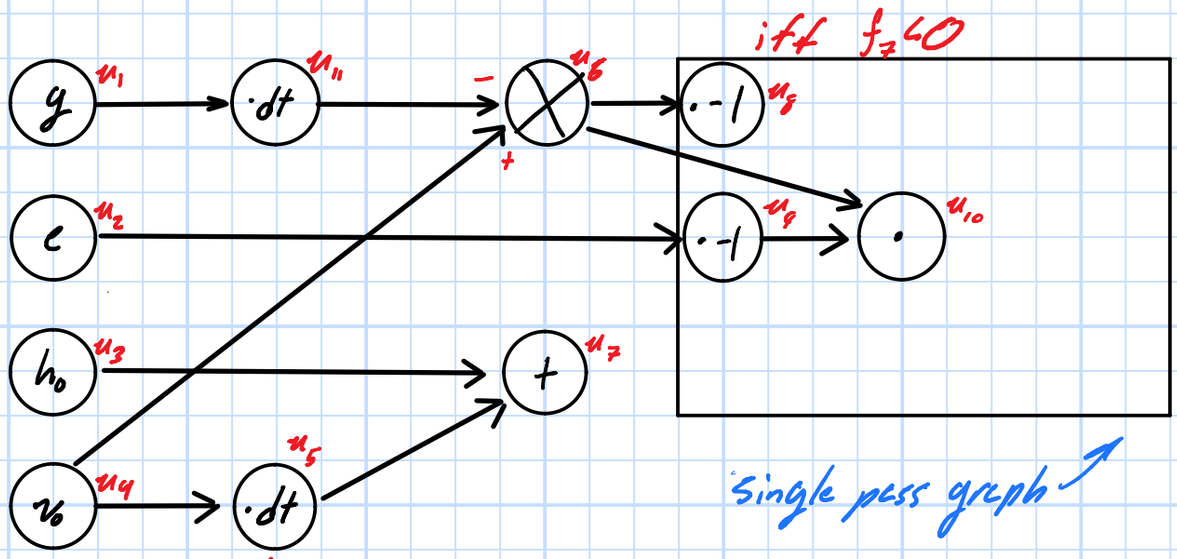
4.4)



multiply by non given constant

For ICs (-10, .9, 0, 5) w/ dt = .5

↳ Step through iterations to find path:

(1): (-10, .9, -5, 5)

