Introduction to Mathematical Modeling using Magnolia

With Applications to Pharmacokinetic Analysis

Section 1: Introduction to Magnolia

Outline

1. **Introduction to Magnolia**
2. Overview of the Magnolia user interface and modeling and analysis workflow
3. Specifying models using the CSL language
4. Scripting simulation runs using the CMD and Python languages
5. Additional analytical methods: parameter estimation, sensitivity analysis, Monte-Carlo analysis

What is Magnolia?

- An easy-to-use software application for modeling systems whose behavior can be described by differential equations
- An interactive environment for data visualization and model exploration
- A workbench for performing analyses such as parameter estimation, sensitivity analysis and Monte-Carlo analysis
- A flexible scripting environment for controlling simulation runs, pre- and post-processing data and programmatically specifying analyses and data visualizations

Magnolia History

- Originally developed in 2015, Magnolia was created as an alternative to tools such as ACSL/acslX and Berkeley Madonna
- Over the last few years, Magnolia has evolved to support new modeling language features, more scripting and analysis capabilities, and a user interface which enables interactive exploration of model behavior
- Presently, Magnolia is in use by industrial, government and academic organizations around the world, and is applied to fields including life sciences, vehicle modeling and industrial process modeling
- Magnolia has also been used as the simulation “engine” in other software applications

How Models are Specified in Magnolia

Magnolia provides several languages which are specifically designed to support different modeling and simulation activities.

For model specification, Magnolia uses a language called “CSL” (Continuous Simulation Language) to describe systems of ordinary differential equations. This language provides numerous built-in operators to simplify the specification and analysis of mathematical models.

```
model Exponential
    derivative

        constant k    = 1.0  ! Decay rate constant
        constant xic  = 10.0 ! Initial condition
        xd = -k*x            ! Derivative
        x  = integ(xd, xic) ! Integrate derivative to compute state

        constant tstop = 10.0
        termt(t >= tstop, 'Stopped on time limit')

    end ! derivative
end ! program
```

Simulation scenarios can be scripted using either the Python language, or a simple command-based language. Python provides a powerful language for scripting complex analyses, while the command-based language provides an easy-to-learn capability for quickly setting up parameter values, interactively executing runs, and plotting outputs.

Magnolia: Under the Hood

- Magnolia (CSL) models are converted into machine-executable code to enable fast execution
  - CSL model code is translated in Java source code, which is compiled using a Java compiler and linked to Magnolia runtime libraries
  - The generated executable is run on a Java virtual machine, which further compiles and optimizes the simulation executable
  - The translate/compile/optimize process happens extremely quickly, even for large, complex models
- A Java implementation of the CVODE solver is used for fast, accurate solution of ODEs
- Python scripting is provided using the Jython engine, enabling seamless integration of model variables, parameters, functions and output trajectories into the Python scripting environment

Magnolia Features

- An equation-based modeling language which supports representation of systems of ODEs, DDEs, and hybrid discrete/continuous behavior
- A rich set of language operators for constructing models
- Mechanisms for creating reusable model components
- Support for large, complex models and fast execution of simulations enabled by code generation and compilation
- A highly interactive user interface for exploring and understanding model behavior/response
- Advanced scripting capabilities using the CMD or Python languages
- Built-in support for parameter estimation (least-squares or maximum likelihood), local and global (Morris, Sobol) sensitivity analysis, Bayesian parameter estimation using Markov Chain Monte-Carlo analysis (M-H random walk), and uncertainty/variability analysis using standard Monte-Carlo approaches
- Modern code development environment features: syntax highlighting editors, integrated help, code revision tracking, steppable code execution, etc.

System Requirements

- Magnolia can be used on most hardware and operating systems which support the Java Development Kit
  - This includes Windows, macOS, and various distributions of Linux running on x64 hardware
- Magnolia requires < 500 Mb of disk space and runs well in systems with 8 Gb of RAM
- Magnolia is a multi-threaded application and can make use of multi-core processors

Download and Support

- Downloads are available from the Magnolia website (free, but registration required)
  - https://www.magnoliasci.com/register/
- For questions regarding download and installation, email team@magnoliasci.com
- Dedicated technical support is not currently available, but questions directed to the team@magnoliasci.com reflector are generally responded to on a best-effort basis

Installation

- A standard Microsoft Windows executable installer is provided for the Windows operating system
  - Just launch the installer executable and follow the instructions on the UI
- A .pkg “package” file is available for macOS
  - Double-click the .pkg file to start the macOS package installer and follow the instructions on the UI
- A generic Linux (i.e., distribution-independent) .tar.gz file is provided for Linux OS flavors
  - Uncompress the file to a location of your choosing and use the included shell script to start Magnolia
  - Due to variations in Linux distributions, we generally create distro-specific Magnolia builds on a case-by-case basis

