**Numerical Techniques**    PSLTDSim utilizes a variety of numerical techniques to perform integration involved with dynamic agents. Some of the employed methods are coded 'by hand', while others utilize Python packages. This appendix is meant to provide more detail on available integration methods for system frequency, cover how Laplace style block diagrams are computed by PSLTDSim, and explain how agents currently handle the integration of running values.

**Combined Swing Equation Solutions**    The options included in PSLTDSim to solve the combined swing equation for a new system frequency are Euler, Adams-Bashforth, and Runge-Kutta. Each of these methods are numerical approximations that provide an accurate approximation to the solution of an initial value problem XXXrefBoyceDiprima. Each method is briefly introduced and the actual code used to solve the combined swing equation is explained.

**Euler Method**    Of the of integration methods available to solve system frequency, the Euler method is the simplest. Using generic math variables, to find the next $y$ value given some function $f(t, y)$ is

$$y_{n+1} = y_n + f(t_n, y_n)t_s, \tag{1}$$

where $t_s$ is desired time step. The next value of $y$ is simply a projection along a line tangent to the derivative at time $t$. The accuracy of this approximation method is related to the time step size.

**RK45 via solve_ivp**    For a more robust integration, a runge kutta method may be employed via the scipy solve_ivp function.

$$
\begin{aligned}
k1 &= f(t_n, \ y_n) \\
k2 &= f(t_n + t_s/2, \ y_n + t_s/2 * k1) \\
k3 &= f(t_n + t_s/2, \ y_n + t_s/2 * k2) \\
k4 &= f(t_n + t_s, \ y_n + t_s * k3) \\
y_{n+1} &= y_n + t_s/6 * (k1 + 2 * k2 + 2 * k3 + k4)
\end{aligned}
\tag{2}
$$

**Adams Bashforth**    An adams bashforth method is a multi-step method that uses more than just the single previous data point (as the Euler and RK method did)

$$y_{n+1} = y_n + 1.5 * f(t_n, y_n)t_s - 0.5 * f(t_{n-1}, y_{n-1})t_s \tag{3}$$

**sig.lsim**    To process a signal through a Laplace block, the block is first transformed into state space matrices and then inserted into sig.lsim for a linear simulation. This is used in all dynamic processes of PSLTDSim including governor states, filter and delay blocks.

**Trapezoidal Integration**    To integrate known values generated each time step, a trapezoidal integration method is used. Given some value $x(t)$ and an integration accumulator variable $xInt$, the trapezoidal method estimates that

$$xInt = xInt + (x(t) + x(t - ts))/2 * ts, \tag{4}$$

where $ts$ is the time step used between calculated values of $x$. This technique is used in a variety of agents, such as the BA agent for IACE, and the window integrator agent.