Manual for the RDM library

RDM: Reaction Diffusion with Matlab

Yuval Zelnik

April 23, 2018

RDM is a library built with the goal of analyzing and simulating Reaction-Diffusion and similar models, using the software Matlab. It is meant to give a simple and intuitive way of handling spatial non-linear models, running various simulations of them, and analyzing and presenting their results.

1 Basics

1.1 Main data structures

RDM is built around 3 main data structures, termed Vs, Ps, and Es. These are used for holding the information about the variables of the state of the model (Vs), for holding information about the model parameters (Ps), and for holding any other auxiliary information (Es). A fourth common data structure is used for holding bifurcation information, often using the shorthand bf. Following is a short description of each of these 4 data structures.

1.1.1 Vs - variable information

This structure is used to hold and convey information about the variables of a given model. Here, the term variables denotes entities of the model that are affected and changed by the dynamics of the model, as opposed to parameters which are imposed on the model. In the context of RDM we consider spatial models, and therefore a state of a model is defined by the values of its variables (one or more) across space. Vs is made of a single matrix, where the first dimension (vertical) is used for different locations in space, and the second dimension (horizontal) is used for different variables. The third dimension is used to hold more then one state, when relevant.

1.1.2 Ps - model parameters

The purpose of the Ps structure is to hold all information related to a specific model. This includes the functions that define its dynamics, the parameters that it includes

(which are used by these functions), and other general features of the model such as size and boundary conditions. The exact definition of which parameter or property fits into this structure, as opposed to the Es one (see next subsection) is not always clear-cut, and largely depends on common practice and the user's preference. A typical structure of Ps is given in Table 1, which includes the typical (but largely insignificant) order of fields.

field	meaning/purpose
functions	
LocFunc	defines the local (typically nonlinear) part of the model
SpaFunc	defines the spatial (often linear) part of the model
IntegFunc	defines the integration scheme
model parameters	
(prm 1)	some parameter of the model
(prm 2)	another parameter of the model
...	(number of parameters depends on model)
Ds	(coefficients of diffusion/spatial derivatives)
general properties	
VarNum	number of variables in model (= second dimension of Vs)
Nx	number of points in grid (on main axis)
Ny	number of points in grid (on secondary axis, if applicable)
Lx	physical size (on main axis)
Ly	physical size (on secondary axis, if applicable)
Bc	boundary conditions (if applicable)
Net	network structure (if applicable)
Locs	location of points (if applicable)

Table 1: Typical structure of Ps.

1.1.3 Es - auxiliary parameters

The Es structure is meant to hold a wide array of information that does not fit into other places and is used by the various functions of RDM. In particular it can contain information about the simulation scenarios to run, the type of analysis to make on the results of these simulations, the way to present results, and many more features. As such it is the most loosely defined structure, which can easily contain dozens of parameters or no information at all. See Tables 4-8 for an exhaustive list of the different fields of the Es structure.

1.1.4 bf - bifurcation information

A bf structure is a two-dimensional matrix, or a cell-array of several such matrices. It hold information used to plot bifurcation diagrams or similar plots, and in general it contains a more limited information of each state (when compared to Vs), but for a wide range of different parameter values. Each row in the matrix is used for one state, and each column corresponds to either a parameter, or some measure of

a state for these parameters (for example, the average or maximal value of one of the variables).

1.2 Main function types

In RDM there are three main types of interface functions (meant to be used directly by the user), 7 types of generic functions (mostly used indirectly), and some other functions that do not clearly fit into these two categories. The interface functions are used for running simulations, plotting results and preforming simple analysis (see Table 2). The generic functions include functions for time integration, functions for defining models, functions to preform tests and calculations, and so forth (see Table 3). Note that all generic functions, and most interface functions (all except for plotbf and plotps) accept as the first 3 inputs the structures {Vs,Ps,Es}. All interface functions (and many generic functions) also allow for an "online update" of parameters, by writing in more inputs after these initial ones.

function	output	purpose
Simulation functions		
runsim	final state	time integration with pre-defined runtime
run2ss	final state, history, runtime	time integration until reaching a steady-state
runframes	states, history	time integration to series of time points
runnewt	state, iteration #, residual	find solution with Newton-Raphson method
runflow	state, bif. information	run a list of functions in sequence
runfind	state, bif. information	run an iterative search on condition
runcont	state, bif. information	continue solution along a branch
runpar	state(s), bif. information	run multiple parallel simulations
Plotting functions		
plotst	(figure handle)	plot a state across space
plotwf	(figure handle)	plot a state across space and time
plotbf	(figure handle)	plot a bifurcation diagram
plotps	(figure handle)	plot a parameter space
Analysis functions		
getode	(non spatial) state	find a non-spatial solution
gettest	test result	run a test function on a given state
getrhs	right-hand-side vector	return the right-hand-side of the model
getjac	Jacobian matrix	return the Jacobian of the model

Table 2: Main interface functions, of three types: simulation, plotting and analysis

function	output	purpose
Model functions: define local (L) and spatial (S) terms		
L_Log	rhs	local terms for Logistic equation
L_GS	rhs	local terms for Gray-Scott model
S_RD	rhs	spatial terms for reaction-diffusion
S_BE	rhs	spatial terms for Burger's equation
Integration functions (I): move forward in time		
I_FDE	state	explicit finite-difference Euler scheme
I_PSRD	state	pseudo-spectral method (for reaction-diffusion)
Test functions: used on a state (T) or on multiple states (C)		
T_Var	variance	calculate the variance in space of a state
T_L2Norm	norm	calculate the L2norm of a state
C_AvgTest	average	average out a test result over multiple time points
C_ReachVal	time	find the time to reach a certain value
Modifying functions (M): new state based on an old one		
M_InitMixSt	state	initialize a state by mixing two given states
M_AddNoise	state	add white noise to a state
Update functions (U): setup or update configuration		
U_SetupRndSpace	Vs,Ps,Es	setup random heterogeneity of parameter in space
U_TakeOutLinks	Vs,Ps,Es	remove links from a network structure

Table 3: Examples of RDM's generic functions: seven categories, recognized by the first letter of the function's name (L,S,I,T,C,M,U).

1.3 Simple demonstration

A simple and short demonstration of how one may use RDM is given by the demo file demo_intro.m. To get an idea of what the different interface functions do, look at demo_interfacefuncs.m. To learn the basic idea of the PDE differentiation/integration, look at demo_pdebasics.m.

2 More details

2.1 List of Es fields

Given below is a list of Es fields, which includes almost all the fields that have been implemented and used. Note that many of these fields are used by multiple functions, and thus their naming choice is meant to be relatively general and also easy to guess. In particular, the names of fields are made of one or more words (or shorthands), each starting with a capitalized letter, with the rest of it uncapitalized. The naming convention attempts to be clear yet not too long, and many words/shorthands are used many times, such as "Prm" (parameter), "Thresh" (threshold), "Func" (function), and "Ts" (time-step).

field	meaning/purpose
general and common fields	
VarInd	variable(s) to show and/or affect
StSmall	baseline size of small change in variable
NonNeg	flag to insure that variables are not negative valued
BfFields	columns in bf structure to read information from
RandSeed	randomization seed to use
JacNum	use numerical Jacobian? (if analytical is not given)
TestFunc	main function used to measure each given state
OlDraw	plot state while running simulation (online)?
PlotFunc	which function to (online) plot state
Verbose	print out information during run?
NoWarning	do not show warnings?
time related	
TsSize	default size of integration time-step
TsMode	mode for changing (or not) the integration time-step
TimeMin	minimum size of integration time-step
TimeMax	maximum time of integration time-step
TsNum	default number of time-steps to take at a time
TimeDst	total time to run a simulation
TimeMin	minimum time of a simulation
TimeMax	maximum time of a simulation
EvalTsPrm	parameters for time-step evaluation

Table 4: List of fields in the Es structure, part 1.

field	meaning/purpose
used by different simulation functions	
BfPrm	List of bifurcation parameters to use
BfRange	range of each bifurcation parameter to consider
BfSmall	baseline size of small change in bifurcation parameter value
BfVal	list of different values for the bifurcation parameters
SsThresh	threshold for defining steady-state
used by run2ss	
SsCheckFunc	function for defining steady-state
used by runframes	
Frames	which frames (time points) to run to?
FramesChoice	return state-data on which time points?
RecurFunc	recurrent function to run at specific time points
RecurFrames	when to run this recurrent function
DynPrm	parameter(s) to change dynamically throughout run
DynVal	values for changed parameter(s)
RepDynVal	repeat dynamical values more than once?
used by runnewt	
InverseMatrix	use inverse matrix for Newton loop?
NewtonRate	slower convergence iterations?
MaxNewtLoop	maximum number of iteration loops
used by runflow	
FuncList	which functions to run in sequence
FuncSpec	designation of each function (auto-detected)
TestMultSt	work-around to test more than one state
MergeBfData	merge bf data together (used by runpar)
used by runfind	
FindFunc	function used in the search algorithm
FindVal	goal value for search algorithm
PrmLim	limits of search region
FindOnlyBf	work-around to return only bf information
used by runcont	
ContMin	minimum good value to go on with continuation
ContUpdate	update function at every iteration
SsFunc	function to define steady-state for continuation
BfMaxDiff	stop continuation if norm's value jumps too far
used by runpar	
RunFunc	function to run for each parameter set
RunsChoice	sublist of parameter sets to run

Table 5: List of fields in the Es structure, part 2. This list contains fields associated with simulation functions

test functions	
UnfThresh	threshold for defining a state uniform
CsiThresh	threshold for stability evaluation by integration
SegThresh	threshold for defining a segment/region
CompFft	list of FFT components to return
LsaThresh	threshold(s) for linear stability analysis
LsaPer	preform linear stability analysis on several periods?
EigNum	How many eigenvalues to request?
UseEig	use eig instead of eigs for linear stability analysis?
AvoidErrors	try to avoid errors using the try mechanism?
calculation functions	
TestFields	which test fields to do calculation on?
CalcRange	which time/frame range to do calculation on?
ReachVal	values(s) to check when they are reached
CalcSpeedPrm	parameters for speed calculation
FlipThresh	threshold for flip calculation
multiple tests/calcs	
TestList	list of test functions
TestOuts	which output to use of test functions
CalcList	list of calculation functions
CalcOuts	which output to use of calculation functions
initialization	
InitFunc	function to use for initialization (I_Init...)
InitPrm	parameter(s) to use for initialization
OdeInit	initialize by running an ode (no space) simulation?
OdeTime	default time for running an ODE (for initialization)
CorrSize	correlation size for initialize random values with correlation
other modiying functions	
ModPrm	parameters for modifying a state
PopThresh	parameters for imposing a population threshold
ShiftPrm	parameters for shifting a state in space
StNoise	parameters for adding noise to a state
RegPrm	parameters for changing regions
setup functions	
PppPrm	parameter for setup of point processes
RescaleFactor	factor(s) to rescale by
RescalePow	power to rescale by
RescalePrm	name(s) of parameter(s) to rescale
RndNetFunc	function to create random network
RndNetPrm	parameters to create random network
RndSpacePrm	name(s) of parameter(s) to randomize in space
RndSpaceVal	values for randomization in space

Table 6: List of fields in the Es structure, part 3.

plotting states	
StAxis	boundaries for values of state variable to show
StInd	which states to show (if more than one)
StLineWidth	width of curves in 1d plots
St1Color	color of each variable in 1d plots
St2Color	color palette for 2d plots
St2Colorbar	add colorbar to 2d plots?
St2Interp	interpolate grid in 2d plots? (for smoother image)
St2Angle	rotate state for plotting?
PlotBare	plot state without axes and such ?
plotting bifurcation	
BfColor	color of each curve in bifurcation plots
BfStyle	line style of curves in bifurcation plots
BfPhases	values of different designations of curves
BfLineWidth	width of curves in bifurcation plots
BfFilter	columns in bf structure to use for filtering out
BfSmooth	smoothing coefficient for bifurcation plots
BfLogColor	use log values for parameter-space plot?
SubPlot	specify layout when plotting several panels
RephaseMode	mode of re-phasing bf data
file related	
FileOut	filename to write to after run
WriteSt	save also state info, or just bif. one?
WriteFreq	write every how many runs?
BfOut	bif. file to write to

Table 7: List of fields in the Es structure, part 4.

external functions (auto, p2p, etc.)	
Auto	mode of use auto files
MultPeriod	use multiple periods for analysis
BadText	try to deal with bad text in auto files?
TransDef	parameters for transforming variables in p2p
VideoQuality	video quality definition when writing video file
VideoSpeed	video speed definition when writing video file
TitlesFrames	when to put titles, when writing video file
TitlesText	which titles to show, when writing video file
internal fields (not for direct use)	
JacMode	flag for asking for Jacobian or right-hand-side
InitActive	flag to denote that initialization is under way
CellData	place to put auxiliary data
PrmInCell	are parameter values stored in cells?
PrmList	generic list of parameters
SetupMode	we are in setup mode, do not go into loop
PartsCollected	which parts of parallel run did we find?
SpaMatUse	spatial matrix is being used?
RecurFramesExtra	holder for recurrent frames

Table 8: List of fields in the Es structure, part 5.