Ansaq Journal for Scientific Publishing

مجلة انساق للنشر العلمي

Fourth_edition

Volume (2) Issue (3)

(1–11) 2026

Using R language to solve Partial deferential equations

Osman Mohieddin Baber Ahmed

(Sudan University of Science and Technology – SUST Faculty of science)

Supervisor : Dr. Abdul Rahim Bashir Hamid

(Associate Professor of Applied Mathematics, University of Gezira, Faculty of Education, Sudan)

Co Supervisor : Dr. Balqis Abdul Aziz Abdul Rahman

(Associate Professor, Sudan University of Science and Technology, Faculty of Science)

Published on: Published online on July 20, 2026

Abstract:

This paper discusses how we use R-language to solve PDE ,wave equation as a model.

Introduction:

Computers are gradually becoming the most powerful human inventions with applications spreading to almost all aspects of human endeavor .one aspect in which computers have had signification impact is education .the advent of educational technology has been regarded as a major revelation within the educational system (of .Asliloy ,1972,fourth Revelation ,1972) however , from the –reading role computers are playing in education, one can argue that discovery of the computer and there world wide web is another revolution .with this development the face of the wale educational system has changed forever.

Computers have been used in education for more than four decades, and they have now become an integral part of the entire educational system: as teacher, study male and more especially as tools to improve the entire teaching and learning processes. this usage is dramatically on the increase ,and predictions have it that this trend will continue to accelerate (Brantford, Brown and cocking 1999).the various ways of computer using in education include ,but are not restricted to :tutorial, drills, simulation, games, tests,

problem solving , and at more advanced level in what known as intelligent CAL(ICAL),intelligent tutorial and computer –controlled video(Alessi and trollip,1985).

In his development, no subject has stronger intrinsic links with computers than mathematics, kaput (1992) noted that the role of computers in mathematics education is so significant that is difficult to describe, and it changes so rapidly that it is difficult to follow.

There are really two kind of languages:

- Languages that are designed to minimize the time programmers spend programming.
- Languages that are designed to minimize the time programmers spend computing.

The first category are languages that are easy to program in .it is composed of interpreted languages: Python, Julia, R, MATLAB… .The most popular free alternative is like Python .in some domains. Like biology or statistics, it might be R. if you feel the visceral need to cough up some cash, and then you could buy a MATL AB license.

They all offer easy ways to plot graphs ,do complex mathematics (eigenvalues, inverse matrix, conjugate gradient, Monte-Carlo),query databases ,the web..

The second category is composed of C/C++ and –God forbid-elunky old FORTRAN.

They are harder to program but should be more computationally efficient. If the programs are well written.

The default option should be minimize the time you spend programming and go with the first category .your time is valuable, the computer's is cheap.so minimize the time you spend programming and maximize the time doing actual physics.as Google puts it (and they do a fair amount of compute):Python where we can ,C++ where we must.

There is a tired argument that to do high performance computing it is absolutely necessary to use low level language such as C and C++.however, most high-performance physic software will devolve the heavy lifting to external libraries, such as BLAS, LAPACK, FFTW, and others.in that case, whether there libraries are called from a language of the first or second category has little implication on the speed code.

R Programming language

R is a programming language that is primarily used for statistics computing and graphics.it is available for free .Users can compile and run R on various operating systems including windows, Mace OS X and Linux.

R is a programming language and free software environment for statistical computing and graphics supported by the foundation for statistical computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis R Programming ,or R .has turned into the most prevalent language for data science and a fundamental tool for finance and analytics –driven organizations, for example , Google, Facebook, and Linkedln. R is a language and environment for statistical computing and design.

R is programming language developed by Ross Ihaka and Robert Gentleman in 1993,R possesses an extensive catalog of statistical and graphical methods, it includes machining learning algorithm ,linear regression, time series, statistical inference to name a few .Most of the R libraries are written in R ,but for heavy computational task, C,C++ and Fortran erodes are preferred.R is not only entrusted by academic, but many large companies also use R Programming language, including Uber, Google, Airbanb , and Facebook and so on. Data analysis with R is done in a series of step: programming, transforming discovering, modeling and communicates the results

- Program is clear and accessible programming tool.
- Transform: R is made up of a collection of libraries designed specifically for data science.
- Discover: investigate the data, refined your hypothesis and analyze them.
- Model: R provides a wide array of tools to capture the right model for your data.
- Communicate: integrate codes, graphs, and outputs to a repute with R markdown or build shiny apps to share with the world

The language is known to be fairly unconventional compared to popular software development language such as C++or java .what makes R stand out from most other languages is that it acts as an interactive statistical environment also allows underscores as variable characters, unlike other language that use them as assignment operators is popular among data scientists.

