Computational Physics Methods and Algorithms

K. M. Kiran Raj1,2,*, Srivatsa Maddodi1,3 & Yogish Pai U1,4

1Research Scholar, College of Computer and Information Science, Srinivas University, Mangaluru, Karnataka, India.
2Lecturer, Department of Computer Science, Sri Venkataramana Swamy College, Bantwal, Karnataka, India. E-mail:
3Professional 2, DXC Analytics, DXC Technology, Bengaluru, Karnataka, India
4Corresponding author e-mail: kiranraj224@gmail.com

Abstract. Computational Physics plays a major role in solving the physical problems numerically, which cannot be achieved using analytical methods due to the time constraint and complexity of the underlying physical systems. Computational physics combines the theoretical and experimental aspects of conventional scientific study. The advancement in the field of computing and numerical analysis helped computational physics to solve the problems associated with molecular modeling, protein folding, atmospheric science more effectively. In this paper, we review some of the important methods and equations used in Computational Physics in order to solve mathematical problems numerically. Integration, Root finding, Ordinary differential equations, Matrix eigenvalue problems, system of linear equations and Partial differential equations along with some of the well-known methods are briefly discussed. Challenges and applications associated with Computational Physics are also reviewed.

Keywords: Integration, Root finding, Ordinary differential equations, linear equations, Integration, Matrix eigenvalue

1. Introduction
Computational Physics is study and implementation of numerical algorithms and techniques using computer programming. It is a subset of computational science. Computational physics can also be used as tool for solving various numerical problems across various domains like computational biophysics, computational astrophysics, computational chemistry etc. Computational physics encompasses numerical integration, numerical solution of differential equation, simulations etc. Computational physics is often needed when we cannot solve the problems analytically and there are too much of data to be processed. This paper has total 5 sections. Section 1 provides the brief introduction of computational physics. Section 2 Methods and Algorithms describes prominent used in computational physics Section 3 lists Applications and Impact of computational physics. Section 4 covers challenges faced by computational physics. Finally this paper concludes with Section 5 Conclusion.
2. Methods and Algorithms
Algorithms are the set of logical steps for a specific computational problem. These involve two steps in the first step we need to transform a problem or equations into a set of logical steps that a computer can follow and in the second step we inform the computer to perform these logical steps. Figure 2. Computational Physics Methods shows different methods used.

![Figure 2 Computational Physics Methods](image)

2.1. Root Finding
Root finding is a method to find root for the continuations equation f(x) = 0. Root Finding method can be used in different fields. Roots of the mathematical equation in found in successive iterative process [2], where solution of one iteration is given as input to next iteration until solution / roots found is close enough to solution. Figure 3. Represents some of the prominent different types of root finding methods [3] [4] [5] [6].
2.1.1. *Bracketing Method*. Bracketing Method will successively allow shrinking the interval until it is sufficiently close enough to solution/root. Bracket is an interval where solution/root is present. Initial values of brackets are guessed depending on the problem. Bracketing Methods are slower but they are reliable, if there is a solution bracketing methods will converge to it [4].

![Diagram of Root Finding Methods](image)

**Figure 3.** Types of Root Finding Method

2.1.1.1 *Bisection Method*. Within bracketing method, Bisection Method is the simplest method of root finding. If \( f(x) \) is a continuous function in between initial guess of interval \( x_l \) and \( x_u \), where \( x_l \) is the lower bracket and \( x_u \) is the upper bracket. Calculate \( x_r \) by taking average of interval. There are 3 conditions, Condition 1. \( f(x_l)f(x_r)=0 \), then \( x_r \) is root/solution and stop iteration. Condition 2 \( f(x_l)f(x_r)>0 \) then \( x_l=x_r \) and Condition 3 \( f(x_l)f(x_r)<0 \) then \( x_u=x_r \) if condition 2 or 3 is true then return to previous step [2][6]. The iteration can be stopped when value of subsequent \( x_r \) becomes relatively smaller or when approximate error becomes lesser than 1%.

2.1.1.2 *False Position (Regula Falsi)*. False position method is very similar to Bisection method but much faster when compared with it. It will be using geometric information. Initial guessing of interval, checking the conditions and updating of intervals are same. Only difference between bisection and false position method is how \( x_r \) is calculated. The value of function at \( x_l \) and \( x_u \) is connected using straight line. The point where line crosses x axis is \( x_r \).

2.1.2. *Open Method*. Only one value is guessed. Solution of iteration is fed to successive iteration to reach solution which will be very closer to root/solution. Faster compared with the bracketing method but unreliable, Open methods may not converge to solution [4].

