τ is taken relative to the centre of the pulse.

The third argument 1/100 indicates that the pulse propagator may only be reused at multiples of the modulation period.

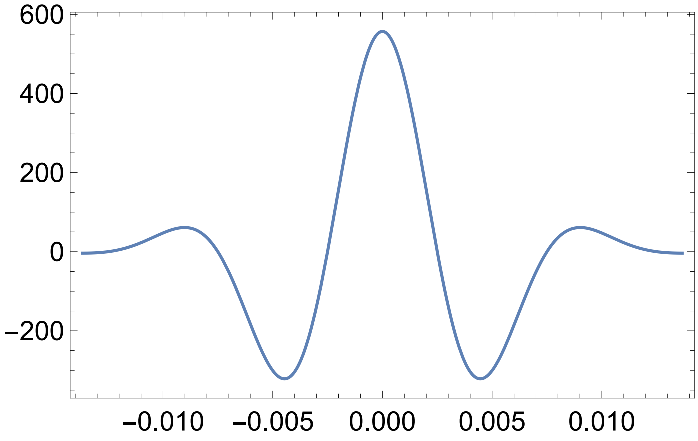
The last argument gives the form of the spin Hamiltonian during the pulse. In this case the modulation depends on the absolute time coordinate as well as the relative time.

```
T = EventDuration[CosineModulatedGaussianPulse1[{π, 0}]]
```

$$\frac{\sqrt{\text{Log}[100]}}{25\,\pi}$$

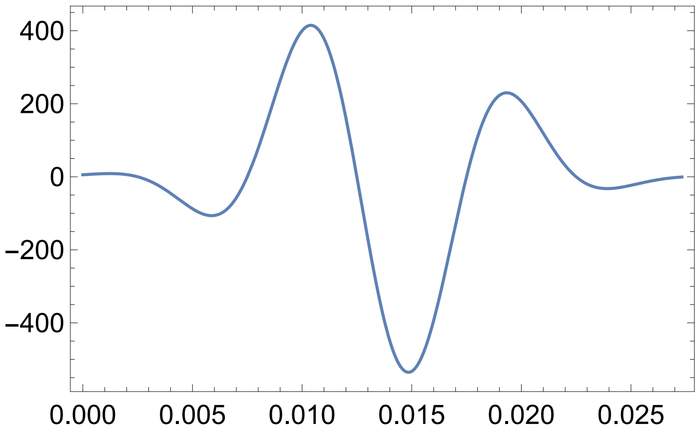
plot CosineModulatedGaussianPulse2

```
Plot[
 Evaluate[
  OperatorAmplitude[opI["x"],
   First[CosineModulatedGaussianPulse2[{π, 0}]][t, -T/2, T/2]
  ]
 ],
 {t, -T/2, T/2}, PlotRange → All, Frame → True
]
```



If the pulse is shifted, the shape is the now different, since the modulation function changes phase.

```
Plot[
 Evaluate[
  OperatorAmplitude[opI["x"],
   First[CosineModulatedGaussianPulse2[{π, 0}]][t, 0, T]
  ]
 ],
 {t, 0, T}, PlotRange → All, Frame → True
]
```



