SBML DSMTS — User Guide

T. W. Evans, C. S. Gillespie, D. J. Wilkinson*

October 21, 2010

1 Introduction

This guide describes how the SBML discrete stochastic models test suite can be used to test a stochastic simulator. Testing a stochastic simulator is hard, since two identical simulators will never produce exactly the same set of results. The only sensible method is to run the simulation lots of times and check that the distribution of outcomes is correct. This can only be tested in a probabilistic way. The latest version of this test suite can be obtained from:

<div align="center">

http://dsmts.googlecode.com/

</div>

To cite the test suite:
Evans, T. W., Gillespie, C. S., Wilkinson, D. J. (2008) The SBML discrete stochastic models test suite, *Bioinformatics*, **24**:285-286.

1.1 Aims and Scope of the Test Suite

The test suite is a set of SBML models each with time course data for the means and standard deviations of the model species. Developers may use the suite to check that their simulators produce results that are consistent with the SBML standard. The test suite assumes that the simulator produces output on a regular time grid. Of course exact stochastic simulators naturally produce output on a non-regular grid corresponding to individual reaction events. However, this "step function" output is easy to map onto a regular time-grid either post-hoc, or during the simulation run itself. Further discussion of this issue can be found in Chapter 6 of Wilkinson (2006)[1], to which the reader is referred for further discussion of stochastic simulation generally, and the use of SBML for discrete stochastic models in particular.

2 Structure of the Test Suite

2.1 Contents of the Test Suite

The test directory contains the following files for each test model:

- `<model>.xml`: the SBML Level 3 test model

- `<model>-mean.csv`: means of the simulation results at $t = 0, 1, \ldots, 50$

- `<model>-sd.csv`: standard deviations of the simulation results at $t = 0, 1, \ldots, 50$

- `<model>.mod`: documentation on the model in the form of SBML-shorthand

- `<model>-mean.pdf`: plot of mean species levels against time

---

*d.j.wilkinson@ncl.ac.uk
[1]See http://www.staff.ncl.ac.uk/d.j.wilkinson/smfsb/

- `<model>-sd.pdf`: plot of the standard deviations of the species levels against time.

The first lines of both the `<model>-mean.csv` and the `<model>-sd.csv` files contain the column headings. The first column heading is *Time*, and the remaining column headings are the species names in the order they appear in the `listOfSpecies` in the `<model>.xml` file. The file `<model>.mod` contains the SBML-shorthand for the model (`http://www.staff.ncl.ac.uk/d.j.wilkinson/software/sbml-sh/`), which is supposed to provide human-readable documentation for each model. The test suite also contains this documentation, and a file (`model-list`) containing a simple list of all models in the test suite (this is useful for batch-testing against all models in the test suite).

3 Testing an Exact Simulator

In order to test the output from a stochastic simulator for a given SBML model, n independent simulation runs of the simulator should be performed. The value of n should not be less than 1,000. However, for the statistical tests to have reasonable power to detect problems, n should be set to at least 10,000 (but this will be very time-consuming for some models) — see the discussion in Section 3.4. The sample means and standard deviations of the species amounts from the simulation runs at $t = 0, 1, \ldots, 50$ can be compared with the corresponding values in the test suite using the statistical tests described below.

3.1 Testing the simulated mean values

A test for the simulated values of a species X is as follows: Let $X_t^{(i)}$ be the value of X_t on the ith run of the simulator, where X_t is the random variable representing X at time t. Put $\mu_t = \mathrm{E}(X_t)$ and $\sigma_t = \sqrt{\mathrm{Var}(X_t)}$. By the Central Limit Theorem, we have

$$\bar{X}_t \sim N(\mu_t, \sigma_t^2/n), \quad \text{where } \bar{X}_t = \frac{1}{n}\sum_{i=1}^{n} X_t^{(i)}.$$

Therefore

$$\bar{X}_t - \mu_t \sim N(0, \sigma_t^2/n),$$

and it follows that

$$Z_t \equiv \sqrt{n}\left(\frac{\bar{X}_t - \mu_t}{\sigma_t}\right) \sim N(0,1).$$

So under the null hypothesis that the simulator is correct, the Z_t values should have a standard normal distribution. In this case, most values will lie in the range (-3,3). Therefore, values of Z_t outside this range correspond to evidence that the simulator is in error.

3.2 Testing the simulated standard deviation values

For this test we need to assume the approximation $X_t \sim N(\mu_t, \sigma_t^2)$. From this approximation, we have:

$$\frac{X_t - \mu_t}{\sigma_t} \sim N(0,1) \Rightarrow \frac{1}{\sigma_t^2}(X_t - \mu_t)^2 \sim \chi_1^2,$$

and so

$$\frac{1}{\sigma_t^2}\sum_{i=1}^{n}(X_t^{(i)} - \mu_t)^2 \sim \chi_n^2.$$

Now put

$$\hat{S}_t^2 \equiv \frac{1}{n}\sum_{i=1}^{n}(X_t^{(i)} - \mu_t^{(i)})^2$$

so that
$$\frac{n\hat{S}_t^2}{\sigma_t^2} \sim \chi_n^2.$$

This means that $n\hat{S}_t^2/\sigma_t^2$ will have an approximate $N(n, 2n)$ distribution. Therefore $\hat{S}_t^2/\sigma_t^2$ will have an approximate $N(1, 2/n)$ distribution. So

$$Y_t \equiv \sqrt{\frac{n}{2}} \left(\frac{S_t^2}{\sigma_t^2} - 1 \right) \sim N(0, 1).$$

Again, values of Y_t outside the expected range for a standard normal variable correspond to evidence that the simulator is in error. However, owing to the approximation used in the development of this test, it is probably best to consider only values outside the range $(-5, 5)$ as evidence of a problem.

3.3 A note on the tests

The tests are designed to test simulator accuracy. It is important to remember that the tests are probabilistic. If lots of tests are carried out, some failures would be expected to occur by chance, even if the simulator is behaving correctly. Also, since the standard deviations test is based on an approximation, the predicted failure probability for this test is an underestimate. Therefore we might expect to fail more standard deviation tests than mean tests. Running the entire test suite with $n = 10,000$, it wouldn't be particularly surprising to fail 2 or 3 mean tests and 5 or 6 standard deviation tests in total (summed over all models and all time points), even if the simulator is correct.

3.4 Choice of the sample size, n

The power to detect problems with simulator implementation is highly dependent of the choice of the sample size, n, that is used. A value of $n = 1,000$ is the absolute minimum value of n that should ever be used, and is intended to be used in conjunction with unit tests that are run every time the code changes just ensure you haven't completely stuffed everything up by making a change to your simulator code. $n = 10,000$ is needed to have any chance of picking up minor issues, and $n = 100,000$, or more likely $n = 1,000,000$ is needed to detect any subtle implementation errors. Obviously you aren't going to want to use such large values of n for the entire suite every time you run unit tests, but they can be used for particular models when you have been working on particular implementation details. Unfortunately Monte Carlo testing is not very CPU-efficient, but that's life.

4 Testing an Approximate Simulator

The test suite is designed first for rigorous testing of exact simulators. However, it should also prove useful to developers of approximate simulators. One way of using the test suite to assess the performance of an approximate simulator is to plot the means and standard deviations as percentages of their true values. For the mean values, a plot of $\bar{X}_t/\mu_t$ against t should be produced for $t = 0, 1, \ldots, 50$ (for $\mu_t \neq 0$). Similarly, a plot of $\hat{S}_t/\sigma_t$ against t should be produced to assess the standard deviation values ($\sigma_t \neq 0$). Ideally, the plots will be close to 1 for all values of t, and values outside the range $[0.98, 1.02]$ (say) is indicative of poor simulator accuracy. For $\mu_t = 0$ or $\sigma_t = 0$, a simple check should be made to ensure that the corresponding sample estimates are (very) close to zero.

5 Models

This section lists each model in the test suite, gives a brief description of it, and provides some indication of what the model is designed to test. It also includes plots of the means and standard deviations for each model.

5.1 dsmts–001 — The birth-death process

5.1.1 dsmts–001–01

This model contains one species, denoted by X. The amount of X present in the system is measured in numbers of molecules. The initial number of molecules of X is 100. The birth reaction

$$X \longrightarrow 2X$$

has rate parameter `Lambda=0.1`, and the death reaction

$$X \longrightarrow \emptyset$$

has rate parameter `Mu=0.11`. These rate parameters are global, which means that they apply to the whole model and not just to a single reaction. Mass-action stochastic kinetics is assumed. The SBML-shorthand for the model is stored in the file `dsmts-001-01.mod`, and this is intended to be human-readable and self-documenting. The SBML file itself is stored in `dsmts-001-01.xml`, and this is generated automatically from the corresponding SBML-shorthand. The mean and standard deviation of the associated Markov process are given in the plots below, and are stored in the files `dsmts-001-01-mean.csv` and `dsmts-001-01-sd.csv` respectively.
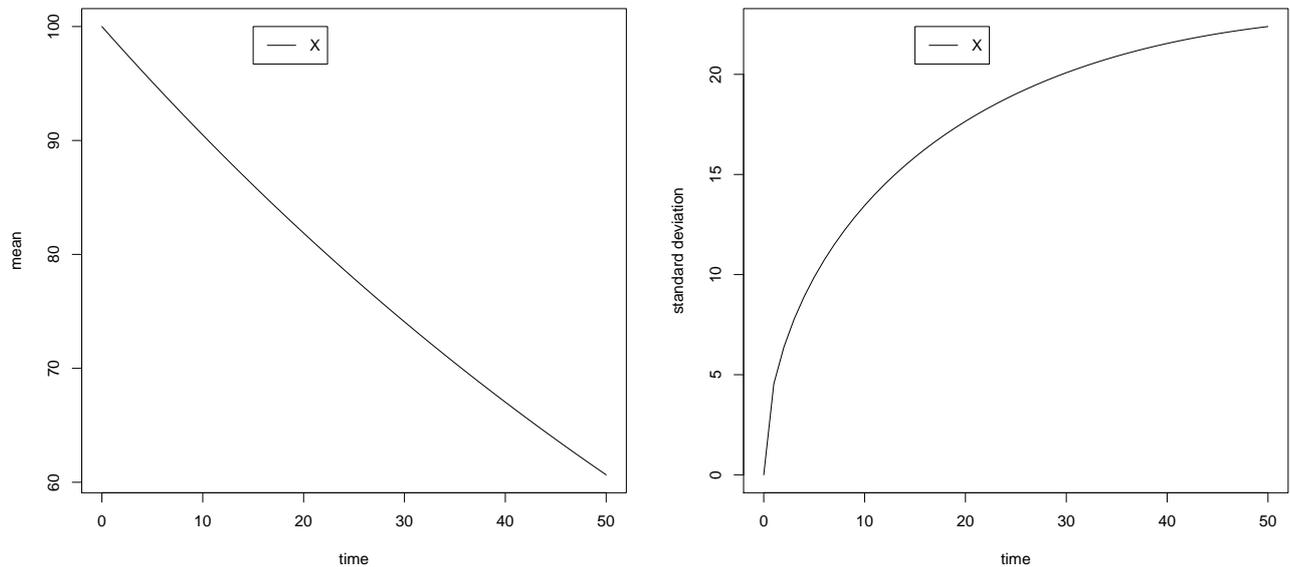
5.1.2 dsmts–001–02

This model is the same as `dsmts-001-01`, except that the rate parameters `Lambda` and `Mu` are declared to be local rather than global. Therefore the rate parameters apply only to their respective reactions, but this does not affect the model output here.
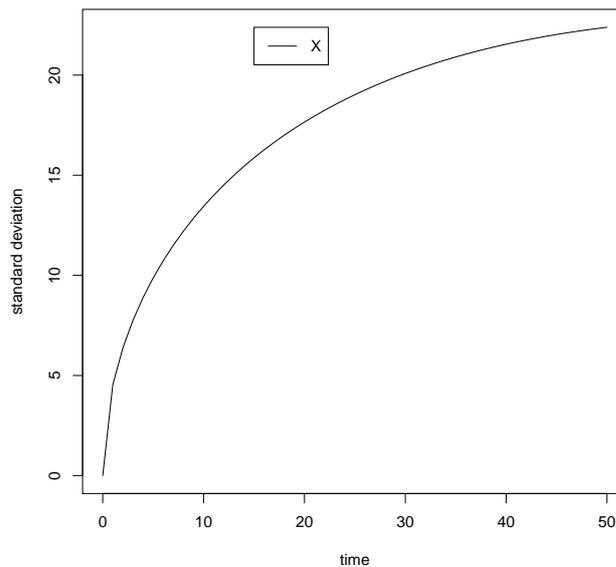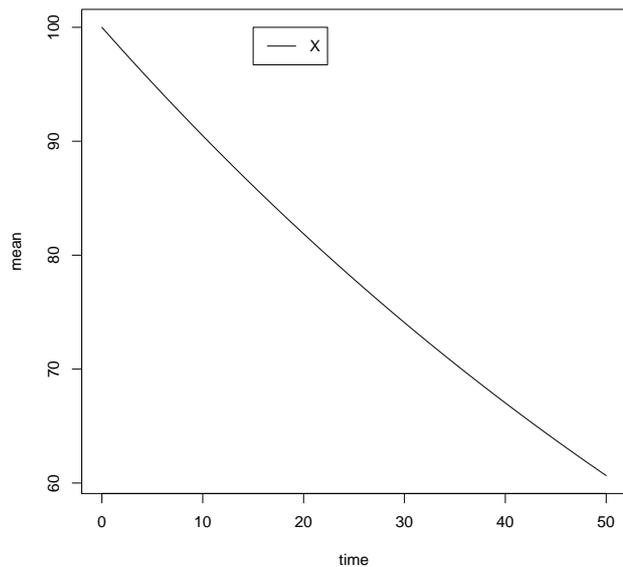
5.1.3 dsmts–001–03