Exercise 1.1

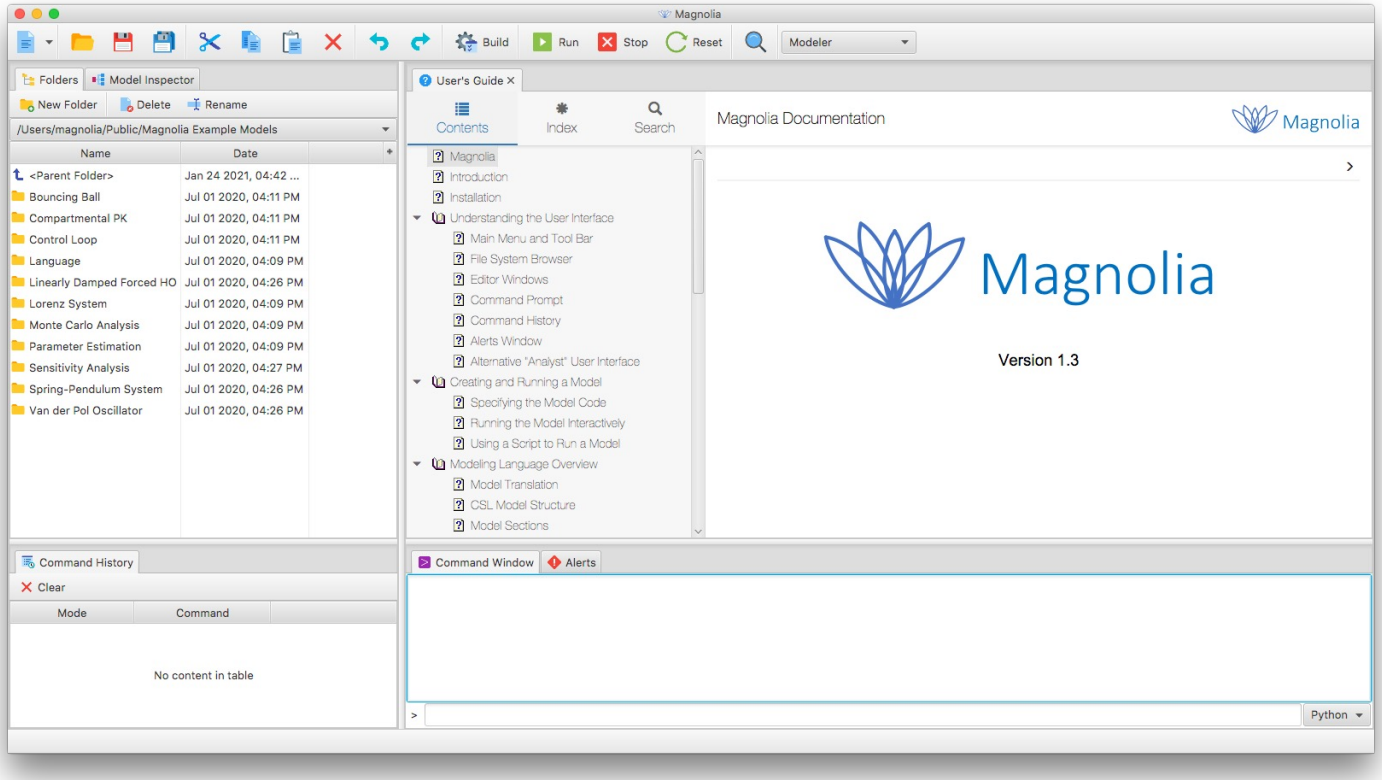
Check for proper installation of Magnolia by starting the application using the method specific to your operating system:

- Mac OS: click the Magnolia icon in the Applications folder or Launchpad screen
- Windows: find the Magnolia icon on the Windows “Start” menu and click it, or double-click the Magnolia icon on the desktop
- Linux: open a shell terminal window and navigate to the folder in which you uncompressed the Magnolia archive file, then start the application by typing the command “./magnolia.sh”

Exercise 1.1 (continued)

Verify that the application starts and the main window is displayed:

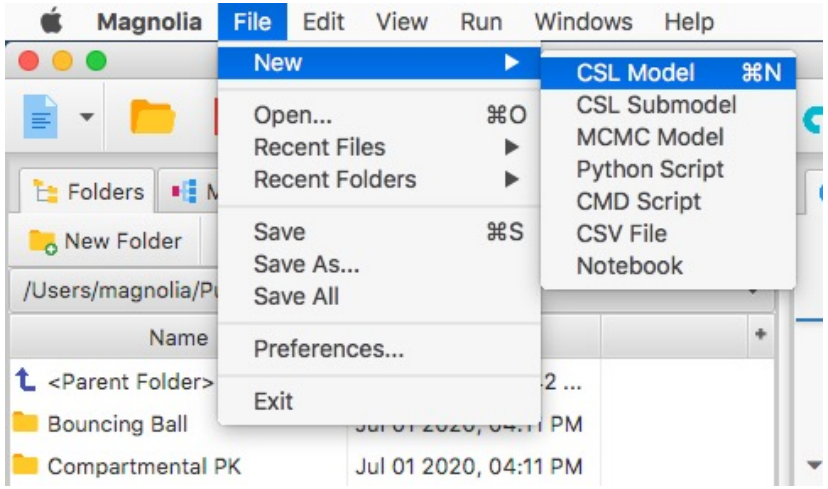
- If this is the first time you've started Magnolia, you should see the User's Guide open in the middle of the application window, and a tree view to the left showing folders containing some example models

