Assignment 7

1.a) $x_{k+1} = x(t_k) + \Delta t \sum_{i=1}^{2} b_i k_i$

$= x(t_k) + \Delta t \left(b_1 k_1 + b_2 k_2 \right)$

$x_{k+1} - x(t_{k+1}) = x(t_k) + \Delta t (b_1 k_1 + b_2 k_2) - (x(t_k) + \Delta t k_1 + \frac{\Delta t^2}{2} \dot{f} + O(\Delta t^3))$

$= \Delta t b_1 k_1 + \Delta t b_2 k_2 - \Delta t k_1 - \frac{\Delta t^2}{2} \dot{f} - O(\Delta t^3)$

taylor expansion gives,

$k_2 = f(x(t_k), u(t_k + c\Delta t)) + a \Delta t \dot{f}(x(t_k), u(t_k + c\Delta t)) + O(\Delta t^2)$

$= f(x(t_k), u(t_k)) + a \Delta t \dot{f}(x(t_k), u(t_k)) + O(\Delta t^2)$

$= k_1 + a \Delta t \dot{f} + O(\Delta t^2)$ where $c \in [0, 1]$

$x_{k+1} - x(t_{k+1}) = e_k = \Delta t b_1 k_1 + \Delta t b_2 (k_1 + a \Delta t \dot{f} + O_1(\Delta t^2)) - \Delta t k_1 - \frac{\Delta t^2}{2} \dot{f} - O_2(\Delta t^3)$

$= \Delta t b_1 k_1 + \Delta t b_2 k_1 + \Delta t^2 b_2 a \dot{f} + \Delta t b_2 O_1(\Delta t^2) - \Delta t k_1 - \frac{\Delta t^2}{2} \dot{f} - O_2(\Delta t^3)$

$\Delta t b_2 O_1(\Delta t^2) - O_2(\Delta t^3) = O(\Delta t^3)$

$= \Delta t (b_1 + b_2 - 1) k_1 + \Delta t^2 (b_2 a - \frac{1}{2}) + O(\Delta t^3)$

$e_k = O(\Delta t^3) \Rightarrow \underline{b_1 + b_2 = 1}, \ \underline{b_2 a = \frac{1}{2}}, \ \underline{c \in [0, 1]}$

1.b) $e_k = O(\Delta t^3)$ implies $\|x_N - x(T)\| = O(\Delta t^2)$, which is the best for ERK2 methods.

if each step $\|x_{k+1} - x(t_{k+1})\|$ has error $O(\Delta t^3)$, and an integration up to time T (with steps Δt) requires $N = \frac{T}{\Delta t}$ total steps, then after N steps the error will be $N\Delta t^3 = T\Delta t^2 = \underline{O(\Delta t^2)}$.

2. a) using $x_{k+1} = x_k + \Delta t \overbrace{\sum_{i=1}^{s} b_i K_i}^{\text{x\_sum}}$

and $K_s = f\left(x_k + \Delta t \underbrace{\sum_{j=1}^{s} a_{sj} K_j}_{\text{k\_sum}},\ u(t_k + c_s \Delta t)\right)$

b)
- increasing Δt causes shittier approximations, with RK1 becoming most shite quickest.
- decreasing Δt yields more accurate approximations.
- RK4 is generally best, followed by RK3, RK2, then RK1.

The orders are as follows

RK1 $\rightarrow$ order 1. $\frac{1}{2}\Delta t \rightarrow \frac{1}{2}$ error

RK2 $\rightarrow$ order 2. $\frac{1}{2}\Delta t \rightarrow \frac{1}{4}$ error

RK3 $\rightarrow$ order 3. $\frac{1}{2}\Delta t \rightarrow \frac{1}{8}$ error

RK4 $\rightarrow$ order 4. $\frac{1}{2}\Delta t \rightarrow \frac{1}{16}$ error.

2.c) simulating 20 sec. with $\Delta t = 0.4$ reveals that

- RK1 and RK2 become unstable when $\lambda < -5$
- RK3 becomes unstable when $\lambda < -6.282$
- RK4 becomes unstable when $\lambda < -6.963$

Generally, $\dot{x} = \lambda x$, $x(0) = x_0$ is stable $\forall \lambda$ s.t.

$$\sum_{k=0}^{N} \frac{(\lambda \Delta t)^k}{k!} \leq 1,$$ where N is the order of the RK.

π 73

3.a)

$$\dot{x} = y$$
$$\dot{y} = \upsilon(1-x^2)y - x, \quad \upsilon = 5$$

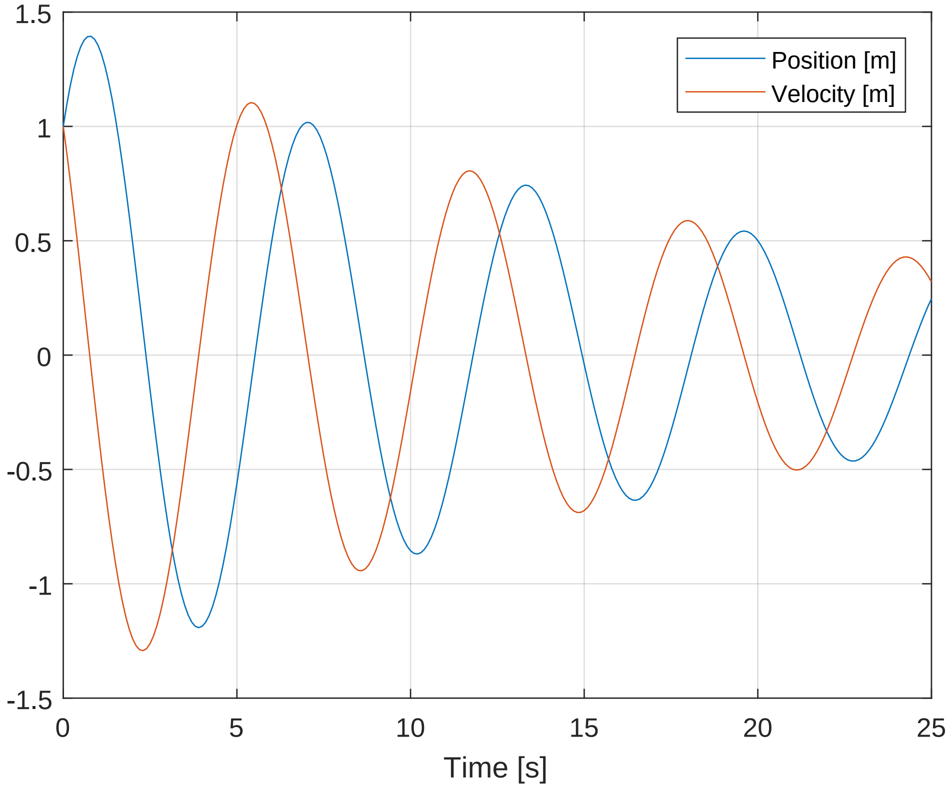
$x(0) = 2$

$y(0) = 0$

plotting using ODE45 yields graphs for x and y. they're both marginally stable.

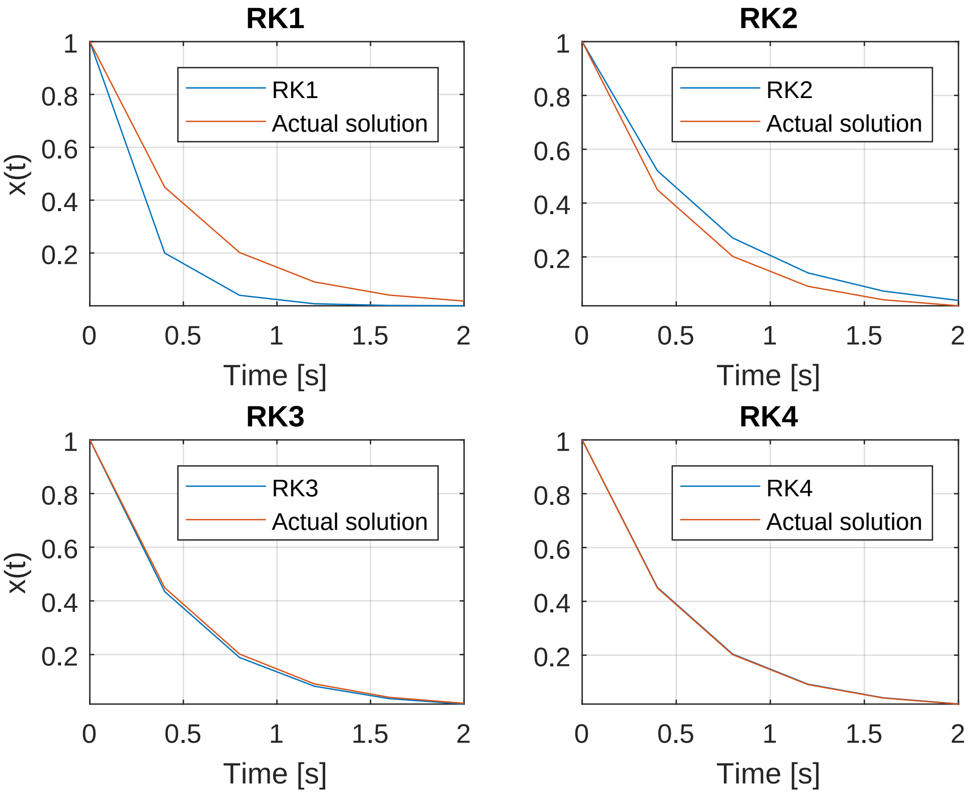
b) I used the RK4 from exc 2 and observed that the system was stable until around $\Delta t = 0.162$. Increasing Δt beyond this point caused the RK4 approximation to go nuts. Lower values of Δt worked just fine. $\Delta t < 1.1$ was ideal, as $\Delta t > 1.1$ caused RK4 to "lag" behind ODE45 after some seconds.

Mass damper, RK3



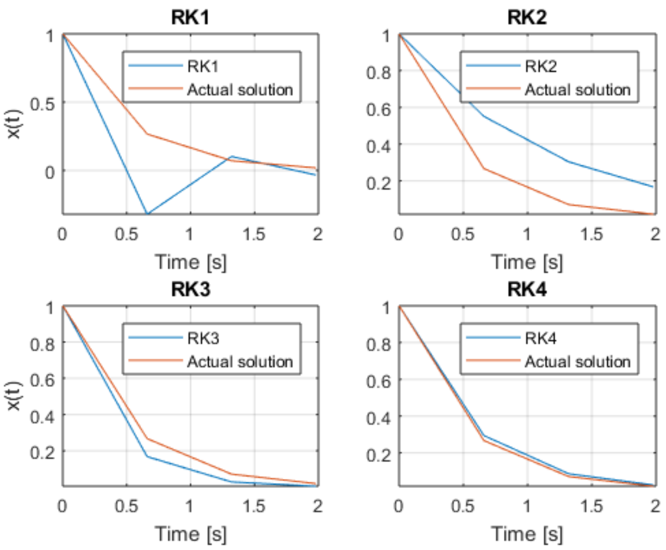
2.a)

the higher the order,
the better the approximation.

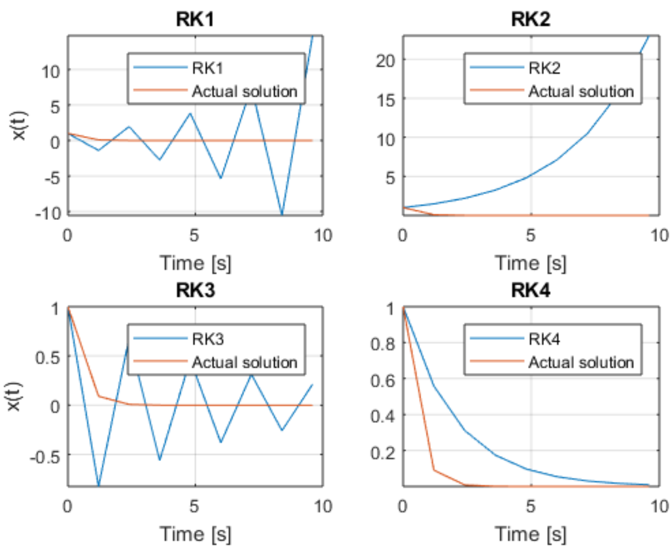


2.b)

dt = 0.66

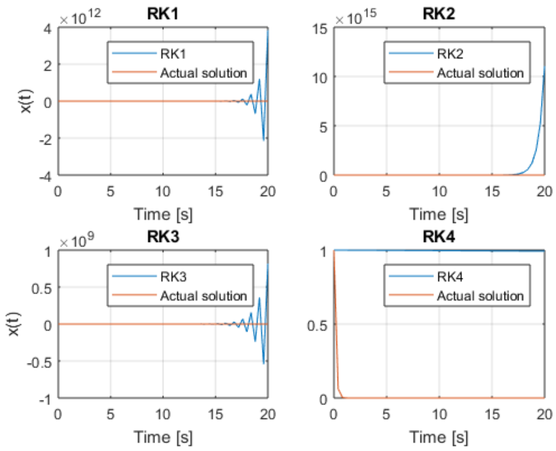


dt = 1.2



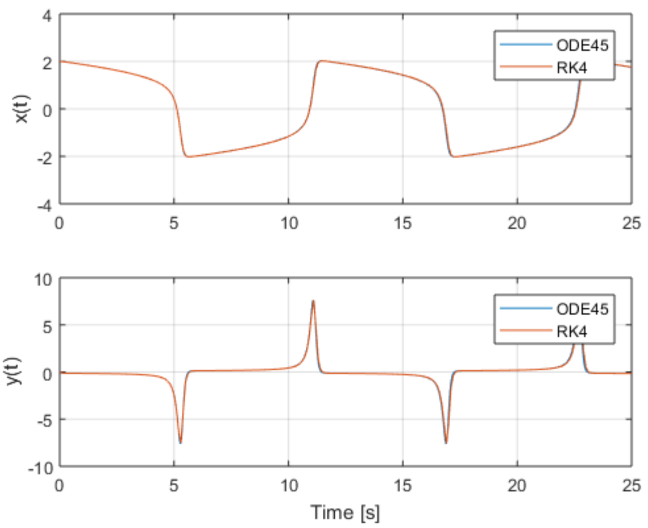
2.c)

lambda = -6.963



3.b)

dt = 0.1



dt = 1.163