This model is the same as `dsmts-001-01`, except that the values of the rate parameters are `Lambda=1` and `Mu=1.1`. The skewed distribution associated with this model means that simulators are likely to fail the suggested standard deviation test for large values of `t` (as the normality assumption underlying the test is clearly invalid in this case).
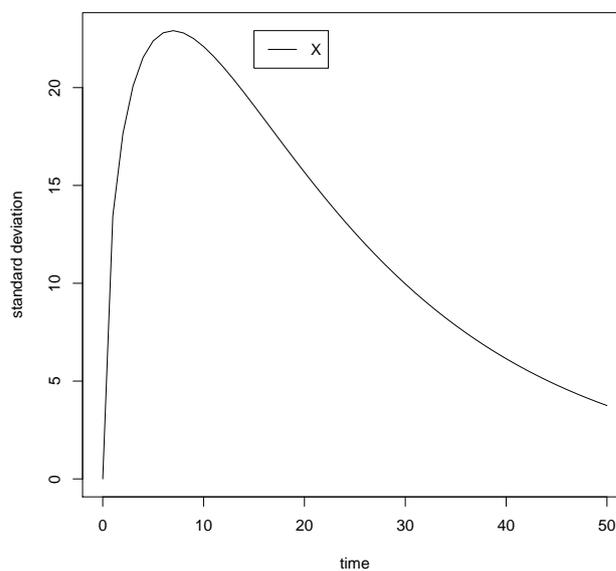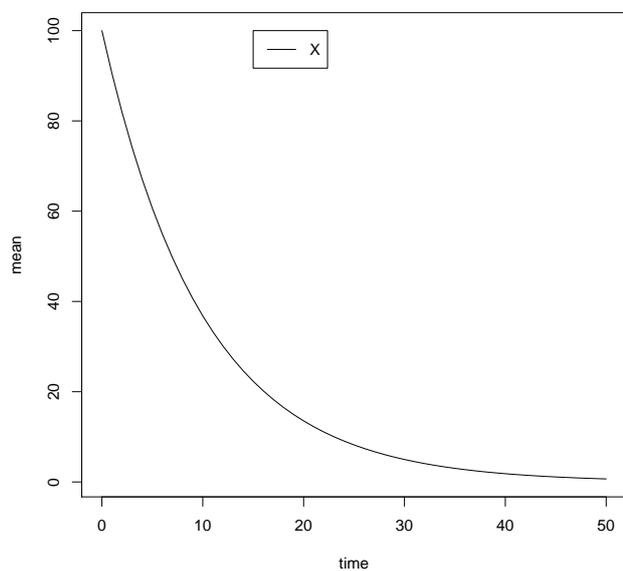
5.1.4 dsmts–001–04

This is the same as `dsmts-001-01`, except that the initial number of molecules of `X` is 10.
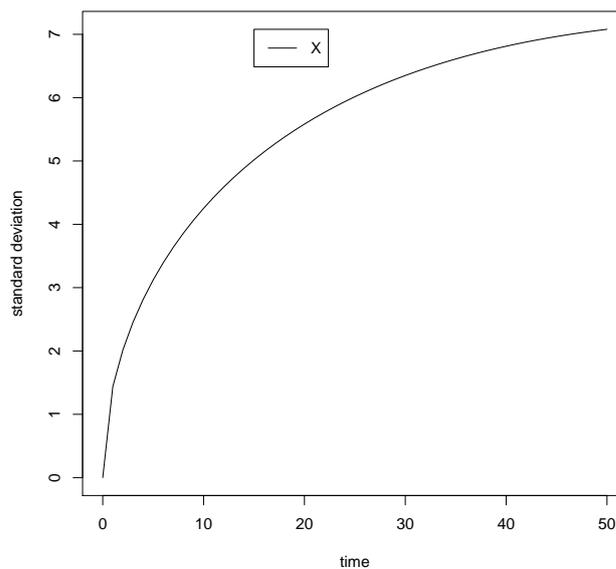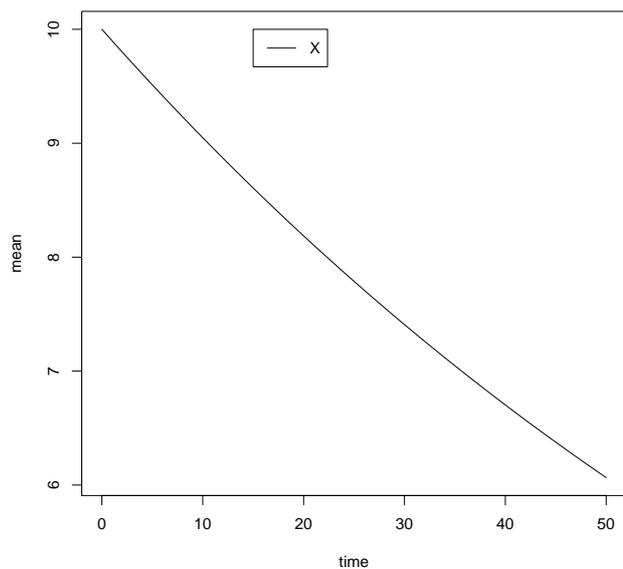
5.1.5 dsmts–001–05

The same as `dsmts-001-01`, except that the initial number of molecules of X is 10,000. Due to the large number of molecules involved, tests using this model will take longer to complete than for many other models in the suite.
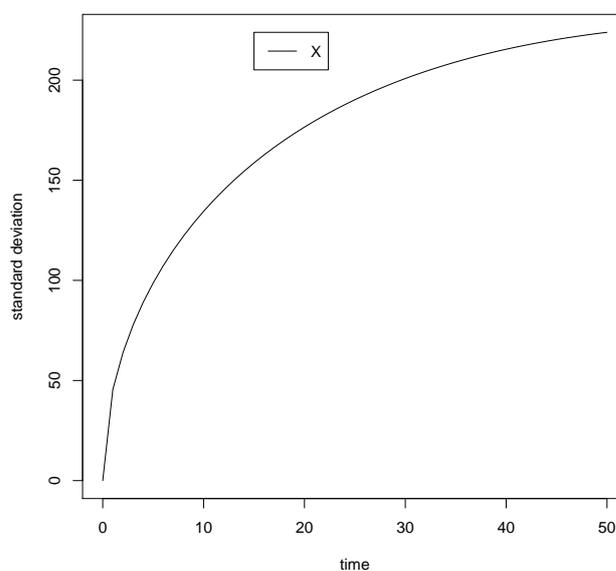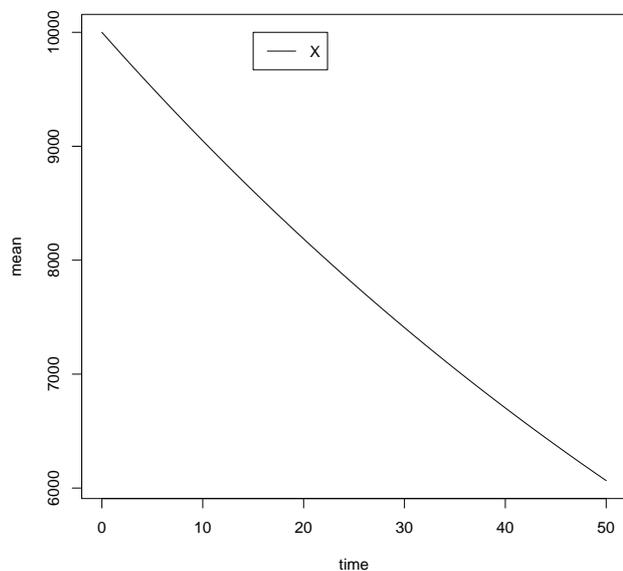
5.1.6 dsmts–001–06

The same as `dsmts-001-01`, except that: (i) There is another species, named `Sink`, with initial amount 0. The `Sink` species is declared to be a boundary condition, which means that its value is not determined by the model reactions but remains constant at its initial level. (ii) The death reaction is now:

$$X \longrightarrow \texttt{Sink}$$
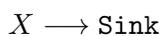
This model checks that the boundary condition attribute is handled correctly.
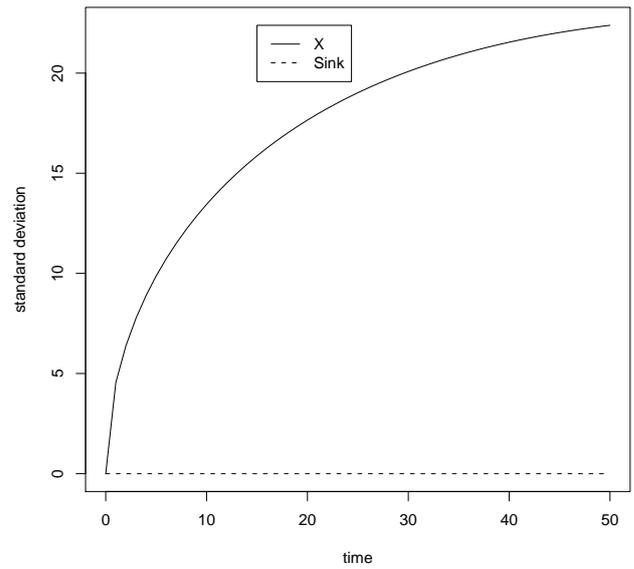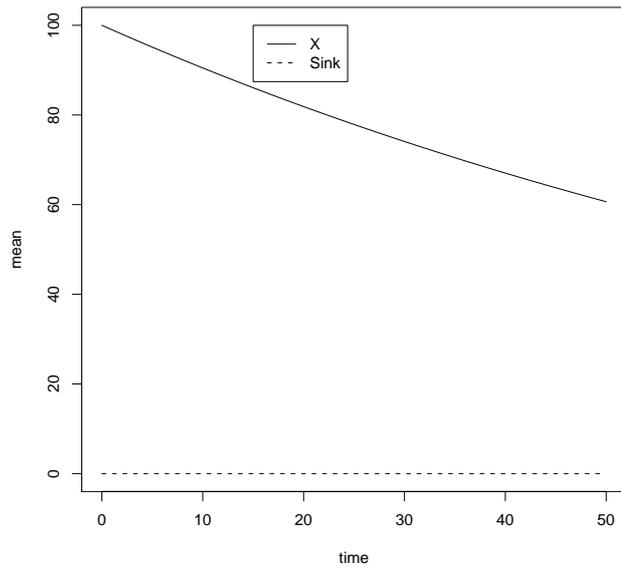
5.1.7 dsmts–001–07

The same as `dsmts-001-06`, except that the `Sink` species is not a boundary condition. This model is the first one involving two time-varying species.
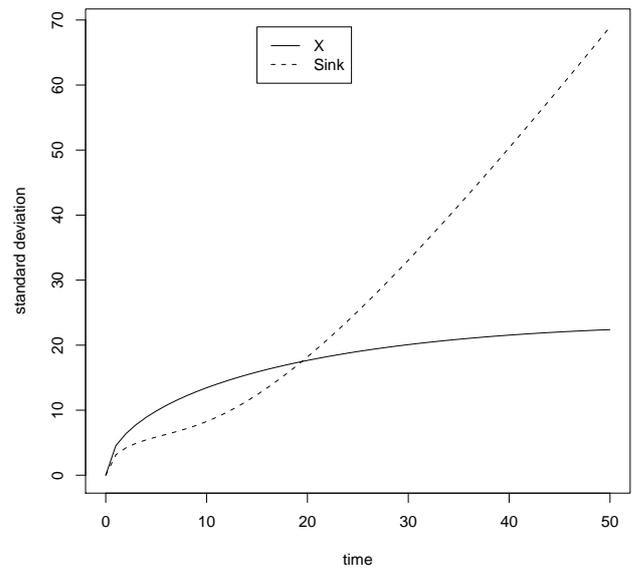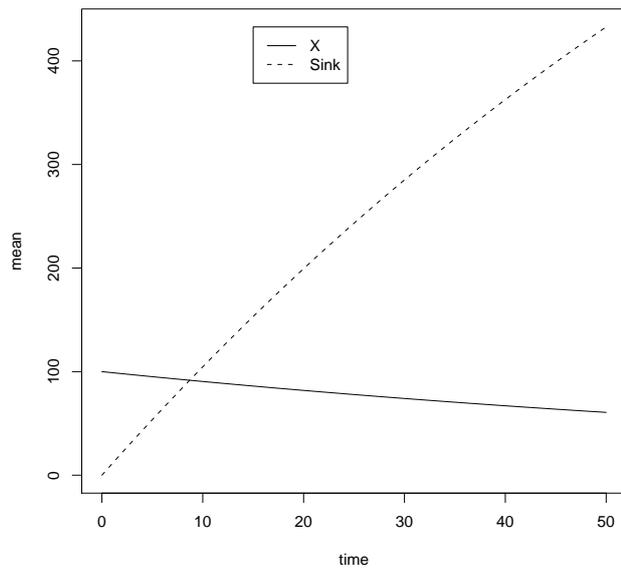
5.1.8 dsmts–001–08

