18.085 Pset #4 Solutions

August Trollbäck

July 2020

Question 1.

As all the $m_i = c_i = 1$, we use matrices $\mathbf{M} = \mathbf{C} = \mathbf{I}_3$. With our usual stiffness matrix $\mathbf{K} = \mathbf{A}^T\mathbf{A}$, our sytem with zero external force can be written as $\mathbf{M}\mathbf{u}'' + \mathbf{K}\mathbf{u} = \mathbf{0}$. This can be rewritten as $\mathbf{u}'' + \mathbf{G}\mathbf{u} = 0$ with $\mathbf{G} = \mathbf{M}^{-1}\mathbf{K}$. To decouple the system, take the eigendecomposition $\mathbf{G} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1}$ so that we can form:

$$\begin{aligned} \mathbf{u}'' + (\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1})\mathbf{u} &= \mathbf{0} \\ (V^{-1}\mathbf{u}'') + \boldsymbol{\Lambda}(\mathbf{V}^{-1}\mathbf{u}) &= \mathbf{0} \\ \tilde{\mathbf{u}}'' + \boldsymbol{\Lambda}\tilde{\mathbf{u}} &= \mathbf{0} \end{aligned}$$

As $\boldsymbol{\Lambda}$ is diagonal we have three independent differential equations for $\tilde{u}_i$. Each has general solution $\tilde{u}_i(t) = A\exp(i\sqrt{\lambda_i}t) + B\exp(-i\sqrt{\lambda_i}t)$. See next page for Julia code that plots the three modes:

In [1]:

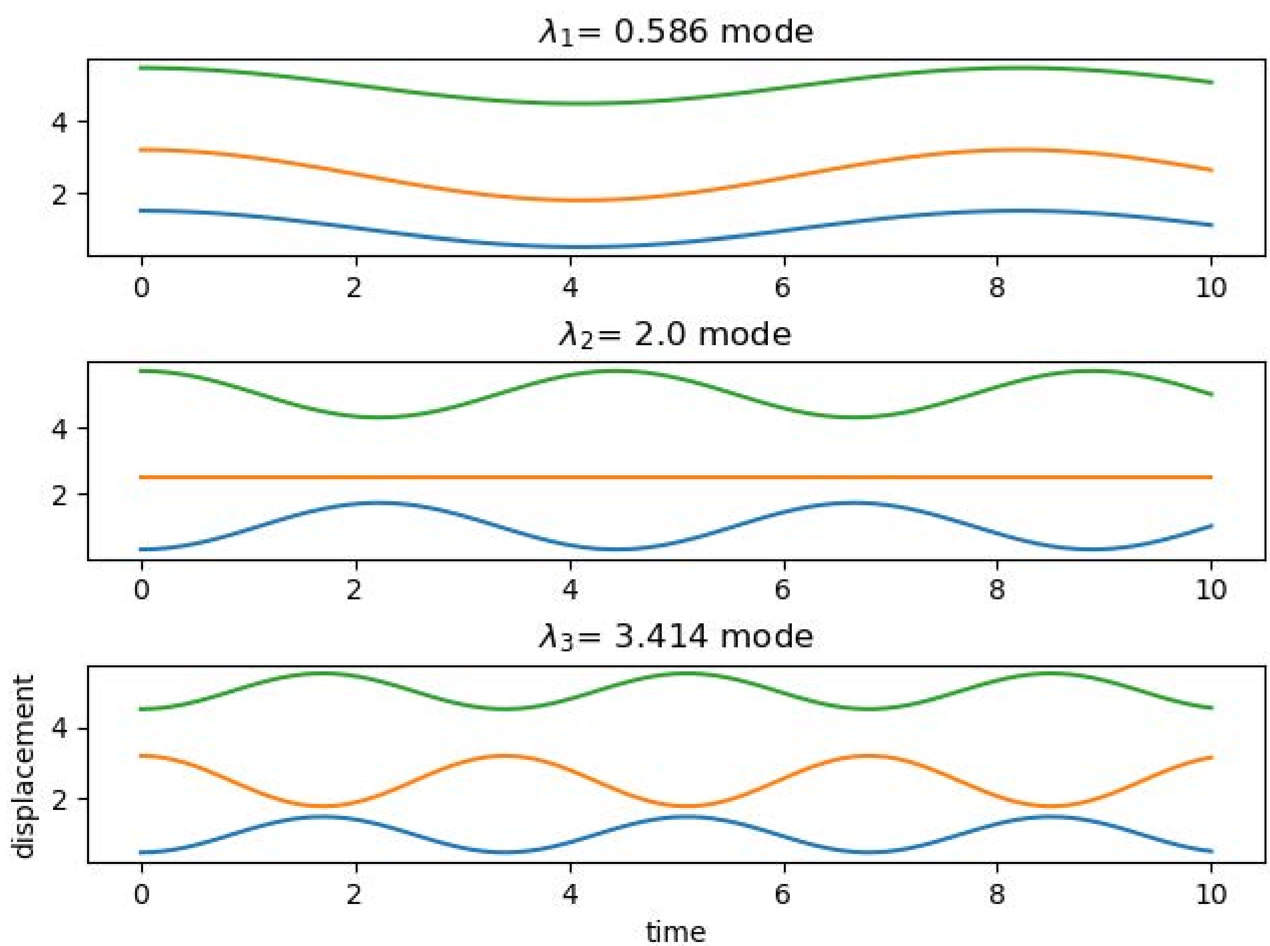
```julia
using LinearAlgebra
using PyPlot
```

