**A First Guide to Computational Modelling in Physics**

This innovative text helps demystify numerical modelling for early-stage physics and engineering students. It takes a hands-on, project-based approach, with each project focusing on an intriguing physics problem taken from classical mechanics, electrodynamics, thermodynamics, astrophysics, and quantum mechanics. To solve these problems, students must apply different numerical methods for themselves, building up their knowledge and practical skills organically. Each project includes a discussion of the fundamentals, the mathematical formulation of the problem, an introduction to the numerical methods and algorithms, and exercises, with solutions available to instructors. The methods presented focus primarily on differential equations, both ordinary and partial, as well as basic mathematical operations. Developed over many years of teaching a computational modelling course, this stand-alone book equips students with an essential numerical modelling toolkit for today's data-driven landscape, and gives them new ways to explore science and engineering.

Paweł Scharoch is University Professor within the Department of Semiconductor Materials Engineering at Wrocław University of Science and Technology, Poland. He has also held positions at Durham University, the Fritz Haber Institute, Berlin, and at Orange Labs Poland. His principal work is on structural and electronic properties of atomic systems. Professor Scharoch also has around thirty years of experience in teaching various courses in general physics and computational physics.

Maciej P. Polak holds a PhD in Physics from Wrocław University of Science and Technology and is currently a dedicated researcher at University of Wisconsin–Madison's Department of Materials Science and Engineering. He works on first-principles modelling of the electronic band structure of highly mismatched semiconductor alloys for their use in opto-electronics and metals for space applications devices. With over thirty published peer-reviewed articles, he consistently strives to push the boundaries in scientific exploration.

Radosław Szymon, MSc, graduated with honours from Wrocław University of Science and Technology in 2022. He is currently pursuing a PhD in semiconductor technology at the same institution, supported by the prestigious Pearl of Science grant. He also enjoys conducting physics simulations, in particular in electromagnetism and quantum mechanics.

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE

Cambridge University Press & Assessment
978-1-009-41312-1 — A First Guide to Computational Modelling in Physics
Pawel Scharoch , Maciej P. Polak , Radosław Szymon ,
Software developed by Katarzyna Holodnik-Malecka
Frontmatter
More Information

# A First Guide to Computational Modelling in Physics

**Paweł Scharoch**
Wrocław University of Science and Technology

**Maciej P. Polak**
University of Wisconsin–Madison

**Radosław Szymon**
Wrocław University of Science and Technology

Software developed by

**Katarzyna Hołodnik-Małecka**
Wrocław University of Science and Technology

CAMBRIDGE UNIVERSITY PRESS

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE

Cambridge University Press & Assessment
978-1-009-41312-1 — A First Guide to Computational Modelling in Physics
Pawel Scharoch , Maciej P. Polak , Radosław Szymon ,
Software developed by Katarzyna Holodnik-Malecka
Frontmatter
More Information

CAMBRIDGE
UNIVERSITY PRESS

Shaftesbury Road, Cambridge CB2 8EA, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre,
New Delhi – 110025, India

103 Penang Road, #05–06/07, Visioncrest Commercial, Singapore 238467

Cambridge University Press is part of Cambridge University Press & Assessment,
a department of the University of Cambridge.

We share the University's mission to contribute to society through the pursuit of
education, learning and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781009413121

DOI: 10.1017/9781009413138

© Paweł Scharoch, Maciej P. Polak and Radosław Szymon 2024
Software components © Katarzyna Hołodnik-Małecka 2024

This publication is in copyright. Subject to statutory exception and to the provisions
of relevant collective licensing agreements, no reproduction of any part may take
place without the written permission of Cambridge University Press & Assessment.

First published 2024

*A catalogue record for this publication is available from the British Library*

*A Cataloging-in-Publication data record for this book is available from the Library of Congress*

ISBN 978-1-009-41312-1 Hardback
ISBN 978-1-009-41310-7 Paperback

Cambridge University Press & Assessment has no responsibility for the persistence
or accuracy of URLs for external or third-party internet websites referred to in this
publication and does not guarantee that any content on such websites is, or will
remain, accurate or appropriate.