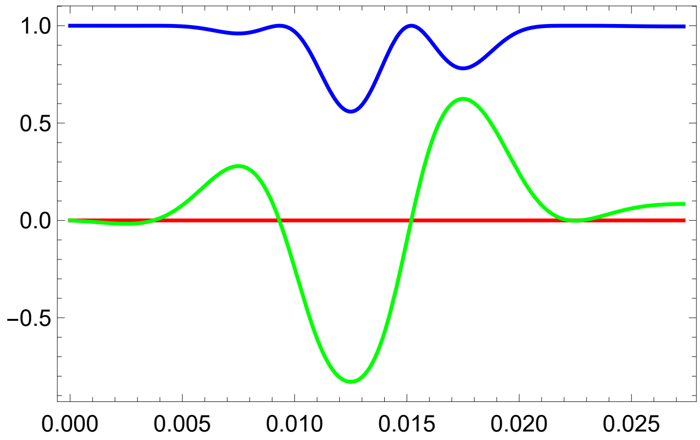
simulate magnetization trajectories

on-resonance

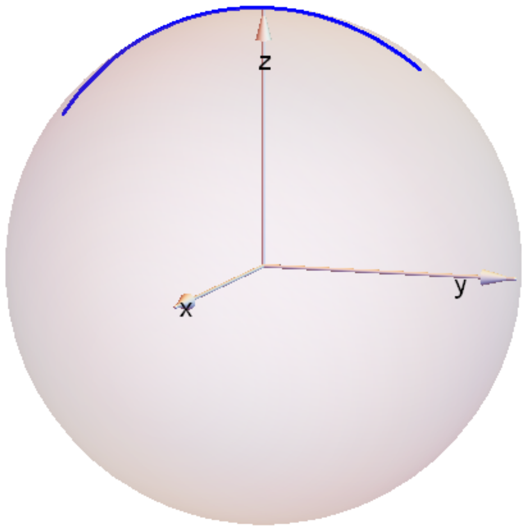
```
{xtraj, ytraj, ztraj} = Trajectory[
  opI["z"] → {opI["x"], opI["y"], opI["z"]}, CosineModulatedGaussianPulse2[{π, 0}]]
```

```
{TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>]}
```

```
Plot[{xtraj[t], ytraj[t], ztraj[t]}, {t, 0, T}, Frame → True,
 PlotRange → All, PlotStyle → {{Thick, Red}, {Thick, Green}, {Thick, Blue}},
 LabelStyle → Directive[Medium, FontFamily → "Helvetica"]]
```



```
Show[
 Graphics3D@{Opacity[0.2], Sphere[{0, 0, 0}, 1]},
 ParametricPlot3D[
  Re@{xtraj[t], ytraj[t], ztraj[t]}, {t, 0, T},
  Boxed → True, Axes → None, PlotStyle → {{Thick, Blue}}
 ],
 Axes3D[], Boxed → False, ViewPoint → {6, 2, 1}, ViewVertical → ez
]
```

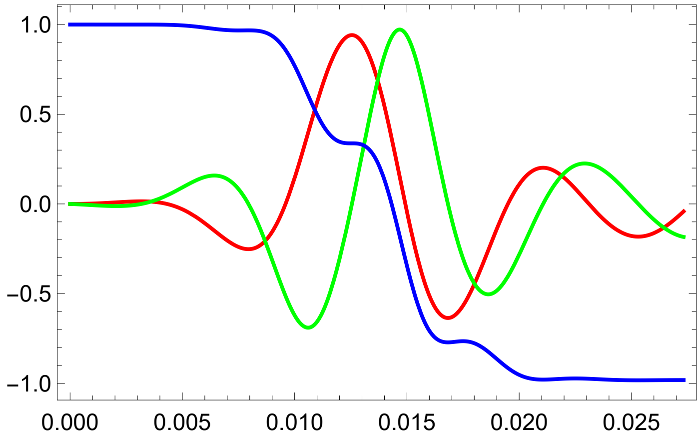


The modulated pulse is ineffective, on-resonance

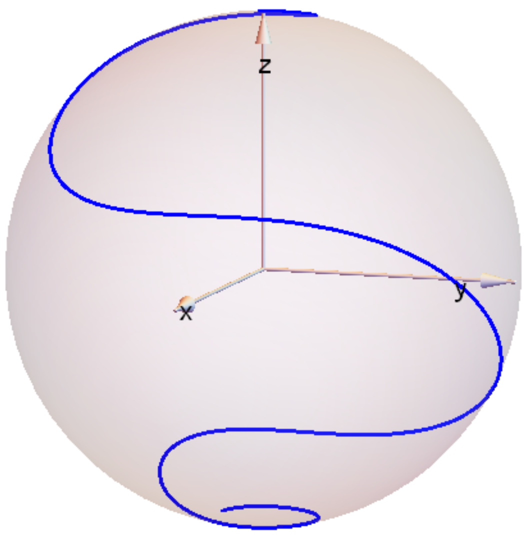
off-resonance

```
{xtraj, ytraj, ztraj} =
 Trajectory[
  opI["z"] → {opI["x"], opI["y"], opI["z"]}, CosineModulatedGaussianPulse2[{π, 0}],
  BackgroundGenerator → 2 π 100 opI["z"]
 ]
{TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 27.3233 × 10^-3}} , <>]}
```

```
Plot[{xtraj[t], ytraj[t], ztraj[t]}, {t, 0, T}, Frame → True,
 PlotRange → All, PlotStyle → {{Thick, Red}, {Thick, Green}, {Thick, Blue}},
 LabelStyle → Directive[Medium, FontFamily → "Helvetica"]]
```