Same as `dsmts-001-01`, except that the `Cell` compartment is declared to have `size=1`. This shouldn't affect anything.
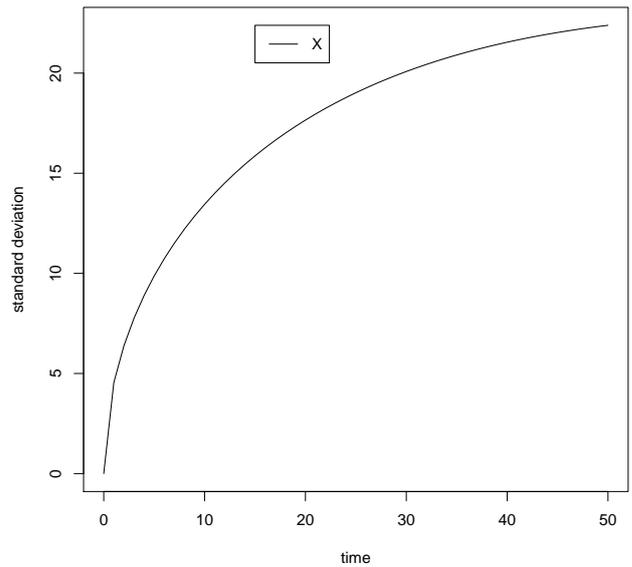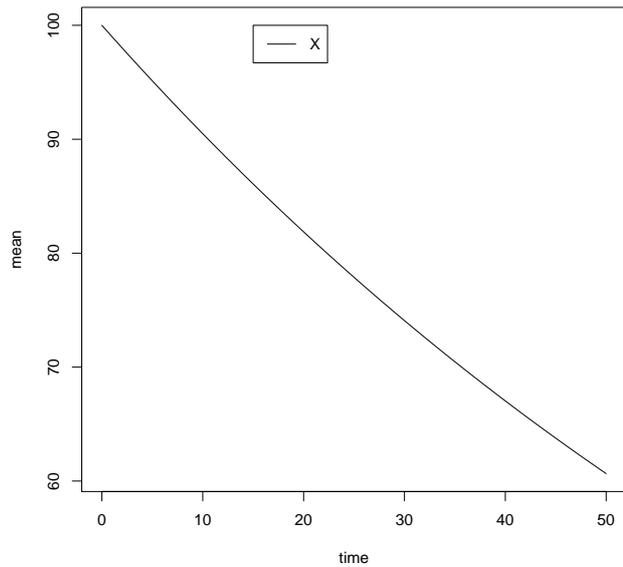
5.1.9 dsmts–001–09

Same as `dsmts-001-01`, except that the `Cell` compartment is declared to have `size=2`. Again, this shouldn't affect anything, due to the presence of the `hasOnlySubstanceUnits` flag in the species declaration.
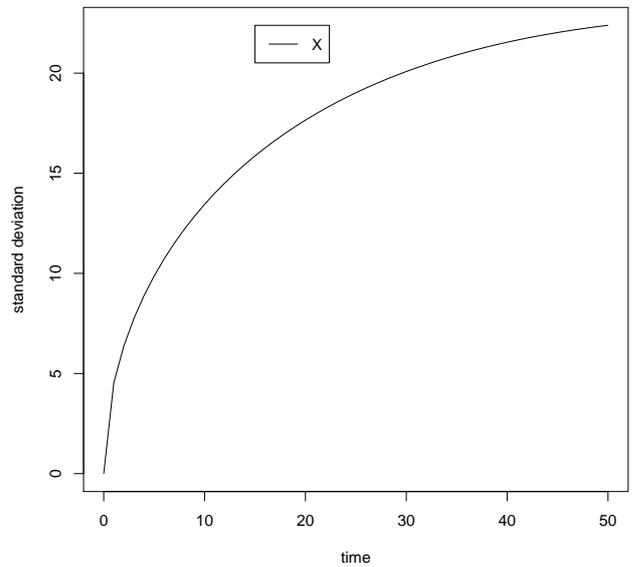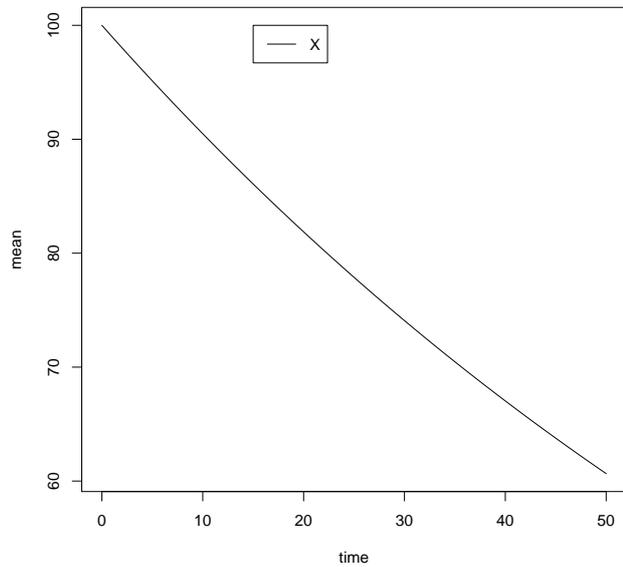
5.1.10 dsmts–001–10

Same as `dsmts-001-01`, except that: (i) the `Cell` compartment is declared to have `size=1` (ii) `hasOnlySubstanceUnits` is not declared to be true. Although many simulators will mis-interpret this model, most will get the right output due to the unit compartmental size.
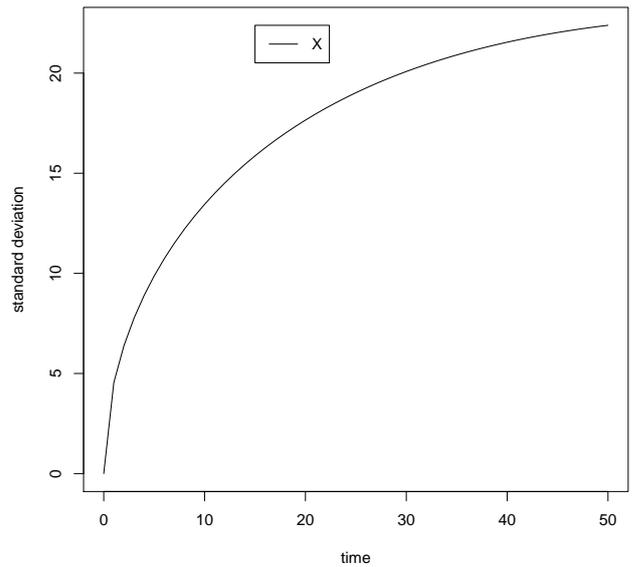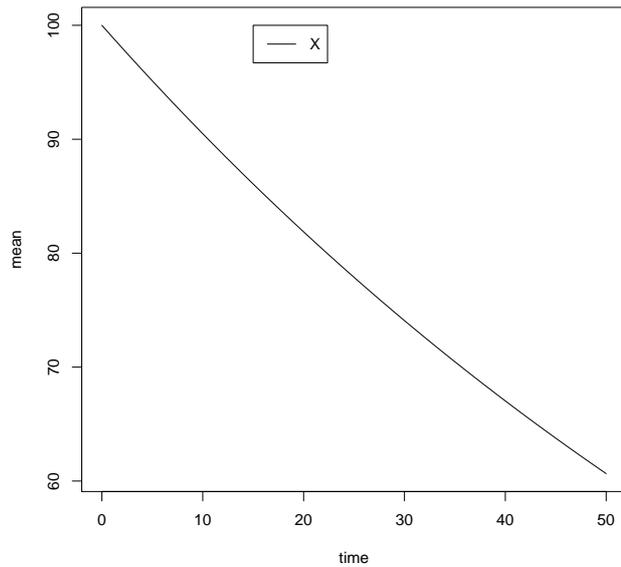
5.1.11 dsmts–001–11

This is the same as `dsmts-001-01`, except that: (i) the `Cell` compartment is declared to have `size=2` (ii) `hasOnlySubstanceUnits` is not declared to be true. Many simulators (which pass the earlier tests) are expected to fail this test, due to a lack of sophisticated handling of the `hasOnlySubstanceUnits` tag. It would be reasonable for a simulator to refuse to accept such a model, but a simulator should not accept the model and then produce incorrect output.
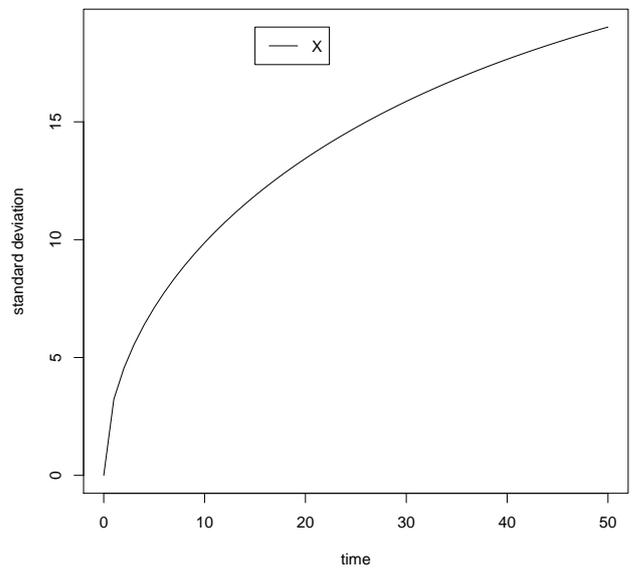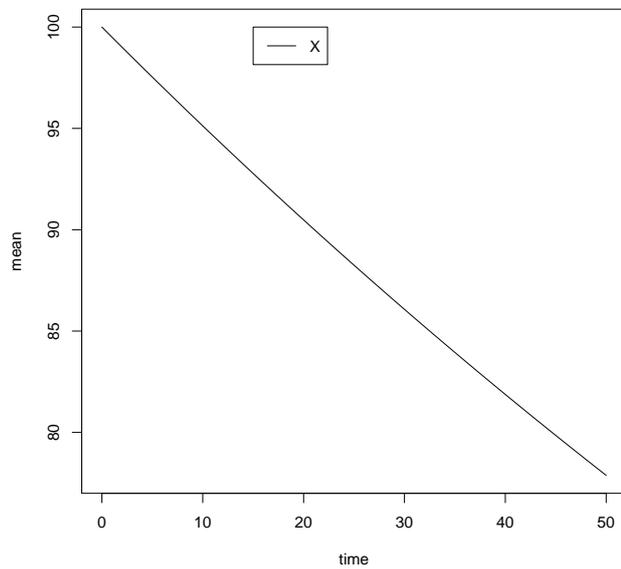
5.1.12 dsmts–001–12

This is the same as `dsmts-001-01`, except that the rate law is written as `Lambda*X*0.5*2`. This is designed to test the math expression parsing.
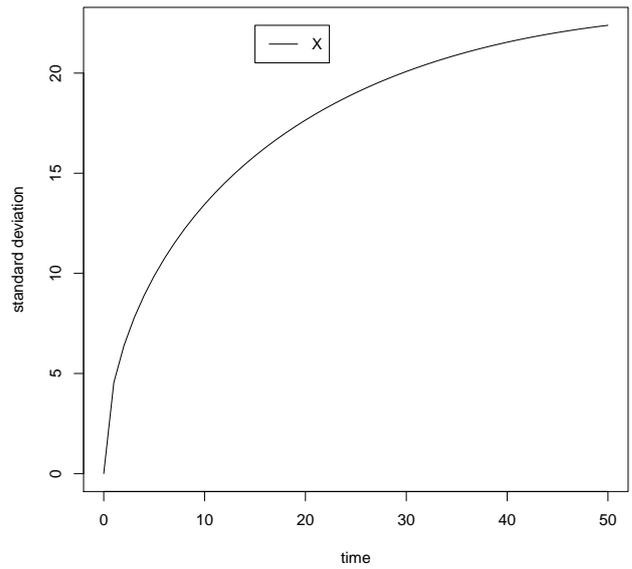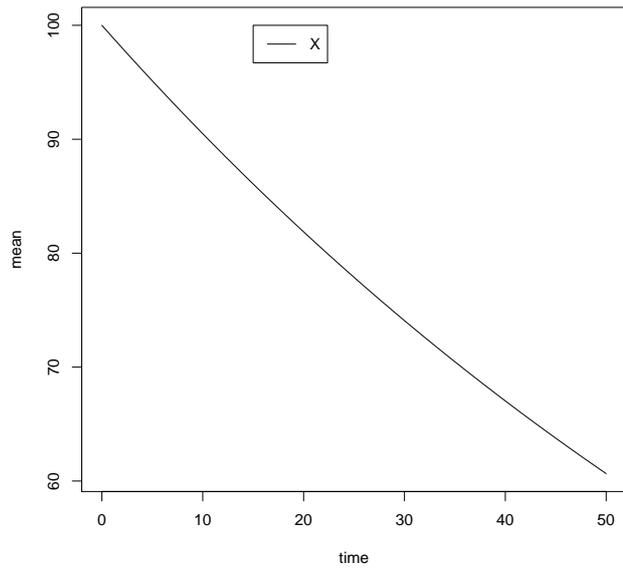
5.1.13 dsmts–001–13