© in this web service Cambridge University Press & Assessment www.cambridge.org

# Contents





© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE

Cambridge University Press & Assessment
978-1-009-41312-1 — A First Guide to Computational Modelling in Physics
Pawel Scharoch , Maciej P. Polak , Radosław Szymon ,
Software developed by Katarzyna Holodnik-Malecka
Frontmatter
More Information





© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE
Cambridge University Press & Assessment
978-1-009-41312-1 — A First Guide to Computational Modelling in Physics
Pawel Scharoch , Maciej P. Polak , Radosław Szymon ,
Software developed by Katarzyna Holodnik-Malecka
Frontmatter
More Information





© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE

Cambridge University Press & Assessment
978-1-009-41312-1 — A First Guide to Computational Modelling in Physics
Pawel Scharoch , Maciej P. Polak , Radosław Szymon ,
Software developed by Katarzyna Holodnik-Malecka
Frontmatter
More Information





© in this web service Cambridge University Press & Assessment www.cambridge.org

# Preface

Numerical modelling is a relatively recent and powerful scientific tool that has experienced remarkable development since the mid-twentieth century, fuelled by advancements in computer technology. Today, it is difficult to find a field in science or engineering where computational methods do not play a crucial role. The list of benefits is extensive, including new opportunities such as predicting the properties of physical systems, exploring properties that are not experimentally accessible, obtaining quantitative data that require massive amounts of mathematical operations, processing vast amounts of data, and facilitating machine learning, among others. These new opportunities are accompanied by the relative ease of application, rapid acquisition of valuable results, and cost-effectiveness. As a result of these features, computational methods have become an independent scientific tool, complementing experiment and theory. On the one hand, they extend mathematical modelling and could not exist without it; on the other hand, they establish their own methodology, often resembling experimental work through the use of virtual systems. Computational methods have proven to be invaluable in supporting experimental research, technological advancements, and even theoretical physics, where mathematical models often require extensive numerical calculations to yield results in the form of quantitative data. Considering the aforementioned points, it is clear that various aspects of numerical modelling should be an integral part of academic education, particularly in the fields of science and engineering. In fact, universities worldwide have been offering such educational components for years, often as independent specialisations, such as Applied Computer Science or Big Data. These specialisations focus on diverse aspects of computer applications in science and technology, with an emphasis on numerical modelling. This course is designed to address the need for numerical modelling education. Its level is tailored so that first-year physics or engineering students can fully engage with the material. It is crucial to introduce these methods early in higher education, as they continue to grow in popularity. The course employs a project-oriented



© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE
Cambridge University Press & Assessment
978-1-009-41312-1 — A First Guide to Computational Modelling in Physics
Pawel Scharoch , Maciej P. Polak , Radosław Szymon ,
Software developed by Katarzyna Holodnik-Malecka
Frontmatter
More Information



teaching approach, as suggested by the title *A First Guide to Computational Modelling in Physics*. Rather than systematically covering numerical methods, the course introduces them as tools necessary for solving specific problems. This approach makes learning the methods more engaging and goal-oriented. Aside from the 'First Steps' section, which teaches students numerical methods for basic mathematical operations (finding zeros and extrema of a function), the course is divided into eight basic and six advanced projects. Each basic project begins with an introduction to the necessary physics background, followed by a presentation and discussion of a specific problem and its mathematical description. Next, the required numerical methods and algorithms are introduced, and finally, a set of exercises is proposed. Advanced projects build upon the previously introduced physics background and provide opportunities for students to conduct their first computational investigations, allowing for the realisation of individual research scenarios. This book is designed as a stand-alone resource, containing all the necessary materials for the completion of each project. While there is no strict requirement to use other sources, students are strongly encouraged to expand their knowledge base by consulting additional literature, as this is an inherent aspect of higher education.

