Exercise 1. Consider the weakly nonlinear oscillator:

$$\frac{d^2u}{dt^2} + u + \epsilon u^5 = 0$$

with $u(0) = 0$ and $u'(0) = A > 0$ and with $0 < \epsilon \ll 1$.

(a) Use a regular perturbation expansion and calculate the first two terms.

(b) Determine at what time the approximation of part (a) fails to hold.

(c) Use a Poincare-Lindstedt expansion and determine the first two terms and frequency corrections.

(d) For $\epsilon = 0.1$ plot the numerical solution (from MATLAB), the regular expansion solution, and the Poincare-Lindstedt solution for $0 \leq t \leq 20$.

Solution 1. (a) We begin by writing

$$u(t) = u_0(t) + \epsilon u_1(t) + \cdots .$$

Plugging this subsitution in the differential equation, we can gather terms of the same order. For example, we have that

$$O(1): \qquad u_{0tt} + u_0 = 0,$$

since all u^5 terms are of order ϵ or higher. We can get the first and second order coeffiecient as

$$\begin{aligned} O(\epsilon): &\qquad u_{1tt} + u_1 = -u_0^5 \\ O(\epsilon^2): &\qquad u_{2tt} + u_2 = -5u_1u_0^4. \end{aligned}$$

We've used the binomial theorem to find the coefficients in the expansion

$$\epsilon u^5 = \epsilon \left(\sum_{n=0}^{\infty} \epsilon^n u_n \right)^5 = \epsilon u_0^5 + \epsilon^2 u_1 u_0^4 + \cdots .$$

Starting with the $O(1)$ equation, we see that with the initial conditions $u_0(0) = 0$ and $u_0'(0) = A > 0$, we have that

$$u_0(t) = A\sin(t).$$

This allows us to simplify the first order equation as

$$\begin{gathered} u_{1tt} + u_1 = -A^5\sin^5(t) = -\frac{A^5}{16}(\sin(5t) - 5\sin(3t) + 10\sin(t)) \\ u_1(0) = 0, \quad u_1'(0) = 0. \end{gathered}$$

This equation has solution

$$u_1(t) = \frac{A^5}{384}\left(-80\sin(t) - 15\sin(3t) + \sin(5t) + 120t\cos(t)\right)$$

(b) Given that our approximation is written as

$$u(t) \approx u_0(t) + \epsilon u_1(t),$$

we see this approximation will grow unbounded as $t \approx \epsilon^{-1}$ due to growth term given by $\frac{120}{384} A^5 t \cos(t)$ above.
(c) We'll now use a Poincare-Lindstedt expansion. This is using the similar expansion of u but we addditionally change variables to $\tau = \omega t$ so that

$$\begin{aligned} \omega &= \omega_0 + \epsilon\omega_1 + \epsilon^2\omega_2 + \cdots \\ u(\tau) &= u_0(\tau) + \epsilon u_1(\tau) + \epsilon^2 u_2(\tau) \\ \omega^2 u_{\tau\tau}(\tau) &+ u(\tau) + \epsilon u(\tau)^5 = 0. \end{aligned}$$

This gives leading order equation

$$\begin{aligned} \omega_0^2 u_{0\tau\tau} + u_0 &= 0 \\ u_0(0) = 0, \quad u_0'(0) &= 0. \end{aligned}$$

Picking $w_0 = 1$ for simplicity, we see that this has leading order solution

$$u_0 = A\sin(\tau).$$

We can now find the first order approximation

$$u_{1\tau\tau} + u_1 = -u_0^5 - 2\omega_1 u_{0\tau\tau}.$$

This gives us a solution

$$u_1(\tau) = \frac{A^5}{384}\left(-80\sin(\tau) - 15\sin(3\tau) + \sin(5\tau) + 120\tau\cos(\tau)\right) + A\omega_1(\tau\cos(\tau) - \sin(\tau)).$$

To remove the secular growth term, we solve

$$\begin{aligned} \left(\frac{120}{384}A^5 + A\omega_1\right)\tau\cos(\tau) &= 0. \\ \omega_1 &= -\frac{120}{384}A^4. \end{aligned}$$

(d) Plotting attached in appendix.