This is the same as `dsmts-001-01`, except that `Lambda=0.2` and the rate law is written as `Lambda*X*0.5`. This is designed to test the math expression parsing.
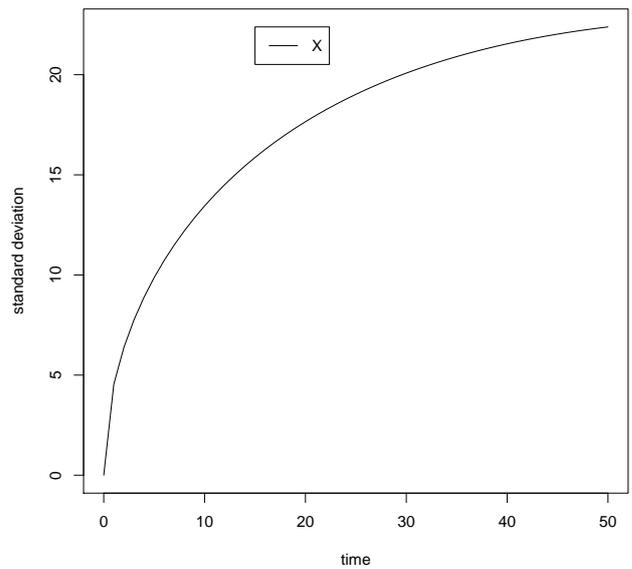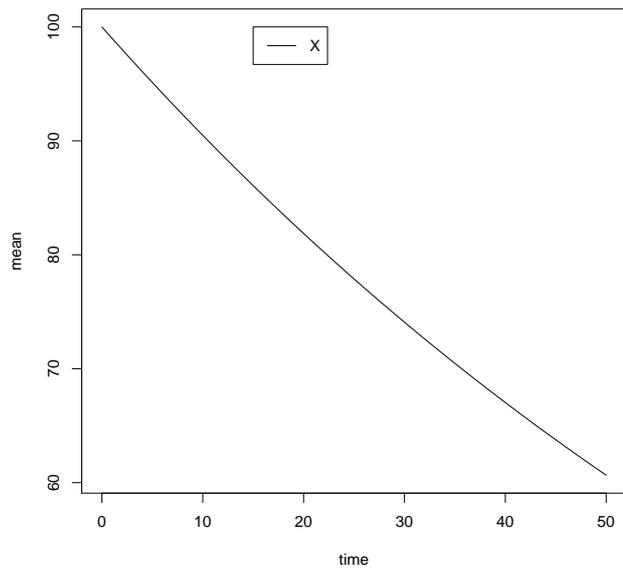
5.1.14 dsmts–001–14

This is the same as `dsmts-001-01`, except that the rate law is written as `Lambda*X/2/0.5`. This is designed to test the math expression parsing.
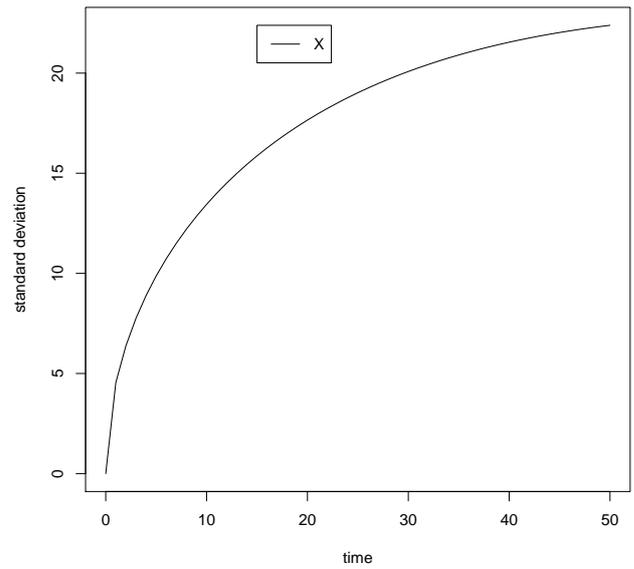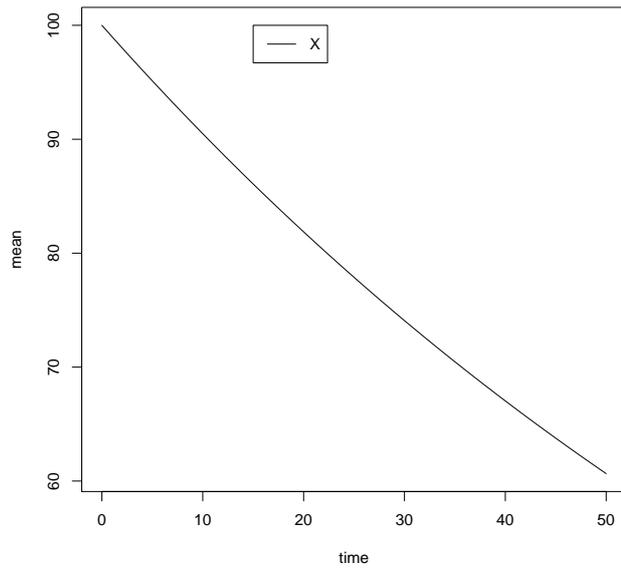
5.1.15 dsmts–001–15

This is the same as `dsmts-001-01`, except that the rate law is written as `Lambda*(X/2)/0.5`. This is designed to test the math expression parsing.
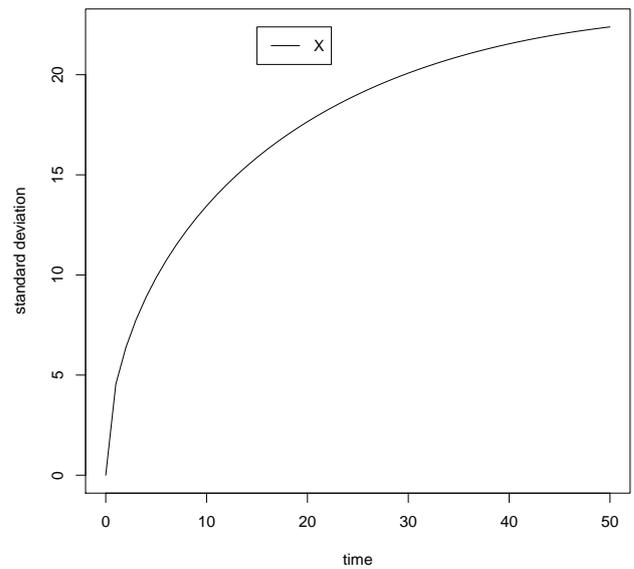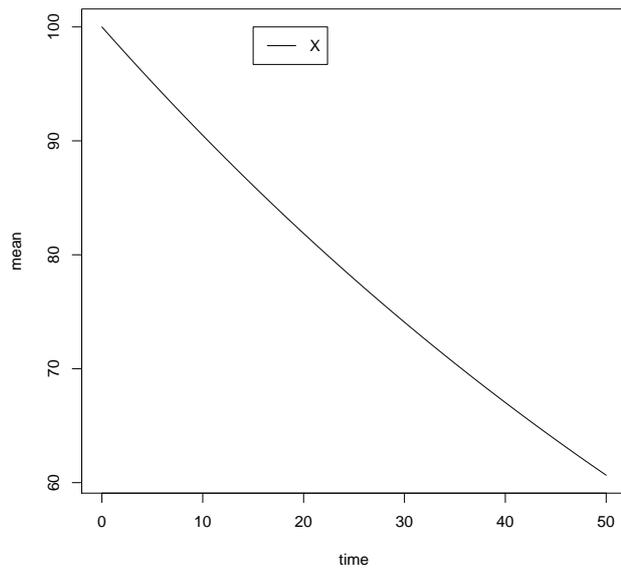
5.1.16 dsmts–001–16

This is the same as `dsmts-001-01`, except that the rate law is written as `Lambda*X/(2/2)`. This is designed to test the math expression parsing.
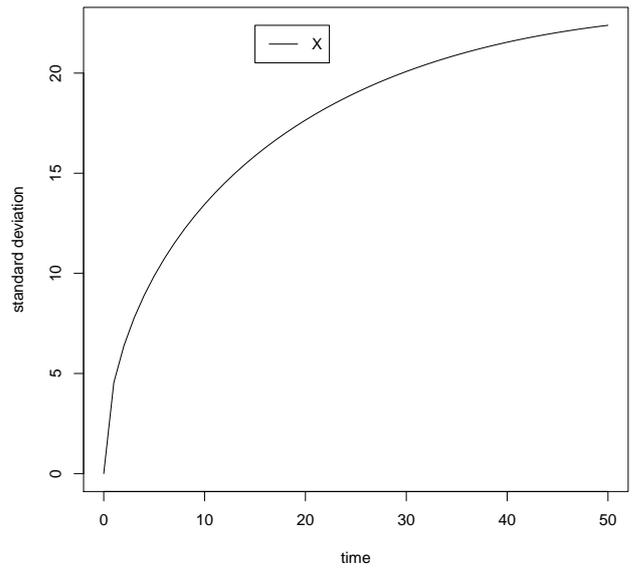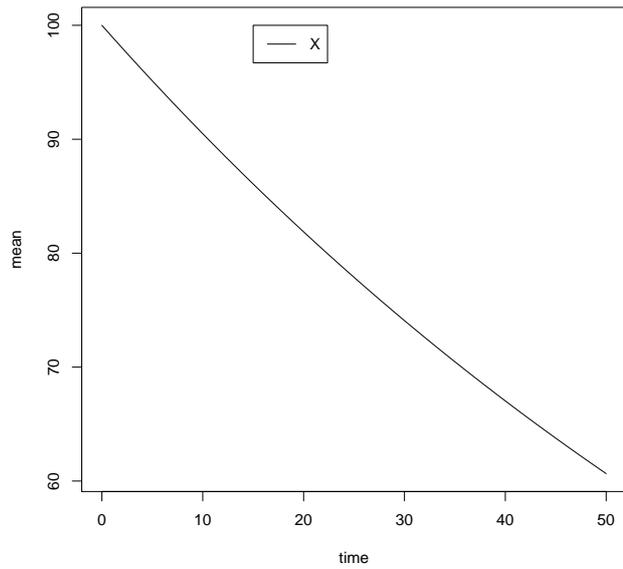
5.1.17 dsmts–001–17

This is the same as `dsmts-001-08`, except that the (unit) compartmental volume is explicitly included in the rate laws (this shouldn't affect anything).
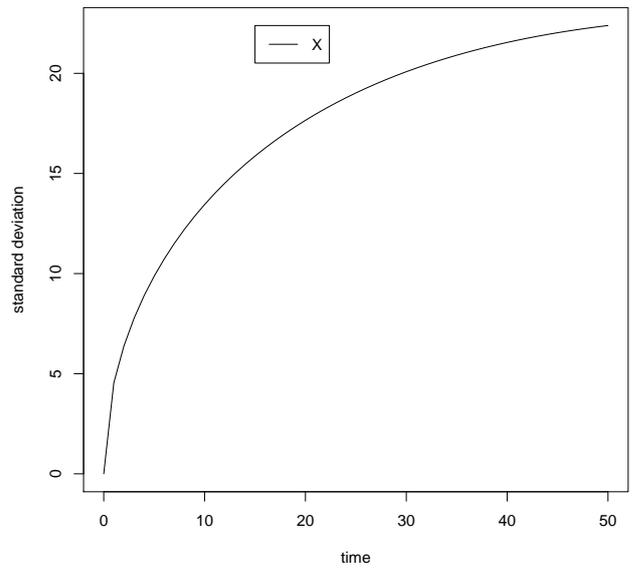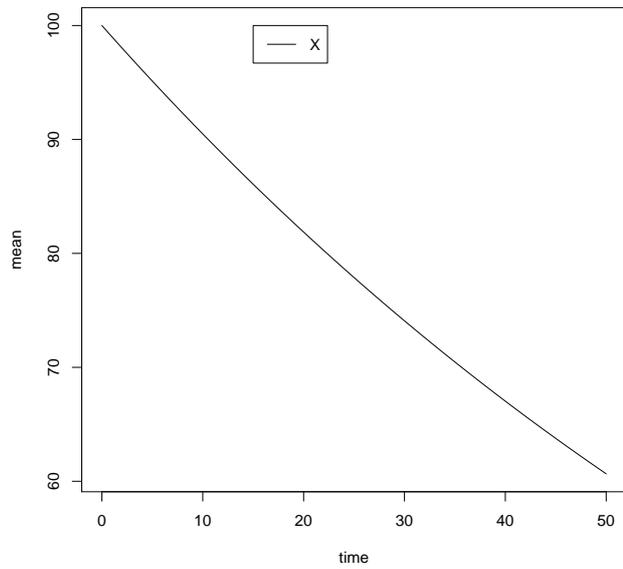
5.1.18 dsmts–001–18

