Plasmon hybridization

We want to solve the equation system

$$\begin{aligned} p_1 =& \epsilon_0\, \epsilon_{out}\, \alpha_1 \left[E^{\mathrm{inc}} + \frac{1}{4\pi\epsilon_0\, \epsilon_{out}} \frac{v}{d^3}\, p_2 \right] \\ p_2 =& \epsilon_0\, \epsilon_{out}\, \alpha_2 \left[E^{\mathrm{inc}} + \frac{1}{4\pi\epsilon_0\, \epsilon_{out}} \frac{v}{d^3}\, p_1 \right] \end{aligned}$$

We rewrite this as

$$\begin{aligned} E^{\mathrm{inc}} =& \frac{1}{\epsilon_0\, \epsilon_{out}\, \alpha_1} p_1 - \frac{1}{4\pi\, \epsilon_0\, \epsilon_{out}} \frac{v}{d^3}\, p_2 \\ E^{\mathrm{inc}} =& \frac{1}{\epsilon_0\, \epsilon_{out}\, \alpha_2} p_2 - \frac{1}{4\pi\, \epsilon_0\, \epsilon_{out}} \frac{v}{d^3}\, p_1 \end{aligned}$$

or as matrix

$$E^{\mathrm{inc}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\epsilon_0\, \epsilon_{out}} \begin{pmatrix} \frac{1}{\alpha_1} & \frac{1}{4\pi} \frac{v}{d^3} \\ \frac{1}{4\pi} \frac{v}{d^3} & \frac{1}{\alpha_2} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}$$

ϵ_in (generic function with 1 method)

```
# Drude model for the metal, all constants are defined below
function ϵ_in(ω)
    return ϵ_∞ - ω_p^2 / (ω * (ω + 1im * γ))
end
```

α (generic function with 1 method)

```
# Polarizability of the particle
function α(R, ω)
    return 4 * pi * R^3 *  (ϵ_in(ω)  - ϵ_out)  / (ϵ_in(ω)  + 2 * ϵ_out)
end
```

Amatrix (generic function with 1 method)

```
# System matrix (2x2 complex)
function Amatrix(ω)
    return [ 1 / α(R1, ω)          -v / ( 4 * pi * d^3)  ;
            -v / ( 4 * pi * d^3)     1 / α(R2, ω)          ] / (ε_0 * ϵ_out)
end
```

```
α_eff (generic function with 1 method)
```

```julia
# effective polaizability
function α_eff(ω)
    amul = ustrip.(u"F^-1 * m^-2",  Amatrix(ω))  # strip units
    Evec = [1;1];                               # units  V/m

    p = (amul \ Evec)                            # solve eq. system for p

    p = ( p * 1u"V/m") /  1u"F^-1 * m^-2"      # put back units, now C m

    return  p / (ε_0 * ϵ_out * 1u"V/m")
end
```

```
Cabs (generic function with 1 method)
```

```julia
# Absorption cross section
function Cabs(ω)
    return ω / c_0 * sum(imag(α_eff(ω))) |> u"nm^2"
end
```

```
Cext (generic function with 1 method)
```

```julia
# Extinction cross section
function Cext(ω)
    return (ω / c_0)^4 / (6 * pi) * sum(abs.(α_eff(ω)).^2)  |> u"nm^2"
end
```

```
ω_res0 = 2.846049894151541 eV
```

```julia
# particle plasmon resonance energy
ω_res0 = ħ * ω_p / sqrt(2 * ϵ_out + ϵ_∞) |> u"eV"
```

```
[2.76864 eV, 2.91474 eV]
```

```julia
# estimated hybridized energies
begin
    η = ( ϵ_∞ - ϵ_out ) / (2 * ϵ_out + ϵ_∞) ;
    g = ν * (sqrt(R1 * R2)/d)^3
    ω_res0_hyb = ω_res0 * [ (1 + g) / (1 + η * g), (1 - g) / (1 - η * g)]
end
```

```
2.5:0.004020100502512563:3.3
```

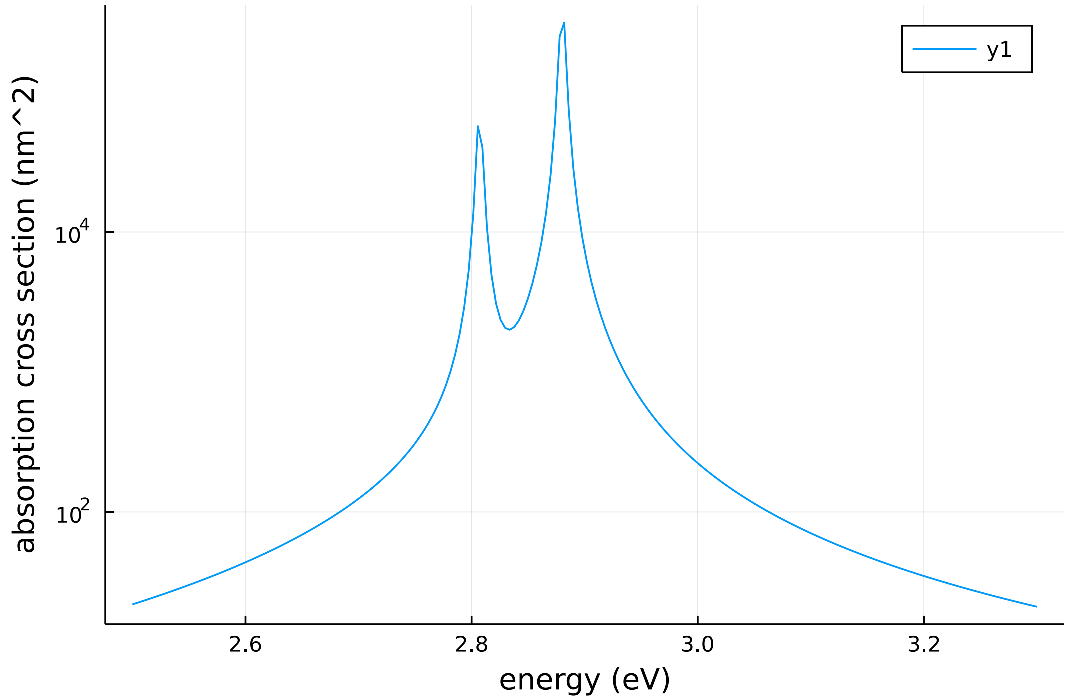
```julia
# some constants
begin
    ω_p = 9u"eV" / ħ;  # frequency units are eV
    γ = 0.005u"eV" / ħ;
    ϵ_∞ = 8;

    ϵ_out = 1;

    ν = -1;
    R1 = 15u"nm";
    R2 = 25u"nm";
    d = d_slider * 1u"nm";

    energy_range = range(2.5, 3.3; length = 200) ; # photon energy range of plot
end
```

44

```julia
@bind d_slider Slider((1:100) ; show_value=true)
```

Everything is OK

d is larger than $R_1 + R_2$




```
plot(energy_range, @. ustrip(u"nm^2", Cabs(energy_range * 1u"eV" / ħ)) ; yaxis=:log,
xlabel = "energy (eV)", ylabel = "absorption cross section (nm^2)")
```

```
# libraries that we need
using Plots, PlutoUI, Unitful
```

```
import PhysicalConstants.CODATA2018: c_0, ε_0, ħ
```