オープンソース数値解析ソフトによる偏微分方程式解法の基礎調査

狙い

(1) １次元でも良いから、汎用的な方程式を解析により解きたい

(2) Salome-mecaは、Wizardsにない動解析の敷居が高い

(参考) 個人的な意見：Excelは、全体を見通すことが難しく、遅い [2]

調査結果の概要

R言語に微分方程式関係のプラグインソフトを追加し、デモのコードやネットでの活用例を参考にすれば、汎用的な解析ができそう。

(1) R言語＋RStudio＋cran（微分方程式関係のパッケージ[3]～[6])

例を参考にして修正すれば、簡単なモデルを解くことができた

(2) 他のソフトについて（情報が少ない）

Scilab＋MmodD、 OpenModelica（操作性が良くないと感じた）

1. R言語 について（GNU Projectの一つ、GPLライセンス）

- オープンソースの統計解析向けプログラミング言語(Wikipedia　より)
- 動作環境：Linux、Windows（x86_64-w64-mingw32/x64）、MacOS
- Linuxではパッケージをソースからビルドするので、コンパイラ設定を変えれば高速化できそう（特にSandy Bridge以降、極めるならR-baseもソースからビルド）
- 膨大な数のパッケージがある（オフィシャルなcranに4000?、バグがあるかもしれないが、githubに開発者の最新版がある）
- クラウドコンピューティング対応 [8]
- 拡張性：C/C++等の関数組み込み、Perlやjavaによるデータ入出力の自動化
- 日本語の書籍の数が多い（一式10冊、20冊のものもある）
- Debianのパッケージリストの1項目　https://packages.debian.org/jessie/

GNOME
The GNOME desktop environment, a powerful, easy to use set of integrated applications.

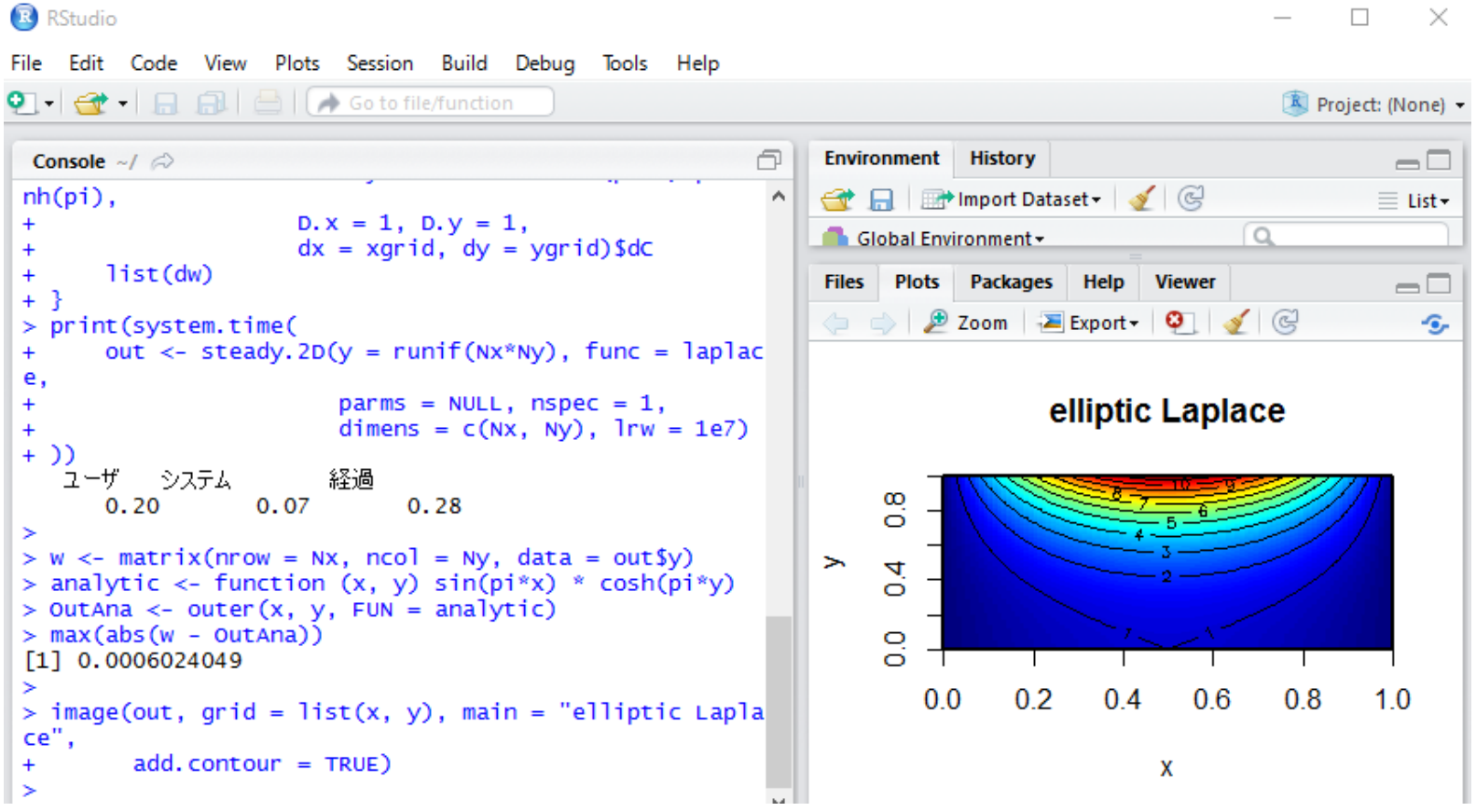
GNU R
Everything about GNU R, a statistical computation and graphics system.

GNUstep
The GNUstep environment.

Graphics
Editors, viewers, converters... Everything to become an artist.

RStudioについて

- Rのための統合環境（パッケージのインストール、ロード、アップデートが容易）
- 動作環境：Windows、MacOS、Ubuntu/Debian、Fedora/Redhat/openSUSE
- 文法アシスト機能付きのエディタがあり、入力が楽になる （小生は、Rのアシスト機能付きのNotepad++でコードを作成し、ctrl+ac, ctrl+vでコピペ）
- オープンソースエディション（無償のDesktop, ローカルPC用、AGPL v3）と商用ライセンス（Server, サーバやクラウド用）がある。



CRAN Task Views https://cran.r-project.org/web/views/

Bayesian	Bayesian Inference
ChemPhys	Chemometrics and Computational Physics
ClinicalTrials	Clinical Trial Design, Monitoring, and Analysis
Cluster	Cluster Analysis & Finite Mixture Models
DifferentialEquations	Differential Equations
Distributions	Probability Distributions
Econometrics	Econometrics
Environmetrics	Natural Language Processing
ExperimentalDesign	Numerical Mathematics
Finance	Official Statistics & Survey Methodology
Genetics	Optimization and Mathematical Programming
Graphics	Analysis of Pharmacokinetic Data
HighPerformanceComputing	Phylogenetics, Especially Comparative Methods
MachineLearning	Psychometric Models and Methods
MedicalImaging	Reproducible Research
MetaAnalysis	Robust Statistical Methods
Multivariate	Statistics for the Social Sciences
	Analysis of Spatial Data
	Handling and Analyzing Spatio-Temporal Data
	Survival Analysis
	Time Series Analysis
	Web Technologies and Services
	gRaphical Models in R

CRAN Task ViewsのDifferential Equations / PDE部分

Partial Differential Equations (PDEs)

PDEs are differential equations in which the unknown quantity is a function of multiple independent variables. A common classification is into elliptic (time-independent), hyperbolic (time-dependent and wavelike), and parabolic (time-dependent and diffusive) equations. One way to solve them is to rewrite the PDEs as a set of coupled ODEs, and then use an efficient solver.

The R-package ReacTran provides functions for converting the PDEs into a set of ODEs. Its main target is in the field of ''reactive transport'' modelling, but it can be used to solve PDEs of the three main types. It provides functions for discretising PDEs on cartesian, polar, cylindrical and spherical grids.

The package deSolve contains dedicated solvers for 1-D, 2-D and 3-D time-varying ODE problems as generated from PDEs (e.g. by ReacTran).

Solvers for 1-D time varying problems can also be found in the package deTestSet.

The package rootSolve contains optimized solvers for 1-D, 2-D and 3-D algebraic problems generated from (time-invariant) PDEs. It can thus be used for solving elliptic equations.

差分法

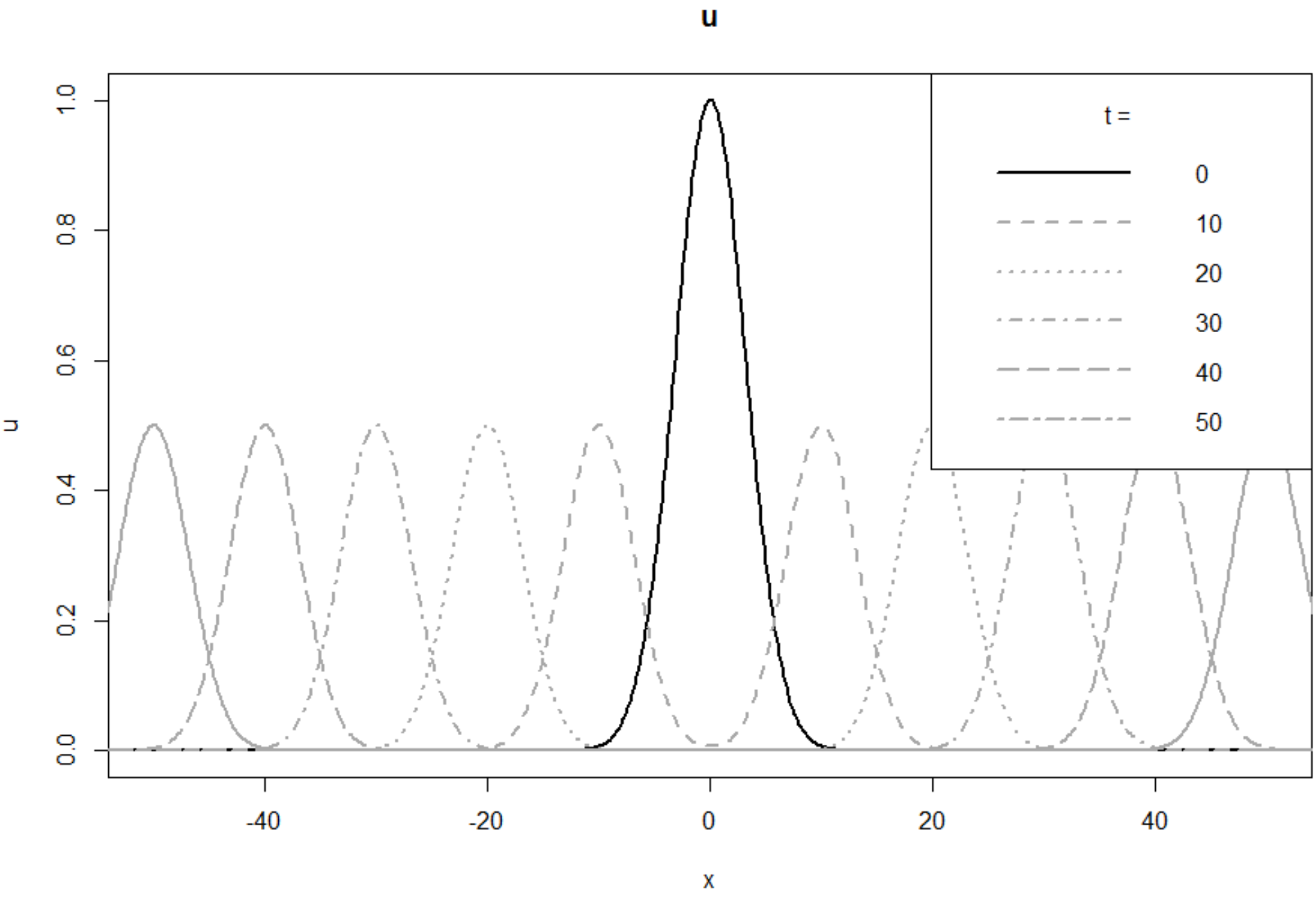
Note that, to date, PDEs in R can only be solved using finite differences. At some point, we hope that finite element and spectral methods will become available.

https://cran.r-project.org/web/views/DifferentialEquations.html

波動方程式のRコード　文献[5]の例

```
#### 波動方程式
##        https://cran.r-project.org/web/packages/diffEq/vignettes/PDEinR.pdf
library("deSolve")
library("ReacTran")
dx    <- 0.2
xgrid <- setup.grid.1D(x.up = -100, x.down = 100, dx.1 = dx)
x     <- xgrid$x.mid; N     <- xgrid$N
lam   <- 0.05; uini  <- exp(-lam*x^2); vini  <- rep(0, N); yini  <- c(uini, vini)
times <- seq (from = 0, to = 50, by = 1)
wave <- function (t, y, parms) {
  u <- y[1:N]
  v <- y[(N+1):(2*N)]
  du <- v
  dv <- tran.1D(C = u, C.up = 0, C.down = 0, D = 1,
  dx = xgrid)$dC
  return(list(c(du, dv)))
}
out <- ode.1D(func = wave, y = yini, times = times,
  parms = NULL, method = "adams",
  dimens = N, names = c("u", "v"))
u <- subset(out, which = "u")
analytic <- function (t, x)
0.5 * (exp(-lam * (x+1*t)^2 ) +exp(-lam * (x-1*t)^2) )
OutAna <- outer(times, x, FUN = analytic)
max(abs(u - OutAna))

outtime <- seq(from = 0, to = 50, by = 10)
matplot.1D(out, which = "u", subset = time %in% outtime,  grid = x, xlab = "x", ylab = "u", type = "l",
      lwd = 2, xlim = c(-50, 50),  col = c("black", rep("darkgrey", 5)))
legend("topright", lty = 1:6, lwd = 2, col = c("black", rep("darkgrey", 5)),  title = "t = ", legend = outtime)
```

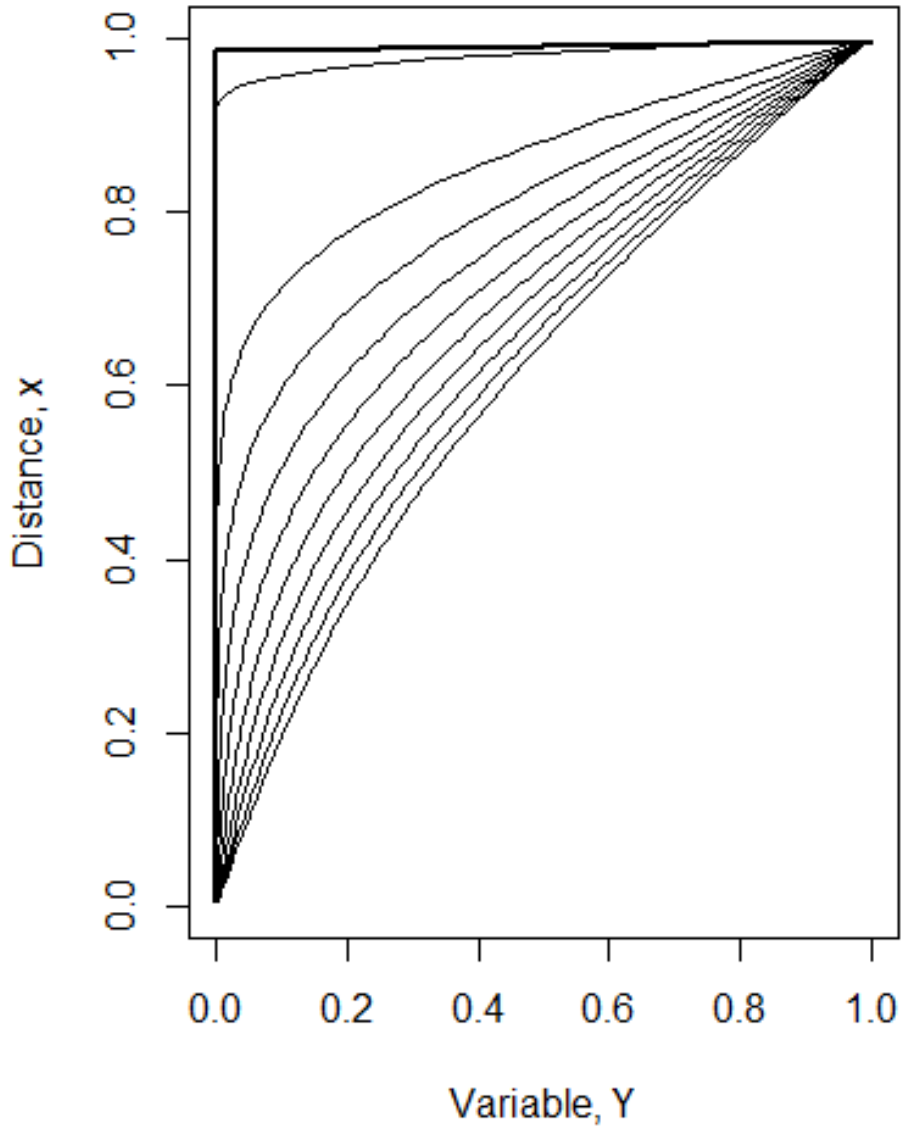
波動方程式の出力グラフ

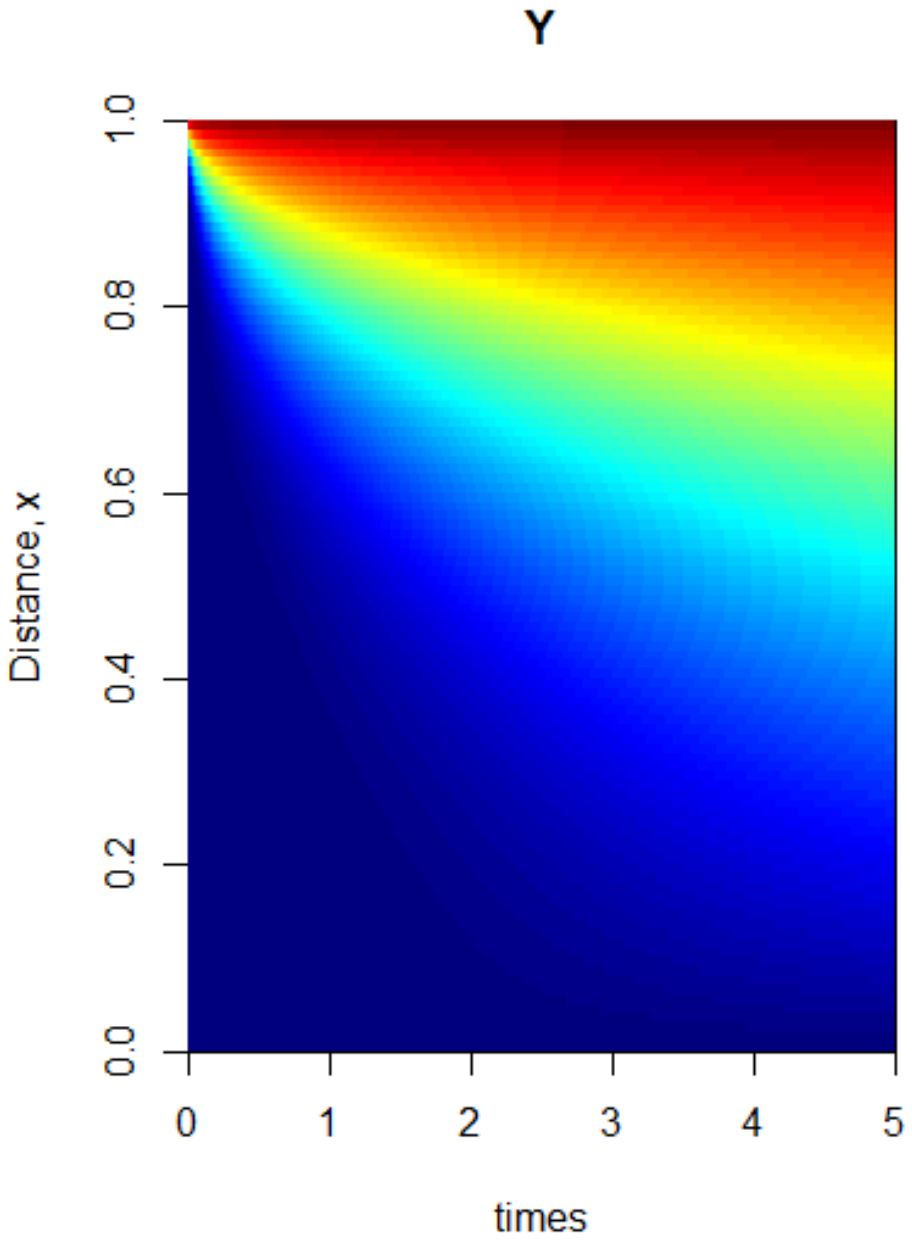


拡散方程式のRコード　文献[5]の例を境界条件変更

```
#### 熱拡散方程式
##        https://cran.r-project.org/web/packages/diffEq/vignettes/PDEinR.pdf
library("deSolve")
library("ReacTran")
N        <- 100
xgrid    <- setup.grid.1D(x.up = 0, x.down = 1, N = N)
x        <- xgrid$x.mid
D.coeff  <- 0.03 # 拡散係数
Diffusion <- function (t, Y, parms){
  tran <- tran.1D(C = Y, C.up =  0, C.down = 1,
  D = D.coeff, dx = xgrid)
  list(dY = tran$dC, flux.up = tran$flux.up,
  flux.down = tran$flux.down)
}
Yini  <- 0*x    # Rでは*が要素の掛け算になる。Xと同じサイズのベクトル作成（全て0）
Yini[N]  <- 1  # N番目の値のみ1にする
times <- seq(from = 0, to = 5, by = 0.01)
print(system.time(
out <- ode.1D(y = Yini, times = times, func = Diffusion,
parms = NULL, dimens = N)              ))

par (mfrow=c(1, 2))
plot(out[1, 2:(N+1)], x, type = "l", lwd = 2,
     xlab = "Variable, Y", ylab = "Distance, x")

for (i in seq(2, length(times), by = 50))
   lines(out[i, 2:(N+1)], x)
   image(out, grid = x, mfrow = NULL, ylab = "Distance, x",
   main = "Y")
```

拡散方程式のグラフ出力

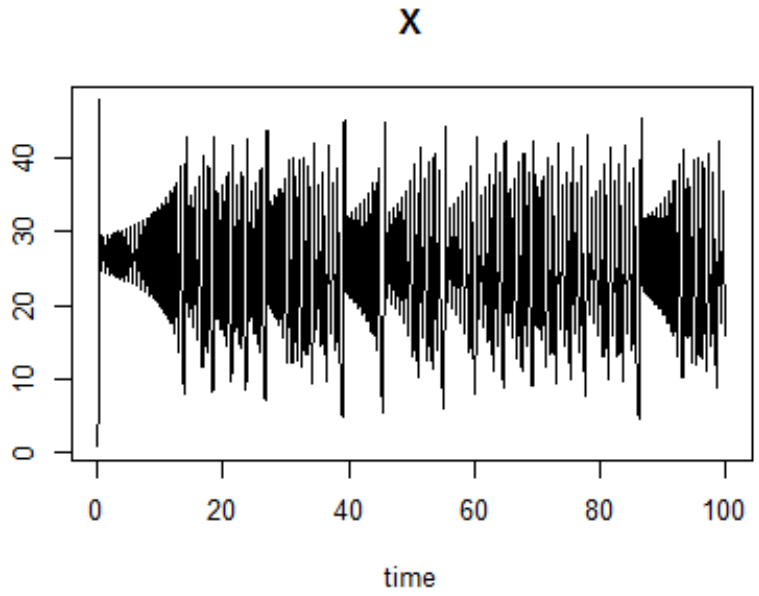


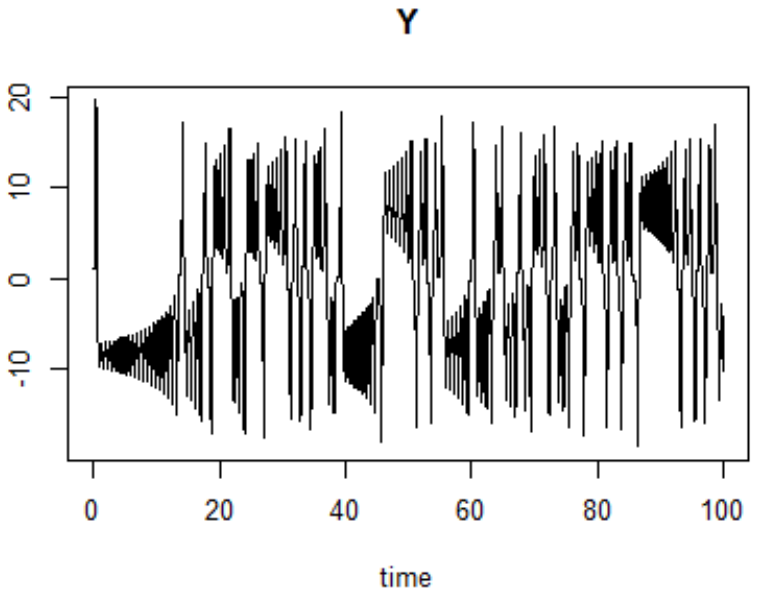


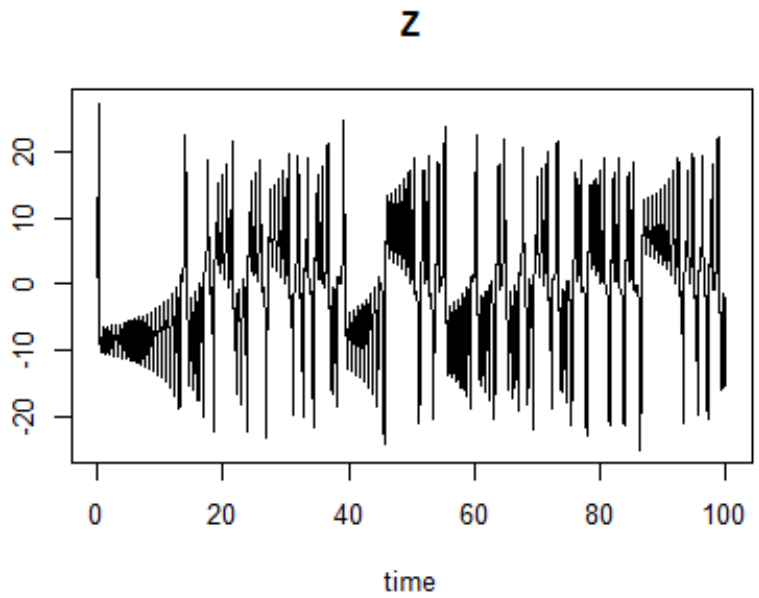
Chaos in the atmosphere / deSolve関数のヘルプ

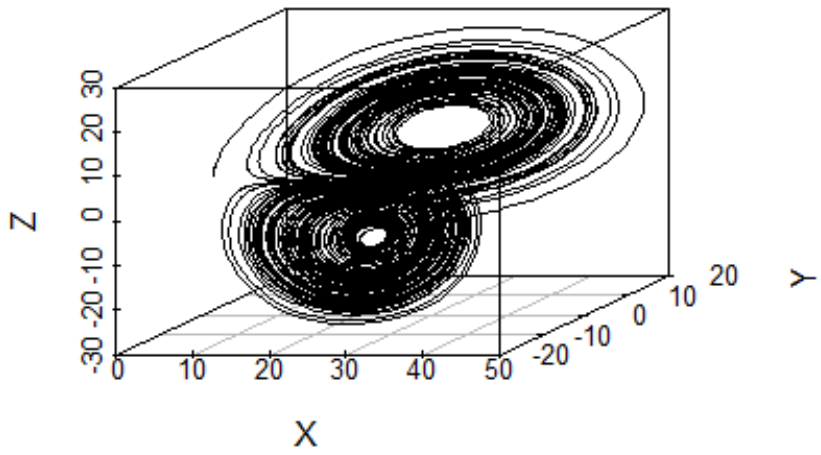
```
library(deSolve)
library(scatterplot3d)
## Chaos in the atmosphere
Lorenz <- function(t, state, parameters) {
  with(as.list(c(state, parameters)), {
    dX <-  a * X + Y * Z
    dY <-  b * (Y - Z)
    dZ <- -X * Y + c * Y - Z
    list(c(dX, dY, dZ))
  })
}
parameters <- c(a = -8/3, b = -10, c = 28)
state      <- c(X = 1, Y = 1, Z = 1)
times      <- seq(0, 100, by = 0.01)

out <- ode(y = state, times = times, func = Lorenz, parms = parameters)
plot(out)
## add a 3D figure if package scatterplot3D is available
if (require(scatterplot3d))
  scatterplot3d(out[,-1], type = "l")
```

Chaos in the atmosphere の出力









2. Scilab ＋ MmodD [9] CeCILL（≒GPL）ライセンス

- MmodD：Modular Modeling is a finite element external module for SCILAB.
- mmodd_install.sceをダウンロードして、スクリプトを実行すればネットから必要なファイルを自動的にダウンロード/インストールする（MmodDとAtoms）などがインストールされる。
- MmodDのhelpとデモが追加される（Paraviewファイルへの変換コマンドあり）
- Windows 10では、stacksize('max')のコマンドが動作しないので、一部のデモが動作しなかった（Ver. 5.5.2, 6.0.0-α1共にNG）。
- Scilab 6.0.0 α版が公開された（現時点ではC言語などとのリンクがNG）。
  直ぐに利用するのでなければ、Scilab 6のリリース待ちで良いと思います。

Description

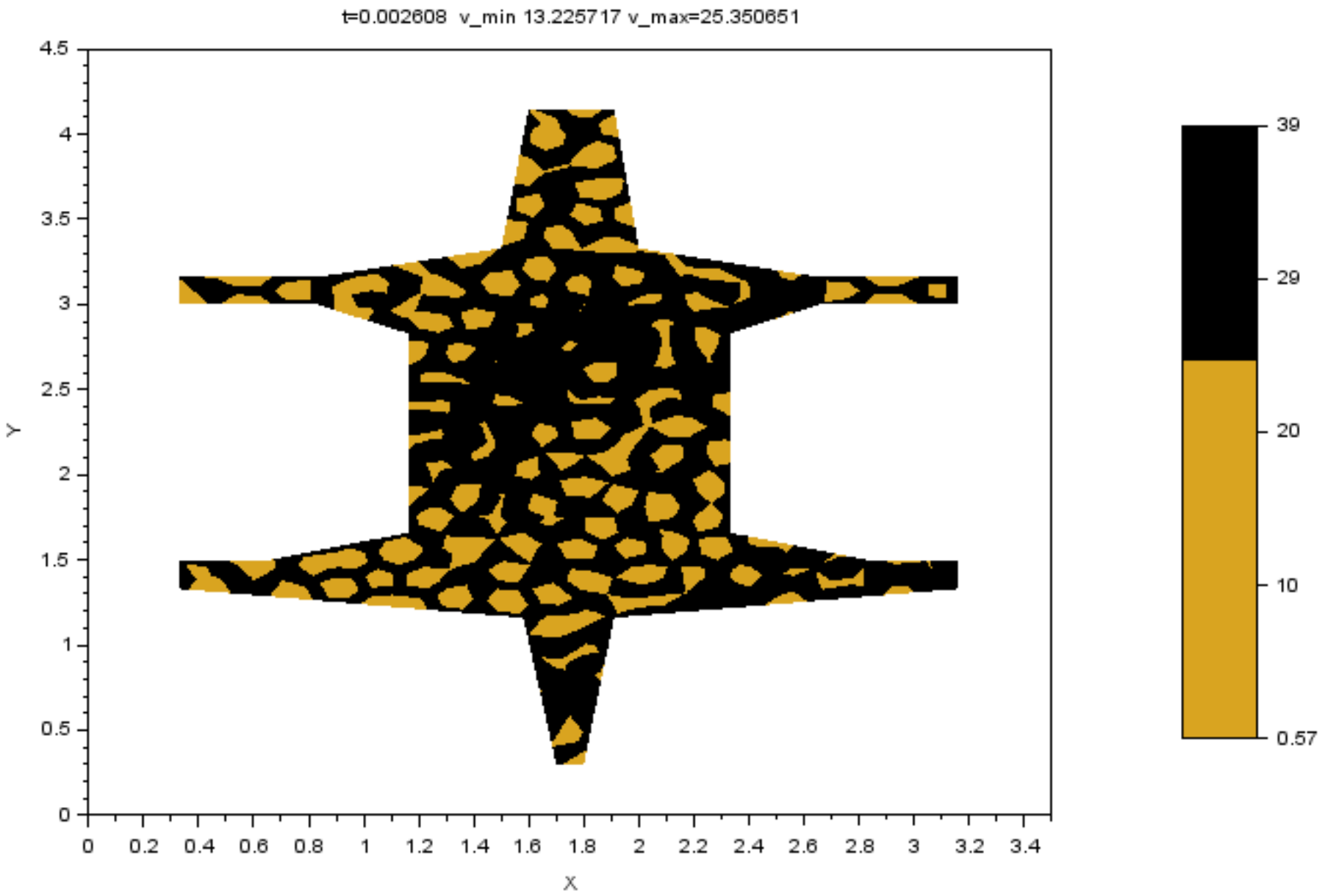
The MmodD project contains tools for the study and solution of partial differential equations (PDEs) in 2d and 3d.
A set of command-line functions and a graphical user interface let you preprocess, solve, and postprocess generic PDEs.

Features

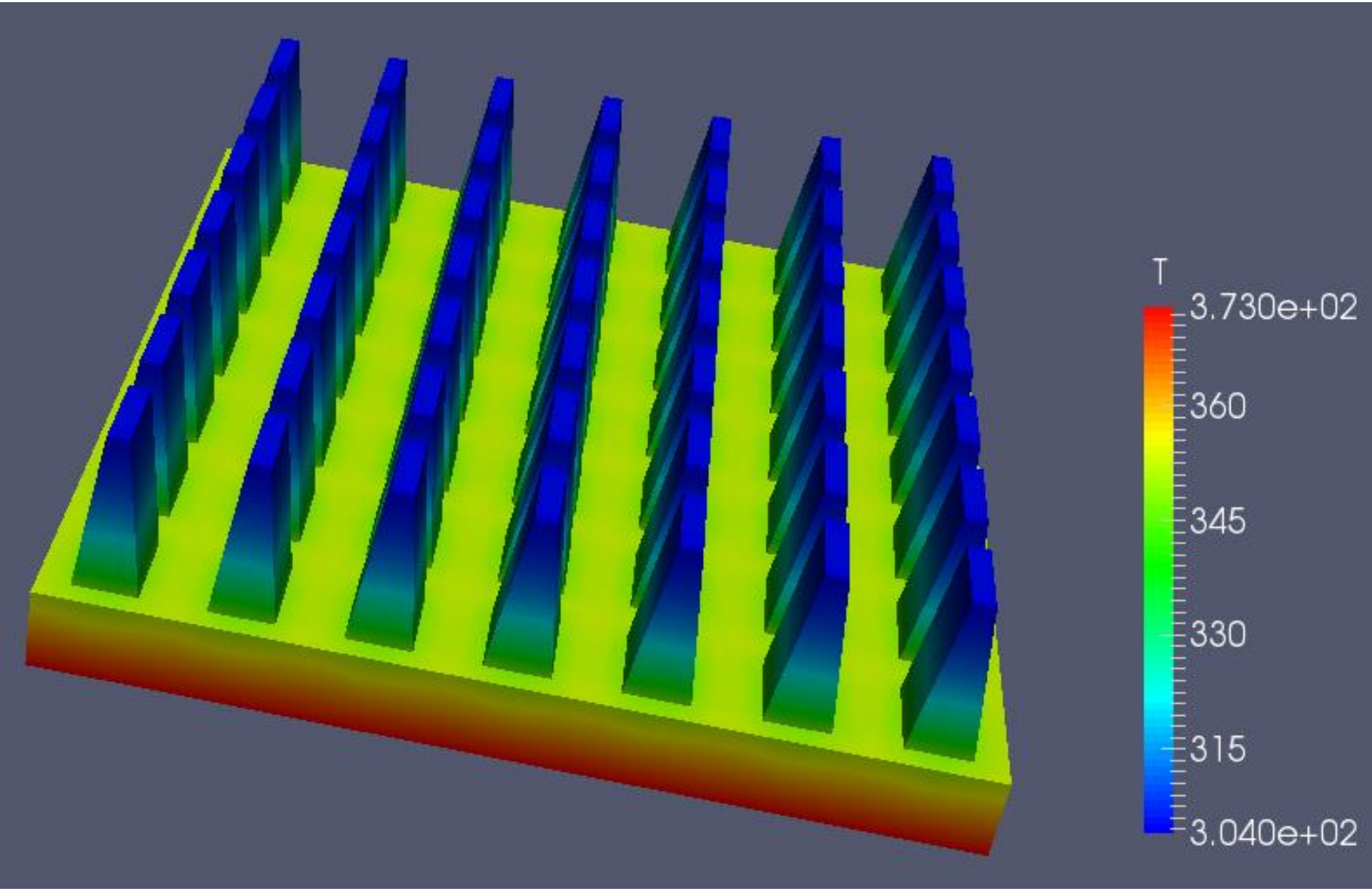
2013年10月からメンテされていないので、Scilab 6で対応するか不明

- Import mesh files from Netgen, Tetgen, gmsh.
- Direct solver UMFPACK (including in scilab)
- Iterative solver : Conjugate gradient, GMRES, Stabilized Bi-conjugated gradient.... (SPARSKIT)
- Incomplete ILU preconditioner.
- Export VTK files for Paraview, VisIt.

デモ:黒と黄色の2部分の拡散アニメーションの1コマ

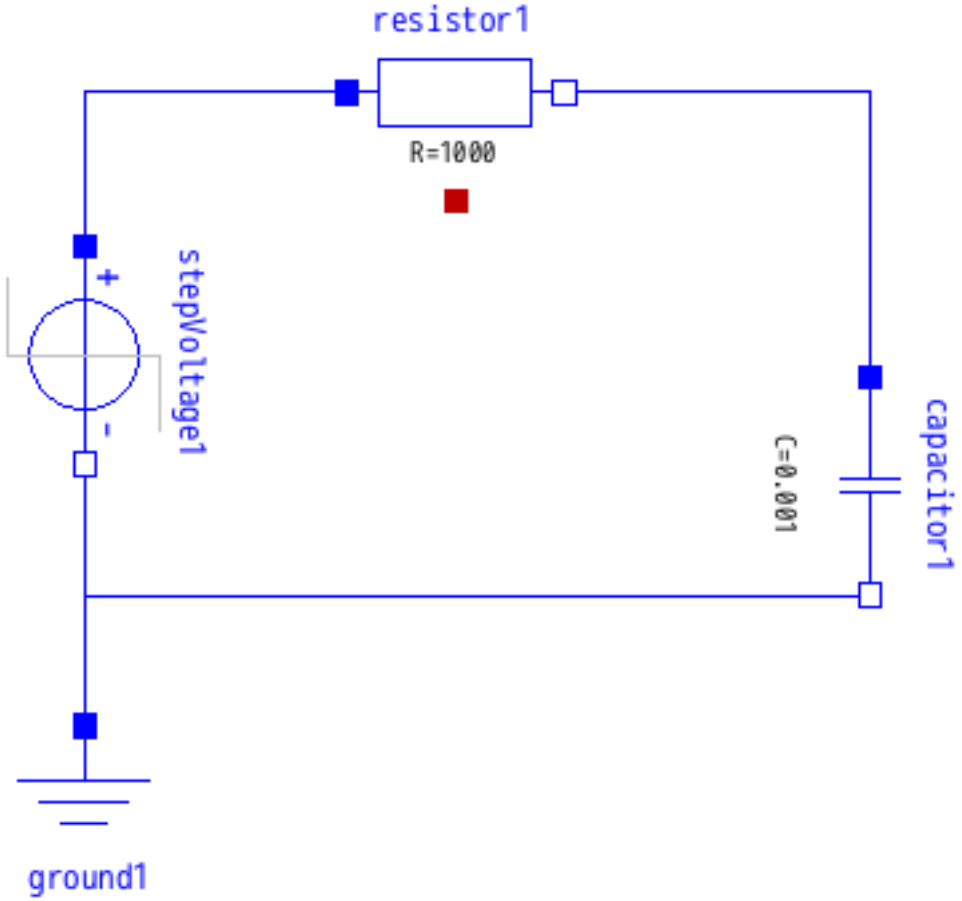


デモ:3D定常熱解析（Paraviewにエクスポートした）



3. OpenModelica

- 今年の6月に日本語訳の書籍「Modelicaによるシステムシミュレーション入門」が出版された[11]
- 内容は、英文のPDFファイル「Introduction to Object-Oriented Modeling and Simulation with Modelica Using OpenModelica」[10]と類似
- 偏微分方程式の発表資料[12]のコマンドを入力したが、文法エラーとなった。



グラフィック画面での部品接続操作が難しい！

2000年以前のPSPICE評価版の方が操作しやすい？

ラベル機能は、部品に入出力のみ？

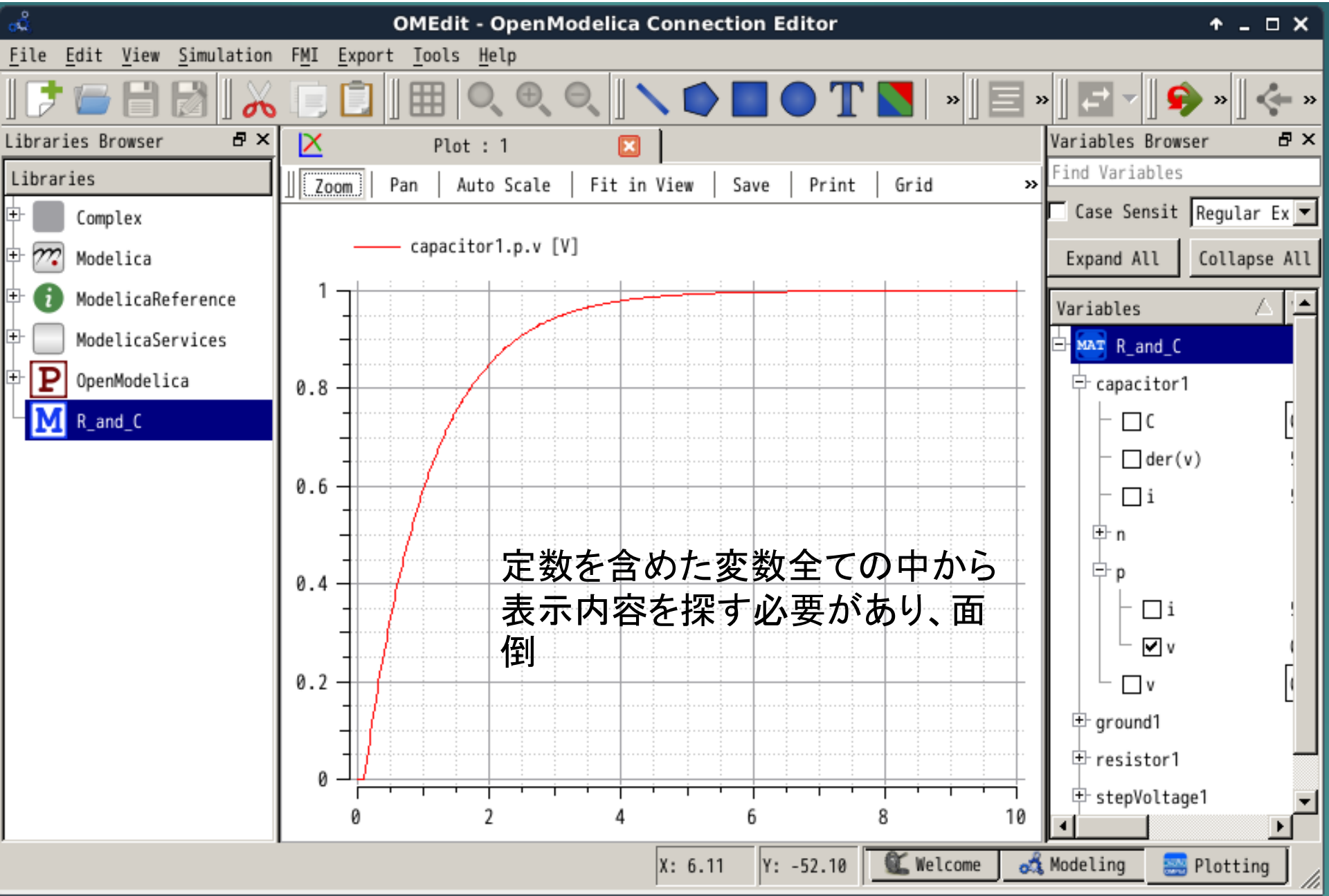
RC回路でのmodelicaコード（グラフィック操作から作成）

```
model R_and_C
  Modelica.Electrical.Analog.Sources.StepVoltage stepVoltage1(V = 1, startTime = 0.1) annotation(Placement(visible = true,
transformation(origin = {-72, 16}, extent = {{-10, -10}, {10, 10}}, rotation = -90)));
  Modelica.Electrical.Analog.Basic.Ground ground1 annotation(Placement(visible = true, transformation(origin = {-72, -28}, extent = {{-
10, -10}, {10, 10}}, rotation = 0)));
  Modelica.Electrical.Analog.Basic.Capacitor capacitor1(C = 0.001) annotation(Placement(visible = true, transformation(origin = {0, 4},
extent = {{-10, -10}, {10, 10}}, rotation = -90)));
  Modelica.Electrical.Analog.Basic.Resistor resistor1(R = 1000) annotation(Placement(visible = true, transformation(origin = {-38, 40},
extent = {{-10, -10}, {10, 10}}, rotation = 0)));
equation
  connect(stepVoltage1.p, resistor1.p) annotation(Line(points = {{-72, 26}, {-72, 26}, {-72, 40}, {-48, 40}, {-48, 40}}, color = {0, 0, 255}));
  connect(resistor1.n, capacitor1.p) annotation(Line(points = {{-28, 40}, {0, 40}, {0, 14}, {0, 14}}, color = {0, 0, 255}));
  connect(ground1.p, stepVoltage1.n) annotation(Line(points = {{-72, -18}, {-72, -18}, {-72, 6}, {-72, 6}}, color = {0, 0, 255}));
  connect(capacitor1.n, ground1.p) annotation(Line(points = {{0, -6}, {-72, -6}, {-72, -18}, {-72, -18}}, color = {0, 0, 255}));
  annotation(Icon(coordinateSystem(extent = {{-100, -100}, {100, 100}}, preserveAspectRatio = true, initialScale = 0.1, grid = {2, 2})),
Diagram(coordinateSystem(extent = {{-100, -100}, {100, 100}}, preserveAspectRatio = true, initialScale = 0.1, grid = {2, 2})));
end R_and_C;
```

部品の位置・回転操作、ラインと接続点の情報が入り、実行するコードの本体が判りにくい

Modelicaのコードは中間コード（通常Cコード）に変換され[11]、GCCでコンパイルする（コンパイル時間はLinuxで行った方が短い：高速）

出力図:キャパシタの電圧



まとめ

1．R言語により1次元の偏微分方程式を解くことができた（使えそう）
- ユーザーが多く、日本語の書籍の数が多い
- RStudioを使用するとパッケージの操作などが容易になる
- 書籍「RStudioではじめるRプログラミング入門」[7]によると、cranのパッケージは十分評価し安定しているとのこと

2. Scilab＋MmodD（Windows 10ではScilabの動作がおかしい）
- 日本語の書籍の数が少なく、インストール方法や基本操作を中心にしたものが多い
- 1年後くらいにメジャーバージョンアップ（Ver. 6）があるので、機能が拡張されてから再検討したい

３．Openmodelica
- グラフィックモードで部品接続の操作性が悪い（使えない？）
- 出力表示機能が洗練されていない
- デモの修正程度の利用にとどまる？

参考資料ーその1

1. 偏微分方程式の講義ノートPDF。解き方や分類の基礎を学ぶ入門用の参考書
   http://d.hatena.ne.jp/language_and_engineering/20140608/PartialDifferentialEquationsPDFLectureNotes

2. 吉田の教材文庫：Excelで学ぶ振動基礎(7h)、他
   http://edu.katzlab.jp/lec/vib7h

3. GNU R / CRAN Task View：微分方程式
   http://www.trifields.jp/r-cran-task-view-differential-equations-685

4. Solving partial differential equations, using R “package ReacTran”
   https://cran.r-project.org/web/packages/ReacTran/vignettes/PDE.pdf

5. Solving Differential Equations in R (book) – PDE examples
   https://cran.r-project.org/web/packages/diffEq/vignettes/PDEinR.pdf

6. Package deSolve: Solving Initial Value Differential Equations in R
   https://cran.r-project.org/web/packages/deSolve/vignettes/deSolve.pdf

7. 書籍：RStudioではじめるRプログラミング入門（Hands-On Programming with R）
   Garrett Grolemund 著、大橋 真也 監訳、長尾 高弘 訳、2015年3月 初版

8. 「R for Cloud Computing」の紹介 2015年1月
   http://www.slideshare.net/hiratake55/r-for-cloud-computing-43609330

参考資料ーその2

９．MmodD（Scilab用偏微分解析用ソフト）
http://forge.mmodd.org/projects/mmodd

１０．Introduction to Object-Oriented Modeling and Simulation with Modelica Using OpenModelica
Peter Fritzson https://www.openmodelica.org/images/docs/tutorials/modelicatutorialfritzson.pdf

１１．書籍：「Modelicaによるシステムシミュレーション入門」 Peter Fritzson著
大畠　明　監訳　　広野　友英　訳、2015年6月初版、 TechShare

１２．Partial Differential Equations in Modelica、Jan Sílar、OpenModelica Workshop 2015
https://openmodelica.org/images/docs/openmodelica2015/OpenModelica2015-talk05-PDEInModelica_silar.pdf