This is the same as `dsmts-001-17`, except that the compartmental volume is set to 0.5.
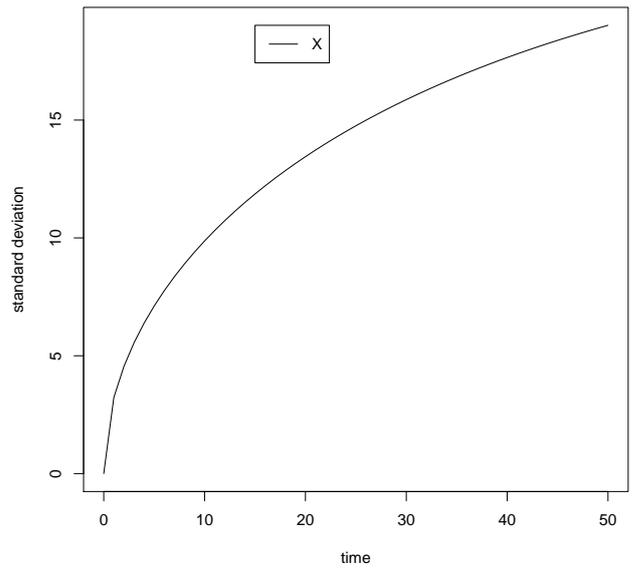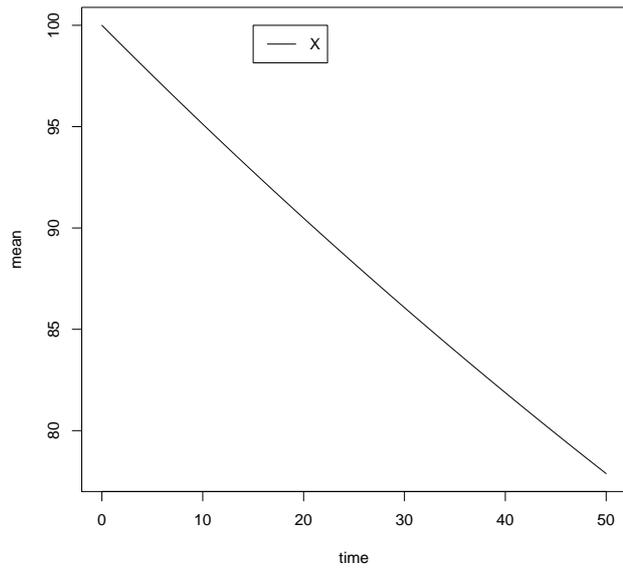
5.1.19 dsmts–001–19

This is the same as `dsmts-001-01`, except that there is an assignment rule that defines a new species y that is $2 \times X$. This is the first model containing an assignment rule.
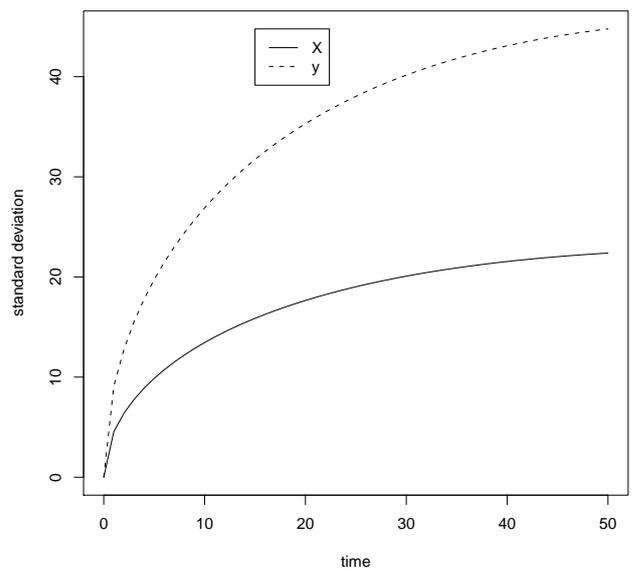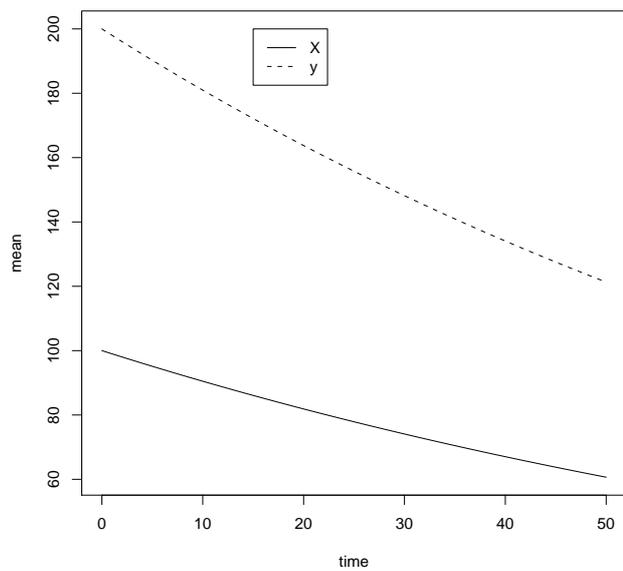
5.2 dsmts–002 — The immigration-death process

5.2.1 dsmts–002–01

This model contains one species, denoted by X. The amount of X present in the system is measured in numbers of molecules. The initial number of molecules of X is 0. The immigration reaction

$$\emptyset \longrightarrow X$$
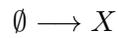
has global rate parameter `Alpha=1`. The death reaction $X \longrightarrow \emptyset$ has global rate parameter `Mu=0.1`. Mass-action stochastic kinetics is assumed.
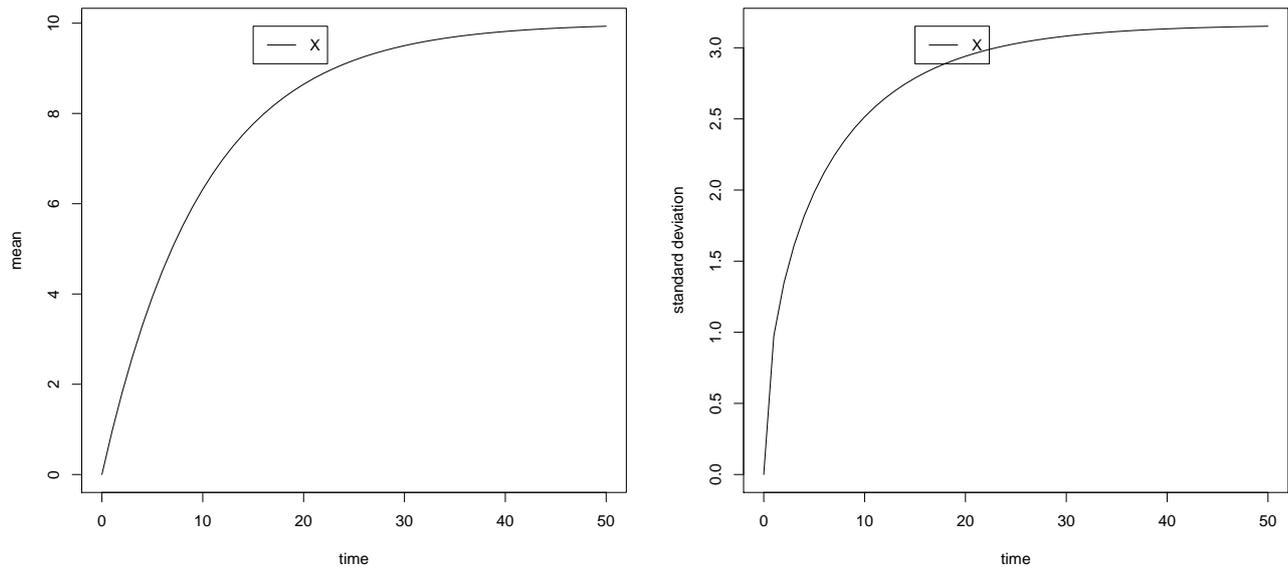
5.2.2 dsmts–002–02

Same as `dsmts-002-01`, except that `Alpha=10`.

14
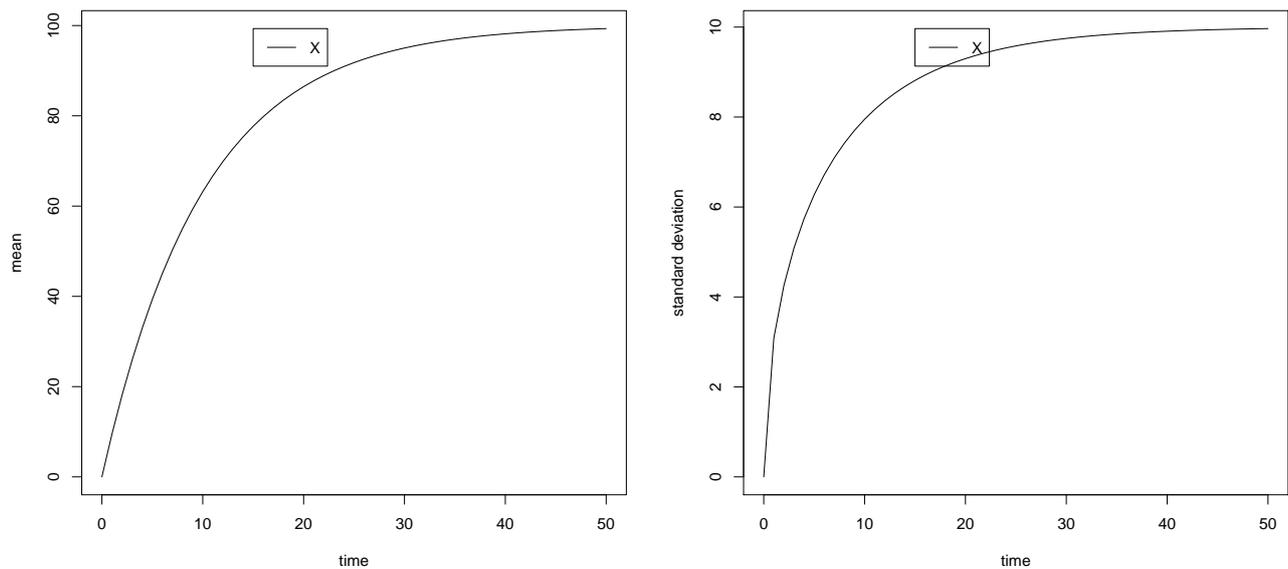
5.2.3 dsmts–002–03

Same as `dsmts-002-02`, except the global parameter `Alpha=10` is overridden by the local parameter `Alpha=5`. This is the first model that checks that local parameters overload global parameters.
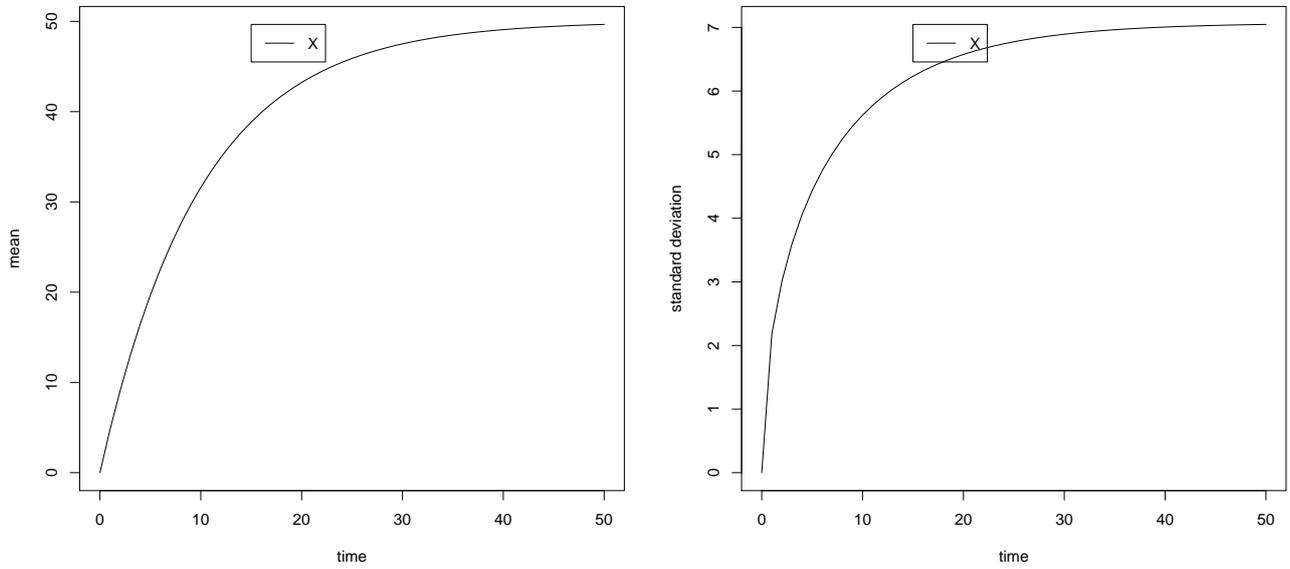
5.2.4 dsmts–002–04

Same as `dsmts-002-01`, except that `Alpha=1000`. This model will be slow to run.
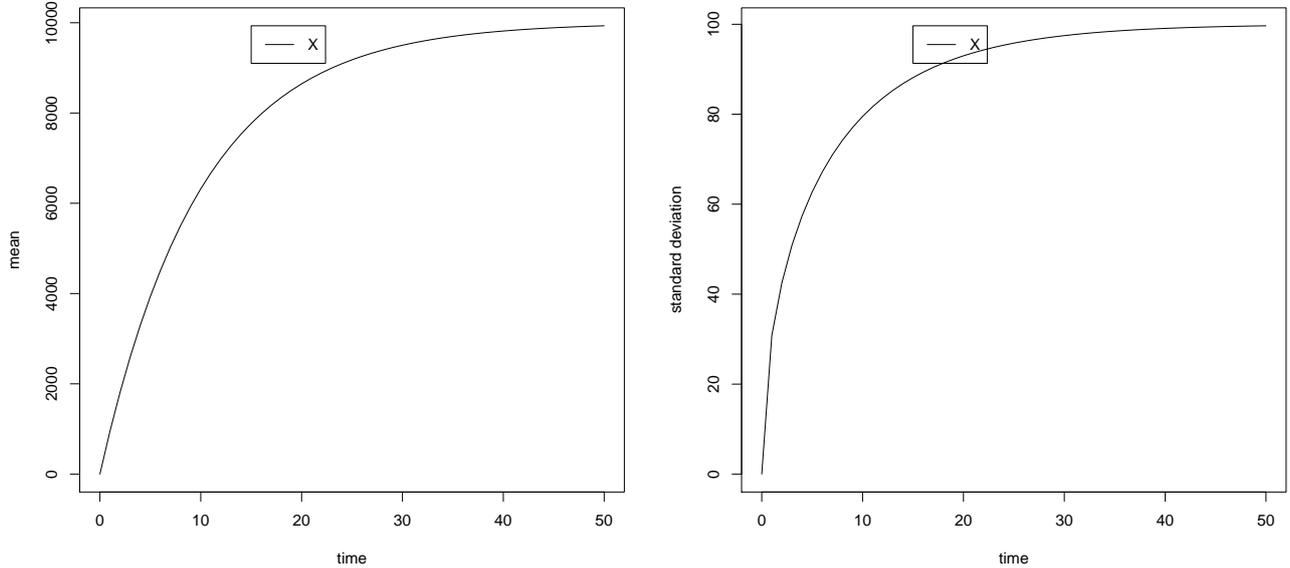
5.2.5 dsmts–002–05