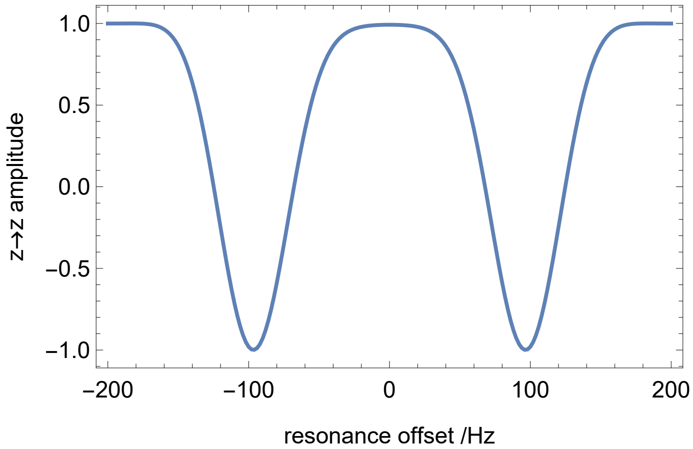



```
Show[
 Graphics3D@{Opacity[0.2], Sphere[{0, 0, 0}, 1]},
 ParametricPlot3D[
  Re@{xtraj[t], ytraj[t], ztraj[t]}, {t, 0, T},
  Boxed → True, Axes → None, PlotStyle → {{Thick, Blue}}
 ],
 Axes3D[], Boxed → False, ViewPoint → {6, 2, 1}, ViewVertical → ez
]
```



simulate magnetization inversion as a function of offset

```
ListPlot[
 Table[
  {νoff,
   TransformationAmplitude[
     opI["z"],
     CosineModulatedGaussianPulse1[{π, 0}],
     BackgroundGenerator → 2 π νoff opI["z"]
    ][opI["z"]]
  },
  {νoff, -200, 200, 2}
 ],
 Frame → True, PlotRange → All, PlotStyle → Thick, Joined → True,
 LabelStyle → Directive[Medium, FontFamily → "Helvetica"],
 FrameLabel → {"resonance offset /Hz", "z→z amplitude"}
]
```



The performance of the two shapes is essentially identical.

simulate magnetization trajectories for pairs of pulses, each with flip angle π/2

This simulation shows that only type #2 pulses may be chained together, since the cosine modulations must be coherent.

pairs of CosineModulatedGaussianPulse1

```
pulsepair1 =
  {CosineModulatedGaussianPulse1[{π/2, 0}],
   CosineModulatedGaussianPulse1[{π/2, 0}]};
```