In [2]:

```julia
A = [
    1 0 0
    -1 1 0
    0 -1 1
    0 0 -1
]
K = A' * A
Λ, V = eigen(K);
```

In [3]:

```julia
ts = range(0, 10, length=100)
u = zeros(3, length(ts))
disp = [1, 2.5, 5]
for i = 1:3
    subplot(310+i)
    u = hcat([V[:,i] .* cos(sqrt(Λ[i]) * t) for t in ts]...) .+ disp
    plot(ts, u[1,:])
    plot(ts, u[2,:])
    plot(ts, u[3,:])
    title("\$\\lambda_{$i} \$= $(round(Λ[i], digits=3)) mode")
end
xlabel("time")
ylabel("displacement")
tight_layout(pad=0.5)
```



Question 2.

$$
\begin{aligned}
u'(t) &= \frac{\omega_0 \sin \omega_0 t - \lambda \sin \lambda t}{m(\omega_0^2 - \lambda^2)} \\
u''(t) &= \frac{\omega_0^2 \cos \omega_0 t - \lambda^2 \cos \lambda t}{m(\omega_0^2 - \lambda^2)}
\end{aligned}
$$

Then we can see u satisfies the differential equation:

$$
\begin{aligned}
mu'' + cu &= \frac{\omega_0^2 \cos \omega_0 t - \lambda^2 \cos \lambda t}{\omega_0^2 - \lambda^2} + c\frac{\cos \lambda t - \cos \omega_0 t}{m(\omega_0^2 - \lambda^2)} \\
&= \frac{\omega_0^2 \cos \omega_0 t - \lambda^2 \cos \lambda t}{\omega_0^2 - \lambda^2} + \lambda^2 \frac{\cos \lambda t - \cos \omega_0 t}{\omega_0^2 - \lambda^2} \\
&= \frac{\omega_0^2 \cos \omega_0 t - \lambda^2 \cos \omega_0 t}{\omega_0^2 - \lambda^2} = \cos \omega_0 t
\end{aligned}
$$

In the limit $\omega_0 \to \lambda$ we can notice the limit form of the derivative appears (with respect to λ). Alternatively L'Hopital can be used.

$$
\begin{aligned}
\lim_{\omega_0 \to \lambda} \frac{\cos \lambda t - \cos \omega_0 t}{m(\omega_0^2 - \lambda^2)} &= \lim_{\omega_0 \to \lambda} \frac{\cos \lambda t - \cos \omega_0 t}{m(\omega_0 - \lambda)(\omega_0 + \lambda)} \\
&= \lim_{\omega_0 \to \lambda} \frac{1}{2\omega_0 m} \frac{-\cos \omega_0 t - (-\cos \lambda t)}{\omega_0 - \lambda} \\
&= \frac{1}{2\omega_0 m} \frac{d}{d\lambda}\Big[-\cos \lambda t\Big] = \frac{1}{2\omega_0 m} t \sin \lambda t \\
&= \frac{t \sin \omega_0 t}{2\omega_0 m}
\end{aligned}
$$

Question 3.

$$
\begin{aligned}
\frac{e^{ix} - e^{-ix}}{2i} &= \frac{\cos(x) + i\sin(x) - \cos(-x) - i\sin(-x)}{2i} \\
&= \frac{\cos(x) - \cos(x) + i\sin(x) + i\sin(x)}{2i} = \sin(x)
\end{aligned}
$$

$$\begin{aligned}\frac{e^{ix}+e^{-ix}}{2} &= \frac{\cos(x)+i\sin(x)+\cos(-x)+i\sin(-x)}{2} \\ &= \frac{\cos(x)+\cos(x)+i\sin(x)-i\sin(x)}{2} = \cos(x)\end{aligned}$$

$$\begin{aligned}\sin(z)=\sin(x+iy) &= \frac{e^{i(x+iy)}-e^{-i(x+iy)}}{2i} = \frac{e^{ix}e^{-y}-e^{-ix}e^{y}}{2i} \\ &= \frac{e^{ix}-e^{-ix}}{2i}\frac{e^{y}+e^{-y}}{2} + i\frac{e^{ix}+e^{-ix}}{2}\frac{e^{y}-e^{-y}}{2} \\ &= \sin(x)\cosh(y)+i\cos(x)\sinh(y)\end{aligned}$$

Question 4.