Exercise 2. Consider Rayleigh's equation:

$$\frac{d^2u}{dt^2} + u + \epsilon\left[-\frac{du}{dt} + \frac{1}{3}\left(\frac{du}{dt}\right)^3\right] = 0$$

which has only one periodic solution called a "limit cycle" $0 < \epsilon \ll 1$. Given

$$u(0) = 0 \quad \text{and} \quad \frac{du}{dt}(0) = A.$$

(a) Use a multiple scale expansion to calculate the leading order behavior.

(b) Use a Poincare-Lindsted expansion and an expansion of $A = A_0 + \epsilon A_1 + \ldots$ to calculate the leading-order solution and the first non-trivial frequency shift for the limit cycle.

(c) For $\epsilon = 0.01, 0.1, 0.2$ and 0.3, plot the numerical solution and the multiple scale expansion for $0 \leq t \leq 40$ and for various values of A for your multiple scale solution. Also plot the limit cycle solution calculated from part (b)

(d) Calculate the error

$$E(t) = |y_{\text{numerical}}(t) - y_{\text{approx}}(t)|$$

as a function of time ($0 \leq t \leq 40$) using $\epsilon = 0.01, 0.1, 0.2$ and 0.3.

Solution 2. (a) Adding an independent scaled time variable $\tau = \epsilon t$ to our differential equation, we get that our leading order term is given by

$$\begin{aligned} u_{0tt} + u_0 &= 0 \\ u_0(0,0) = 0, \quad u_{0t}(0,0) &= A. \end{aligned}$$

Solving this, gives the leading order behavior as

$$\begin{aligned} u_0(t,\tau) &= A(\tau)\sin(t) + B(\tau)\cos(t) \\ A(0) = A, &\quad B(0) = 0. \end{aligned}$$

Writing out the $O(\epsilon)$ terms of the expansion, we can derive formulas for $A(\tau)$ and $B(\tau)$. Writing $\rho(\tau) = 4A^2/(A^2 + (4 - A^2)\exp(-\tau))$ as in (268) in the lecture notes, we have

$$\begin{aligned} A(\tau) &= \frac{2A}{\sqrt{A^2 + (4 - A^2)\exp(-\tau)}} \\ B(\tau) &= 0. \end{aligned}$$

(b) Now using a PL, we'll expand

$$\begin{aligned} \omega &= \omega_0 + \epsilon\omega_1 + \epsilon^2\omega_2 + \cdots \\ u(\tau) &= u_0(\tau) + \epsilon u_1(\tau) + \epsilon^2 u_2(\tau) \\ A &= A_0 + \epsilon A_1 + \epsilon^2 A_2 \\ \omega^2 u_{\tau\tau} + u - \epsilon\omega u_\tau &+ \epsilon\omega^3 u_\tau^3/3 = 0 \end{aligned}$$

This allows us to write the leading order solution as

$$u_{0\tau\tau} + u_0 = 0,$$
$$u_0(0,0) = 0, u_{0t}(0,0) = A_0$$

where once again I've set $\omega_0 = 1$. This has solution

$$u_0 = A_0 \sin(\tau).$$

Now for the $O(\epsilon)$ term,

$$\begin{aligned} u_{1\tau\tau} + u_1 = u_{0\tau} - u_{0\tau}^3/3 - 2\omega_1 u_{0\tau\tau} = 0, \\ u_1(0,0) = 0, \quad u_{1\tau}(0,0) = A_1. \end{aligned}$$

Plugging in our previous solution, we can simplify the baove equation as

$$\begin{aligned} u_{1\tau\tau} + u_1 &= A_0 \cos(\tau) - A_0^3 \cos^3(\tau)/3 + 2\omega_1 \sin(\tau) \\ &= \left(A_0 - \frac{A_0^3}{4}\right)\cos(\tau) + 2\omega_1 \sin(\tau) - \frac{A_0^3}{12}\cos(3x). \end{aligned}$$

In order to rid ourselves of the secular growth here (caused by the $\sin(\tau)$ and $\cos(\tau)$), we require that

$$A_0 = 2, \quad \omega_1 = 0.$$

This leaves us with

$$\begin{aligned} u_{1\tau\tau} + u_1 = -\frac{2}{3}\left(\cos(3\tau)\right) \\ u_1(0,0) = 0, \quad u_{1\tau}(0,0) = A_1. \end{aligned}$$

This gives solution

$$u_1(\tau) = A_1 \sin(\tau) + \frac{2}{3}\cos(\tau)\cos(3\tau) - \frac{2}{3}\cos(3\tau)$$