2.2. *System of Linear Equation*
In linear equation we will be trying to find solution for unknown variable in an equation. System of linear equation can have 1 or more equation [7].

2.2.1. *Gauss Elimination Method*. The paragraph text follows on from the subsubsection heading but should not be in italic. Carl Friedrich Gauss in 19th century recommended this method. Gauss Elimination method mainly 2 steps Elimination method and Back substitution method. In elimination method unknown variable will be eliminated and in back substitution method the value of the
unknown variable which is found will be substituted in an equation to find different unknown variable. To overcome disadvantages several variation introduced like scaling and pivoting. Gauss-Jordan method is another variation of Gauss Elimination Method [8].

2.2.2. LU Decomposition Method. LU Decomposition Method decomposes matrix A as multiplication of 2 matrices L and U, where L represent lower triangular matrix and U represents upper triangular matrix [9]. It mainly consists of 3 steps decomposition, forward substitution and backward substitution.

\[ AX = B \]  
\[ \text{---Eq(1)} \]

In decomposition step, matrix A is decomposed to matrix L and U. \( A = LU \) Eq(1) can be rewritten as \( LUX = B \). Renaming \( UX = D \). In forward substitution value of D is found by equation \( LD = B \). Value of X can be found by backward substitution \( UX = D \). LU Decomposition is more efficient when forward substitution and backward substitution steps are more efficient. LU Decomposition method decomposition method will not go into iteration which will save considerable amount of time [10].

2.3. Ordinary Differential Equation

Some text. Ordinary differential equations (ODEs) are derived from physical science which proven to be essential approaches to solve physical problems [11]. ODE comprises few ordinary derivatives of a function and a single independent variable (like \( y \))[12]. Classification of ODEs have been done based on the order which has been determined based on the order of the highest derivative exists in any of the equation. The lack of availability of analytical solutions for the wide range of differential equations that emerges in physics and engineering applications makes numerical methods indispensable. Runge-Kutta Methods, Linear Multistep Methods etc are example for the numerical methods used in ODEs.

2.3.1. Runge-Kutta Methods. Runge-Kutta methods aids in numerically assessing solutions to differential equations which would be in the form of \( y' = f(x, y) \) [13]. They were first introduced in 1900 by Carle Runge and Martin Kutta. John Butcher contributed further developments in 1960. Runge-Kutta algorithm is widely used by researchers and scholars since it is known for accuracy when used in applications pertaining to scientific and engineering field. The Runge-Kutta methods is a group of methods with common structure. The Runge–Kutta methods are a collection of implicit and explicit iterative methods and popular due to its simplicity and efficacy. This method can be considered as compelling predictor–corrector methods. It may follow single predictor step and one or more corrector steps [14].

The explicit Runge-Kutta method is not as stable and accurate as implicit Runge–Kutta methods [15]. Some of the well-known explicit methods are Ralston's method, Forward Euler, Heun's method, Kutta's third-order method Explicit midpoint method, Generic second-order method, etc. Backward Euler, Implicit midpoint, Crank-Nicolson method, Diagonally Implicit Runge Kutta methods and Gauss–Legendre methods are some of the implicit methods.

2.4. Integration

Numerical Integration computes the integral of a function using numerical techniques [16]. It involves a wide range of algorithms for calculating the numerical value of a definite integral.

\[ A = \int_{a}^{b} f(x) \, dx \approx \sum_{j} w_{j} f(x_{j}) \]

The area from x axis to curve \( y = f(x) \). The positions of the points, \( x_{j} \), at which heights are calculated are termed abscissae in numerical integration terminology, and the widths, \( w_{j} \), are named weights. To a given degree of accuracy, computing an approximate solution to a definite integral is a real challenge in numerical integration. [17].
2.4.1. Monte Carlo integration. In physics and other areas where these methods are used, the word “Monte Carlo” refers to the use of random numbers. The simplest of a broad range of “Monte Carlo methods” is a Monte Carlo integration. Here uniform random sampling method is used to determine the averages \[ \frac{1}{8} \]. An integral can be stated as an average of the integrand over the range, or volume, of integration, and Monte Carlo integration is based on this fact. Two of the easily applicable methods to multi-dimensional integrals are Monte-Carlo and quasi-Monte Carlo methods.

2.4.2. Romberg’s method. In the field of numerical analysis, the definite integral is estimated using Romberg’s method. Taking a sequence approximate solutions to an integral and calculating an enhanced approximation is an extrapolation technique of Romberg integration. It assumes that the function that is being integrated is sufficiently differentiable[19].