Same as `dsmts-002-02`, except that: (i) There are two additional species: `Source` and `Sink`. `Source` and `Sink` are both boundary conditions, and they both have initial amount 0. (ii) The immigration reaction is now:

$$Source \longrightarrow X.$$
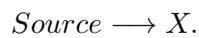
(iii)The death reaction is now:

$$X \longrightarrow Sink$$
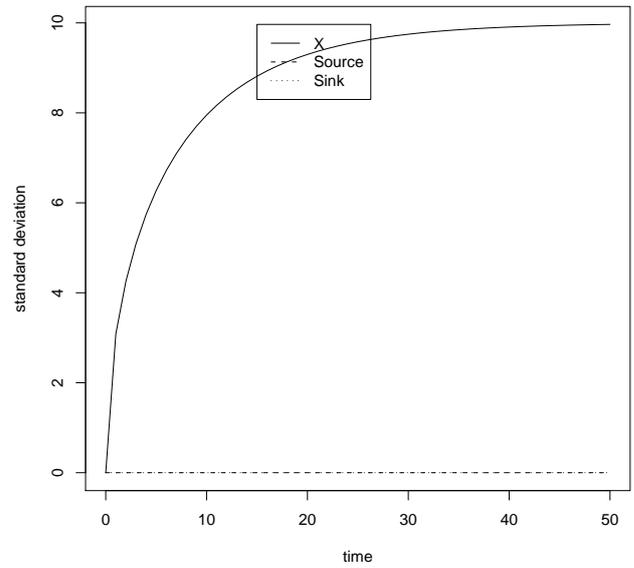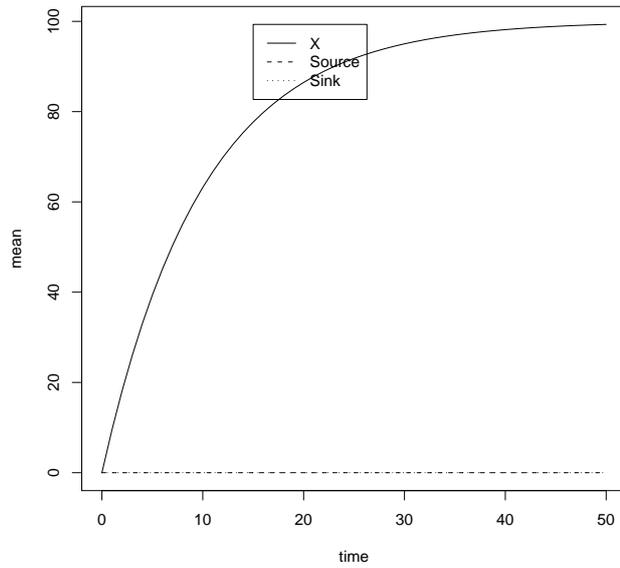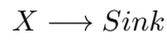
5.2.6 dsmts–002–06

Same as `dsmts-002-05`, except that the `Sink` species is not a boundary condition.
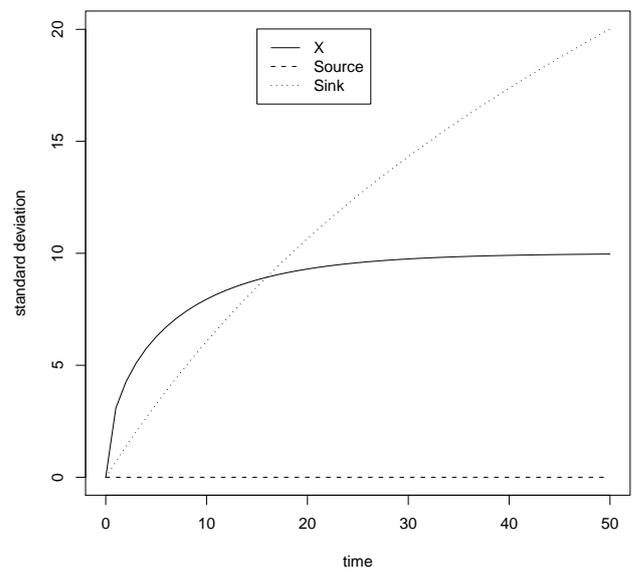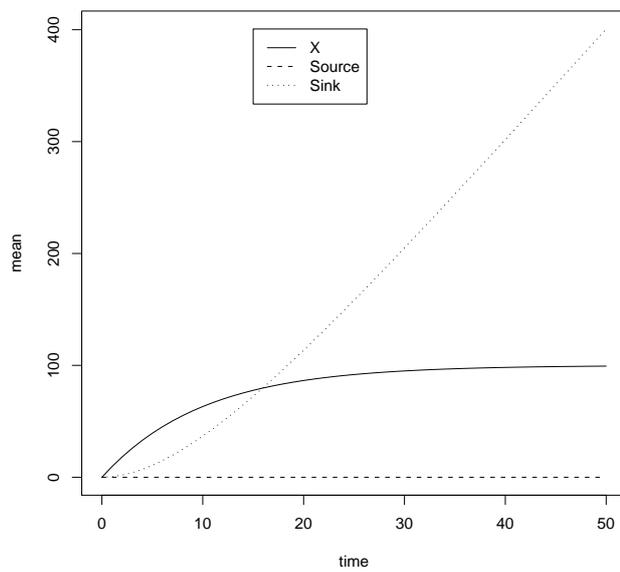

5.2.7 dsmts–002–07

Same as `dsmts-002-05`,except that the Sink species is a constant boundary condition.
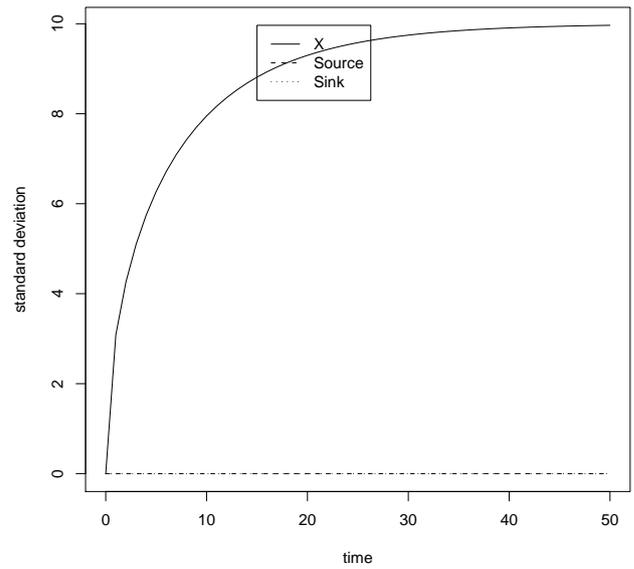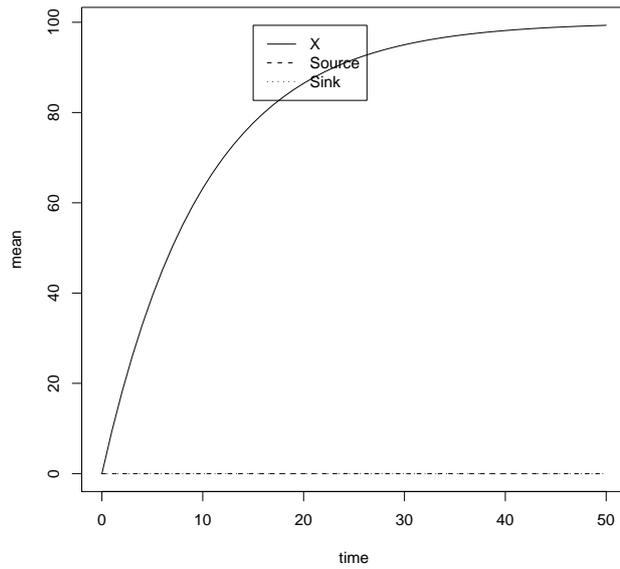
5.2.8 dsmts–002–08

Same as `dsmts-002-01`, except that: (i) The global rate parameter `k` is set equal to 2. (ii) In the immigration rate, `k` is locally set equal to 1(iii) In the death rate, `k` is locally set equal to 0.1. This model is designed to test parameter overloading.
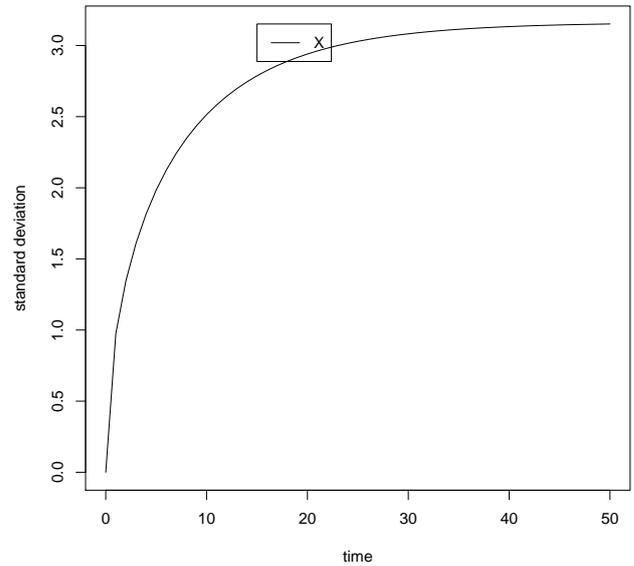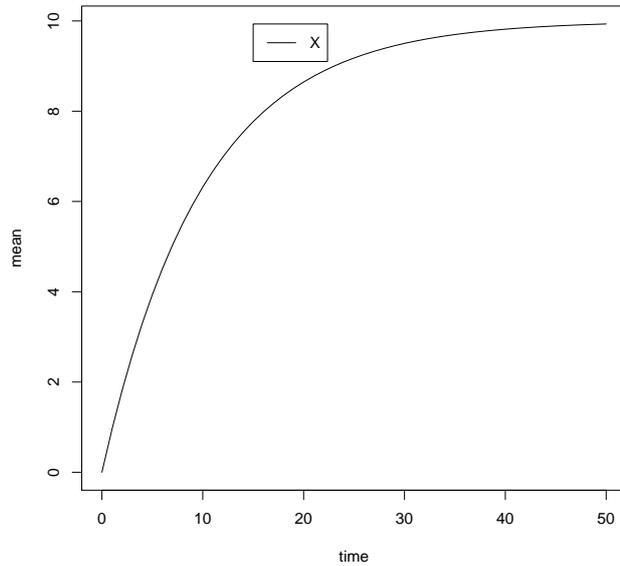
5.2.9 dsmts–002–09

Same as `dsmts-002-01`, except that an event is triggered at time 25 to reset X to a value of 50. This is the first model that includes an event. It is a timed event.
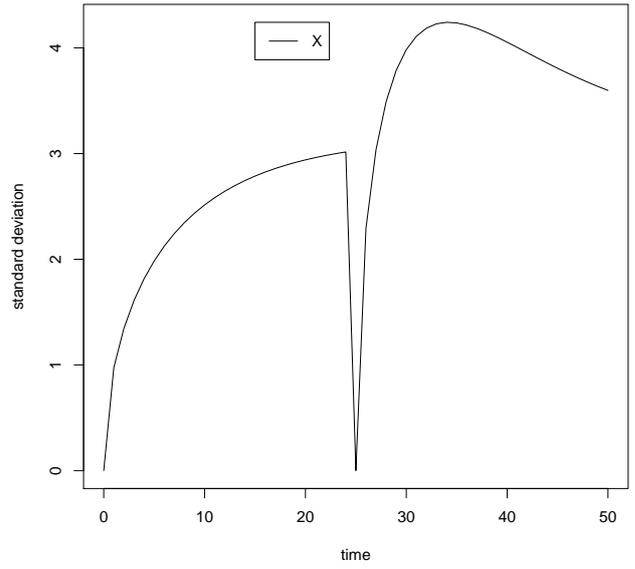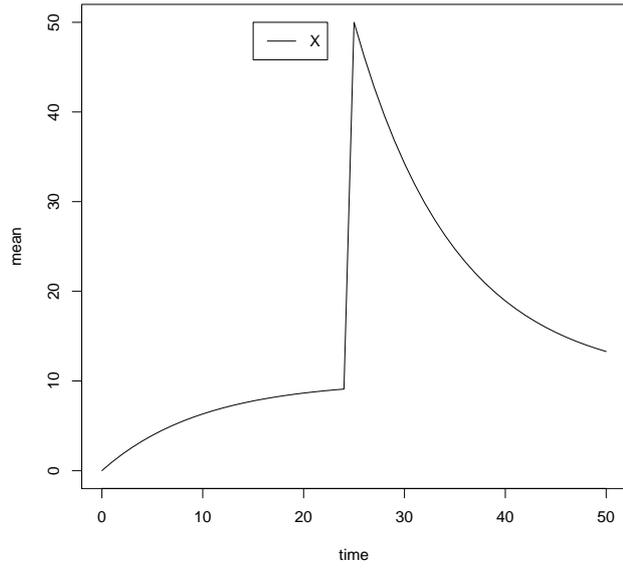
5.2.10 dsmts–002–10