Going one more level up, we can solve for the $O(\epsilon^2)$ term as

$$2\omega_2 u_{0\tau\tau} + u_{2\tau\tau} + u_2 - u_{1\tau} + u_{0\tau}^2 u_{1\tau} = 0.$$

which we simplify to

$$\begin{aligned} u_{2\tau\tau} + u_2 &= u_{1\tau} - 2\omega_2 u_{0\tau\tau} - u_{0\tau}^2 u_{1\tau} \\ &= -2A_1 \cos(\tau) - A_1 \cos(3\tau) + \frac{1}{4}\sin(\tau) + 4\omega_2 \sin(\tau) + \frac{1}{6}\sin(3\tau) + \frac{1}{4}\sin(5\tau) \end{aligned}$$

Once again, canceling secular growth terms requires

$$\begin{aligned} -2A_1 = 0 \quad \frac{1}{4} = 4\omega_2 = 0 \\ A_1 = 0, \quad \omega_2 = -\frac{1}{16}. \end{aligned}$$

(c, d) Plots attached in appendix.

HW-4-Code-Figgins

February 12, 2021

[1]:
```
using DifferentialEquations, Plots
```

0.1 Exercise 1

[2]:
```
#Initial Conditions
A = 1.0
  = 0.1

u0 = [0.0]
du0 = [A]
tspan = (0.0, 20.0)
p = ( )

function weak_nl(ddu, du, u, p, t)
      = p[1]
    ddu .= -(u +  *u .^ 5)
end

prob = SecondOrderODEProblem(weak_nl, du0, u0, tspan,  p)
sol = solve(prob, Tsit5())

Exer1_plot = plot(sol, vars = [2],
            label = "DifferentialEquations.jl",
            linewidth = 1.5,
            title = "Exercise (1d)");
```

[3]:
```
## Regular Perturbation Expansion
function exer_1_perturb(t,  , A)
    u0 = A*sin(t)
    u1 = (A^5 / 384) * (-80*sin(t) - 15*sin(3*t) + sin(5*t) + 120*t*cos(t))
    return u0 +  *u1
end

Exer1_plot = plot!(t -> exer_1_perturb(t,  , A), 0, 20,
            label = "Regular Peturbation Expansion",
            linewidth = 1.5,
            linestyle = :dash,
```

```
                color = "green");
```

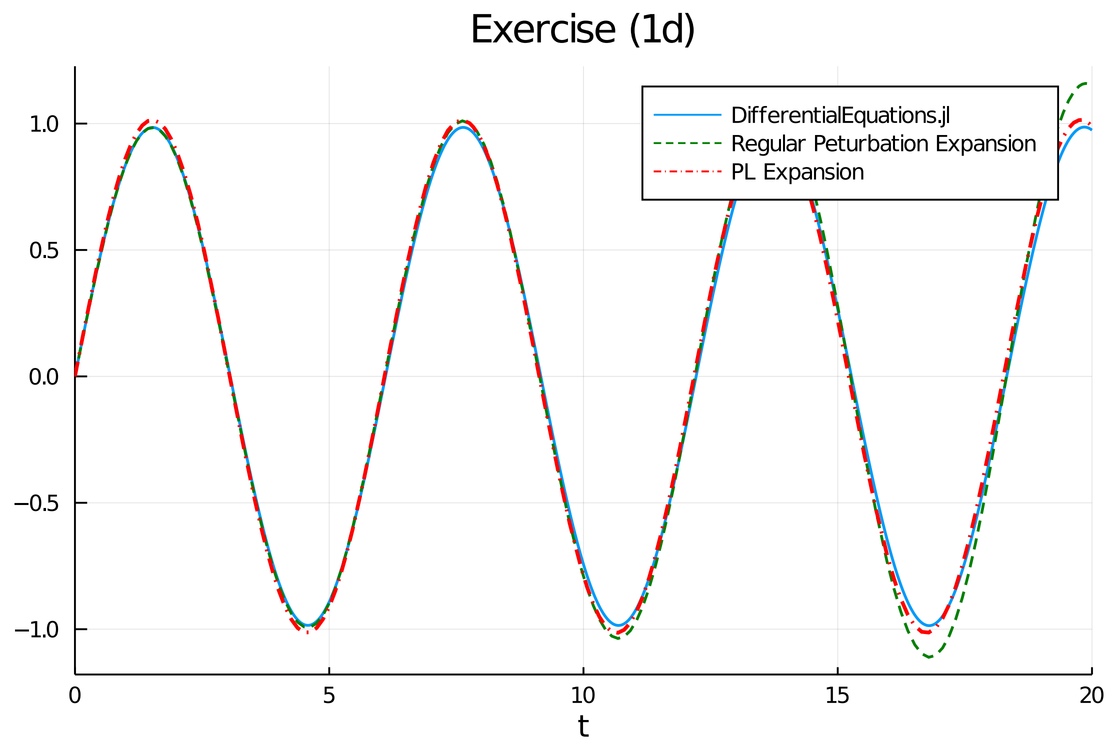
[4]:

```
## Poincare-Lindstedt
function exer_1_PL(t,  , A)
     1 = - 120/384 * A^4
      = 1 +  * 1
      = t/
    u0 = A*sin( )
    u1 = (A^5 / 384) * (-80*sin( ) - 15*sin(3* ) + sin(5* ))
    u1 += A* 1*(- sin( ))
    return u0 +  *u1
end

Exer1_plot = plot!(t -> exer_1_PL(t,  , A), 0, 20,
                label = "PL Expansion",
                linewidth = 2.0,
                linestyle = :dashdot,
                color = "red");
```

