ADCompMod
Age/Duration specific Compartment Models
(Version 0.9 α)

Christian Almeder

14th May 2001

Abstract

This paper describes the underlying concept of the Matlab program ADCompMod and the way it can be used to investigate a special type of Mathematical Models by application of numerical methods. The current α-version of ADCompMod is a test version, so it may contain bugs. In order to improve this software please report any bug or comments about missing features via e-mail to

calmeder@eos.tuwien.ac.at

I hope this software is a useful tool for your work and assists your research.

Contents

1 Introduction

1.1 General Description

This software was designed to assist the investigations of age or duration specific dynamical systems with aggregated and non-aggregated feedback components. Except of very simple models, it is hardly possible to do any useful theoretical analysis on such models. Hence, this program offers the possibility to apply numerical methods on such models and compare the results of different experiments. A useful tool for such an analysis is the parameter variation, which is an easy way to gain more insights on the behavior of the system and the sensitiveness of important parameters.

The software is based on a very general model formulation, which allows to use it for a wide range of applications. The user must provide the whole model description in a specific, for the system usable form. Because of the open design of this package, the user can use only parts of the software package in order to build own application which are designed for specific problems.

The whole program was implemented in Matlab programming language using Matlab R12 (version 6). It uses only the main part of Matlab without any toolboxes and it should run on any platform, where Matlab is available. (Till now it is tested on MS Windows 98 and Linux.) The model description has to be supplied in the form of several M-functions. So the whole command set of Matlab can be used to design the user-specific functions. Furthermore the Matlab environment can be used to do post-calculations and analyze the results.

1.2 Formal Model Concept

We consider a system $\mathbf{S} = (S_1, \dots S_n)$ of different states, where $S_i(t, x)$ with $i = 1 \dots n$ are quantities depending on the time t and a second parameter x, which can either be the global age or the duration in this state. In both cases x has the form $x = t + c$, where c indicates the time of birth (age case) or the time of state entrance (duration case). To distinguish between an age specific or a duration specific system, we introduce the age-duration-index

$$\mathrm{ADI} = \begin{cases} 1 & \text{age-specific case} \\ 0 & \text{duration-specific case} \end{cases} \tag{1.1}$$

There are some parameters describing the transitions between the different states, and flows into and out of the system:

$\mu_{ij} = \mu_{ij}(t, x, \mathbf{S}, \mathbf{F})$ denotes the transition from state i to j depending on time t, the parameter x, the state of the system $\mathbf{S}$, and the overall feedback $\mathbf{F}$ of the system.

$\alpha_i = \alpha_i(t, x, \mathbf{S}, \mathbf{F})$ represents the removal or outflow term from the state i.

$\beta_i = \beta_i(t, x, \mathbf{S}, \mathbf{F})$ is the inflow term into the state i.

Hence, the dynamic of the system can be described through

$$\frac{\partial S_i(t,x)}{\partial t} + \frac{\partial S_i(t,x)}{\partial x} = -\sum_j \mu_{ij}\left(t, x, \mathbf{S}, \mathbf{F}\right) - \alpha_i\left(t, x, \mathbf{S}, \mathbf{F}\right) + \mathrm{ADI} \sum_j \mu_{ji}\left(t, x, \mathbf{S}, \mathbf{F}\right) \tag{1.2}$$

$$S_i(t,0) = \int_0^\omega \left[\beta_i\left(t, x, \mathbf{S}, \mathbf{F}\right) + (1 - \mathrm{ADI}) \sum_j \mu_{ji}\left(t, x, \mathbf{S}, \mathbf{F}\right)\right] dx \tag{1.3}$$

$$S_i(0,x) = S_i^0(x) \tag{1.4}$$

The difference between an age and a duration specific system is the way of entering a new state. In the age case the transitions into the new state are included in the partial differential equation (1.2), because the age is a preserved property. But in the duration case the transitions occur in the boundary condition (1.3), because the duration parameter has to be reset.

The feedback component $\mathbf{F}$ describes the actual state of the system with respect to a certain aspect. The general form is:

$$\mathbf{F}\left(t, x, \mathbf{S}\right) = \int_0^\omega \mathbf{f}\left(t, x, x', \mathbf{S}(t, x')\right) dx' \tag{1.5}$$

As indicated, $\mathbf{F}$ is a vector function, so that it is possible to include different types of feedback. A simple example is that the feedback function consists only of the total amount in each state at time t (so that $\mathbf{f} = \mathbf{S}$).

The restriction $x \in [0, \omega]$ in this system is chosen due to the numerical implementation and easier analysis, but in real applications it is very seldom to have an infinite age or duration range. (e.g. Epidemiological models are often described using an infinite age range, but it is no restriction, if the age range is cut off at 150 years.)

1.3 Examples for applications

Epidemiological models (age specific case)

One of the simplest epidemiological models fitting in the scope of our model description is the S-I-S model (see [1]). The model equations are as follows:

$$\frac{\partial s(t,a)}{\partial t} + \frac{\partial s(t,a)}{\partial a} = -\lambda(t,a)s(t,a) - \mu(a)s(t,a) + \gamma(a)i(t,a) \tag{1.6}$$

$$\frac{\partial i(t,a)}{\partial t} + \frac{\partial i(t,a)}{\partial a} = +\lambda(t,a)s(t,a) - \mu(a)i(t,a) - \gamma(a)i(t,a) \tag{1.7}$$

$$\lambda(t,a) = k_0(a)i(t,a) + \int_0^\infty k(a,a')i(t,a')da' \tag{1.8}$$

$$s(t,0) = \int_0^\infty \beta(a)s(t,a)da \quad i(t,0) = 0 \tag{1.9}$$

$$s(0,a) = s_0(a)da \quad i(0,a) = i_0(a) \tag{1.10}$$

where $s(t,a)$ represents the group of susceptibles, $i(t,a)$ the infected, $\lambda(t,a)$ the infection rate (or transition rate from s to i, $\mu(a)$ the death rate in both groups, and $\gamma(a)$ the cure rate (or transition rate from i to s).

Age-structured population model with applications to social services planning (duration specific case)

This model is described in the paper of Haurie et al. [2]. The population dynamics are described as follows:

$$\frac{\partial y_i(t,\tau)}{\partial t} = -\frac{\partial y_i(t,\tau)}{\partial \tau} - \sum_{j=1}^{m} Q_{ij}(t,\tau)y_i t,\tau, \tag{1.11}$$

$$y_i(t,0) = b_i(t) + \int_0^\infty \sum_{j=1}^{m} Q_{ij}(t,\tau)y_j(t,\tau)d\tau, \tag{1.12}$$

$$\text{for } i \in \{1,\ldots,m\}$$

In this case $y_i(t,\tau)$ is the population density of individuals in state j at time t, who have been in this state since time $t-\tau$. $Q_{ij}(t,\tau)$ the transition rate from state i to state j and $Q_{ij}(t,\tau)y_i(t,\tau)$ is the corresponding expression to μ_{ij} in the general formulation.

Initiation for illicit drugs (mixed case)

This example shows how to combine age and duration specific aspects in one model. For that purpose it is necessary to split up the different states into two groups, the age specific and the duration specific ones. Furthermore, it is necessary to have transition from age specific to duration specific groups, but not the other way, because after entering an duration specific state, the age parameter is lost.

Remark. *This type of model is not considered explicitly in the actual version of ADCompMod. But if the transitions for the duration specific states are included in the inflow functions, this type can be implemented, too.*

The model for the initiation of illicit drugs consists only of to groups, the non-user population $P(t,a)$ (age specific) and the addicts $A(t,d)$ (duration specific). The following equations describe the model dynamics

$$\frac{\partial P(t,a)}{\partial t} + \frac{\partial P(t,a)}{\partial a} = -\mu\left(t,a,R(t,a)\right)P(t,a) \tag{1.13}$$

$$\frac{\partial A(t,d)}{\partial t} + \frac{\partial A(t,d)}{\partial d} = -\alpha(t,d)A(t,d) \tag{1.14}$$

$$P(t,0) = k$$

$$A(t,0) = \int_0^{\omega_a} \mu\left(t,a,R(t,a)\right)P(t,a)da \tag{1.15}$$

$$P(0,a) = P_0(a) \quad A(0,d) = 0 \tag{1.16}$$

$$\tag{1.17}$$

The rate of initiation is

$$\mu(t,a) = \mu_a \Phi\left(R(t,a)\right)\Psi(w(t,a)) \tag{1.18}$$

with

$$R(t,a) = \frac{\int_0^{\omega_d} m(a,d)A(t,d)dd}{\int_0^{\omega_a} P(t,a')da'} \tag{1.19}$$

2 Numerical Methods

Using the method of lines the partial differential equation (1.2) is transformed into a system of ordinary differential equations by discretizing the age (duration) and substituting the partial derivatives w.r.t. x by finite approximations. Hence, instead of a continuous age (duration) variable x we have now a finite series of age (duration) classes $(x_k)_{k=0..N_x}$. [3]

So instead of a state vector $\mathbf{S}$, we now have a state matrix $\mathbf{S^{(x)}}$ where each row represents the the different age (duration) classes for a single state. For each element of this matrix a ordinary differential equation can be formulated:

$$\begin{aligned}\frac{dS_i^{(x_k)}(t)}{dt}t = -\boldsymbol{\Delta_x}\left\{S_i^{(x_k)}(t)\right\} - \sum_j \mu_{ij}\left(t, x_k, \mathbf{S}^{(x_k)}, \mathbf{F}^{(x_k)}\right) - \\ -\alpha_i\left(t, x_k, \mathbf{S}^{(x_k)}, \mathbf{F}^{(x_k)}\right) + \mathrm{ADI}\sum_j \mu_{ji}\left(t, x_k, \mathbf{S}^{(x_k)}, \mathbf{F}^{(x_k)}\right)\end{aligned} \tag{2.1}$$

where $\boldsymbol{\Delta_x}$ denotes an appropriate finite difference operator w.r.t. x.

Analogous transformation must be applied to the boundary conditions and the feedback function. The integral in the definition of the feedback function must be approximated by a quadrature formula using the values at the x_k.

This ODE system can be solved now with any standard algorithm for ordinary differential equations. ADCompMod provides all Matlab ODE solvers for that purpose, but in the usual case the a Runge-Kutta-Fehlberg algorithm is suitable.

A Using ADCompMod

A.1 Installation

The current distribution consists of the following files:

adcompmod.m
: adcompmod program (main function)

adcompmod.fig
: Matlab figure data for main window

adcompmod_calcconfig.cfg
: config file for calculation parameters

adcompmod_odefile.m
: definition of ODE-system (results from the discretized PDEs)

private/adcompmod_privat_calcbar.m
: function for handling a user abort of calculation

adcompmod_privat_calcconfig.m
: function for setting the calculation parameters

adcompmod_privat_calcconfig.fig
Matlab figure data for calculation parameter window

adcompmod_privat_calcprog.m
function for handling the calculation progress

adcompmod_privat_calcprog.fig
Matlab figure data for calcuation progress window

adcompmod_showres.m
function for visualizing the resutls

adcompmod_simulate.m
function for performing simulation runs

private/differentiationop.m
function for the discrete differential operator

feedback-def.mdef
function definition for user-provided feedback function

help.pdf
this paper

inflow-def.mdef
function definition for user-provided inflow function

initfun-def.mdef
function definition for user-provided initializing function

private/integralop.m
function for the discrete integration operator

outflowfun-def.mdef
function definition for user-provided outflow function

outfun-def.mdef
function definition for user-provided output function

private/rawreadconfigfile.m
function for reading raw data from any ASCII configuration file

private/readcalcconfigfile.m
function for reading calculation parameters from ASCII configuration file

private/readconfigfile.m
function for reading model configuration from ASCII configuration file

transfun-def.mdef
function definition for user-provided transition function

private/writecalcconfigfile.m
function for writing calculation parameters to ASCII configuration file

private/writeconfigfile.m
function for writing model configuration to ASCII configuration file

All files must be in one directory, except the private functions which reside in the *private* subdirectory. The main directory (not the private subdirectors) must be included in the Matlab path. In order to view the help file from within the program the Adobe Acrobat Reader must be installed on the system. (A free version can be downloaded from *http://www.adobe.com/*.)

The program can now be started by typing *adcompmod* in the Matlab command window.

A.2 Main Window

A.2.1 The Menu *File*

New
sets all parameters to default values and sets all user-provided functions as undefined.

Open...
opens a model data file (Matlab .mat file). This file type can also contain results.

Save
saves the actual model data and results to the actual data file.

Save as...
saves the actual model data and results in a model data file (Matlab .mat file)

Import Textfile...
imports model data from an ASCII model configuration file. This file type contains no results.

Export Textfile...
exports actual model data to an ASCII model configuration file. This file type contains no results.

Exit
closes the program.

A.2.2 The Menu *Calculation*

Simulate
performs a single simulation run.

Variation Run
performs several simulation runs with different parameter sets.

Show Results
visualizes the results in separate window.

Simulation Parameters...
changes calculation parameters. For more details see section A.3

A.2.3 The Menu *Help*

This item starts the Adobe Acrobat Reader and displays this manual.

A.2.4 Number of States

Here the number of states (n in section 1.2) used in the model is selected.

A.2.5 Age / Duration – Model

Here the Age-Duration-Index ADI is set (see section 1.2).

A.2.6 Age (Duration) – Limits

In this section the limits for the age (duration) parameter are set. The lower and upper boundaries and the number of classes for the discretization have to be set. At the bottom of this field the difference between the age (duration) classes are is displayed.

A.2.7 Time

The start and end time for the simulation run must be set here.

A.2.8 Functional Model Definition

In this field the user-provided functions are handled. There are 3 buttons for each function:

New
: This button opens a dialog to specify a location and a name for the user-specified function (Matlab .m file). The function definition is copied to the given destination and the file is opened for editing.

Select
: This button opens a dialog for choosing a m-function.

Edit
: This button opens the Matlab editor for editing the user-provided function.

A.2.9 Control Buttons

There are 4 buttons for a quick control of the program:

Simulate
: see menu *Calculation*

Variation Run
: see menu *Calculation*

Show Results
: see menu *Calculation*

Exit
: closes the program.

A.3 Calculation Parameters Window

This windows allows to set the global calculation parameters.

A.3.1 Approximation Order

Here the approximation formulas for the derivatives w.r.t. age (duration) and for the intregration of the feedback component can be set. It makes no sense to set the order for the integration higher than that for the derivatives. In order to avoid numerical oscillation it is useful to use upwind finite differences for the approximation of the derivatives.

A.3.2 Accuracy

In this frame the Matlab ODE solver can be chosen (see the Matlab documentation for more information). The relative and absolute tolerance determine the accuracy of the results of the ODE solver and the maximal time step limits the time step for one integration step.

A.3.3 Avoid Negative Values

Click this checkbox, if you want to avoid negative state values. Sometimes it is possible that the results contain negative values for the states, though those negative values make no sense and are no right result for the model equations. This fact is due to some numerical errors. If this option is chosen, all negative values are cut off during the simulation.

B Known Problems and Bugs

B.1 Model files cannot be moved to other directories

Reason: The full path to the user-defined functions is stored in the *.mat and *.cfg files.

Solution: Move the all model files to the new directory; open the model file; select all user-defined functions in the new directory.

B.2 Calculation configuration file

When saving the "Simulation Parameters..." (CALCULATION menu) the configuration file is save into the actual (matlab) directory, but for the calculation the configuration file in the root path of ADCompMod is used.

Reason: When the "Simulation Parameters..." - dialog is opened the program looks for a configuration file in the actual directory. If there is no such file, it takes the configuration file from the root directory. But the Saving button creates always a configuration file in the actual directory.

Solution: Before changing the Simulation Parameters change in the MATLAB Command Window to the root directory of ADCompMod.

B.3 The Help Menu works only on Unix machines (with Adobe Acrobat Reader installed)

Reason: ADCompMod calls the program *acroread* without any path information. Usually on Unix machines, the Adobe Acrobat Reader can be started this way, but the MS Windows version of the reader has a different name and is usually not in the systems search path.

Solution: At the moment the only way is to locate in the file *adcompmod.m* the *dos*-command and fill in the right path and name for the acrobat reader.

References

[1] S.N. Busenberg, M. Iannelli, H.R. Thieme. Global behavior of an age-structured epidemic model. *SIAM J. Math. Anal.*, 22(4):1065–1080, 1991.

[2] A. Haurie, S. Sethi, R. Hartl. Optimal control of an age-structured population model with applications to social services planning. *Large Scale Systems*, 6:133–158, 1984.

[3] W.E. Schiesser. *The Numerical Method of Lines*. Academic Press, 1991.