The methods presented in this book primarily focus on differential equations, both ordinary (ODE) and partial (PDE), while also covering basic mathematical operations, derivatives, and quadrature, as introduced in the second project (Diffraction on a Slit). The book addresses typical problems such as initial value problems (IVP), boundary value problems (BVP), and eigenvalue problems (EVP). Most of the methods discussed are based on variable discretisation and recursive algorithms for ODEs. For PDEs, the methods of finite difference (FD) and finite elements (FE) are explained, along with selected matrix methods for solving systems of equations (FD) and chosen iterative optimisation methods (FE). In the case of PDEs, system symmetry is utilised to reduce dimensionality from 3D to 1D, simplifying implementation and allowing projects to be completed within reasonable time constraints.

An additional benefit of this course is the effective learning of physics, as it offers strong motivation for revisiting and expanding upon one's knowledge. Each project is prefaced by a brief presentation of the relevant physical background. While these presentations are not exhaustive, they are designed to be coherent, provide sufficient information for project execution, and inspire deeper study.

The authors would like to extend their gratitude to Anna Latosinska for her creative review of the and to Dawid Dworzanski for developing the project 'Evolution of a Wave Function in a 1D Quantum Well',

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE
Cambridge University Press & Assessment
978-1-009-41312-1 — A First Guide to Computational Modelling in Physics
Pawel Scharoch , Maciej P. Polak , Radosław Szymon ,
Software developed by Katarzyna Holodnik-Malecka
Frontmatter
More Information



and to Kamil Wrzos and Piotr Tokarczyk for their contribution to the project ‘Hydrogen Star’. Additionally, they acknowledge the significant contributions of Technical Physics and Quantum Engineering students from the Faculty of Fundamental Problems of Technology at Wroclaw University of Science and Technology, who contributed to development and verification of computer codes in C++ and Python.

© in this web service Cambridge University Press & Assessment www.cambridge.org

# How to Use the Book

Students participating in this course are expected to possess a general understanding of physics at an academic level, be familiar with a chosen operating system, know how to use a selected graphics application, and have basic programming skills in a chosen language.

Before embarking on the projects, it is recommended that students read the 'First Steps' chapter and complete the exercises therein. The basic mathematical operations discussed in this chapter (finding zeros and extrema of a function) are relatively simple yet highly useful. Completing these exercises will familiarise students with the working environment, including the computer, operating system, programming environment, and graphical tools, thereby making it easier to work on subsequent projects. The first eight projects are *Basic*, each containing four sections: 1. Physical Background, 2. The Problem, 3. Numerical Methods, and 4. Exercises. Project 4 includes an additional section, 'Reduction of a Single Planet Motion in a Central Field to 1D'. Advanced projects (9–14) build upon the physical background introduced earlier and typically contain only problem definitions, occasionally featuring a description of the numerical algorithm if not previously introduced, and exercises. When working on basic projects, students should begin by reviewing or learning the physics background and familiarising themselves with the physical system under consideration, along with its mathematical description (Sections 1 and 2). If necessary, students should consult additional sources to address any difficulties in understanding, although the material provided in the book should be sufficient for completing the projects. The next section, 'Numerical Methods', is crucial to achieving the primary goal of the course. In this section, numerical algorithms are either fully derived or their underlying concepts are explained to facilitate their conscientious use and help prevent errors arising from the digital character of analysis. Students are also encouraged to derive numerical formulas on their own, following the given ideas (which often appear



© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE

Cambridge University Press & Assessment
978-1-009-41312-1 — A First Guide to Computational Modelling in Physics
Pawel Scharoch , Maciej P. Polak , Radosław Szymon ,
Software developed by Katarzyna Holodnik-Malecka
Frontmatter
More Information



as exercises). Understanding the rationale behind a specific numerical formula is more important than being able to derive it. The final section contains exercises that should be completed using a computer program.