2.5. Partial Differential Equation

Partial Differential Equation (PDE) is one of the methods of computational physics. PDE is a differential equation method that contains unknown multivariate columns along with their partial derivatives [20]. PDE can be used in wide variety of applications such as quantum mechanics, gravitation mechanics, electrodynamics etc. There are three main types of PDE [21].

2.5.1. Parabolic Partial Differential Equation. One Variable has first order derivative and all the other variables have second order derivative. Parabolic PDE are used to define variety of time dependent phenomenon. Examples includes diffusion, time dependent Schrodinger equation, heat equation. Suppose \( u = u(x, t) \) satisfies the second order partial differential equation \[ A u_{xx} + B u_{xt} + C u_{tt} + D u_x + E u_t + F u = G \] In Which \( A,...,G \) are functions the equation is said to be parabolic if

\[
a(x,t)c(x,t) - \frac{b(x,t)^2}{4} = 0
\]

2.5.1.1 Diffusion Equation. It is a kind of Parabolic differential equation in which macroscopic behavior of micro particles in Brownian motion is described [22]. It describes evolution in time like particle density, temperature gradient etc.

2.5.1.2 Elliptic Partial Differential Equation. All the variables have second order derivative with same signs. Examples includes Poisson, time independent Schroedinger equation, Laplace equation. Suppose \( u = u(x, t) \) satisfies the second order partial differential equation \[ A u_{xx} + B u_{xt} + C u_{tt} + D u_x + E u_t + F u = G \] In Which \( A,...,G \) are functions the equation is said to be elliptic if

\[
a(x,t)c(x,t) - \frac{b(x,t)^2}{4} > 0
\]

2.5.1.3 Poisson and Laplace Equation. Poisson and Laplace equation are two order elliptic partial differential equation. Poisson equation is a generalized form of Laplace equation. These equations describe potential field is known one can calculate gravitational field or electrostatic field [23].

2.5.2. Hyperbolic Partial Differential Equation. All the variables have second order derivative with different signs. Example includes wave equations. Suppose \( u = u(x, t) \) satisfies the second order partial differential equation \[ A u_{xx} + B u_{xt} + C u_{tt} + D u_x + E u_t + F u = G \] In Which \( A,...,G \) are functions the equation is said to be hyperbolic if
\[ a(x, t)c(x, t) - \frac{b(x,t)^2}{4} < 0 \]

2.5.2.1 Wave Equation. This is a type of hyperbolic partial differential equation which is used for the description of waves like sound waves or water waves etc. [24].

2.6. Matrix Eigenvalue Problem
An eigenvalue problem is when a linear operation is performed on a vector and the resulting output is the same vector multiplied by a scalar. Typically, a linear operator will have multiple scalar numbers that will be returned for a single vector and can have many vectors that this will happen with. These vectors are called eigen vectors and the scalar numbers are the eigenvalues associated with each eigen vector. Eigen Vector is a non-zero vector and is also called characteristic vector. When a linear transformation is applied it changes at most by a scalar factor. Eigen value is a factor by which eigenvector is scaled. Application of Eigenvalues includes Face recognition, vibration analysis, analysis of financial data etc. Many algorithms/methods can be used to solve the matrix eigenvalue problem.

2.6.1. Power Iteration. The paragraph text follows on from the subsubsection heading but should not be in italic Power iteration is an iterative method and is useful for estimating largest or smallest eigenvalue and its corresponding eigen vector. This technique can also be used to find remaining eigenvalues [25]. Companies like Google use power iteration method for page rank of the webpages in search engine. Twitter uses power iteration for recommending whom to follow.

2.6.2. Jacobi Eigenvalue Algorithm. This is one of the oldest algorithms for computing eigenvalues the algorithm was introduced by Jacobi in the year 1845. Jacobi algorithm is an iterative method for computing eigenvalues and eigen vector for symmetric matrix [26].

2.6.3. Divide and Conquer Eigenvalue Algorithm. This type of algorithm breaks an eigenvalue problem into two problems of equal size then these two problems are solved recursively [27].

2.6.4. Inverse Iteration. This is an iterative method in which an approximate value for eigenvector can be found if an approximate value for the corresponding eigenvalue are already known [28].

3. Applications & Impact
3.1. Understanding of Complex Problem
The main impact of computational physics is development of simulation models of computational physics which helps to understand complex natural phenomenon. In complex systems changes cannot be applied on real world since many real world factors are involved, simulation will provide option of understanding the system without effecting real world. Simulation helps to provide / try many solutions and they will not affect real world because of which we can follow trial and error method. Computational physics allows the development more depth-conscious models for natural phenomena [29].