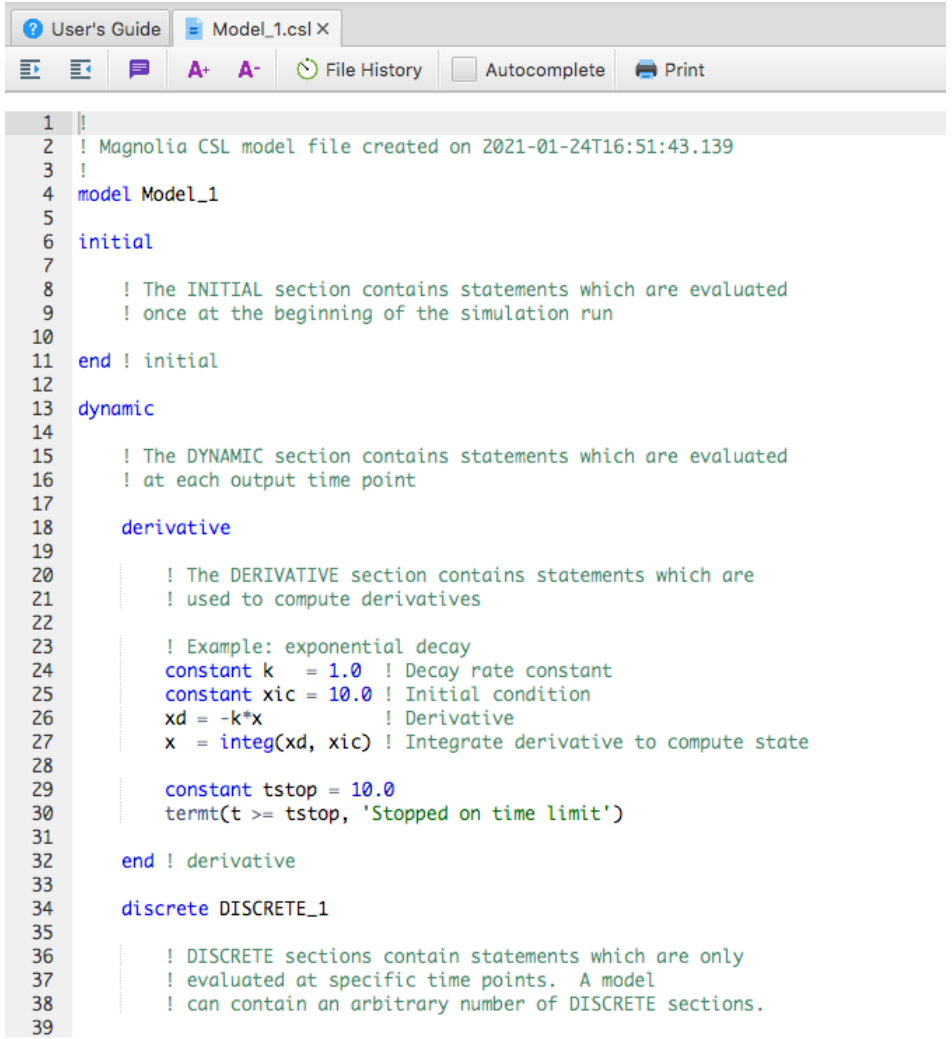


Exercise 1.1 (continued)

Create a new model:

- On the File menu, select New -> CSL Model
- Verify that the code editor opens and displays a “template” example model