[5]:

```
Exer1_plot
```

[5]:



0.2 Exercise 2

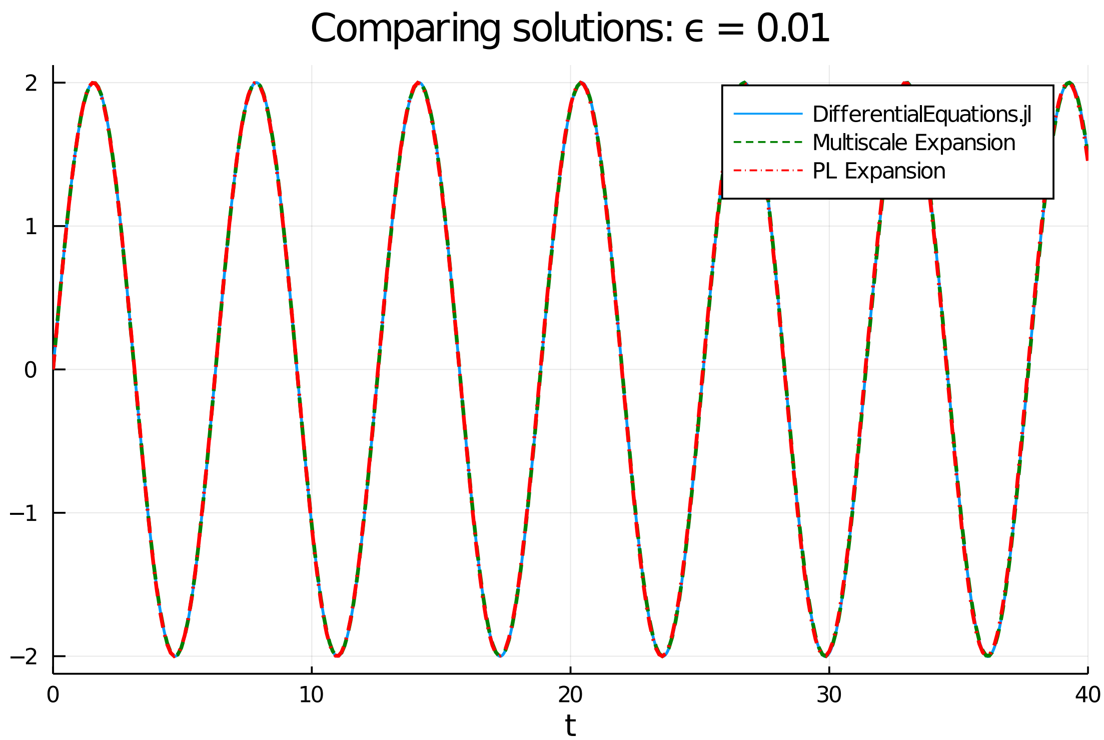
[6]:
```
#Initial Conditions
A = 2.0
  = 0.01

u0 = [0.0]
du0 = [A]
tspan = (0.0, 40.0)
p = ( )

function rayleigh(ddu, du, u, p, t)
      = p[1]
    ddu .= -u +  *du - (1/3)* *du .^3
end

prob = SecondOrderODEProblem(rayleigh, du0, u0, tspan,  p)
sol = solve(prob, Tsit5())

Exer2_plot = plot(sol, vars = [2],
            label = "DifferentialEquations.jl",
            linewidth = 1.5,
            title = "Comparing solutions:   = 0.01");
```