Very similar to `dsmts-002-09`, except that the event is triggered at time 22.5 and the reset value is 20. The novelty here is that the event is triggered at a time not corresponding to a time step.
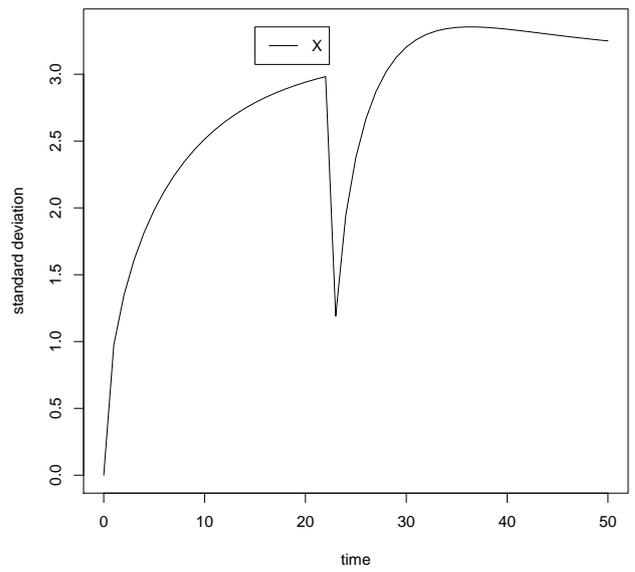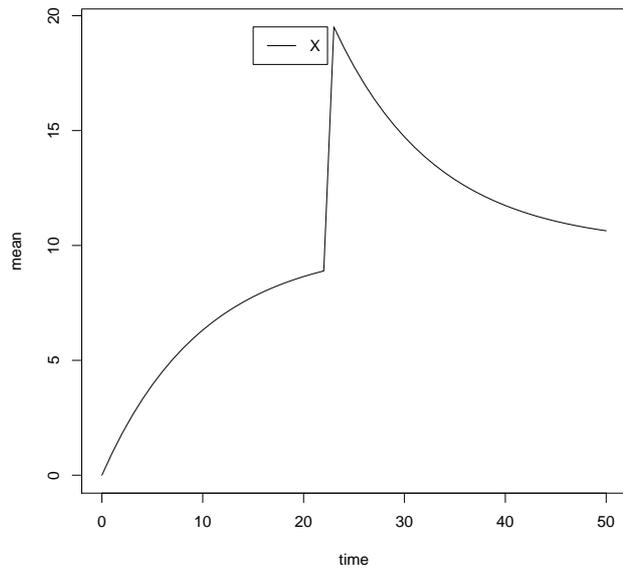
## 5.3 dsmts–003 —Dimerisation

5.3.1 dsmts–003–01

This model contains two species, denoted by `P` and `P2`, representing a dimerisation process. The initial numbers of molecules of P and P2 are 100 and 0 respectively. The dimerization reaction

$$2P \longrightarrow P2$$

has global rate parameter `k1=0.001`. Note that this is a second order reaction with rate law `k1*P*(P-1)/2`. The dissociation reaction

$$P2 \longrightarrow 2P$$

has global rate parameter `k2=0.01`.
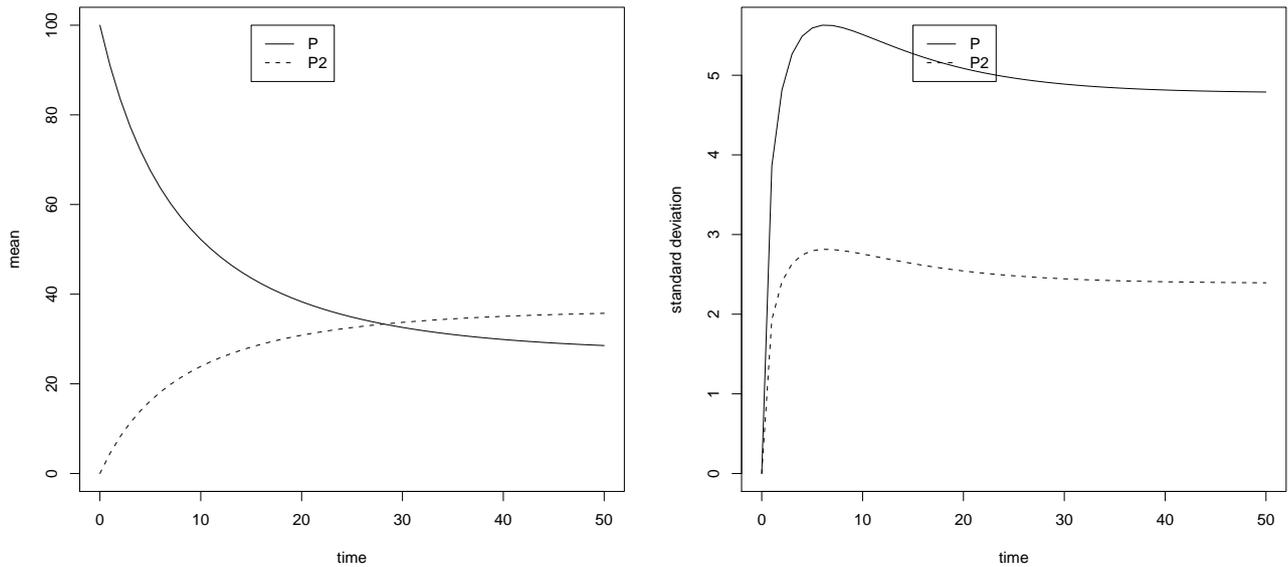
5.3.2 dsmts–003–02

Same as `dsmts-003-01`, except that: (i) the initial number of molecules of `P` is 1000 (ii) the values of the rate parameters are `k1=0.0002` and `k2=0.004`.
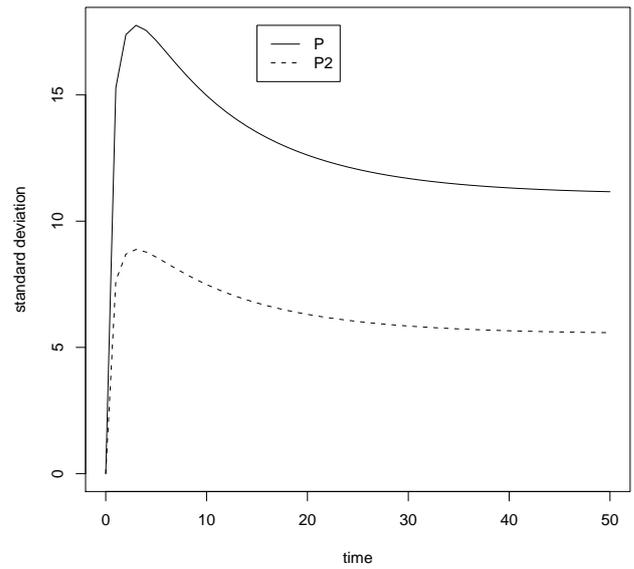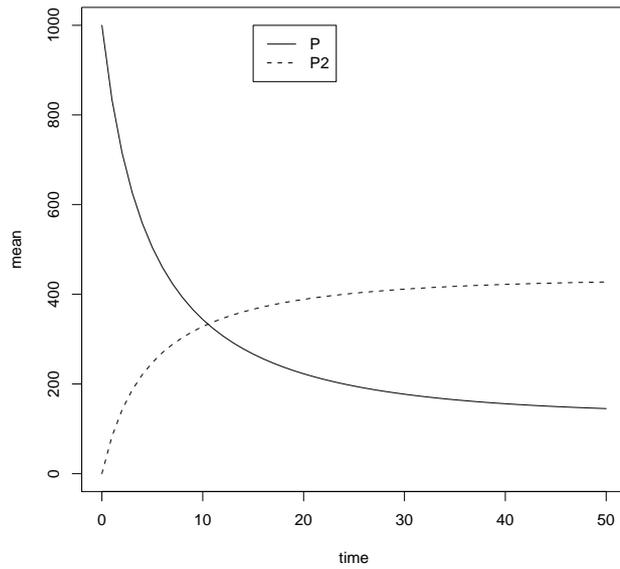
5.3.3 dsmts–003–03

Same as `dsmts-003-01`, except that in the event that the time parameter `t` becomes greater than or equal to 25, the species populations are reset to the following values: `P := 100`, `P2 := 0`. This is the first model that tests SBML `event`s with multiple assignments.
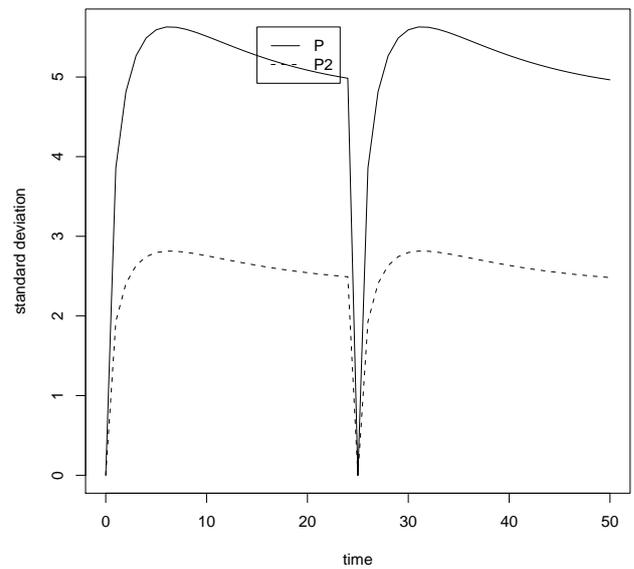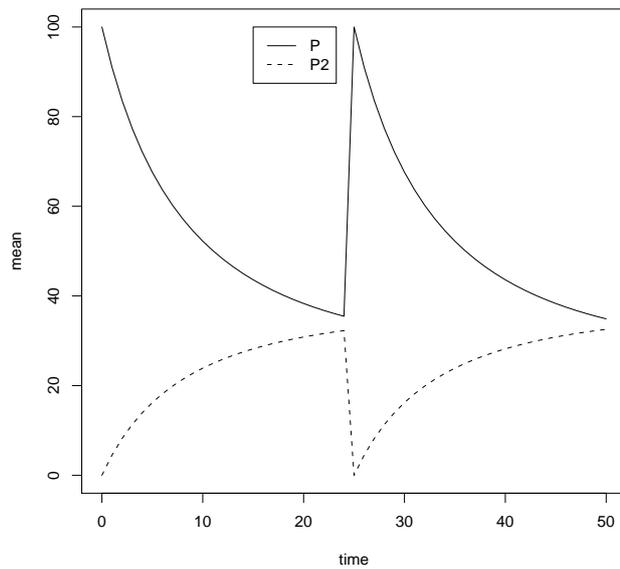

5.3.4 dsmts–003–04

Same as `dsmts-003-01`, except that in the event that the number of `P2` molecules becomes greater than 30, the species populations are reset to the following values: `P := 100`, `P2 := 0`. This is the first test of SBML event handling where the event is triggered by a species — not a timed event.
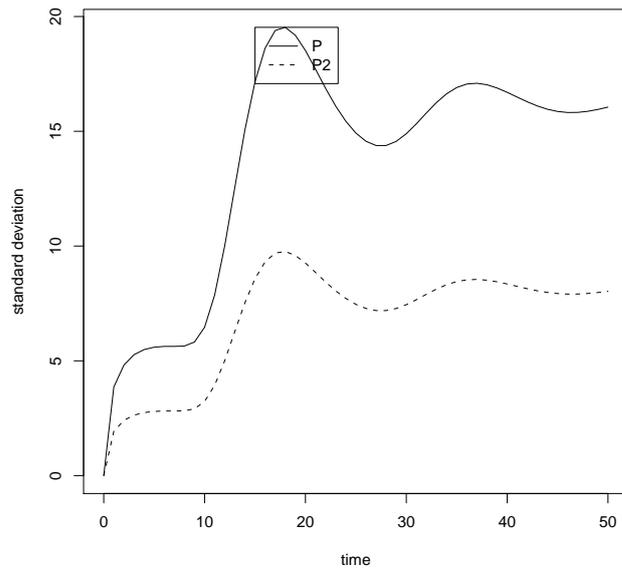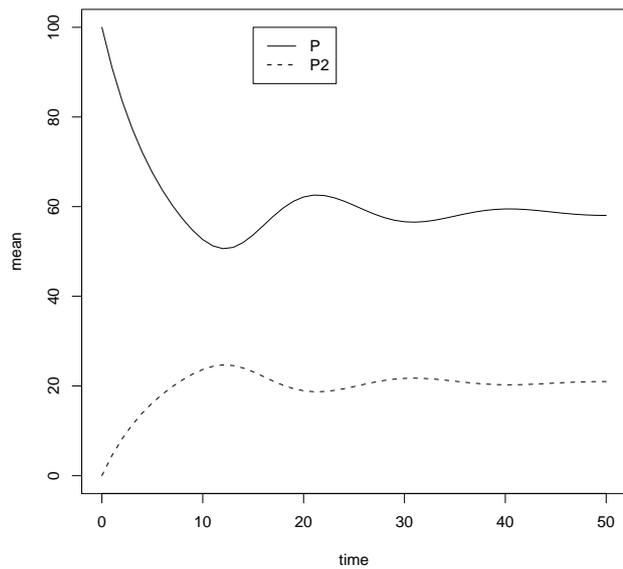
5.3.5 dsmts–003–05