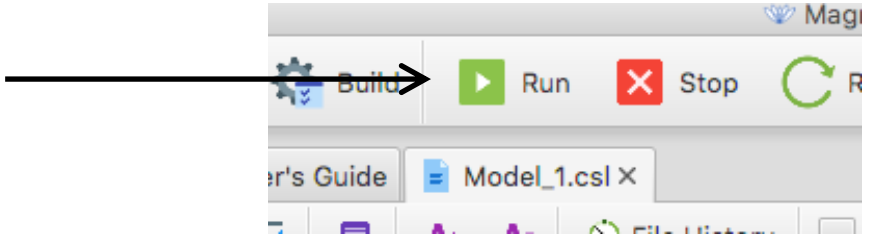


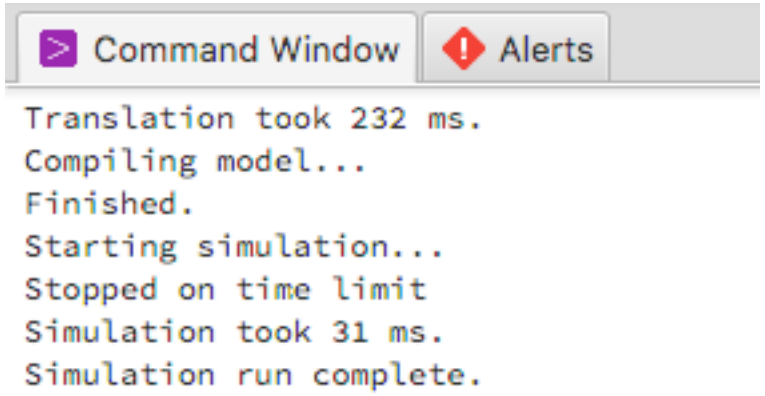


Exercise 1.1 (continued)

Build and run the model you just created:

- Select the “Run” button on the toolbar to build and run the model
  - You’ll be prompted to save the model first; just navigate to a convenient folder and give the model file a name
- Verify that no error messages are displayed in the output window, and that the simulation runs to completion





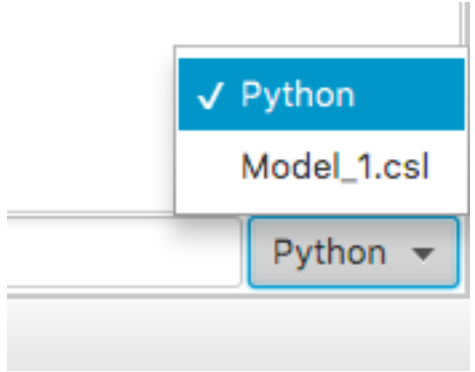
Exercise 1.1 (continued)

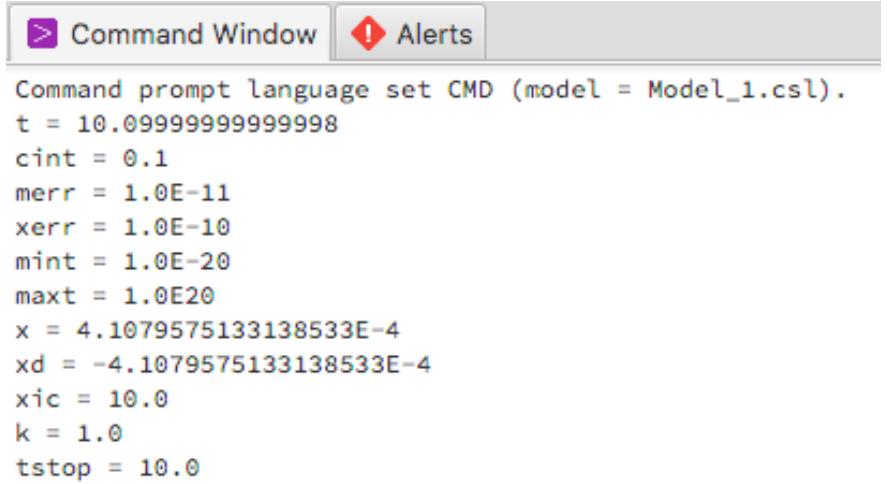
Verify that a (default) plot is displayed:

Exercise 1.1 (continued)

Issue a command at the prompt to ensure the scripting engines are working correctly:

- To the right of the command line near the bottom of the main window, locate the dropdown list and select “Model_1.csl”
  - This tells Magnolia which model you want to control at the command prompt, in the event that you have multiple models loaded at once
- On the command line, type: display @all <enter>
- Verify that the output window displays the values of all model quantities





Exercise 1.1 (continued)

Troubleshooting installation problems:

- If the application fails to start
  - Verify that you're using a compatible operating system: Magnolia requires 64-bit Mac OS, Windows 10, or Linux
  - Verify that your user account has sufficient privileges to run Magnolia (depending on installation folder), and that you have write access to the home folder associated with your user account
- If the application starts, but you can't build or run models
  - Open the Magnolia "About" dialog (main menu -> Help -> About) and make sure Magnolia is using the correct Java JDK by confirming the following lines in the text box:

If any of these settings are incorrect, contact team@magnoliasci.com for additional assistance

```
javafx.version: 8.0.202
java.runtime.name: Java(TM) SE Runtime Environment
java.vm.name: Java HotSpot(TM) 64-Bit Server VM
java.runtime.version: 1.8.0_202-b08
os.arch: x86_64
sun.java.command: org.magnoliasci.MagnoliaModeler
```