[7]:
```
## Poincare-Lindstedt
function exer_2_MS(t,  , A)
      =  *t
    A  = 2 * A / sqrt(A^2 + (4 - A^2)*exp(- ))
    u0 = A *sin(t)
    return u0
end

Exer2_plot = plot!(t -> exer_2_MS(t,  , A), 0, 40,
            label = "Multiscale Expansion",
            linewidth = 2.0,
            linestyle = :dash,
            color = "green");
```

[8]:
```
## Poincare-Lindstedt
function exer_2_PL(t,  , A)
    A0 = 2
     2 = - 1/16
      = 1 +  * 2
      = t/
    u0 = A0*sin( )
    u1 = 2/3 * (cos( ) - 1)*cos(3* )
    return u0 +  *u1
```

```
end

Exer2_plot = plot!(t -> exer_2_PL(t,  , A), 0, 40,
            label = "PL Expansion",
            linewidth = 2.0,
            linestyle = :dashdot,
            color = "red")
```

[8]:

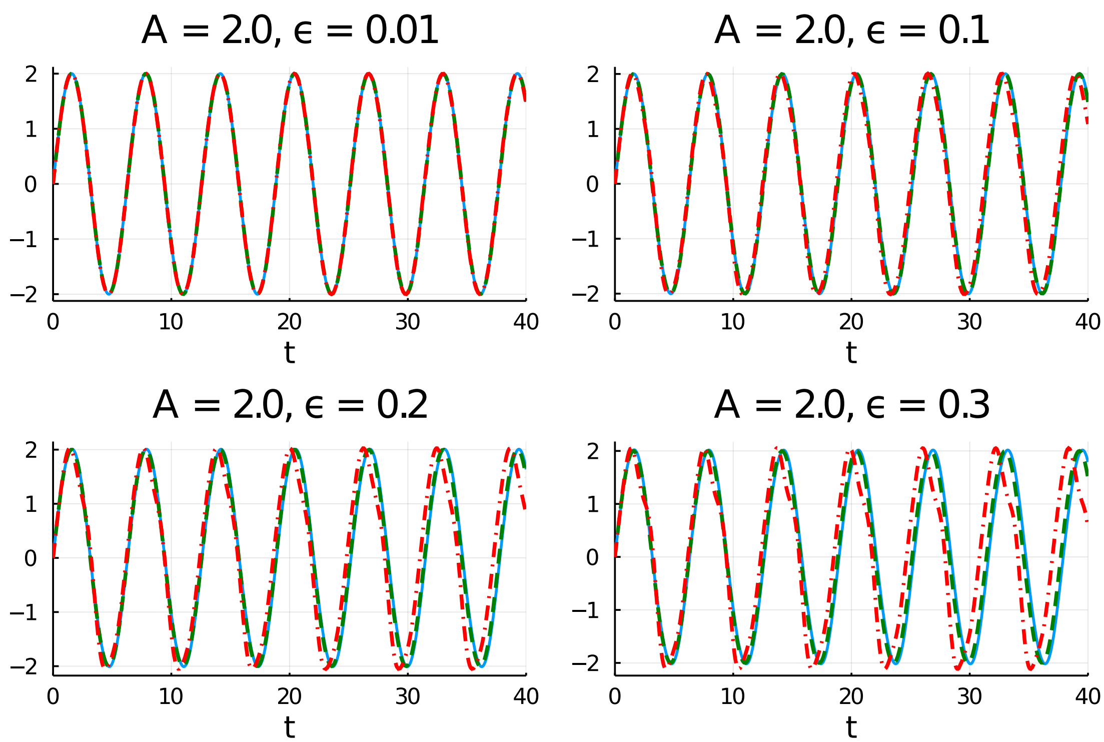


[9]:

```
Exer2_plot_vec = []

A = 2.0
u0 = [0.0]
du0 = [A]

for   in [0.01, 0.1, 0.2, 0.3]

    prob = SecondOrderODEProblem(rayleigh, du0, u0, tspan,  [ ])
    sol = solve(prob, Tsit5())

    Exer2_plot = plot(sol, vars = [2],
                label = "DifferentialEquations.jl",
                linewidth = 1.5,
                legend = :none,
```

```
                title = "A = $A,   = $ ")

    Exer2_plot = plot!(t -> exer_2_MS(t,  , A), 0, 40,
                label = "Multiscale Expansion",
                linewidth = 2.0,