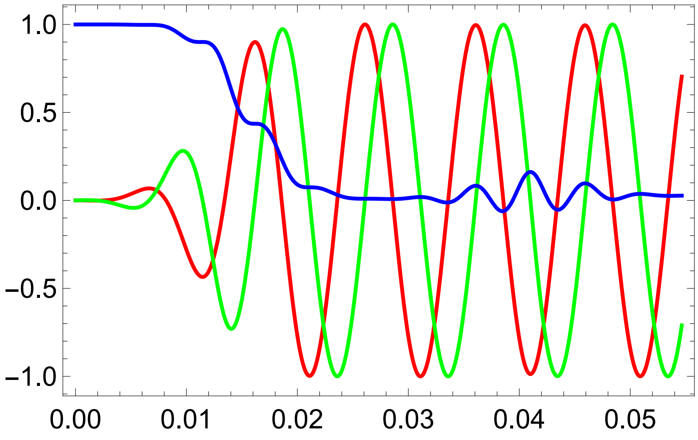
```
T = EventDuration[pulsepair1]
```

$$\frac{2\sqrt{\text{Log}[100]}}{25\,\pi}$$

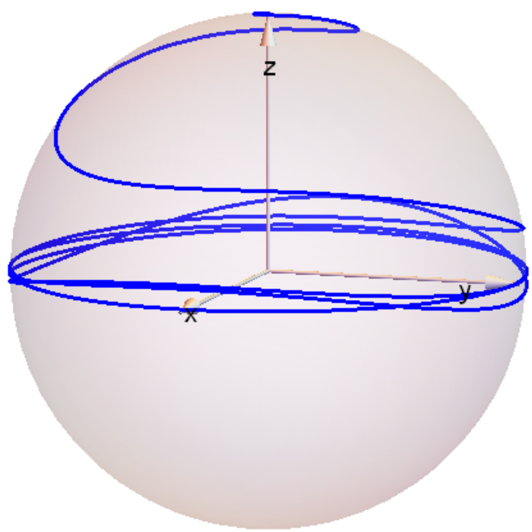
```
{xtraj, ytraj, ztraj} = Trajectory[{opI["x"], opI["y"], opI["z"]}, pulsepair1,
    BackgroundGenerator → 2 π 100 opI["z"]
   ][opI["z"]]
```

```
{TrajectoryFunction[ {{0, 54.6466 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 54.6466 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 54.6466 × 10^-3}} , <>]}
```

```
Plot[{xtraj[t], ytraj[t], ztraj[t]}, {t, 0, T}, Frame → True,
 PlotRange → All, PlotStyle → {{Thick, Red}, {Thick, Green}, {Thick, Blue}}]
```




```
Show[
 Graphics3D@{Opacity[0.2], Sphere[{0, 0, 0}, 1]},
 ParametricPlot3D[
  Re@{xtraj[t], ytraj[t], ztraj[t]}, {t, 0, T},
  Boxed → True, Axes → None, PlotStyle → {{Thick, Blue}}
 ],
 Axes3D[], Boxed → False, ViewPoint → {6, 2, 1}, ViewVertical → ez
]
```



The two pulses do not reinforce each other properly, since they are not phase-coherent

pairs of CosineModulatedGaussianPulse2

```
pulsepair2 =
  {CosineModulatedGaussianPulse2[{π/2, 0}],
   CosineModulatedGaussianPulse2[{π/2, 0}]};
```

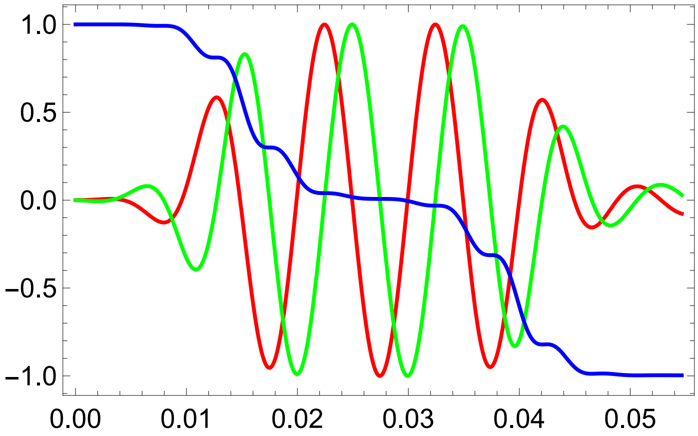
```
T = EventDuration[pulsepair2]
```

$$\frac{2\sqrt{\mathrm{Log}[100]}}{25\pi}$$

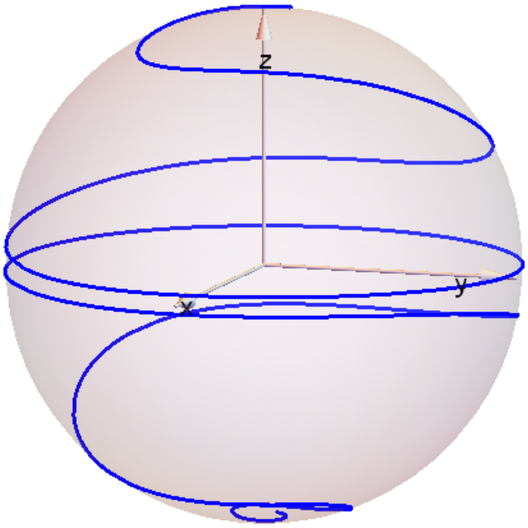
```
{xtraj, ytraj, ztraj} = Trajectory[{opI["x"], opI["y"], opI["z"]}, pulsepair2,
    BackgroundGenerator → 2 π 100 opI["z"]
  ][opI["z"]]
```

```
{TrajectoryFunction[ {{0, 54.6466 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 54.6466 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 54.6466 × 10^-3}} , <>]}
```

```
Plot[{xtraj[t], ytraj[t], ztraj[t]}, {t, 0, T}, Frame → True,
 PlotRange → All, PlotStyle → {{Thick, Red}, {Thick, Green}, {Thick, Blue}}]
```



```
Show[
 Graphics3D@{Opacity[0.2], Sphere[{0, 0, 0}, 1]},
 ParametricPlot3D[
  Re@{xtraj[t], ytraj[t], ztraj[t]}, {t, 0, T},
  Boxed → True, Axes → None, PlotStyle → {{Thick, Blue}}
 ],
 Axes3D[], Boxed → False, ViewPoint → {6, 2, 1}, ViewVertical → ez
]
```



This time the pulses interfere constructively