Same as `dsmts-003-01`, except that P has been removed from the model using the conservation law present in the system. This tests several things, including complex math expression parsing.
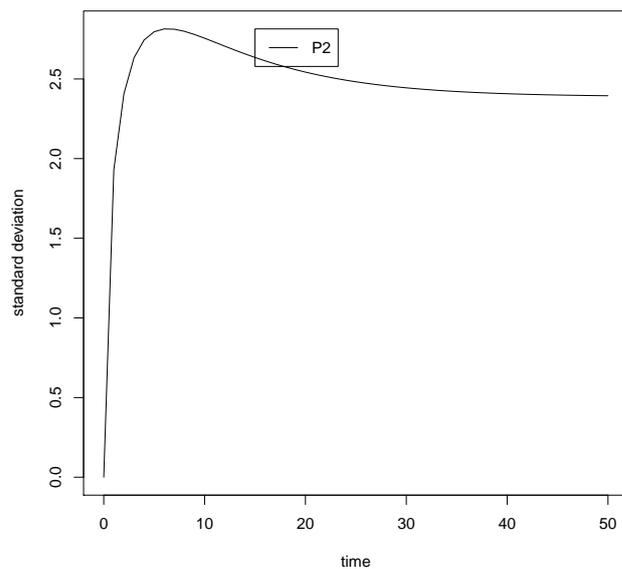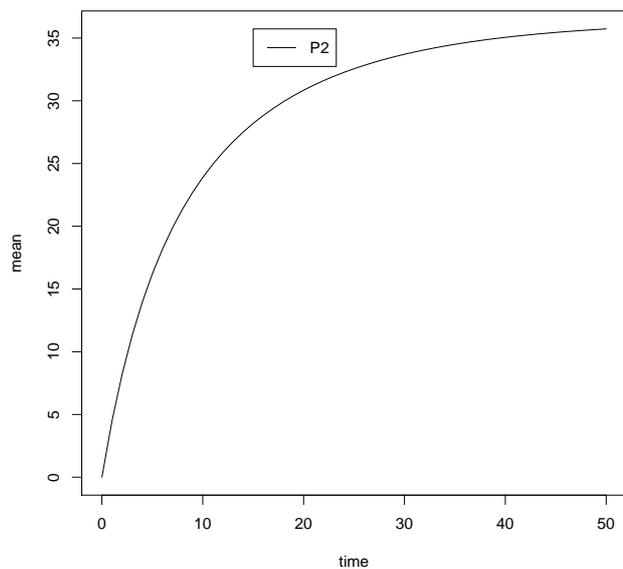

5.3.6 dsmts–003–06

Same as `dsmts-003-05`, except that the rate law is written slightly differently. This is testing the expression parser.
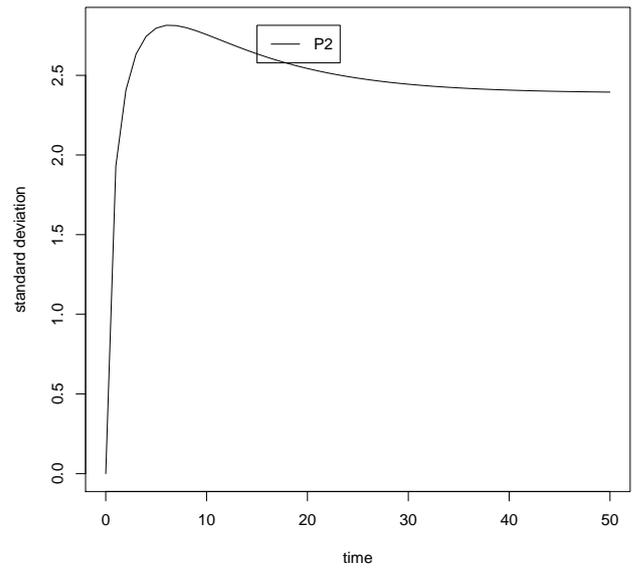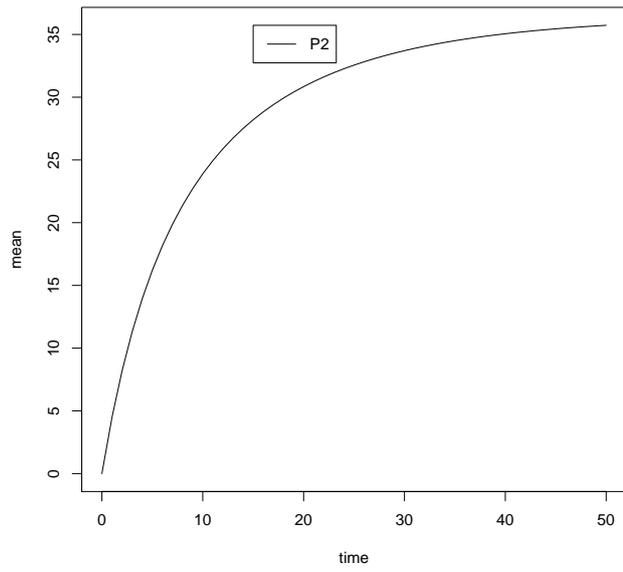
5.3.7 dsmts–003–07

Same as `dsmts-003-06`, except that the rate law is written very slightly differently. This is testing the expression parser.
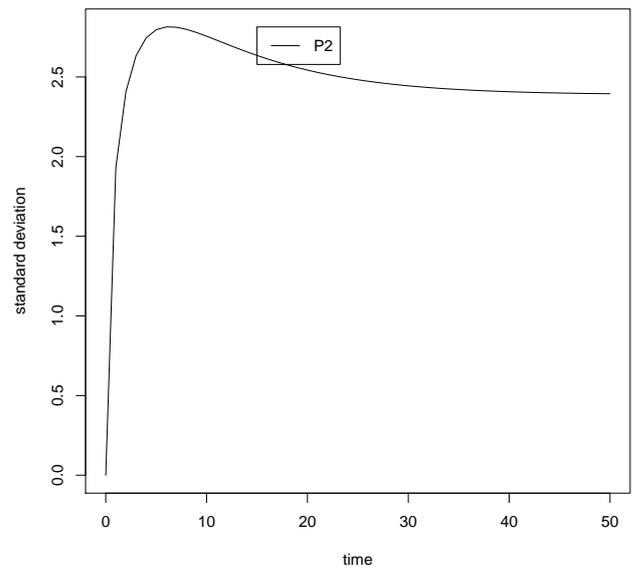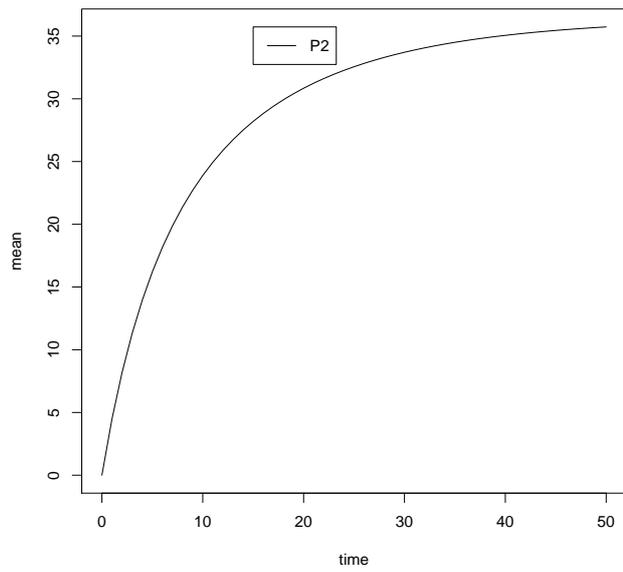
5.4 dsmts–004 — The batch immigration-death process

5.4.1 dsmts–004–01

This model contains one species, denoted by X. The amount of X present in the system is measured in numbers of molecules. The initial number of molecules of X is 0. The immigration reaction

$$\emptyset \longrightarrow 5X$$
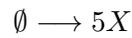
has global rate parameter `Alpha=1`. The death reaction $X \longrightarrow \emptyset$ has global rate parameter`Mu=0.2`. Mass-action stochastic kinetics is assumed.
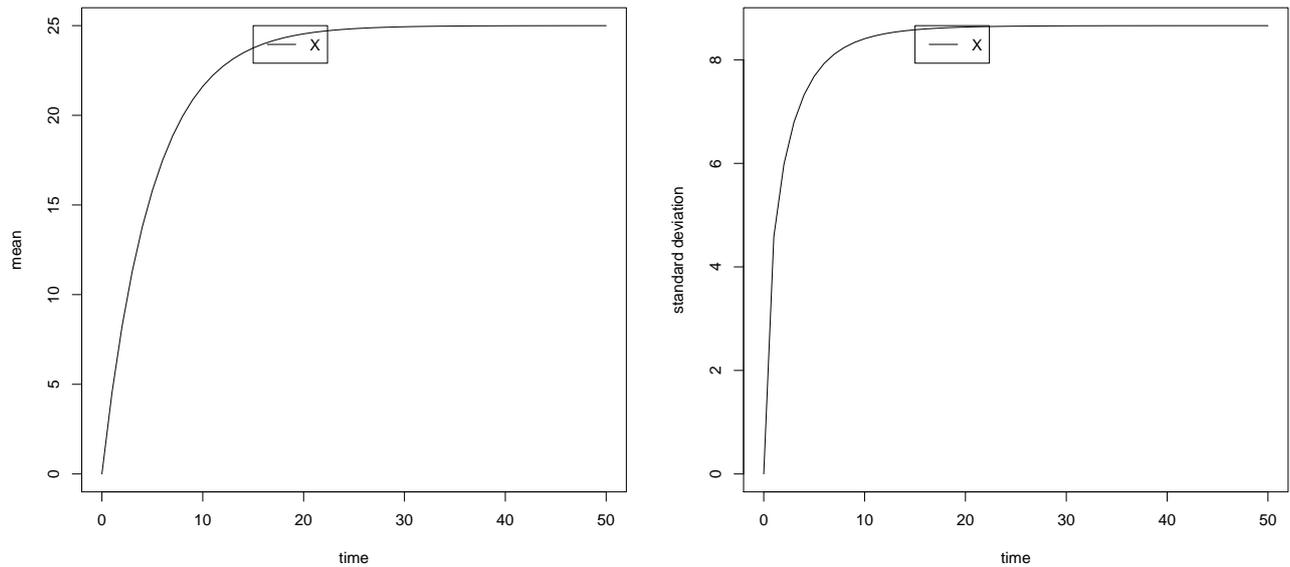
5.4.2 dsmts–004–02

Same as `dsmts-004-01`, except that the immigration reaction is

$$\emptyset \longrightarrow 10X$$
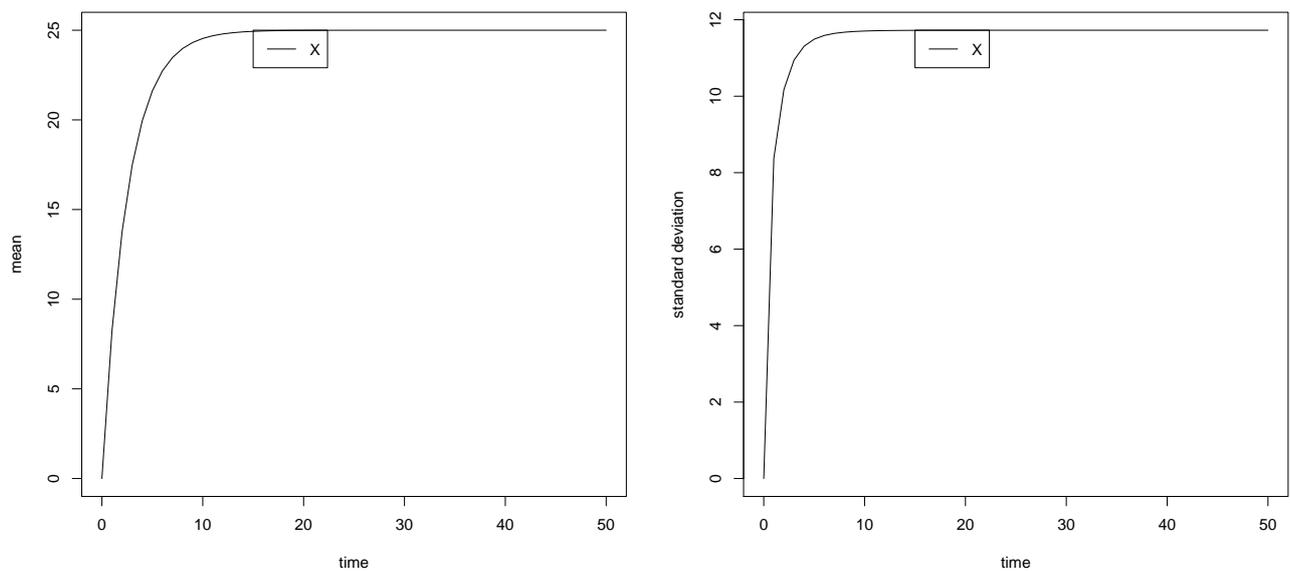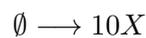
5.4.3 dsmts–004–03

Same as `dsmts-004-01`, except that the immigration reaction is

$$\emptyset \longrightarrow 100X$$