A new language for complex ODE models

Michael R. Dunlavey, Shuhua Hu, Robert H. Leary

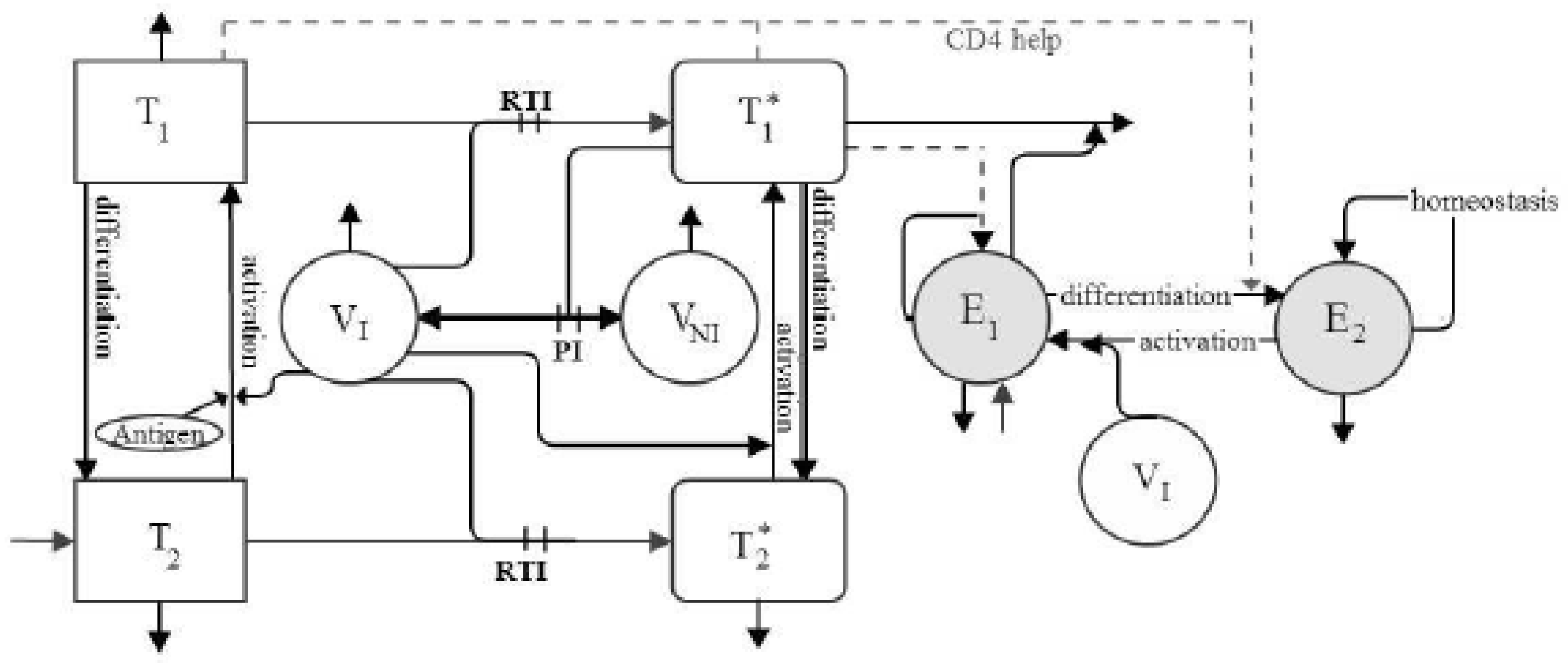
Certara/Pharsight® Corporation

Objective

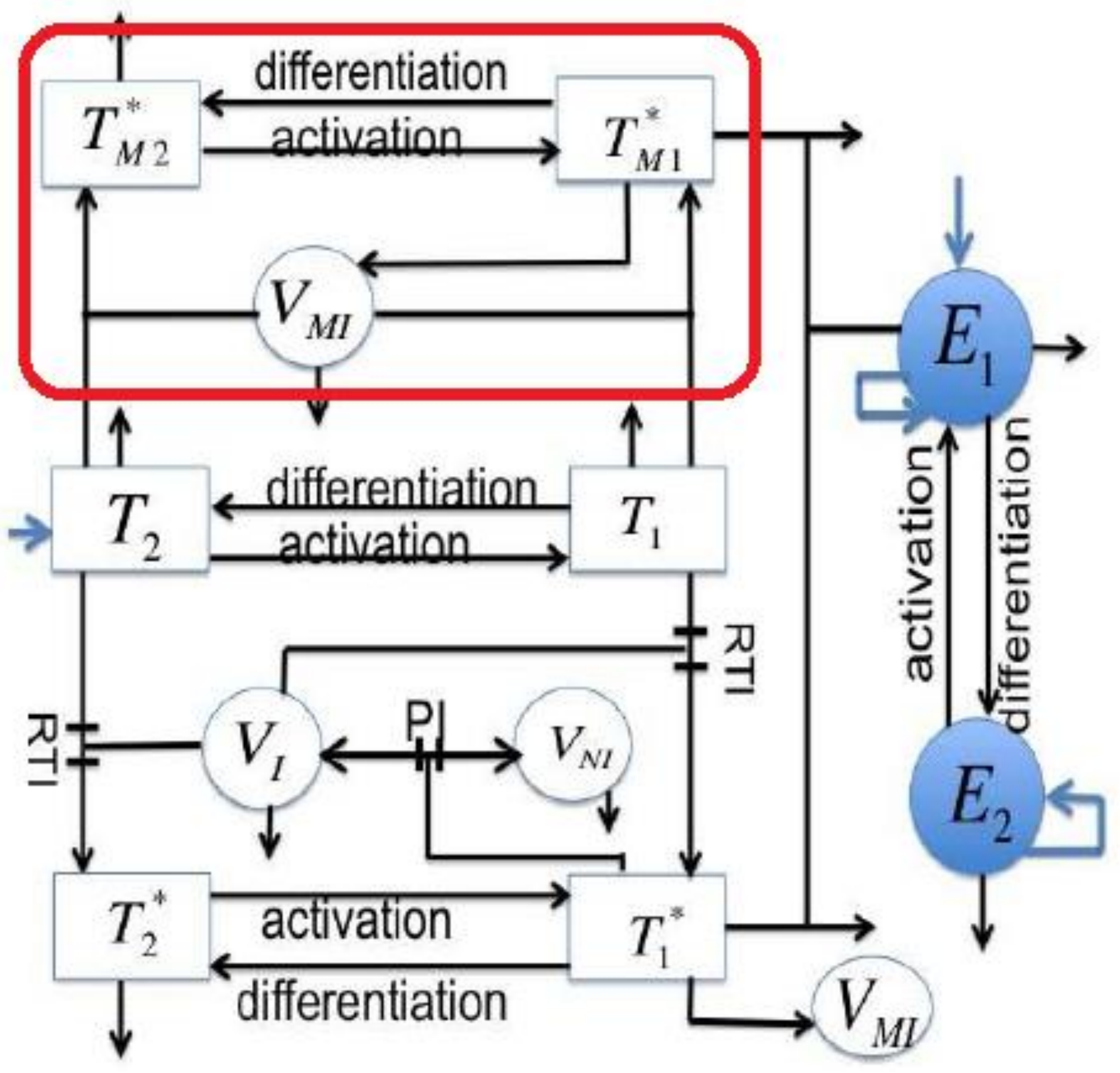
Complex models such as HIV/HCV/HBV, and possibly some diabetes models as in the DDMoRe model repository[1], when represented as ordinary differential equations (ODE), dramatically increase in length and width as changes are made, such as additional mutations or treatments. The differential equations have many repeated terms that are the same except for a sign, or almost the same as other terms. The functional purpose of each term is often hard to discern. This weighs against the verifiability and modifiability of such models. The objective is to find a surface representation (the language that is seen and edited) that reduces these problems.