3.2. Software
The Computational physicist’s for simulation uses high-level programming languages like FOTRAN, C, C++, and Python which in turn developed listed programming languages [29]. Software packages like MATHEMATICA, MAPLE, EXCEL, and MATLAB were improved to support the different needs of researchers in the field of computational physics [29]. Improvement in parallel program codes, Message Passing Interface (MPI Graphics Processing Units (GPUs), CUDA, OpenMP and OpenCL. Improvement in Statistical testing framework like R framework and ecosystem [30].
3.3. *IT Infrastructure*
Simulation of complex problems / natural phenomenon helped to collect new information about systems and understand them. Computational physics opened new fields resulted in availability of new data and improvement of servers, IT related infrastructure, hardware and algorithms with new idea [30] [31].

4. **Challenges**

4.1. **Academics**
Computational physics field is younger than 1 century, people do not know about applications and importance of it. Developing a computational physics curriculum isn’t easy because it includes physics, computer science, and mathematics. Students or faculties may not be interested in specific subjects or fields [29].

4.2. **Simulation**
Deciding whether to use existing program or self-written program, existing model / simulation or new model / simulation is difficult [29] since computational program addresses simple / complex natural phenomenon and factors affecting them are different. The components present in one system will be different to one another which makes harder for building simulation models. Making sure that computational model / simulation that is being tested imitates the natural phenomenon is a major challenge [31].

4.3. **Software and Hardware**
New efficient algorithms are developed to support complexity of computational physics. To develop software or algorithm needs the sufficient information from all fields. Invention / improvement of algorithms are costly and time consuming [30]. To support fast computation, parallel program, time efficient process hardware components should also be improved to support computational physics. Developing / improving hardware is costly, time consuming and requires support of other engineering fields.

4.4. **Interdisciplinary Field**
The quality of theoretical model of computational physics depends on physics, computer science, and mathematics. Because these fields depend on each other, all fields need to improve for better quality. Computational physics is interdisciplinary field [31], because of which finding solution to complex problem is difficult and requires cooperation from researchers from different fields. Creating a common platforms for an interdisciplinary field for exchange of new ideas, information and approaches is difficult [30]. Introduction of new techniques and integrating them in computational physics is difficult since it involves different fields [30] [31].

5. **Conclusion**
An attempt has been made to present the concept of Computational Physics, methods and algorithms. Applications and challenges associated with Computational Physics have been discussed. Attention to development of new algorithms should be given. Although Computational Physics has made tremendous progress in recent years with the help of high-speed computing and cloud technology, there is a need to create a common platform for an interdisciplinary field.

**References**
[1] Wikipedia contributors. (2020, May 21). Computational physics. In Wikipedia, The Free Encyclopedia. Retrieved 03:24, June 2, 2020, from https://en.wikipedia.org/w/index.php?title=Computational_physics&oldid=957917414
[2] Dey, A. (2015). Mathematical Model Formulation and Comparison Study of Various Methods
of Root-Finding Problems. IOSR Journal of Mathematics Ver. III, 11(2), 2278–5728. https://doi.org/10.9790/5728-11236471

[3] Introduction to Numerical Methods/Roots of Equations. (2020, April 16). Wikibooks, The Free Textbook Project. Retrieved 04:34, May 30, 2020 from https://en.wikibooks.org/w/index.php?title=Introduction_to_Numerical_Methods/Roots_of_Equations&oldid=3677012.

[4] Richmond, M. (2003, March 25). Finding Roots by “Open” Methods. Retrieved May 28, 2020, from http://spiff.rit.edu/classes/phys317/lectures/open_root/open_root.html.

[5] Mehtre, V. V., & Sharma, S. (2019). Root Finding Methods: Newton Raphson Method. International Journal for Research in Applied Science and Engineering Technology, 7(11), 411–414. https://doi.org/10.22214/ijiraset.2019.11065.

[6] Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). Numerical Recipes: The Art of Scientific Computing (3rd ed.). Cambridge: CAMBRIDGE UNIVERSITY PRESS. ISBN-13 978-0-511-33555-6.

[7] Wikipedia contributors. (2020, May 7). System of linear equations. In Wikipedia, The Free Encyclopedia. Retrieved 05:20, May 30, 2020, from https://en.wikipedia.org/w/index.php?title=System_of_linear_equations&oldid=955443396.

[8] Megha. (2016). Comparative analysis of gauss elimination and gauss-Jordan elimination. International Journal of Multidisciplinary Education and Research, 1(3), 72–77.

[9] Wikipedia contributors. (2020, May 11). LU decomposition. In Wikipedia, The Free Encyclopedia. Retrieved 05:34, May 30, 2020, from https://en.wikipedia.org/w/index.php?title=LU_decomposition&oldid=956137500.

