Some additional information:

1. Some people might think that Numer.ai already utilizes $\dot{X}$

values buried in X

because a lot of financial data is not stationary. In order to transform the data into the obfuscated range zero to one, the data must first be made stationary. As is discussed in Marcos Lopez de Prado's book, some form of derivative is taken. MLDP suggests the fractional derivative. The fractional derivative computation involves finite differences of the data to various orders. But whatever form of derivative is used to bring some parts of the data to stationarity, this is not what we are looking for. We identify any value first brought to stationarity as X

and the first difference to be $\dot{X}$

. Doing so allows much more powerful theoretical treatment of the data. Even if Numer.ai has already computed actual $\dot{X}$

values as I have here defined, they cannot be used in a Lagrangian treatment without being explicitly identified as first order derivatives.

1. There is a typo after the arrow in the Euler-Lagrange equation above.

2. It would take less than a line of code to compute the $\dot{X}$

s, if they have not yet already be computed in the data stream.

1. One could also consider using $M(\dot{X}, X)$

or just $M(X)$

in the Lagrangian. Up to now I have attempted to use the latter form but that formulation is lacking from a dynamical standpoint. To do this right we really need to know which values in the data are the $\dot{X}$

if they are there or they need to be computed if they are not.

1. Besides the normal process of obfuscation the other reason that it would be very difficult to reconstitute the time series of the rows is that it is not necessary to identify which columns of the $\dot{X}$

s belong to which columns of the X

s but only that the columns of data be identified as X

s and $\dot{X}$

s.