                linestyle = :dash,
                color = "green");

    Exer2_plot = plot!(t -> exer_2_PL(t,  , A), 0, 40,
                label = "PL Expansion",
                linewidth = 2.0,
                linestyle = :dashdot,
                color = "red");
    push!(Exer2_plot_vec, Exer2_plot)
end

plot(Exer2_plot_vec..., layout = 4)
```

[9]:

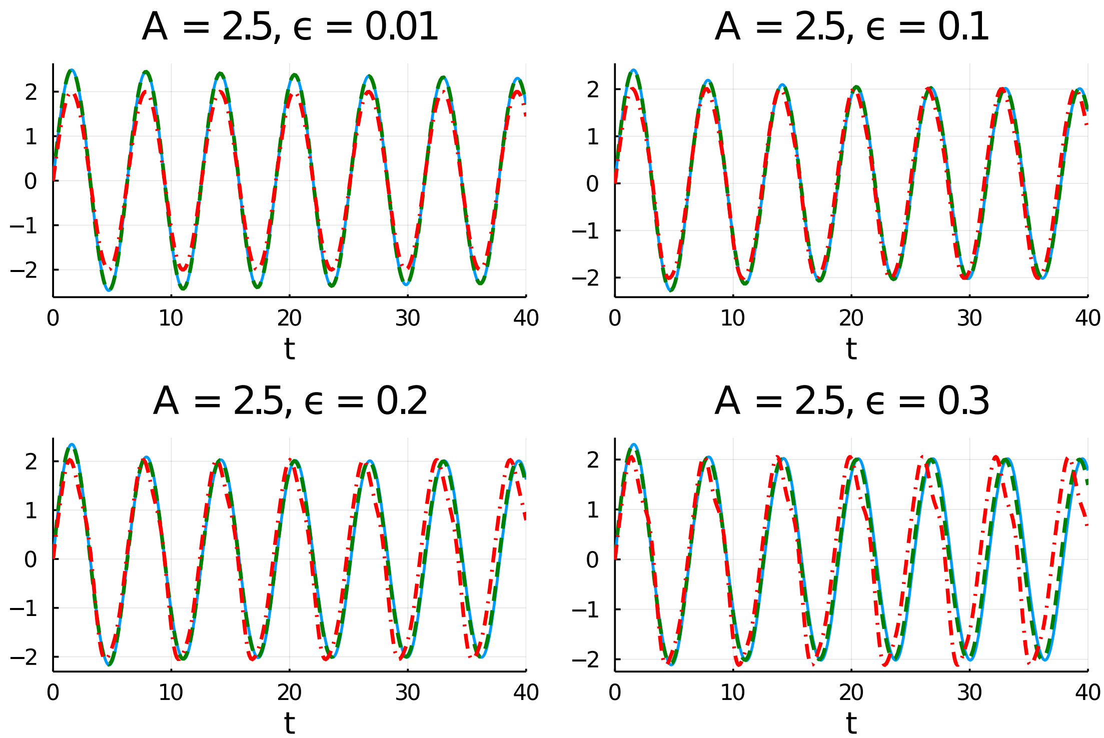


[10]:

```
Exer2_plot_vec = []

A = 2.5
u0 = [0.0]
du0 = [A]
```

```
for   in [0.01, 0.1, 0.2, 0.3]

    prob = SecondOrderODEProblem(rayleigh, du0, u0, tspan,  [ ])
    sol = solve(prob, Tsit5())

    Exer2_plot = plot(sol, vars = [2],
                label = "DifferentialEquations.jl",
                linewidth = 1.5,
                legend = :none,
                title = "A = $A,    = $ ")

    Exer2_plot = plot!(t -> exer_2_MS(t,  , A), 0, 40,
                label = "Multiscale Expansion",
                linewidth = 2.0,
                linestyle = :dash,
                color = "green");

    Exer2_plot = plot!(t -> exer_2_PL(t,  , A), 0, 40,
                label = "PL Expansion",
                linewidth = 2.0,
                linestyle = :dashdot,
                color = "red");
    push!(Exer2_plot_vec, Exer2_plot)
end

plot(Exer2_plot_vec..., layout = 4)
```

