Problem 1

Part 1

```
begin
    # PUBLIC PACKAGES
    using Symbolics ✓ , LinearAlgebra ✓ , DifferentialEquations ✓

    # PRIVATE PACKAGES
    include("/home/burtonyale/Documents/repos
/OptimalSpacecraftTrajectories/src/OptimalSpacecraftTrajectories.jl")
    import .OptimalSpacecraftTrajectories
    const OST = OptimalSpacecraftTrajectories

    md"""### Problem 1
    #### Part 1"""
end
```

```
using Plots ✓
```

$$\cos\left(16 - x_1 x_2 x_3\right)$$

```
begin
    @variables x[1:3]
    z = 4
    f = cos(-x[1]*x[2]*x[3] + z^2)
end
```

f′ = Symbolics.Num[

1: $\sin\left(16 - x_1 x_2 x_3\right) x_2 x_3$

2: $\sin\left(16 - x_1 x_2 x_3\right) x_1 x_3$

3: $\sin\left(16 - x_1 x_2 x_3\right) x_1 x_2$

]

```
f′ = Symbolics.derivative.(f, [x[1], x[2], x[3]]; simplify=true)
```

```
f′il = build_function(f′, x, expression=false)[1];
```

```
f′_actual = Float64[
                1:  -3.26413
                2:  -1.63206
                3:  -1.08804
            ]
```

```
f′_actual = f′il([1, 2, 3])
```

Part 2

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

```
function cplxDiff(f, x₀, h)
    f′ = []
    L = length(x₀)
    for i = 1:L
        H = zeros(Complex{Float64}, L)
        H[i] = h*im
        push!(f′, imag(f(x₀ + H))/h)
    end
    return f′
end
```

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

```
func(x) = cos(-x[1]*x[2]*x[3] + z^2)
```

```
0.0
```

```
norm(f′_actual - cplxDiff(func, [1.0, 2.0, 3.0], 1e-20))
```

Part 3

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

```
function cntrDiff(f, x₀, h, N)
    f′ = []
    L = length(x₀)

    if N == 3
        g = (x, δx) -> (f(x + δx) - f(x - δx))/(2*h)
    elseif N == 5
        g = (x, δx) -> (f(x - 2*δx) - 8*f(x - δx) + 8*f(x + δx) - f(
2*δx))/(12*h)
    elseif N == 7
        g = (x, δx) -> (-f(x - 3*δx) + 9*f(x - 2*δx) - 45*f(x - δx)
45*f(x + δx) - 9*f(x + 2*δx) + f(x + 3*δx))/(60*h)
    end

    for i = 1:L
        δx = zeros(L)
        δx[i] = h
        push!(f′, g(x₀, δx))
    end
    return f′
end
```

-3.2641264694904804 -1.6320633081884361 -1.0880422145226332

```
md"$(cntrDiff(func, [1.0, 2.0, 3.0], 1e-4, 3))"
```

-3.2641266653377072 -1.6320633326697789 -1.0880422217761827

```
md"$(cntrDiff(func, [1.0, 2.0, 3.0], 1e-4, 5))"
```

-3.264126665337763 -1.6320633326698715 -1.0880422217775336

```
md"$(cntrDiff(func, [1.0, 2.0, 3.0], 1e-4, 7))"
```

Part 4

```
f″ =
```

$$-{x_3}^2{x_2}^2\cos\left(16-x_1x_2x_3\right)$$

```
f″ = Symbolics.derivative.(f′[1], x[1]; simplify=true)
```

```
f″il = build_function(f″, x, expression=false);
```

```
f″_actual = 30.206575046752288
```

3.552713678800501e-15

```
f″_actual - cplxDiff(f′il, [1.0, 2.0, 3.0], 1e-20)[1][1]
```

Problem 2

Part 1

Period = 8.451040406976288 TU

```
begin
    μ = 1 # LU³/TU²
    x₀ = [1.0, 1.0, 0.1, # LU
          0.1, 0.7, 0.3] # LU/TU
    a = 1 / (2/norm(x₀[1:3]) - (norm(x₀[4:6])^2)/μ)
    # angmom = norm(x₀[1:3]×x₀[4:6])
    # rad = x₀[1:3]; vel = x₀[4:6]
    # velr = rad⋅vel / norm(rad)
    # e = norm((1/μ) * ((norm(vel)^2 - μ/norm(rad)^2) .-
norm(rad)*velr*vel))
    # a = angmom^2 / μ / (1-e^2)
    T = 2*π*sqrt(a^3 / μ)
    md"Period = $(T) TU"
    # md"$(a)"
end
```