Exercises in this course are divided into three categories: *basic* (which should be completed by all students), *supplementary* (designed to deepen and strengthen knowledge and skills), and *advanced* (intended for students with a particular interest in the subject matter). Within the basic exercises, emphasis is placed on crucial steps in computational work:

1. **Testing the program**. This involves running the program for cases where the results are known from other sources (e.g., analytical solutions) and verifying that the program reproduces these results. This step is typically associated with the initial runs of the program after the code has been written and cleared of any errors that may have arisen.
2. **Testing the effect of control parameters and establishing their correct values**. This step is an essential aspect of computational work. The creation and implementation of numerical algorithms always involve the digitisation of analog formulas, which introduces technical parameters controlling the procedure, such as time steps or grid parameters for spatial variables. Incorrect values for these parameters can adversely affect the results and lead to errors. A fundamental method for assessing the impact of control parameters is the convergence test, which examines results as a function of a given parameter value. The issue becomes more complex when multiple control parameters are involved, as correlations between them may exist. However, the aim of this course is mostly to highlight the problem. Methods for evaluating the influence of control parameters by examining physical quantities (e.g., conservation laws) will be further described in the book.
3. **Virtual experiments**. This stage represents the primary objective of all preceding work. Once we are confident that the program is functioning correctly and have determined the appropriate values for control parameters, we can begin investigating the properties of a system. This process usually involves conducting experiments on a virtual system, observing its behaviour under various physical conditions (e.g. the motion of planets in a planetary system with different initial conditions).

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE

Cambridge University Press & Assessment
978-1-009-41312-1 — A First Guide to Computational Modelling in Physics
Pawel Scharoch , Maciej P. Polak , Radosław Szymon ,
Software developed by Katarzyna Holodnik-Malecka
Frontmatter
More Information



Preliminary versions of the codes can be found in the online repository.

**Figure 0.1** Internet repository; *https://wppt.pwr.edu.pl/PhysModelCodes*

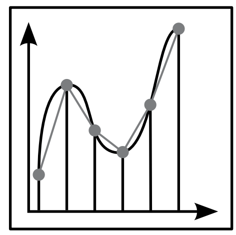
It is recommended that students write the first two to three codes in their chosen programming language to test and practice their programming skills and, if necessary, expand their knowledge. Model codes have been prepared in Python, complete with detailed descriptions (Python Notebooks), but Fortran and C++ versions are also available if preferred. The programs have been designed for clarity rather than optimisation in terms of efficiency. Some exercises require code modifications, providing an opportunity to practice programming skills further.

In addition to the eight basic projects, the book offers six advanced projects. These projects allow students to apply the methods learned throughout the course to the analysis of more complex physical systems. It is recommended that students with a particular interest in computational research undertake a chosen advanced project individually. All work throughout the course can be completed using free tools available on the Internet (e.g. `Spyder` for Python, `Force` for FORTRAN, `Gnuplot` for graphics).

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE
Cambridge University Press & Assessment
978-1-009-41312-1 — A First Guide to Computational Modelling in Physics
Pawel Scharoch , Maciej P. Polak , Radosław Szymon ,
Software developed by Katarzyna Holodnik-Malecka
Frontmatter
More Information

# First Steps



During this first computer lab session, students will learn or review the basic elements of their chosen programming language, programming environment, and graphical program. They will be introduced to numerical methods for fundamental mathematical operations, such as finding the roots and the minimum or maximum of a 1D function. Furthermore, students will practice essential programming operations, including loop instructions and tabulating data from 1D and 2D functions.

## Basic Mathematical Operations

## 0.1 Finding Roots of a 1D Function

### 0.1.1 Bisection Method

Let's start from the case where we are sure that a single root of a function $f(x)$ is somewhere within the interval $(x_l, x_r)$ (Figure 0.2). That means we already know the root of the function with an uncertainty of $\Delta x = (x_r - x_l)$. To reduce the uncertainty, we cut the interval from the previous step in half, and find the point in the middle $x_m = (x_r - x_l)/2$. We then get two new intervals, two times shorter than the previous one. However, the root of the function may be present only in one of them (it may happen that $x_m$ is already the root, and this should be checked in the program). In order to identify the one within which the function crosses zero we check the condition $f(x_m) \cdot f(x_r) < 0$ (the function changes its sign). If the condition is true, the root is in the $(x_l, x_r)$ interval, if not, then it is in the other one. We denote the ends of the new interval again by $(x_l, x_r)$ by replacing $x_m = x_l$ (or $x_m = x_r$) and repeat the procedure. After $n$ steps, the length of the domain segment containing zero is reduced by of factor $2^n$ and when it is less than the assumed

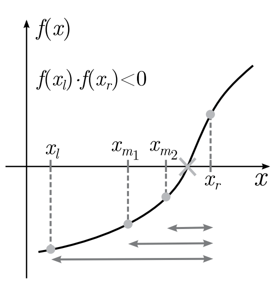



**Figure 0.2** The bisection method



© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE
Cambridge University Press & Assessment
978-1-009-41312-1 — A First Guide to Computational Modelling in Physics
Pawel Scharoch , Maciej P. Polak , Radosław Szymon ,
Software developed by Katarzyna Holodnik-Malecka
Frontmatter
More Information



uncertainty $\epsilon$, that is the condition $\Delta x < \epsilon$ is fulfilled, the whole procedure stops. Alternatively, we can predict how many steps ($n$) are needed from the condition that after $n$ steps the initial interval is reduced by a factor of $2^n$, and perform only this number of steps.

### 0.1.2 Secant and Newton–Rhapson Methods

The bisection method is very safe but not the most efficient. The popular (and often more effective) alternatives are the secant and Newton–Raphson methods (Figure 0.3). The Newton–Raphson method is useful when together with the function $f(x)$ its analytical derivative is also known. Using the derivative we construct a linear approximation of the function $f(x)$ at the point $x_n$. Then we approximate the root of $f(x)$ by the zero of the linear function $(x_{n+1})$, $x_{n+1} = x_n - f(x_n)/f'(x_n)$. This root is treated as the new starting point and the iteration procedure continues until a few subsequent changes in $x_n$ are smaller than the assumed uncertainty $\epsilon$. The algorithm can be applied only if a power series expansion of the function $f(x)$ in the vicinity of its root contains a linear term. Of course, we usually do not know in advance if this is the case. Thus this procedure, although quickly convergent, should be applied with caution. The secant method is similar to Newton–Raphson except that the derivative of a function is found numerically, using, for example a three-point scheme (see Project 2). However, it should be noted that this scheme requires several evaluations of the function $f(x)$ at each iteration step, and this may appear to be less efficient than the safer bisection method.

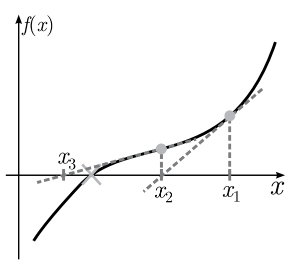



**Figure 0.3** The Newton–Raphson method

## 0.2 Finding Minimum of a 1D Function

### 0.2.1 Golden Section Search

Determining the minimum (or maximum) of a 1D function is a crucial task, as many optimisation methods in multidimensional spaces rely on directional minimisation. It is worth noting that finding the minimum

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE
Cambridge University Press & Assessment
978-1-009-41312-1 — A First Guide to Computational Modelling in Physics
Pawel Scharoch , Maciej P. Polak , Radosław Szymon ,
Software developed by Katarzyna Holodnik-Malecka
Frontmatter
More Information



is equivalent to finding the maximum since changing the sign of the function converts maxima to minima and vice versa. Therefore, our discussion will primarily focus on identifying the minimum.