See next pages for phase plots. As the plot shows, this system is **unstable**.

Question 5.

See next pages for phase plots. As the plot shows, this system is **stable**.

In [1]:

```julia
using PyPlot
using LinearAlgebra
```

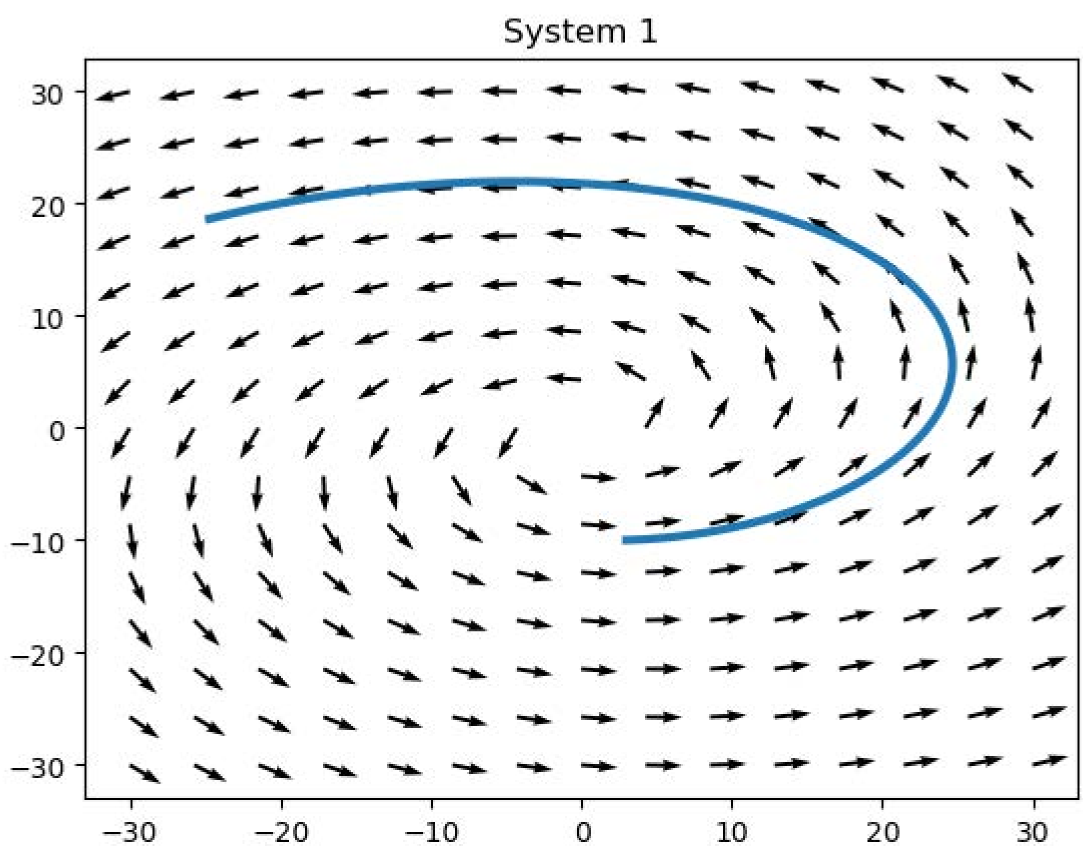
In [29]:

```julia
function plotsystem(A, u₀, name, window=10, time=1)
    Λ, V = eigen(A)

    axis = range(-window, window, length=15)
    meshgrid(x, y) = (repeat(x, outer=length(y)), repeat(y, inner=length(x)))
    x_ax, y_ax = meshgrid(axis, axis)
    UVs = A * [x_ax y_ax]'
    UVs ./= sqrt.(sum(UVs.^2, dims=1))
    u, v = UVs[1,:], UVs[2,:]

    quiver(x_ax, y_ax, u, v)
    title(name)

    ts = range(0, time, length=100)
    c = V \ u₀
    u_sol = hcat([V * (c .* exp.(Λ .* t)) for t in ts]...)
    plot(real.(u_sol[1,:]), real.(u_sol[2,:]), linewidth=3)
end
```

Out[29]: plotsystem (generic function with 3 methods)

In [32]:

```julia
A₁ = [3 -13 ; 5 1]
u₀ = [3, -10]
sol = plotsystem(A₁, u₀, "System 1", 30, 0.45);
```



In [33]:

```
A₁ = [-2 2 ; -2 1]
u₀ = [3, -10]
sol = plotsystem(A₁, u₀, "System 2", 15, 10);
```