Part 2

EoM! (generic function with 1 method)

```
function EoM!(du, u, p, t)
    μ, = p
    r⃗ = u[1:3]
    v⃗ = u[4:6]
    du[1:3] = v⃗
    du[4:6] = -μ*(r⃗/norm(r⃗)^3)
    # Φ = reshape(u[7:end], (6, 6))
    # A = μ*
    return du
end
```

```
prob =
ODEProblem with uType Vector{Float64} and tType Float64. In-place: true
timespan: (0.0, 8.451040406976288)
u0: 6-element Vector{Float64}:
 1.0
 1.0
 0.1
 0.1
 0.7
 0.3
```

```
prob = ODEProblem(EoM!, x₀, (0.0, T), (μ))
```

Integrator: Tsit5 | Tol: 1e-13 | Miss: 1.5771215576378597e-13 LU

```
sol = solve(prob, reltol=1e-13, abstol=1e-13); md"Integrator: Tsit5 | Tol: 1e-13 | Miss: $(norm(sol[1][1:3] - sol[end][1:3])) LU"
```

```
plot(sol, vars=(1,2),
    xlabel="Position-X (LU)", ylabel="Position-Y (LU)",
zlabel="Position-Z (LU)",
label="Orbit", format=:png, aspect_ratio=:equal); savefig("~/Documen
/repos/OptimalSpacecraftTrajectories/src/HW/hw1_2.2eq.png")
```

Part 3

Function Setup

```
begin
    function genSTM(diffFunc, EoM, x₀, xₜ, h)
        L = length(x₀)
        ∂x₀ = reshape(hcat(diffFunc(EoM, x₀, h)...), (L, L))
        ∂xₜ = reshape(hcat(diffFunc(EoM, xₜ, h)...), (L, L))
        Φₜ = ∂xₜ/∂x₀
    end
    EoM_shrthnd1 = (u) -> EoM!(zeros(Complex{Float64}, 6), u, (μ), (
T))
    EoM_shrthnd2 = (u) -> EoM!(zeros(6), u, (μ), (0.0, T))
    EoM_shrthnd1 = (u) -> EoM!(zeros(Complex{Float64}, 6), u, (μ), (
T))
    EoM_shrthnd2 = (u) -> EoM!(zeros(6), u, (μ), (0.0, T))
    EoM_shrthnd3 = (u) -> EoM!(zeros(6), u, (μ), (0.0, T))
    md"Function Setup"
end
```

Calculating STMs

```julia
begin
    # COMPLEX STEP METHOD
    Φa = genSTM(cplxDiff, EoM_shrthnd1, x₀, sol[end], 1e-13);

    # VARIATIONAL METHOD
    ⊗(u, v) = u*transpose(v)
    function EoMSTM!(du, u, p, t)
        μ, = p
        r⃗ = u[1:3]
        r = sqrt(u[1]^2 + u[2]^2 + u[3]^2)
        v⃗ = u[4:6]
        Φ = reshape(u[7:42], (6, 6))

        Idnt = [1.0 0 0; 0 1 0; 0 0 1]
        G = μ/r^5 * ( (3 * r⃗⊗r⃗) - (r^2 * Idnt) ) # ← from Bates
        # G = OST.jacobian(r⃗, μ) # Same value as above
        zmtrx = 0.0*Idnt

        A = [zmtrx Idnt; G zmtrx]
        Φ̇ = A*Φ

        du[1:3] = v⃗
        du[4:6] = -μ*(r⃗/r^3)
        du[7:42] = reshape(Φ̇, 36)
        return du
    end
    probb = ODEProblem(EoMSTM!, vcat(x₀, reshape(1.0*Matrix(I, 6, 6)
36)), (0.0, T), (μ))
    solb = solve(probb, reltol=1e-8, abstol=1e-8);
    Φb = reshape(solb[end][7:42], (6, 6));

    # 3 POINT FINITE DIFFERENCE
    Φc = genSTM((f, x, h) -> cntrDiff(f, x, h, 3), EoM_shrthnd2, x₀,
sol[end], 1e-13)

    # 5 POINT FINITE DIFFERENCE
    Φd = genSTM((f, x, h) -> cntrDiff(f, x, h, 5), EoM_shrthnd2, x₀,
sol[end], 1e-13)
    md"Calculating STMs"
end
```

Part 4

```
6×6 Matrix{Float64}:
  1.0  0.0  0.0  -0.0  -0.0  -0.0
  0.0  1.0  0.0  -0.0  -0.0  -0.0
  0.0  0.0  1.0  -0.0  -0.0  -0.0
  0.0  0.0  0.0   1.0  -0.0   0.0
 -0.0  0.0  0.0  -0.0   1.0   0.0
 -0.0  0.0  0.0  -0.0  -0.0   1.0
```

```
begin
    ∂x₀ = reshape(hcat(cplxDiff((u) -> EoM!(zeros(Complex{Float64},
u, (μ), (0.0, T)), x₀, 1e-13)...), (6, 6))
    ∂xₜ = reshape(hcat(cplxDiff((u) -> EoM!(zeros(Complex{Float64},
u, (μ), (0.0, T)), sol[end], 1e-13)...), (6, 6))
    Φ = ∂xₜ/∂x₀
end
```