The problem

Below is a diagram of an HIV model [2] in which virus V is either noninfectious or infectious but responsive to antiretroviral treatment. Treatments are Reverse Transcriptase Inhibitor (RTI) and Protease Inhibitor (PI). T (denoting CD4+) cells are either infected or not, and are either resting (memory or naïve) or active. E (denoting HIV-specific CD8+) cells are either resting or not. The ODE model requires eight differential equations with many terms and parameters.



Below is the same model [3] with an additional mutation, so V is either noninfectious, infectious but responsive to antiretroviral therapy (denoted mutation A), or infectious with a mutation making it resistant to antiretroviral therapy (denoted mutation B). This requires adding a virus V compartment and two CD4+ T cell compartments. That adds three differential equations and additional terms and parameters.



One can see that as more mutations or T cell conditions are added, the complexity of the diagram and the differential equations becomes unmanageable.

The language

Dimensions:

```
  dimension activity( active, resting )
  dimension infection( noninf, infA )
```

These statements say that there is a dimension called “activity” with two values – active and resting. There is also a dimension called “infection” with two values – noninfectious, and infectious A (denoted noninf and infA). To add the mutation, an additional infection condition is added, denoted infB:

```
  dimension infection( noninf, infA, infB )
```

States:

```
  state T(activity, infection)
    = {108, 59.8, 60, 431, .294, .3}
  state V(infection) = {3570, 69300, 60000}
  state E(activity) = {.0682, .691}
```

These statements define T (six CD4+ compartments), V (two virus compartments, and E (two CD8+ compartments). The numbers are their initial values. The underlining indicates code added for the mutation.

Parameters:
There are numerous parameters. The following declares `d` which will be the decay rate for active and resting CD4+ T cells:

```
  parm d(activity) = {.0912, .0031}
```

Flows:
Rather than have a single differential equation for each compartment, the model is divided into functional sections. For example:

```
  parm delE(activity) = {.0597, .00145}
  decayrate E(i), delE(i)
```

says that HIV-specific CD8+ (denoted E) cells (both resting and active) decay at rates given by parameter `delE`. Variables `i` and `j` act as universal quantifiers over all values of the corresponding dimension. A more complex example is:

```
  decayrate T(i, j)
    , (i==active && j!=noninf ? 0 : d(i))
```

meaning that all CD4+ T cells decay at rate `d(i)` *except for active infected cells*.

A more complex statement does 2nd-order binding:

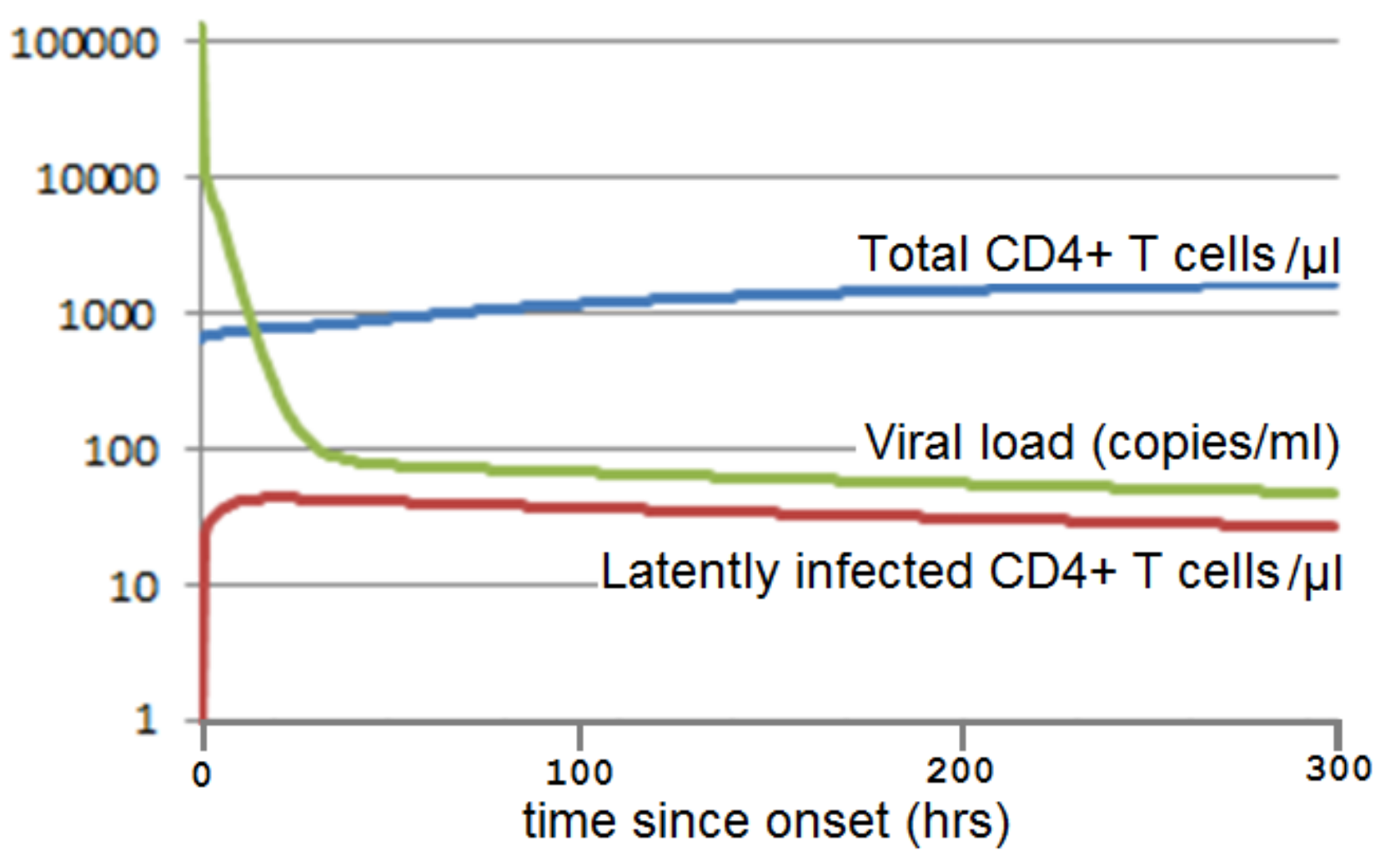
```
  bind T(i, noninf), V(infB), T(i, infB)
    , (1 - xi1*(i==resting ? f : 1))*k(i)
    , 1, rho(i)*1000, 1
```

representing the infection process for mutated virus. It says noninfected CD4+ T cells (both resting and active) bind with mutated virus to form CD4+ T cells with the B infection. The second line gives the rate, which is multiplied by the first two compartments. (`xi1` is efficacy of RTI.) That term replaces six terms in the ODE representation, three for resting CD4+ T cells, and three for active. The third line gives ad-hoc multipliers to be applied to each term.

To perform summation over sets of states, the `*` operator may be used. For example, `V(*)` is viral load, the sum of all V compartments.

Results

To modify the model, 105 lines of code were unchanged, 6 lines were modified, and 8 lines were added. This should assist in verifiability. The code is translated into C/C++ and executes. The following simulation begins with onset of viremia and treatment:



Conclusions

A language is demonstrated that suggests a surface representation for complex ODE models that minimizes redundant terms, leading to reduced effort to modify and verify, relative to differential equations.

References

[1] DDMoRe Model Repository,
http://repository.ddmore.eu/models

[2] Banks, H.T., Hu, S., Thompson, W.C. Modeling and inverse problems in the presence of uncertainty, CRC Press, 2014

[3] Hu, S., personal communication

Contact

Michael R. Dunlavey, PhD, Certara/Pharsight Corporation,
Ph: +1 781.449.2719, Email: Mike.Dunlavey@Certara.com