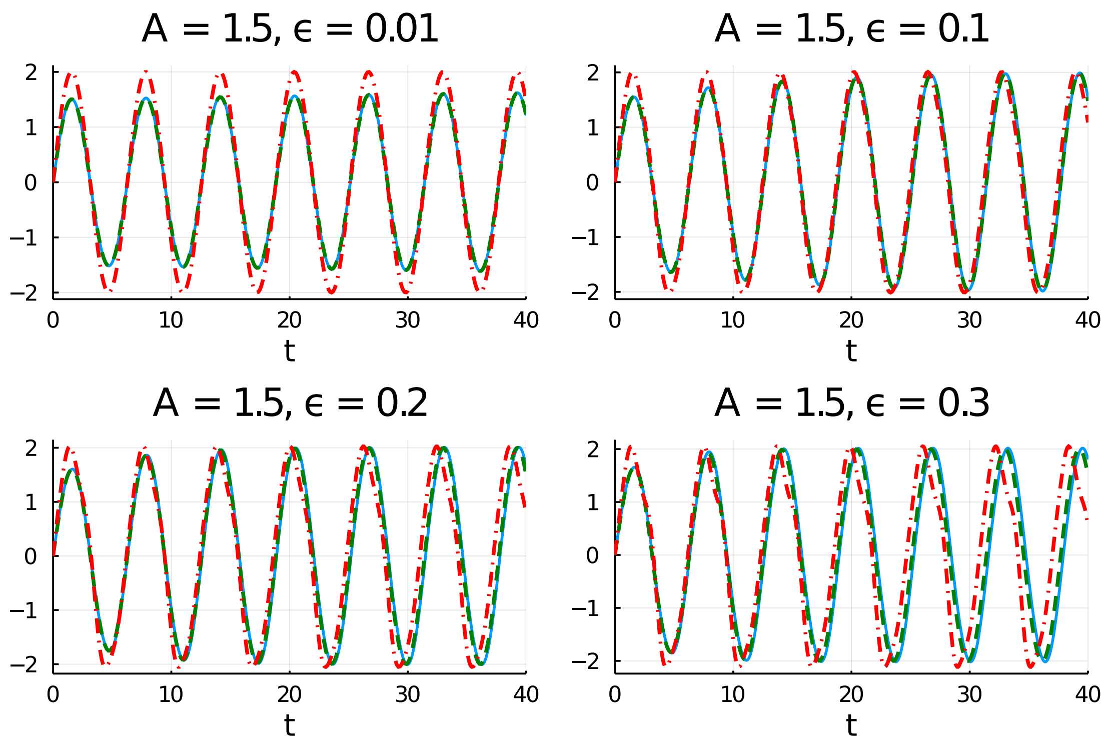
[10]:



[11]:
```
Exer2_plot_vec = []

A = 1.5
u0 = [0.0]
du0 = [A]

for   in [0.01, 0.1, 0.2, 0.3]

    prob = SecondOrderODEProblem(rayleigh, du0, u0, tspan,  [ ])
    sol = solve(prob, Tsit5())

    Exer2_plot = plot(sol, vars = [2],
                label = "DifferentialEquations.jl",
                linewidth = 1.5,
                legend = :none,
                title = "A = $A,   = $ ")

    Exer2_plot = plot!(t -> exer_2_MS(t,  , A), 0, 40,
                label = "Multiscale Expansion",
                linewidth = 2.0,
                linestyle = :dash,
                color = "green");
```

```
    Exer2_plot = plot!(t -> exer_2_PL(t,  , A), 0, 40,
                label = "PL Expansion",
                linewidth = 2.0,
                linestyle = :dashdot,
                color = "red");
    push!(Exer2_plot_vec, Exer2_plot)
end

plot(Exer2_plot_vec..., layout = 4)
```

[11]:



[12]:

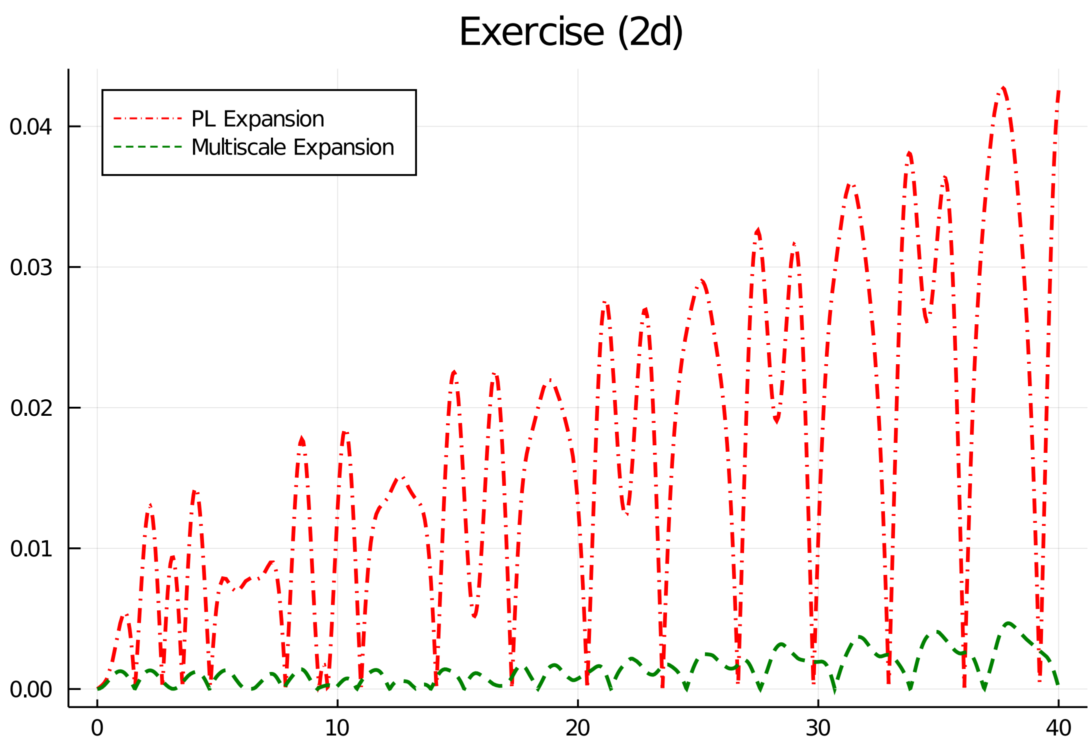
```
## Plotting Error
```

[13]:

```
#Initial Conditions
A = 2.0
  = 0.01

u0 = [0.0]
du0 = [A]
tspan = (0.0, 40.0)
p = ( )

prob = SecondOrderODEProblem(rayleigh, du0, u0, tspan,  p)
sol = solve(prob, Tsit5())
```

```
error_PL(t) = abs(exer_2_PL(t,  , A) .- sol(t)[2])
Exer2_error_plot = plot(t -> error_PL(t), 0, 40,
            label = "PL Expansion",
            linewidth = 2.0,
            linestyle = :dashdot,
            color = "red",
            title = "Exercise (2d) ")

error_MS(t) = abs(exer_2_MS(t,  , A) .- sol(t)[2])
Exer2_error_plot = plot!(t -> error_MS(t), 0, 40,
            label = "Multiscale Expansion",
            linewidth = 2.0,
            linestyle = :dash,
            legend = :topleft,
            color = "green")
```

[13]:



[14]:
```
Exer2_error_plot_vec = []

A = 2.0
u0 = [0.0]
du0 = [A]
```

```
for   in [0.01, 0.1, 0.2, 0.3]

    prob = SecondOrderODEProblem(rayleigh, du0, u0, tspan,   [ ])
    sol = solve(prob, Tsit5())

    error_PL(t) = abs(exer_2_PL(t,  , A) .- sol(t)[2])
    Exer2_error_plot = plot(t -> error_PL(t), 0, 40,
                label = "PL Expansion",
                linewidth = 2.0,
                linestyle = :dashdot,
                color = "red",
                title = "Error: A = $A,   = $  ")

    error_MS(t) = abs(exer_2_MS(t,  , A) .- sol(t)[2])
    Exer2_error_plot = plot!(t -> error_MS(t), 0, 40,
                label = "Multiscale Expansion",
                linewidth = 2.0,
                linestyle = :dash,
                legend = :topleft,
                color = "green")

    push!(Exer2_error_plot_vec, Exer2_error_plot)
end
plot(Exer2_error_plot_vec..., layout = 4)
```

[14]: