In order to find the values of b_1, b_2 and c we need to expand the functions for x_{k+1} (the estimated value using RK2) and $x(t_{k+1})$ (the real trajectory value) and compare for which they are equal.

$$x(t_{k+1}) = x(t_k) + \underline{1} \cdot \Delta t f(x(t_k), u_k) + \underline{\tfrac{1}{2}} \Delta t^2 \dot{f}(x(t_k), uk) + \mathcal{O}(\Delta t^3) \qquad where\ \dot{f} = \frac{\partial f}{\partial x} f \tag{1}$$

$$x(t_{k+1} = x(t_k) + \Delta t f(x(t_k), u_k) + \frac{\Delta t^2}{2} (\frac{\partial f}{\partial x} \cdot f(x(t_k), u_k)) \tag{2}$$

$$x_{k+1} = \Delta t \cdot b_1 \cdot f(x(t_k), u_k) + \Delta t \cdot b_2 \cdot \underbrace{f(x(t_k) + a\Delta t \cdot K_1, u_k)}_{taylor\ expansion\ using\ 7.22} + \mathcal{O}(\Delta t^3) \tag{3}$$

$$x_{k+1} = x(t_k) + \underline{(b_1 + b_2)} \Delta t \cdot f(x(t_k), u_k) + \underline{a \cdot b_2} \Delta t^2 \cdot \frac{\partial f}{\partial x} f(x(t_k), u_k) + \mathcal{O}(\Delta t^3) \tag{4}$$

To find the RK2 estimate and its constant we had to first find the derivative of the given polynomial $x(t)$ which equals the f in the RK2. And then derive the K1, K2 and lastly x_{k+1} as below. The RK2 cannot solve polynomial of order higher than 2 because it's not constructed to.

$$n = 1 \begin{cases} x(t) = x(t_k) + \underline{1} \cdot \alpha_1 (t - t_k) \\ f(t) = \dot{x}(t) = \alpha_1 \end{cases} \qquad \begin{aligned} K_1 &= f(t_k) = \alpha_1 \\ K_2 &= f(t_k + c\Delta t) = \alpha_1 \\ x_{k+1} &= x(t_k) + \Delta t \alpha_1 \underline{(b_1 + b_2)} \end{aligned}$$

$$n = 2 \begin{cases} x(t) = x(t_k) + \alpha_1 (t - t_k) + \underline{\tfrac{1}{2}} \alpha_2 (t - t_k)^2 \\ f(t) = \dot{x}(t) = \alpha_1 + \alpha_2 (t - t_k)^2 \end{cases} \qquad \begin{aligned} K_1 &= f(t_k) = \alpha_1 \\ K_2 &= f(t_k + c\Delta t) = \alpha_1 + \alpha_2 c \Delta t \\ x_{k+1} &= x(t_k) + \Delta t \alpha_1 (b_1 + b_2) + \underline{b_2 c} \cdot \alpha_2 \Delta t^2 \end{aligned}$$

From shown above the RK2 has $\mathcal{O}(\Delta t^3)$ which relates to the one-step error between the estimated and the real function.

By using the formulas and equations from the lecture notes book, page 166 we could construct RK1, RK2 and RK4 just by using more or less K's. The different methods were plotted using the test function for visual inspection. To plot the error relative to the true solution we used the example as the hint suggested by taking the norm of the difference between the true solutions last value and our solutions last value, in that way we get the global error. We did this for different Δt's and for all methods and then the solution were ploted.

To check for which λ that the solution still will be stable we used the formula defined in the lecture notes page 170, equation (7.41). We used the formula to calcualte the sum "S" and then solved for which λ where the solution is equal to 1. The solution was: $RK_1 = -20$, $RK_2 = -20$ and $RK_4 = -27.8529$. The solutions for the corresponding λ where also ploted.

The ode45 command in Matlab was used to find the solutions to the described system. The solution to the described system using ode45 can later be observed by plotting.

By finding the solution to the ode45 with the new tolerances and then plotting against our RK4 solution we observed that when using the original 0.1 as δt there was a slight visible difference. When trying $\delta t = 0.01$ we couldn't observe any difference.

To define our IRK function we used the formulas and equations from the lecture note, page 180-181. Where the definition for both r and the algorithm is defined.

To find the lambda for which the solution is stable for the described system we used equation (8.30) page 185 in the lecture notes. We ploted the different sum values for the different λ and could observe that the solution never becomes unstable no matter the λ.

When deploying our solutions to both IRK4 and RK4 on the vdp dynamics and plotting we can observe that there is very little to no difference. We have to zoom in a lot to see that they actually differ a little.