Journal of University Studies for inclusive Research (USRIJ)
مجلة الدراسات الجامعية للبحوث الشاملة
ISSN: 2707-7675

Journal of University Studies for Inclusive Research
Vol.5, Issue 53 (2026), 158480- 158494

USRIJ Pvt. Ltd

+In the name of ALLA

Sudan University of Science and Technology - SUST

Faculty of science

Department Mathematics+

استخدام ماتلاب لحل المعادلات التفاضلية الجزئية

الاسم: عثمان محي الدين بابكر أحمد

المشرف الأول: د. عبدالرحيم بشير حامد

أستاذ مساعد، الرياضيات التطبيقية، جامعة الجزيرة، كلية التربية، السودان، رئيس قسم

المشرف الثاني: د. بلقيس عبدالعزيز عبدالرحمن

أستاذ مساعد، جامعة السودان للعلوم والتكنولوجيا، كلية العلوم، رئيس قسم

Journal of University Studies for inclusive Research (USRIJ)
مجلة الدراسات الجامعية للبحوث الشاملة
ISSN: 2707-7675

الملخص

تُعدّ المعادلات التفاضلية الجزئية أدوات أساسية في نمذجة الظواهر الفيزيائية والبيولوجية والهندسية، إذ تُجسّد العمليات التي تتطور عبر المكان والزمان. تقتصر الحلول التحليلية عمومًا على الحالات المبسطة أو الخطية، مما يجعل الأساليب الحسابية ضرورية للأنظمة المعقدة أو غير الخطية. وقد برز برنامج **MATLAB**كمنصة رائدة بفضل تصميمه الموجه نحو المصفوفات، وحلوله المدمجة، وقدراته التصويرية القوية. اكتسب **MATLAB**مكانة مرموقة في التعليم والبحوث التطبيقية نظرًا لسهولة استخدامه، وقدرته على إنشاء نماذج أولية سريعة، ومكتبته الواسعة من الأدوات. في الوقت نفسه، فإن تكلفته العالية وقيود أدائه في عمليات المحاكاة واسعة النطاق تدفع إلى مقارنته ببدائل مفتوحة المصدر مثل **Python**و**Julia**، على الرغم من أن **MATLAB**لا يزال يحظى بتقدير كبير لوضوحه واستقراره وتكامله بين الحساب العددي والتصوير. يهدف هذا البحث إلى إثبات فعالية **MATLAB** في حل المعادلات التفاضلية الجزئية غير الخطية من خلال تطبيق وتحليل نظام تفاعل–انتشار.

أبرزت النتائج مزايا بنية **MATLAB**القائمة على المصفوفات، والتي سهّلت تطبيق النموذج بإيجاز. ووفرت أدوات التصور المتكاملة رؤى واضحة حول تطور النظام، مما عزز قيمته لأغراض البحث والتدريس. وبشكل عام، أكدت النتائج ملاءمة **MATLAB**كمنصة متعددة الاستخدامات وموثوقة لنمذجة وتحليل الأنظمة الديناميكية المعقدة التي تحكمها المعادلات التفاضلية الجزئية. في الختام، يثبت **MATLAB**أنه منصة فعالة ومتعددة الاستخدامات لحل المعادلات التفاضلية الجزئية غير الخطية، إذ يجمع بين الدقة

Journal of University Studies for inclusive Research (USRIJ)
مجلة الدراسات الجامعية للبحوث الشاملة
ISSN: 2707-7675

الحسابية وقدرات التصور القوية. وتعزز نقاط قوته في سرعة إنشاء النماذج الأولية وسهولة الوصول إليه أهميته المستمرة في الرياضيات التطبيقية والحوسبة العلمية على الرغم من بعض القيود.

Using MATLAB to Solve Partial differential Equations

NAME: Osman Mohieddin Babiker Ahmed

SUPERVISOR: Dr. Abd El-Rahim Bashir Hamed

Associated Professor, Applied Mathematics, Uof G, Faculty of Education, Sudan. Head of Department

d.abdelrahim@gmail.com

CO-SUPERVISOR: Dr. Belgiss Abdelaziz Abdelrhman

Associated Professor, SUST – Faculty of Science, Sudan, Head of Department

Belgiss2014@gmail.com

Journal of University Studies for inclusive Research (USRIJ)
مجلة الدراسات الجامعية للبحوث الشاملة
ISSN: 2707-7675

Abstract

Partial differential equations (PDEs) serve as essential tools in modeling physical, biological, and engineering phenomena by capturing processes that evolve over space and time. Analytical solutions are generally limited to simplified or linear cases, making computational approaches indispensable for complex or nonlinear systems. With MATLAB emerging as a leading platform due to its matrix-oriented design, built-in solvers, and strong visualization capabilities. MATLAB has gained prominence in education and applied research because of its accessibility, rapid prototyping capacity, and extensive library of toolboxes. At the same time, its proprietary cost and performance limitations on large-scale simulations invite comparison with open-source alternatives such as Python and Julia, though MATLAB remains highly valued for its clarity, stability, and integration of numerical computation with visualization. The aim is to demonstrate the effectiveness of MATLAB in solving nonlinear partial differential equations through the implementation and analysis of a reaction–diffusion system.

The results highlighted the advantages of MATLAB's matrix-based structure, which facilitated concise implementation of the model. Its integrated visualization tools provided clear insights into the evolution of the system, strengthening its value for both research and pedagogical purposes. Overall, the findings underscored MATLAB's suitability as a versatile and reliable platform for modeling and analyzing complex dynamical systems governed by PDEs. In conclusion, MATLAB proves to be an effective and versatile platform for solving

Journal of University Studies for inclusive Research (USRIJ)
مجلة الدراسات الجامعية للبحوث الشاملة
ISSN: 2707-7675

nonlinear partial differential equations, combining computational accuracy with powerful visualization capabilities. Its strengths in rapid prototyping and accessibility reinforce its continued relevance in applied mathematics and scientific computing despite certain limitations.

Keywords

Machine Learning, Applied Mathematics, Math Computing, Differential Equations, MATLAB, Computers to Solve PDEs.

Introduction

Ever since early days of computing, the use of computers to solve scientific or physical problems has been a driving face in the effort to advance the field of computing. As early as igloos, one can problems ([CF63,KA68]);. These early efforts mostly failed because of the lack of computing power and limitations of early programming languages.

While scientific applications still drive the need for move and move powerful hardware, they also demand the development of move and move powerful software systems. Scientific computing as practical today ranges from writing special-case solvers in assembly languages FORTRAN-type language to applying advanced software systems with multimedia interfaces that quickly compute solutions using parattel wa-chines on a network and provide virtual reality

Journal of University Studies for inclusive Research (USRIJ)
مجلة الدراسات الجامعية للبحوث الشاملة
ISSN: 2707-7675

visualization to help understand the software phenomena under study. This thesis studies the problem of developing such advanced software platforms for scientific applications whose math MATLAB is widely used in all applied mathematics, in education and research at universities, and in the industry. MATLAB stands for MATrix Laboratory and the software is built up around vectors and matrices. This makes the software particularly useful for linear algebra hut MATLAB is also a great tool for solving algebraic and differential equations and for numerical integration. MATLAB has powerful graphic tools and can produce rice pictures in both 2D and 3D. it is also a programming language, and is one of the easiest programming languages for writing mathematical programs. MATLAB also has some tool boxes useful for signal processing, image processing, optimization, etc.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

-Math and computation

-Algorithm development

-Modeling, simulation, and prototyping

-Data analysis, exploration, and visualization

-Scientific and engineering graphics

-Application development, including Graphical User Interface building

Journal of University Studies for inclusive Research (USRIJ)
مجلة الدراسات الجامعية للبحوث الشاملة
ISSN: 2707-7675

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or Fortran.ematical models are based on partial differential equations (PDES).

Example .

Consider the nonlinear system of partial differential equations

$u_{1t} = u_{1xx} + u_1(1 - u_1 - u_2)$

$u_{2t} = u_{2xx} + u_2(1 - u_1 - u_2)$,

$u_{1x}(t, 0) = 0; \quad u_1(t, 1) = 1$

$u_2(t, 0) = 0; \quad u_{2x}(t, 1) = 0,$

$u_1(0, x) = x^2$

$$u_2(0, x) = x(x - 2). \qquad (1.2)$$

(This is a non-dimensionalized form of a PDE model for two competing populations.) As

with solving ODE in MATLAB, the basic syntax for solving systems is the same as for

Journal of University Studies for inclusive Research (USRIJ)
مجلة الدراسات الجامعية للبحوث الشاملة
ISSN: 2707-7675

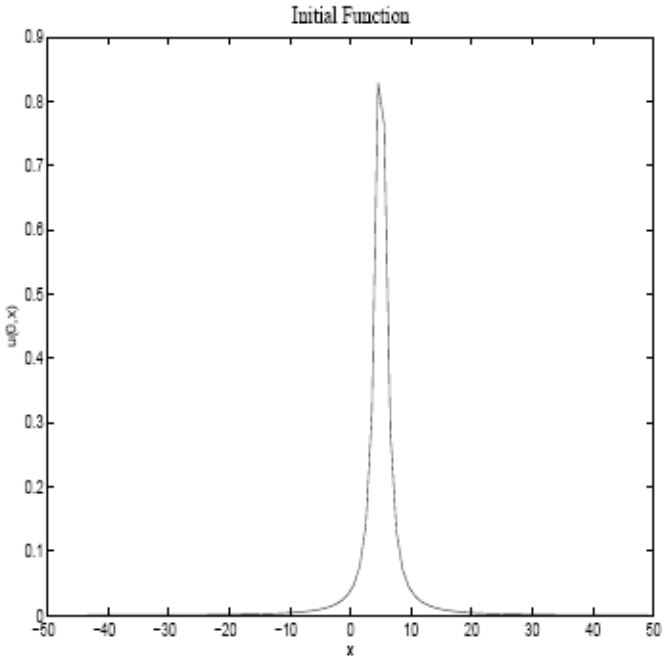

Figure 1.3: Initial Condition for Example 1.2.

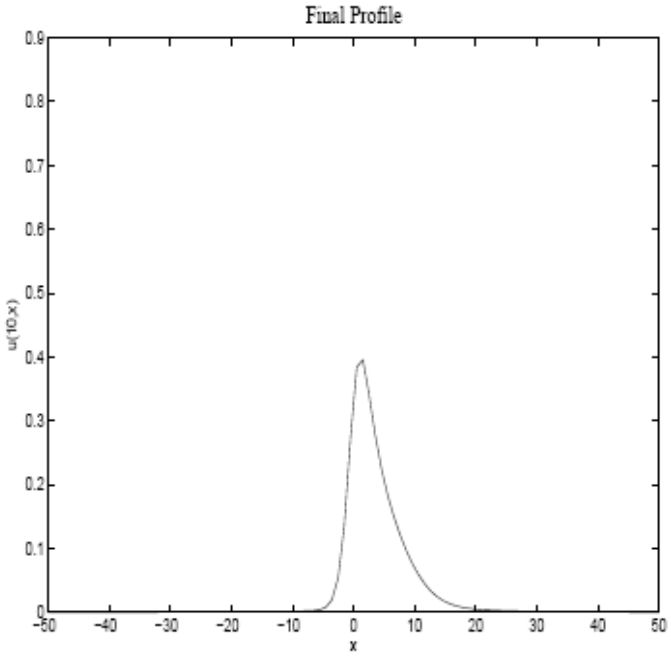

Figure 1.4: Final profile for Example 1.2 solution.

Journal of University Studies for inclusive Research (USRIJ)
مجلة الدراسات الجامعية للبحوث الشاملة
ISSN: 2707-7675

solving single equations, where each scalar is simply replaced by an analogous vector. In

particular, MATLAB specifies a system of n PDE as

$$c_1(x,t,u,u_x)u_{1t} = x^{-m}\frac{\partial}{\partial x}\left(x^m b_1(x, t, u, u_x)\right) + s_1(x, t, u, u_x)$$

$$c_2(x,t,u,u_x)u_{2t} = x^{-m}\frac{\partial}{\partial x}\left(x^m b_2(x, t, u, u_x)\right) + s_2(x, t, u, u_x)$$

$$c_n(x,t,u,u_x)u_{nt} = x^{-m}\frac{\partial}{\partial x}\left(x^m b_n(x, t, u, u_x)\right) + s_n(x, t, u, u_x)$$

(observe that the functions c_k, b_k, and s_k can depend on all components of u and u_x) with

boundary conditions

$p_1(x_l, t, u) + q_1(x_l, t) \cdot b_1(x_l, t, u, u_x) = 0$

$p_1(x_r, t, u) + q_1(x_r, t) \cdot b_1(x_r, t, u, u_x) = 0$

$p_2(x_l, t, u) + q_2(x_l, t) \cdot b_2(x_l, t, u, u_x) = 0$

$p_2(x_r, t, u) + q_2(x_r, t) \cdot b_2(x_r, t, u, u_x) = 0$

...

$p_n(x_l, t, u) + q_n(x_l, t) \cdot b_n(x_l, t, u, u_x) = 0$

$p_n(x_r, t, u) + q_n(x_r, t) \cdot b_n(x_r, t, u, u_x) = 0,$

and initial conditions

$u_1(0, x) = f_1(x)$

$u_2(0, x) = f_2(x)$

...

$u_n(0, x) = f_n(x).$

Journal of University Studies for inclusive Research (USRIJ)
مجلة الدراسات الجامعية للبحوث الشاملة
ISSN: 2707-7675

In our example equation, we have

$$c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}: \quad \mathrm{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}: \quad \mathrm{s} = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} u_1(1-u_1-u_2) \\ u_2(1-u_1-u_2) \end{pmatrix}$$

which we specify with the MATLAB M-file *eqn2.m*.

```
function [c,b,s] = eqn2(x,t,u,DuDx)
%EQN2: MATLAB M-file that contains the coefficents for
%a system of two PDE in time and one space dimension.
c = [1; 1];
b = [1; 1] .* DuDx;
s = [u(1)*(1-u(1)-u(2)); u(2)*(1-u(1)-u(2))];
```

For our boundary conditions, we have

$$P(0,t,u) = \begin{pmatrix} P_1 \\ P_2 \end{pmatrix} = \begin{pmatrix} 0 \\ u_2 \end{pmatrix}: \qquad \mathrm{q}(0,\mathrm{t}) = \begin{pmatrix} \mathrm{q}_1 \\ \mathrm{q}_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$P(1,t,u) = \begin{pmatrix} P_1 \\ P_2 \end{pmatrix} = \begin{pmatrix} u_1 - 1 \\ 0 \end{pmatrix}: \qquad \mathrm{q}(1,\mathrm{t}) = \begin{pmatrix} \mathrm{q}_1 \\ \mathrm{q}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

which we specify in the function M-file *bc2.m*.

```
function [pl,ql,pr,qr] = bc2(xl,ul,xr,ur,t)
%BC2: MATLAB function M-file that defines boundary conditions
%for a system of two PDE in time and one space dimension.
pl = [0; ul(2)];
ql = [1; 0];
```

Journal of University Studies for inclusive Research (USRIJ)
مجلة الدراسات الجامعية للبحوث الشاملة
ISSN: 2707-7675

```
pr = [ur(1)-1; 0];
qr = [0; 1];
```

For our initial conditions, we have

$u_1(0, x) = x^2$

$u_2(0, x) = x(x - 2)$,

which we specify in the function M-file *initial2.m*.

```
function value = initial2(x);
%INITIAL2: MATLAB function M-file that defines initial conditions
%for a system of two PDE in time and one space variable.
value = [x^2; x*(x-2)];
```

We solve equation (1.2) and plot its solutions with *pde2.m* (see Figure 1.5).

```
%PDE2: MATLAB script M-file that solves the PDE
%stored in eqn2.m, bc2.m, and initial2.m
m = 0;
x = linspace(0,1,10);
t = linspace(0,1,10);
sol = pdepe(m,@eqn2,@initial2,@bc2,x,t);
u1 = sol(:,:,1);
u2 = sol(:,:,2);
subplot(2,1,1)
surf(x,t,u1);
title('u1(x,t)');
xlabel('Distance x');
ylabel('Time t');
subplot(2,1,2)
```

Journal of University Studies for inclusive Research (USRIJ)
مجلة الدراسات الجامعية للبحوث الشاملة
ISSN: 2707-7675

```
surf(x,t,u2);
title('u2(x,t)');
xlabel('Distance x');
ylabel('Time t');
```

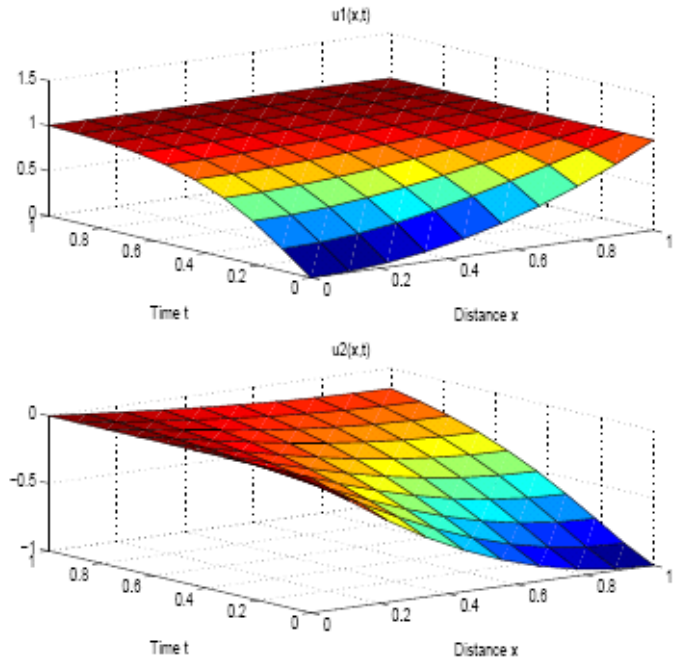

Figure: Mesh plot of solutions for Example.

Objective

The aim is to demonstrate the effectiveness of MATLAB in solving nonlinear partial differential equations through the implementation and analysis of a reaction–diffusion system.

Methodology

Journal of University Studies for inclusive Research (USRIJ)
مجلة الدراسات الجامعية للبحوث الشاملة
ISSN: 2707-7675

- ❖ The nonlinear reaction–diffusion system representing two competing populations was formulated as a coupled system of parabolic partial differential equations with specified initial and boundary conditions.
- ❖ MATLAB's built-in solver was employed to obtain numerical solutions by defining coefficient functions, boundary conditions, and initial states within separate function files.
- ❖ The spatial and temporal domains were discretized, and the solver was applied to generate solution profiles across the defined grid.
- ❖ The numerical outcomes were then visualized using MATLAB's plotting tools, including initial distributions, final steady-state profiles, and mesh plots to capture the complete spatio-temporal dynamics of the system.

Results:

We prove using this example that MATLAB is :

- Mathematical/Scientific computing
- Allows you to rapidly prototype concepts
- Designed completely around mathematical computing
- Invaluable for signal processing
- Incredibly broad array of useful libraries
- Simplest and most concise language for anything involving matrix operations

Journal of University Studies for inclusive Research (USRIJ)
مجلة الدراسات الجامعية للبحوث الشاملة
ISSN: 2707-7675

- Works very well for anything that is simply represented as a numeric feature matrix
- Huge pain to use for anything that isn't simply represented as a numeric feature matrix
- Lacking a good open source ecosystem
- MATLAB, R,Python & Octave are all becoming sought after skills given the role they play as tools to resolve Machine Learning Problems. Note however that they are not just languages. They also represent a set of libraries/packages which matter more when choosing one of them than the difference in syntax. In case of Matlab, it may also cost you money to use those packages/libs.

Abbreviations:

Matlab : Matrix Laboratory

PDE: Partial Differential Equation

PDEs: Partial Differential Equations

References

1. Kant, E., Daube, F., MacCregur, W., & Wald, J. (1992). Knowledge-based program generation for mathematical modeling. In E. N. Houstis, J. R. Rice, & R.

Vichnevetsky (Eds.), Expert systems for scientific computing (pp. 371‑392). North-Holland. 51276 Cited by: 544

2. Koelbel, C. H., Loveman, D. B., Schreiber, R. S., Steele, G. L., Jr., & Zosel, M. E. (1994). The High Performance Fortran handbook. MIT Press. Cited by: 9

3. Kamel, M. S., Ma, K. S., & Enright, W. H. (1993). ODEXPERT: An expert system to select numerical solvers for initial value ODE systems. ACM Transactions on Mathematical Software, 19(1), 44‑62. https://doi.org/10.1145/151271.151276 Cited by: 544

4. König, S., & Ullrich, C. (1990). An expert system for the economical application of self-validating methods for linear equations. In E. N. Houstis, J. R. Rice, & R. Vichnevetsky (Eds.), Intelligent mathematical software systems (pp. 195‑220). North-Holland.

5. MacNeal-Schwendler Corporation. (1991). MSC/NASTRAN user’s manual (Vol. 1). Author.