Part 5

```
2.4494897427834053
```

```
norm(Φ)
```

Part 6

{a,b} = 25.87295976748877 ← I've double checked my method in MATLAB and usin
couple of papers, I am unsure where this error is coming from

{a,c} = 0.0026762560780534987

{a,d} = 0.003485831114903648

```
md"""#### Part 6
{a,b} = $(norm(Φb - Φa)) ← I've double checked my method in MATLAB a
using a couple of papers, I am unsure where this error is coming fro

{a,c} = $(norm(Φc - Φa))

{a,d} = $(norm(Φd - Φa))"""
```

Part 7

```
ode5 (generic function with 1 method)
```

```
function ode5(f, tspan, y₀)
    h = diff(tspan)[1];
    neq = length(y₀)
    N = length(tspan)
    Y = zeros(neq, N)

    C = [0.2, 0.3, 0.8, 8/9, 1]
    A = transpose([0.2 0 0 0 0;
         3/40 9/40 0 0 0;
         44/45 -56/15 32/9 0 0;
         19372/6561 -25360/2187 64448/6561 -212/729 0;
         9017/3168 -355/33 46732/5247 49/176 -5103/18656])
    B = [35/384, 0, 500/1113, 125/192, -2187/6784, 11/84]

    nstages = length(B)
    F = zeros(neq, nstages)

    Y[:, 1] = y₀
    for i = 2:N
        ti = tspan[i-1]
        hi = h[i-1]
        yi = Y[:, i-1]

        F[:, 1] = f(ti, yi)
        for stageccc = 2:nstages
            tstage = ti + C[stage-1]*hi
            ystage = ti + F[:, 1:stage-1]*(hi*A[1:stage-1, stage-1])
            F[:, stage] = f(tstage, ystage)
        end
        Y[:, i] = yi + F*(hi*B)
    end
    Y = transpose(Y)
end
```

```
# On MATLAB
```

Part 8

$$f^{\prime\prime} = -\frac{f_{-2} - 16f_{-1} + 30f_{0} - 16f_{1} + f_{2}}{-12h^2}$$

```
function cntrDiff′(f, x₀, h)
    f″ = []
    L = length(x₀)

    for i = 1:L, j = 1:L
        δx = δy = zeros(L)
        δx[i] += h
        δy[j] += h
        # push!(f″, ( f(x₀ + δx + δy) - f(x₀ + δx) - f(x₀ + δy) + f(
)/h^2)
        push!(f″, -(f(x₀ - 2*δx) - 16*f(x₀ - δx) + 30*f(x₀) - 16*f(x
δx) + f(x₀ + 2*δx))/(12*h^2))
    end
    f″ = reshape(f″, (6, 6))
    return f″
end; md"$f^{\prime\prime}  = -\frac{f_{-2} - 16f_{-1} + 30f_{0} - 16
+ f_{2}}{-12h^2}$"
```

Part 9

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

```
function KE(u, μ, T, idx)
    sol = solve(ODEProblem(EoM!, u, (0.0, T), (μ)), reltol=1e-13,
abstol=1e-13)
    if idx == -1
        r⃗ = sol[end][1:3]
        v⃗ = sol[end][4:6]
    else
        r⃗ = sol[idx][1:3]
        v⃗ = sol[idx][4:6]
    end
    εₖ = 0.5*norm(v⃗)^2
end
```

```
0.29499999999996035
```

```
KE(x₀, μ, T, -1)
```

```
6×6 Matrix{Any}:
 30.8753  39.755   10.5566   17.4601    86.5413  25.1257
 39.755   45.9582  15.0279   24.6       97.8826  32.1899
 10.5566  15.0279  -2.0744    2.36418   49.7998   5.08131
 17.4601  24.6      2.36418   3.07213   68.5336  14.036
 86.5413  97.8826  49.7998   68.5336   170.047   80.2224
 25.1257  32.1899   5.08131  14.036     80.2224  14.8214
```

```julia
begin
    # ∂²x₀ = cntrDiff′((u) -> KE(u, μ, T, 1), x₀, 1e-6)
    ∂²xₜ = cntrDiff′((u) -> KE(u, μ, 5, -1), x₀, 1e-6)
    # hes = ∂²xₜ/∂²x₀
end
```

Part 10

```
320.97984910875914
```

```julia
norm(∂²xₜ)
```