Since data science is dealing in statistics, R will be perfect tools for implementing various operations .R is used for Extract, Transform, and loading .it offers an interface for many databases like SQL and even spreadsheets, R interface with No SQL databases and analyzes unstructured data.

As technology improves, the data companies or research in situations collect has become more and more complex, and R has been adopted by many as the language of choice to analyze data is great for machine learning, data visualization and analysis, and some areas of scientific computing.

Example: The Wave Equation in R:

single equation:

The wave equation is a classic example of a partial differential equation. It comes in several variants and has applications beyond the name. In principle, the wave equation describes the path of a wave traveling through a medium. For a one-dimensional wave equation, this describes a wave traveling on a string, like a violin's string. In two-dimensions, the wave equation describes a wave on a membrane, like a drumhead. And for three dimensions, it describes the propagation of sound through the air. The equation can also describe light waves.

For our purposes, we will look at a one-dimensional light wave using a second-order differential equation. And to further simplify the problem, we will assume the string being modeled is fixed at both ends. This is the same circumstance as a string on a musical instrument, or a jumprope with the ends held in place.

This process is implemented in wave, shown in a moment. We take an approach here that instead of generating a first second step using the simpler method, we actually take a step backwards and create a ``zeroth'' step, as shown in the function. This helps with bookkeeping, rather than forcing a recording of both the first and second steps, before entering the loop, we record the first and continue into the loop for

steps. Inside the loop, the step is broken down into three distinct parts and the next

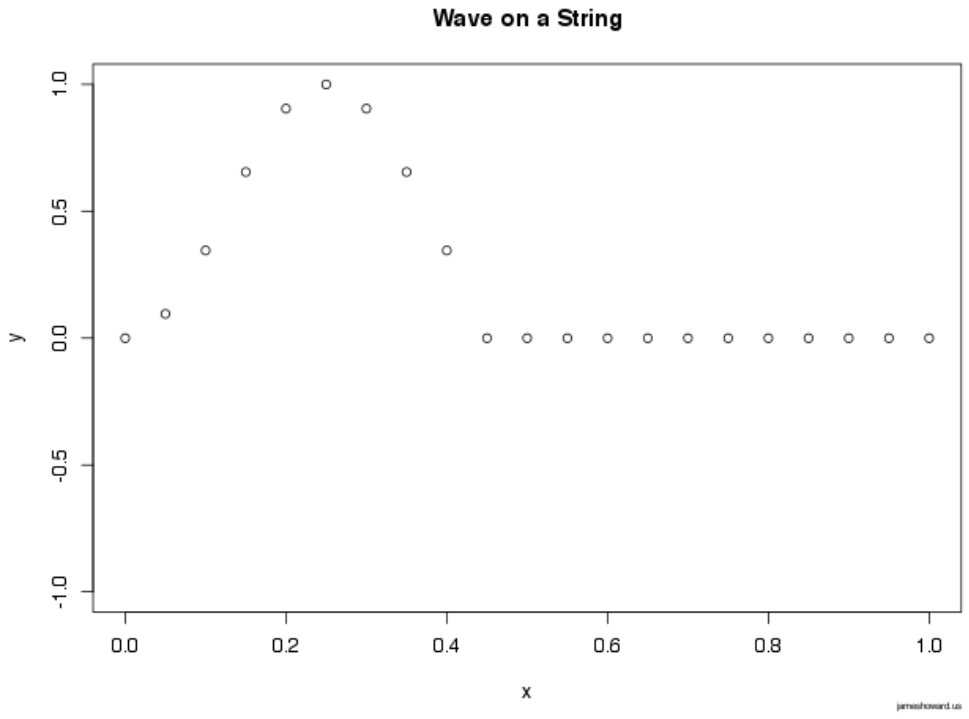
iteration is assembled from those parts. Then it is recorded in the output array and both the previous and current iterations are incremented.

```
wave <- function(u, alpha, xdelta, tdelta, n) {
m <- length(u)
uarray <- matrix(u, nrow = 1)
newu <- u

h <- ((alpha * tdelta) / (xdelta))^2

## Initial the zeroth timestep
oldu <- rep(0, m)
oldu[2:(m - 1)] <- u[2:(m - 1)] + h *
(u[1:(m - 2)] - 2 * u[2:(m - 1)] + u[3:m]) / 2

## Now iterate
for(i in 1:n) {
ustep1 <- (2 * u - oldu)
ustep2 <- u[1:(m - 2)] - 2 * u[2:(m - 1)] + u[3:m]
newu <- ustep1 + h * c(0, ustep2, 0)
oldu <- u
u <- newu
uarray <- rbind(uarray, u)
}

return(uarray)
}
```

We can see this at work in the following example. Of interest, we set the initial state in u to be a partial sine wave. The left side, the lower half of the array, is the positive half of a sine wave. The right half is zeroed out, leading to a wave that travels down the string. The process fixes the ends of the string at zero, making them unmoving. The wave will reach the end of the line and bounce back to the start.

```
speed <- 2
x0 <- 0
xdelta <- .05
x <- seq(x0, 1, xdelta)
m <- length(x)
u <- sin(x * pi * 2)
u[11:21] <- 0
tdelta <- .02
n <- 40
z <- wave(u, speed, xdelta, tdelta, n)</pre>
```

The image below comes from animating the output.



Implementation in R

```
• #Define simulation parameters
x = seq(0,1,1/99) #Spatial grid
dt = 0.5 #Time step
tMax = 200 #Simulation time
c = 0.015 #Wave speed
fLeft = function(t) 0 #Left boundary condition
fRight = f(t) 0.1*sin(t/10) #Right boundary condition
fPosInitial = f(x) exp(-200*(x-0.5)^2) #Initial position
fVelInitial = f(x) 0*x #Initial velocity
#Run simulation
dx = x[2]-x[1]
r = c*dt/dx
n = length(x)
t = seq(0,tMax,dt) #Time vector
m = length(t) + 1
```

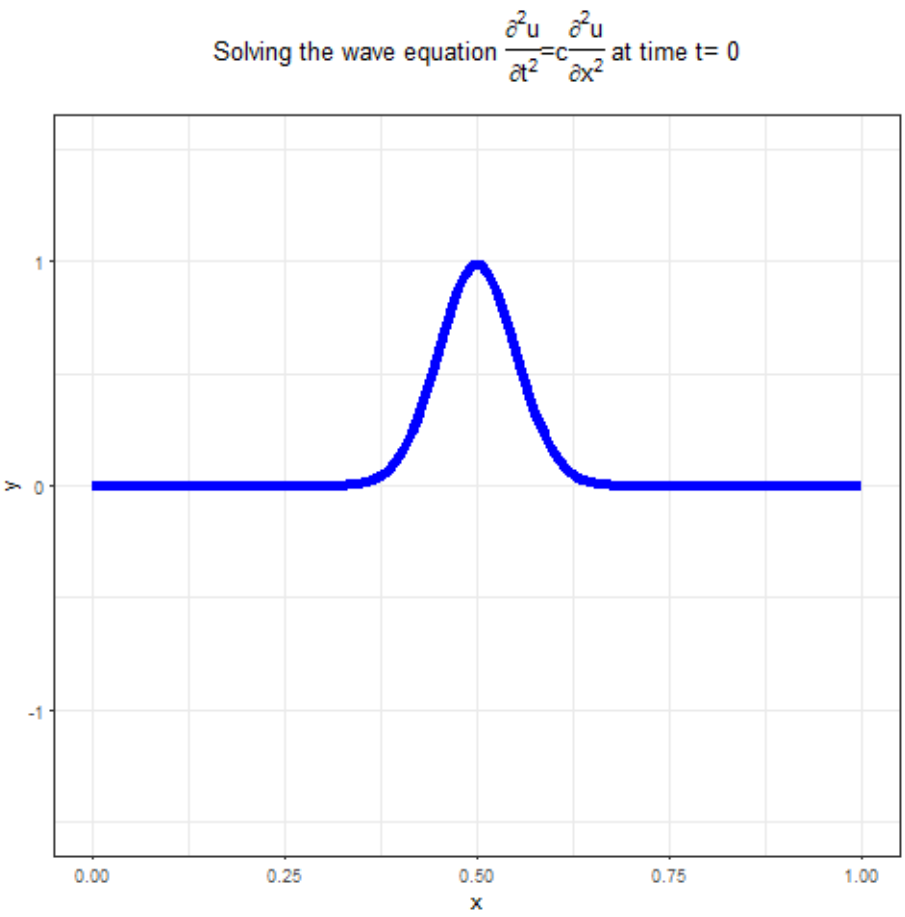
- Now iteratively compute *u(x,t)* by imposing the following **boundary conditions**
- 1. *u(0,t) = 0*

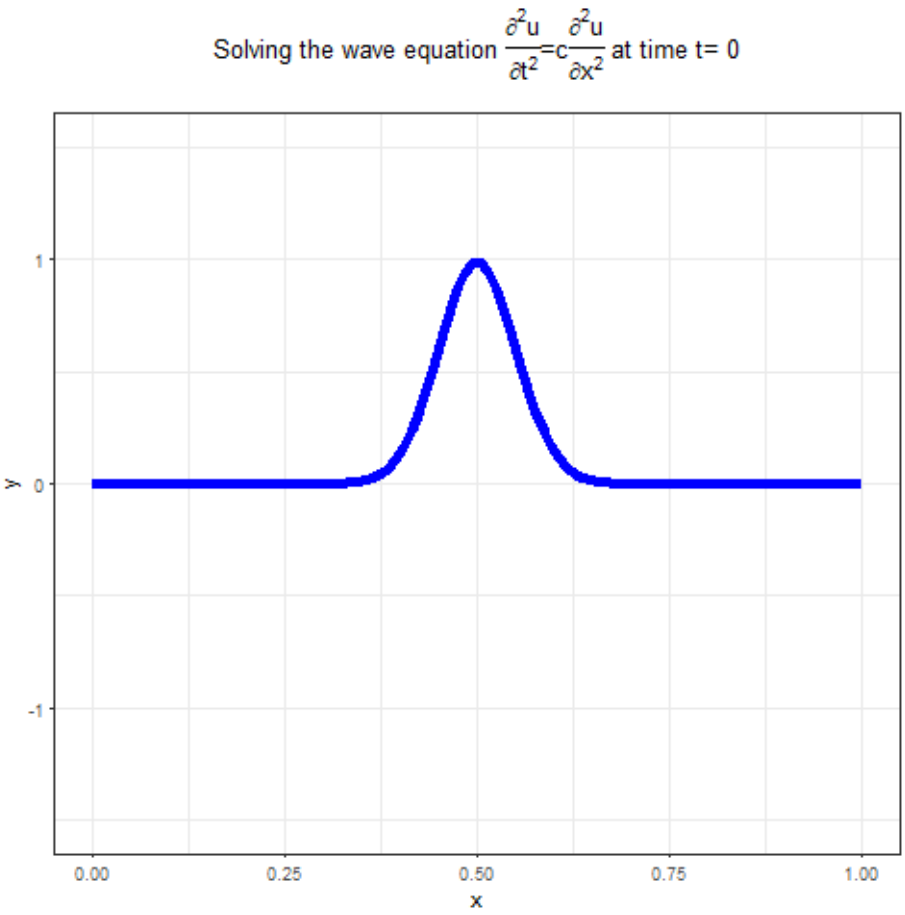
2. *u(L,t) = (1/10).sin(t/10)*

- along with the following **initial conditions**
- 1. $u(x,0) = exp(-500.(x-1/2)2)$

2. $\partial u(x,0)/\partial t = 0.x$

- as defined in the above code snippet.
- #Create tri-diagonal matrix A here, as described in the algorithm
#Impose initial condition
u <- matrix(rep(0, n*m), nrow=m)
u[1,] = fPosInitial(x)
u[1,1] = fLeft(0)
u[1,n] = fRight(0)
for (i **in** 1:length(t)) {
#Find solution at next time step
if (i == 1) {
u[2,] = 1/2*(A%*%u[1,]) + dt*fVelInitial(x)
} **else** {
u[i+1,] = (A%*%u[i,])-u[i-1,]
}
u[i+1,1] = fLeft(t[i]) #Impose left B.C.
u[i+1,n] = fRight(t[i]) #Impose right B.C.
}
- The following animation shows the output of the above implementation of the solution of wave PDE using R, it shows how the waves propagate, given a set of BCs and ICs.


