*Two level system semiclassically with spontaneous decay introduced phenomenologically*
*HGSFP Graduate Days October 2018*
*Adriana Palffy*

Equations to solve:

1. Spontaneous decay from e to g

$\frac{d}{dt}\rho_{ee} = -\gamma * \rho_{ee} + i\Omega\,(\rho_{eg} - \rho_{ge})$

$\frac{d}{dt}\rho_{gg} = \gamma * \rho_{ee} - i\Omega\,(\rho_{eg} - \rho_{ge})$

$\frac{d}{dt}\rho_{eg} = (i\Delta - \gamma / 2) * \rho_{eg} + i\Omega\,(\rho_{ee} - \rho_{gg})$

$\frac{d}{dt}\rho_{ge} = (-i\Delta - \gamma / 2) * \rho_{ge} - i\Omega\,(\rho_{ee} - \rho_{gg})$

Initially, the atom is in the ground state.

2. Spontaneous decay from e/g outside the system

$\frac{d}{dt}\rho_{ee} = -\gamma e * \rho_{ee} + i\Omega\,(\rho_{eg} - \rho_{ge})$

$\frac{d}{dt}\rho_{gg} = -\gamma g * \rho_{ee} - i\Omega\,(\rho_{eg} - \rho_{ge})$

$\frac{d}{dt}\rho_{eg} = (i\Delta - (\gamma g + \gamma e) / 2) * \rho_{eg} + i\Omega\,(\rho_{ee} - \rho_{gg})$

$\frac{d}{dt}\rho_{ge} = (-i\Delta - (\gamma g + \gamma e) / 2) * \rho_{ge} - i\Omega\,(\rho_{ee} - \rho_{gg})$

Initially, the atom is in the ground state.

## Parameters

```
OmegaRabi = 1;
GammaRate = 1;
GammaRateE = 1;
GammaRateG = 0.3;
Detuning = 0;
PulseDuration = 10;
```

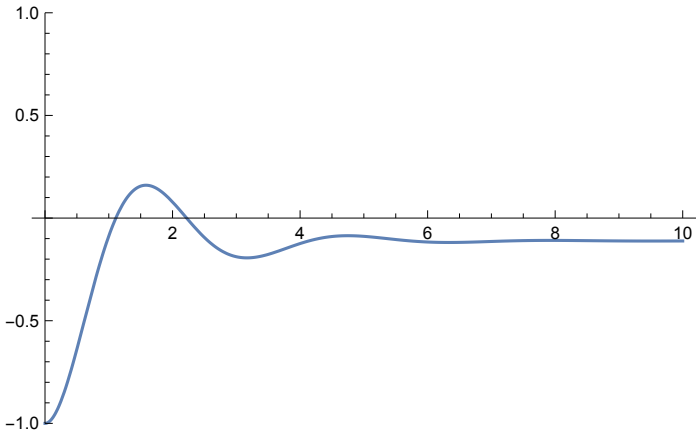
## The Bloch Equations System 1

```
Eqns1 = {D[ρgg[t], t] == GammaRate * ρee[t] - I * OmegaRabi *
      (ρeg[t] - ρge[t]),                                    D[ρge[t], t] ==
   -i * Detuning * ρge[t] + i * OmegaRabi * (ρgg[t] - ρee[t]) - GammaRate * ρge[t] / 2,
  D[ρee[t], t] == -GammaRate * ρee[t] + I * OmegaRabi * (ρeg[t] - ρge[t]),
  D[ρeg[t], t] == i * Detuning * ρeg[t] +
    i * OmegaRabi * (ρee[t] - ρgg[t]) - GammaRate * ρeg[t] / 2,
  ρgg[0] == 1, ρee[0] == 0, ρge[0] == 0, ρeg[0] == 0};
```

```
ToSolve = {ρgg, ρee , ρge, ρeg};
```

```
sol1 = DSolve[Eqns1, ToSolve, t];
```

```
Plot[Re[ρee[t] /. sol1] - Re[ρgg[t] /. sol1],
 {t, 0, PulseDuration}, PlotRange → {-1, 1}]
```




## The Bloch Equations System 2

```
Eqns2 = {D[ρgg[t], t] == -GammaRateG * ρgg[t] -
      I * OmegaRabi * (ρeg[t] - ρge[t]),
   D[ρge[t], t] == -i * Detuning * ρge[t] + i * OmegaRabi * (ρgg[t] - ρee[t]) -
      (GammaRateE + GammaRateG) * ρge[t] / 2,
   D[ρee[t], t] == -GammaRateE * ρee[t] + I * OmegaRabi * (ρeg[t] - ρge[t]),
   D[ρeg[t], t] == i * Detuning * ρeg[t] + i * OmegaRabi * (ρee[t] - ρgg[t]) -
      (GammaRateE + GammaRateG) * ρeg[t] / 2,
   ρgg[0] == 1, ρee[0] == 0, ρge[0] == 0, ρeg[0] == 0};
```

```
ToSolve = {ρgg, ρee , ρge, ρeg};
```

```
sol2 = DSolve[Eqns2, ToSolve, t];
```

```
Plot[Re[ρee[t] /. sol2] - Re[ρgg[t] /. sol2],
 {t, 0, PulseDuration}, PlotRange → {-1, 1}]
```