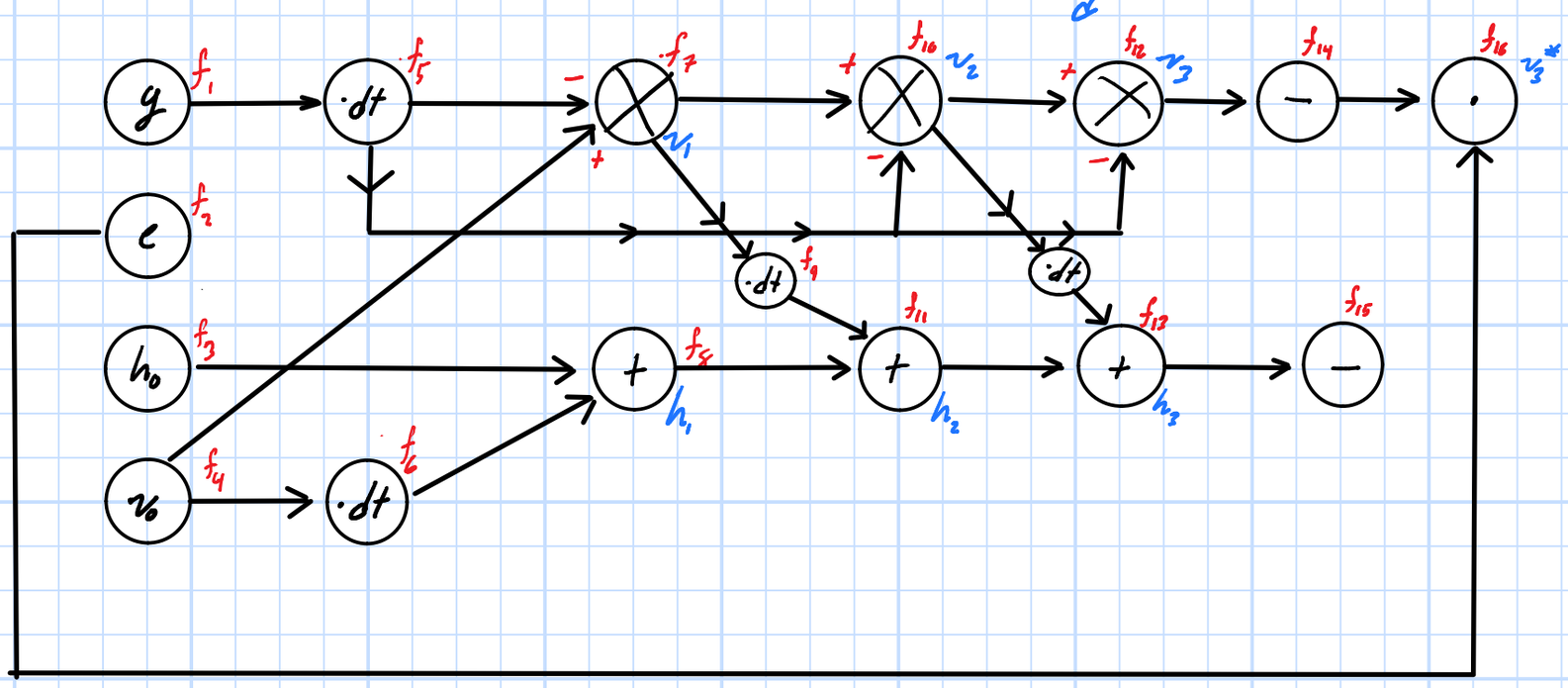
(2): (-10, .9, -10, 2.5) if statement

(3): (-10, .9, -15, -2.5) ⇐ triggered

(3*): (-10, .9, 13.5, 2.5)

From the above graph this looks like

full path for our ICs



$f_1 = g$

$f_2 = e$

$f_3 = h_0$

$f_4 = v_0$

$f_5 = f_1\,dt$

$f_6 = f_4\,dt$

$f_7 = f_4 - f_5$

$f_8 = f_6 + f_3$

$f_9 = f_7\,dt$

$f_{10} = f_7 - f_5$

$f_{11} = f_8 + f_9$

$f_{12} = f_{10} + f_5$

$f_{13} = f_{10} + f_{11}$

$f_{14} = -f_{12}$

$f_{15} = -f_{13}$

$f_{16} = f_{14} f_2$

f_{16} is output node.

$df_{16}/df_2 = f_{14}$, $df_{16}/df_{14} = f_2$

$df_{14}/df_{12} = -1$

$df_{12}/df_5 = df_{12}/df_{10} = 1$

$df_5/df_1 = dt$

$df_{10}/df_5 = -1$, $df_{10}/df_7 = 1$

$df_7/df_4 = 1$, $df_7/df_5 = -1$

the third input, f_3 does not map to our output $\therefore dv_f/dh_0 = 0$

$$df_{16}/df_1 = (df_{16}/df_{14})(df_{14}/df_{12})(df_{12}/df_5 + df_{12}/df_{10}(df_{10}/df_5 + df_{10}/df_7(df_7/df_5)))(df_5/df_1)$$

$$= (e)(-1)(1 + (-1 + (-1))|(-1)| = -e = dv_f/dg$$

$$df_{16}/df_2 = f_{14} = -f_{12} = f_{10} + f_5 = f_7 + f_5 - f_5 = f_4 - f_5 = v_0 - g\,dt = dv_f/de$$

$$df_{16}/df_4 = (df_{16}/df_{14})(df_{14}/df_{12})(df_{12}/df_{10})(df_{10}/df_7)(df_7/df_4)$$

$$= (e)(-1)(1)(1)(1) = -e = dv_f/dv_0$$

$$\begin{bmatrix} dv_f/dg \\ dv_f/de \\ dv_f/dv_0 \\ dv_f/dh_0 \end{bmatrix}_{g=10,\ e=.9,\ v_0=0,\ h_0=5} = \begin{bmatrix} -.9 \\ -5 \\ -.9 \\ 0 \end{bmatrix}_{\text{this work}} = \begin{bmatrix} 1.35 \\ 15 \\ -.9 \\ 0 \end{bmatrix}_{\text{Autograd}} \begin{matrix} \text{wrong} \\ \text{wrong} \\ \text{right} \\ \text{right} \end{matrix}$$

my guesses as to why this attempt got the first two wrong is that I didn't properly account for every contribution or that Autograd handles the loop differently - if you simulate this while varying g the result you'll get by trying to numerically find this derivative will depend highly on your step size

tl;dr I think autograd is numerically differentiating (the backwards pass) here but our timestep is so large that autograds attempt to do the numerical derivatives for the backward pass has numerical stability issues