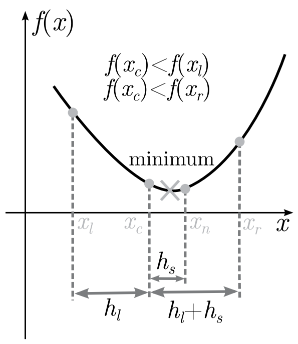



**Figure 0.4** The Golden Section Search

To identify the domain interval containing a minimum in the interval, assuming that there is only one, we need a checking point $x_c$ within the interval $(x_l, x_r)$ (Figure 0.4). We can be sure that the minimum is present in the interval if $f(x_c) < f(x_l)$ and $f(x_c) < f(x_r)$. In constructing the algorithm one should focus on a proper choice of the new, fourth checking point $x_n$. By introducing the fourth point in the interval $(x_l, x_r)$ we obtain two overlapping regions, each defined by three points. We can then identify the one containing the minimum using the condition above. To assure the optimal convergence, the two regions should have the shortest possible and equal length (at each step). This can be achieved if at each step the fourth point splits the longer of the two segments $(x_l, x_c)$ and $(x_c, x_r)$ in golden section ratio, that is $h_l/(h_l + h_s) = h_s/h_l$ ($h_l$ is the longer segment, $h_s$ is the shorter segment). By substituting $x = h_s/h_l$ into the above golden section condition we get the equation $x^2 + x - 1 = 0$, whose one of the solutions $(\sqrt{5} - 1)/2 \approx 0.62$ is the Golden Section proportion, known in art and architecture from ancient times. Thus, after $n$ steps the length of starting interval reduces by factor $(0.62)^n$, which is a little bit slower than in the bisection algorithm of finding the root. The Golden Section Search is an always convergent, safe, and effective method of finding a minimum, but it is not the most efficient.

### 0.2.2 Other Methods

If the analytical form of the first derivative is known, then the minimum (or maximum) can be found by finding a root of the derivative. The derivative can be also approximated by a numerical formula.

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE
Cambridge University Press & Assessment
978-1-009-41312-1 — A First Guide to Computational Modelling in Physics
Pawel Scharoch , Maciej P. Polak , Radosław Szymon ,
Software developed by Katarzyna Holodnik-Malecka
Frontmatter
More Information



The method of parabolas is based on a parabolic approximation, is analogous to the secant method for finding roots, and is often very effective. Using three points, $x_l, x_c, x_r$ we can unambiguously locally approximate the function with a parabola. The position of the parabola's minimum is already an approximation of the function minimum but can be treated as the fourth control point $x_c$. The condition $f(x_c) < f(x_l)$ and $f(x_c) < f(x_r)$ allows to identify the segment containing the minimum. The function is then approximated with a parabola again in the new interval, and the procedure is repeated until a required uncertainty $\epsilon$, in relation to the length of the last interval, is achieved.

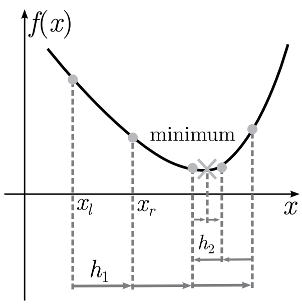



**Figure 0.5** The Simplex method

The 1D Simplex method (Figure 0.5) is the simplest method to search for the minimum which uses a *test window* rather than three points. Starting from a certain point, the domain is scanned in the direction where function decreases, let's say to the right, by moving a window $(x_l, x_r)$ of certain length $h = (x_r - x_l)$. The domain interval containing minimum is found if the function begins to increase, that is $f(x_l) < f(x_r)$. At this point the window length is cut in half and the search continues from the $x_r$ where the increase has been noticed, but in the opposite direction, until the condition $f(x_r) < f(x_l)$ is fulfilled, which, again, is a signal to turn around. The procedure continues until $h < \epsilon$, where $\epsilon$ is the assumed uncertainty. A great advantage of the method is that it can serve for finding interval containing the minimum in general, and then other, more effective methods can be applied to find it precisely.

## 0.3 Exercises

*Obligatory*

1. Using the program FTABLE tabulate your own function and visualise it with a graphical program.

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE

Cambridge University Press & Assessment
978-1-009-41312-1 — A First Guide to Computational Modelling in Physics
Pawel Scharoch , Maciej P. Polak , Radosław Szymon ,
Software developed by Katarzyna Holodnik-Malecka
Frontmatter
More Information



2. Modify the FTABLE code so that it can tabulate a 2D function; set your own function and visualise it.
3. Test the BISEC code by finding the roots of a chosen second-order polynomial and comparing the results with analytical solutions.
4. Find the value of the number $\pi$ as a zero of the $\sin(x)$ function. What precision can you achieve? Explain why.

*Advanced*

1. Write a 1DMINIMUM code that finds the minimum of a 1D function using one of the algorithms presented above (Golden Search, Parabolas or 1D Simplex). Test the program by finding the minimum of a chosen second-order polynomial and compare the results with analytical solution. Find the value of number $\pi$ as a minimum of $\cos(x)$ function. What precision can you achieve? Explain why.

© in this web service Cambridge University Press & Assessment www.cambridge.org