Solving the wave equation $\frac{\partial^2 u}{\partial t^2}=c\frac{\partial^2 u}{\partial x^2}$ at time t= 0
1
0
-1
y
0.00
0.25
0.50
0.75
1.00
x



-

Results

- A PDE can be solved in any computer language you want to solve it in R**.**
- R is simple and effective programming language which has been well-developed as well as R is data analysis software is a well –designed, easy and effective language that has the concepts of conditionals, looping, user-defined recursive procedures and v what is R Programming Language? Introduction & Basics?

References

1/ Vern Paxon and Chris Saltmarch Glish : A uses –level software bus for lossely –coupled distributed systems,Hn Proceedings of the winter 1993 Use nix Conference ,page 217-276.USENIX Association,1993.

2/J.Purtilo,R.Snodgrass,and A.L.solf, software bus organization Reference model and comparison of two existing system.Technical Repot TR-8-ARPA Model intenconneection Formalism working group,1991.

(file: 11thumper.ec.umd.edu/file/doe/refimodel.ps.Z)

3/J.M .Purtilo the polylith software bus. ACM. Transections on programming language and system,16(1):151-174,1994.

4/ K.Sayre and M.A.Gray Backtalk : A generalized dynamic eommunication system for DA1software –practice –practice Experience,23(9):1043-1057,1993.

5/Sum .Soft .Open network 1.1 User's Guide .Sun Microsystems ,inc,1991.

6/Sun Soft .Tool talk 1.1 User's Guide .Sun microsystems .1nc, Mountain view .CA,1993.

7/lecture notes in computational science and engineering .Anders log. Kent-Ander mardal Garth N.Well editors Springer veriager berlin Heideloery 2012.

8/ Petzold LR (1983). “Automatic Selection of Methods for Solving Stiff and Nonstiff Systems of
Ordinary Differential Equations.” SIAM Journal on Scientific and Statistical Computing,
4, 136–148.
9/ Press WH, Teukolsky SA, Vetterling WT, Flannery BP (1992). Numerical Recipes in FOR-
TRAN. The Art of Scientific Computing. Cambridge University Press, 2nd edition.
10/ R Development Core Team (2008). R: A Language and Environment for Statistical Computing.

11/ R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http:
//www.R-project.org.
Shampine L, Thompson S (2000). Solving Delay Differential Equations with dde23. URL
http://www.runet.edu/~thompson/webddes/tutorial.pdf.
Karline Soetaert, Thomas Petzoldt, R. Woodrow Setzer 49
Soetaert K, Cash JR, Mazzia F (2010a). bvpSolve: Solvers for Boundary Value Problems
of Ordinary Differential Equations. R package version 1.2, URL http://CRAN.R-project.org/package=bvpSolve.
Soetaert K, Herman PMJ (2009). A Practical Guide to Ecological Modelling. Using R as a
Simulation Platform. Springer. ISBN 978-1-4020-8623-6.
Soetaert K, Meysman F (2010). ReacTran: Reactive transport modelling in 1D, 2D and 3D.
R package version 1.3.
Soetaert K, Petzoldt T, Setzer R (2010b). "Solving Differential Equations in R: Package deSolve." Journal of Statistical Software, 33(9), 1–25. ISSN 1548-7660. URL http://www.
jstatsoft.org/v33/i09.
Soetaert K, Petzoldt T, Setzer RW (2008). R package deSolve: Writing Code in Compiled
Languages. deSolve vignette - R package version 1.8.
Soetaert K, Petzoldt T, Setzer RW(2010c). deSolve: General solvers for initial value problems
of ordinary differential equations (ODE), partial differential equations (PDE), differential
algebraic equations (DAE) and delay differential equations (DDE). R package version 1.8.