Simple comparison of pulse integration options

This derives from working with point loads acting on a beam. As all point loads are distributed to some extent it was convenient to treat all point loads as distributed i.e. much like gravity with units of N/m. A single pulse train function, based on Heaviside operators, was used to gather all loads into a single $Pulse\_Train$ expression. To help the integration routine, cosine pulses with the same area as the point loads were used.

- Fp_i are the point loads acting on the beam

$$Fp := \begin{bmatrix} 1000 \\ 500 \\ 200 \\ 400 \\ 657 \end{bmatrix} \cdot N$$

- p is the number of point loads acting on the beam

$$p := \mathrm{rows}(Fp) = 5$$

- Δl_i is width of the cosine pulse -i.e. the extent of load distribution

$$\Delta l := \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \cdot cm$$

Play with width Δl

- xp_i are the locations along the beam where each loads is applied

$$xp := \begin{bmatrix} 0.25 \\ 0.4 \\ 0.6 \\ 0.7 \\ 0.86 \end{bmatrix} \cdot m$$

- x is the distance along the beam (independent variable)

$$x := 0 \cdot m, 10^{-3} \cdot m .. 1 \cdot m$$

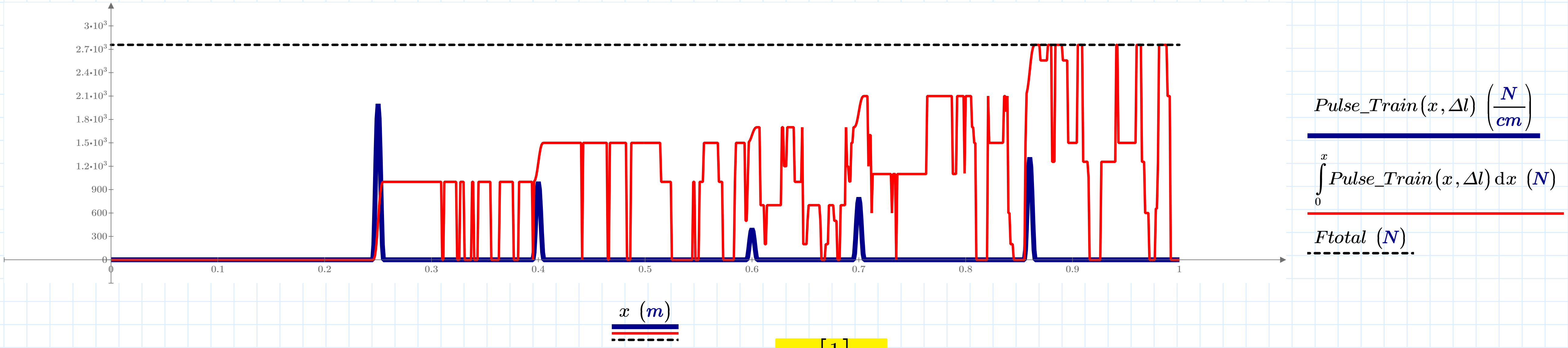
- Total load

$$Ftotal := \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}^{\mathrm{T}} \cdot Fp = \left(2.757 \cdot 10^{3}\right) N$$

- Cosine Pulse Train Function

$$Pulse\_Train(x, \Delta l) := \left(\sum_{i=0}^{p-1} \frac{Fp_i}{\Delta l_i} \cdot \left(\Phi\left(x - xp_i + \frac{\Delta l_i}{2}\right) - \Phi\left(x - xp_i - \frac{\Delta l_i}{2}\right) \right) \cdot \left(1 + \cos\left(\frac{\left(x - xp_i\right) \cdot 2 \cdot \pi}{\Delta l_i} \right) \right) \right)$$

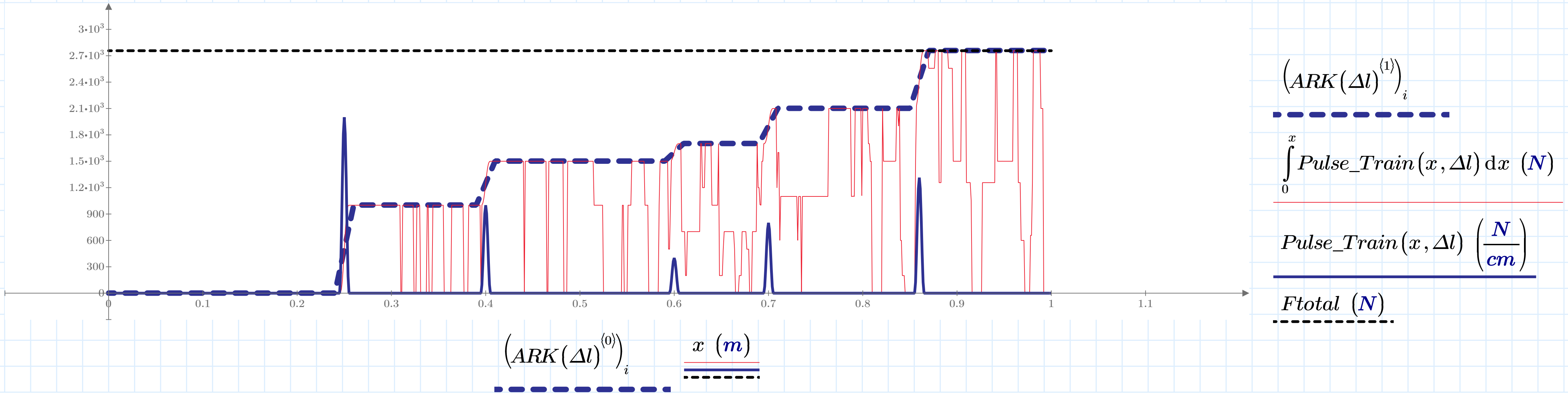
Using the $\int \blacksquare \, \mathrm{d}x$ operator - things fall apart with $\Delta l = \begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix} \, cm$



Using the adaptive Runge-Kutta - things work OK with $\Delta l = \begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix} \, cm$

$intvls := 100$ $\quad$ $i := 0, 1 .. intvls$

$$ARK(\Delta l) := \left\| \begin{array}{l} D(x, Force) \leftarrow Pulse\_Train(x, \Delta l) \\ output \leftarrow \mathrm{Rkadapt}(1, 0, 1, intvls, D) \end{array} \right.$$



But eventually - the adaptive RK starts missing pulse areas when the Δl are too narrow

$$\Delta l := \begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}\cdot mm$$