[10] MZ, A., & K, M. (2018). LU-Decomposition Computerized Method to Solve Linear Programming Problems. Journal of Applied & Computational Mathematics, 07(02). https://doi.org/10.4172/2168-9679.1000391.

[11] Chicone, C., 1999. Ordinary Differential Equations with Applications. New York: Springer.

[12] An Introduction to ordinary differential equations - Math Insight. (n.d). In Mathinsight. Retrieved May 30, 2020, from https://mathinsight.org/ordinary_differential_equation_introduction.

[13] Ordinary differential equations-What's The Difference Between Explicit And Implicit Runge-Kutta Methods?. (n.d). In Mathematics Stack Exchange. Retrieved May 30, 2020, from https://math.stackexchange.com/questions/61638/whats-the-difference-between-explicit-and-implicit-runge-kutta-methods.

[14] Numerical Methods., 2015. Wave Fields in Real Media, 509–573. doi:10.1016/b978-0-08-099999-9.00099-1.

[15] A. Samsudin, N. Rosli, and A. N. Ariffin, “Stability analysis of explicit and implicit stochastic Runge-Kutta methods for stochastic differential equations,” Journal of Physics: Conference Series, vol. 890, no. 1, Article ID 012084, 2017.

[16] Numerical Integration: Romberg Integration. (n.d). In towards data science. Retrieved May 30, 2020, from https://towardsdatascience.com/numerical-integration-romberg-integration-3f54002ab538.

[17] Numerical Integration- wikipedia. (n.d). In Wikipedia. Retrieved May 30, 2020, from https://en.wikipedia.org/wiki/Numerical_integration.

[18] Sandvik, A. W., 2018. Numerical Integration and Monte Carlo Integration. Retrieved May 30, 2020, from http://physics.bu.edu/~py502/lectures2/integration.pdf.

[19] Topic 13.3: Romberg Integration. (n.d). In Ece.uwaterloo Retrieved May 31, 2020, from https://ece.uwaterloo.ca/~dwharder/NumericalAnalysis/13Integration/romberg/complete.html.

[20] Tao Pang. 1999. An introduction to computational physics. (1999). 1st Edition https://doi.org/10.1119/1.19198.
[21] Enzo Tonti: Why starting from differential equations for computational physics? J. Computational. Physics., 257, 1260–1290, doi: 10.1016/j.jcp.2013.08.016, 201
[22] Beghin, Luisa; Garra, Roberto. 2019. "A Note on the Generalized Relativistic Diffusion Equation." Mathematics 7, no. 11: 1009
[23] Y. Y. Li, Y. Zhao, G. N. Xie, D. Baleanu, X. J. Yang, and K. Zhao, “Local fractional Poisson and Laplace equations with applications to electrostatics in fractal domain,” Advances in Mathematical Physics, vol. 2014, Article ID 590574, 5 pages, 2014.
[24] G. C. Ducati & E. Capelas DE Oliveira (1995) The generalized wave equation revisited, International Journal of Mathematical Education in Science and Technology, 26:5, 755-785, DOI:10.1080/0020739950260514
[25] Thomas E. Booth (2006) Power Iteration Method for the Several Largest Eigenvalues and Eigenfunctions, Nuclear Science and Engineering, 154:1, 48-62, DOI:10.13182/NSE05-05
[26] T. Wang et al., "Implementing the Jacobi Algorithm for Solving Eigenvalues of Symmetric Matrices with CUDA," 2012 IEEE Seventh International Conference on Networking, Architecture, and Storage, Xiamen, Fujian, 2012, pp. 69-78, DOI:10.1109/NAS.2012.12.
[27] D.R. Smith. The design of divide and conquer algorithms Sci. Comput. Program., 5 (1) (1985), pp. 37-58
[28] Ipsen, Ilse. (1997). Computing an Eigenvector With Inverse Iteration. SIAM Review. 39. DOI: 10.1137/S0036144596300773.
[29] Sadiku, M. N. O., Shadare, A. E., & Musa, S. M. (2017). Computational Physics: An Introduction. International Journal of Engineering Research, 6(9), 427. https://doi.org/10.5958/2319-6890.2017.00054.x
[30] Hamacher, K. (2013). Grand challenges in computational physics. Frontiers in Physics, 4, 1–2. https://doi.org/10.1021/ct8001046
[31] Klingenberg, C. (2013). Grand challenges in computational physics. Frontiers in Physics, 1, 1–2. https://doi.org/10.3389/fphy.2013.00002