

Article

New Multi-Step Iterative Methods for Solving Systems of Nonlinear Equations and Their Application on GNSS Pseudorange Equations

Kalyanasundaram Madhu ^{1,†} , Arul Elango ^{2,*} , René Jr Landry ² and Mo'tassem Al-arydah ¹

¹ Department of Mathematics, Khalifa University, Abu Dhabi P.O. Box 127788, UAE; kalyanasundaram.mathu@ku.ac.ae (K.M.); motassem.alarydah@ku.ac.ae (M.A.-a)

² LASSENA Laboratory, Department of Electrical Engineering, Ecole de Technologie Supérieure, Montréal, QC H3C 1K3, Canada; renejr.landry@etsmtl.ca

* Correspondence: ganapathy.elango@lassena.etsmtl.ca

† These authors contributed equally to this work.

Received: 2 September 2020; Accepted: 18 October 2020; Published: 22 October 2020



Abstract: A two-step fifth and a multi-step $5 + 3r$ order iterative method are derived, $r \geq 1$ for finding the solution of system of nonlinear equations. The new two-step fifth order method requires two functions, two first order derivatives, and the multi-step methods needs a additional function per step. The performance of this method has been tested with finding solutions to several test problems then applied to solving pseudorange nonlinear equations on Global Navigation Satellite Signal (GNSS). To solve the problem, at least four satellite's measurements are needed to locate the user position and receiver time offset. In this work, a number of satellites from 4 to 8 are considered such that the number of equations is more than the number of unknown variables to calculate the user position. Moreover, the Geometrical Dilution of Precision (GDOP) values are computed based on the satellite selection algorithm (fuzzy logic method) which could be able to bring the best suitable combination of satellites. We have restricted the number of satellites to 4 to 6 for solving the pseudorange equations to get better GDOP value even after increasing the number of satellites beyond six also yields a 0.4075 GDOP value. Actually, the conventional methods utilized in the position calculation module of the GNSS receiver typically converge with six iterations for finding the user position whereas the proposed method takes only three iterations which really decreases the computation time which provide quicker position calculation. A practical study was done to evaluate the computation efficiency index (CE) and efficiency index (IE) of the new model. From the simulation outcomes, it has been noted that the new method is more efficient and converges 33% faster than the conventional iterative methods with good accuracy of 92%.

Keywords: navigation equations; higher-order; convergence order; multi-step method; GNSS; GDOP

MSC: 65H05

1. Introduction

Numerical analysis is a comprehensive subject which is interconnected with applied mathematics, various fields of science and engineering, medical, etc. The most elemental and primal problem in this subject is to find the efficient and precise approximate solution $\mathbf{x}^{(*)} \in D$ of a systems of nonlinear $F(\mathbf{x}) = 0$, where $F : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$, and this type of problem will be solved by the famous Newton's method ($2^{nd}NR$) which has second order convergence [1] by

$$y(\mathbf{x}^{(n)}) = \mathbf{x}^{(n)} - [F'(\mathbf{x}^{(n)})]^{-1}F(\mathbf{x}^{(n)}), \quad n = 0, 1, 2, \dots \quad (1)$$

where $F'(\mathbf{x}^{(n)})$ is the Jacobian matrix of the function and requires the evaluation of one F , one F' per iteration. Traub [2] gave suggestion a multi-point schemes are best one to increase the order of convergence when absent of second derivatives, such methods have given in the literature; see [3–7]. Traub [2] gave a double-step Newton's type method ($3^{rd}TM$) having convergence order 3 by two F , one F' evaluations

$$z(\mathbf{x}^{(n)}) = y(\mathbf{x}^{(n)}) - [F'(\mathbf{x}^{(n)})]^{-1}F(y(\mathbf{x}^{(n)})), \quad (2)$$

where $y(\mathbf{x}^{(n)})$ is given in Equation (1). The two-step Newton's method ($4^{th}NR$) with convergence order four

$$\mathbf{x}^{(n+1)} = y(\mathbf{x}^{(n)}) - [F'(y(\mathbf{x}^{(n)}))]^{-1}F(y(\mathbf{x}^{(n)})), \quad (3)$$

was reconstructed by Noor et al. [6], where F and F' are evaluated two times each. Abad et al. [3] suggested the Newton type methods to get a 3-step 4th order method ($4^{th}ACT$), where two evaluations of F , two F' are used

$$\mathbf{x}^{(n+1)} = y(\mathbf{x}^{(n)}) - [F'(z(\mathbf{x}^{(n)}))]^{-1}F(y(\mathbf{x}^{(n)})). \quad (4)$$

where $z(\mathbf{x}^{(n)})$ is given in Equation (2). Sharma et al. [7] presented a two-step 4th order method ($4^{th}SGS$), where 1 time F and 2 times F' are evaluated per cycle, and it is given here

$$\begin{aligned} \mathbf{x}^{(n+1)} &= \mathbf{x}^{(n)} - \left(-\frac{1}{2}I + \frac{9}{8}[F'(y_1(\mathbf{x}^{(n)}))]^{-1}F'(\mathbf{x}^{(n)}) + \frac{3}{8}[F'(\mathbf{x}^{(n)})]^{-1}F'(y_1(\mathbf{x}^{(n)})) \right) u(\mathbf{x}^{(n)}), \\ y_1(\mathbf{x}^{(n)}) &= \mathbf{x}^{(n)} - \frac{2}{3}u(\mathbf{x}^{(n)}), u(\mathbf{x}^{(n)}) = [F'(\mathbf{x}^{(n)})]^{-1}F(\mathbf{x}^{(n)}). \end{aligned} \quad (5)$$

Babajee et al. [8] gave a two-step 4th order scheme ($4^{th}BCST$), where 1 time F , 2 time F' are evaluated per cycle and its given below

$$\begin{aligned} \mathbf{x}^{(n+1)} &= \mathbf{x}^{(n)} - W(\mathbf{x}^{(n)}) \left(\frac{1}{2}(F'(\mathbf{x}^{(n)}) + F'(y(\mathbf{x}^{(n)}))) \right)^{-1} F(\mathbf{x}^{(n)}), \\ W(\mathbf{x}^{(n)}) &= I - \frac{1}{4}(\tau(\mathbf{x}^{(n)}) - I) + \frac{3}{4}(\tau(\mathbf{x}^{(n)}) - I)^2, \tau(\mathbf{x}^{(n)}) = F'(\mathbf{x}^{(n)})^{-1}F'(y(\mathbf{x}^{(n)})). \end{aligned} \quad (6)$$

Abad et al. [3], a different combination was used to get a three-step fifth order method, where three functions, two Jacobian matrices and their inverses were evaluated, and it is given below

$$\mathbf{x}^{(n+1)} = z(\mathbf{x}^{(n)}) - [F'(y(\mathbf{x}^{(n)}))]^{-1}F(z(\mathbf{x}^{(n)})). \quad (7)$$

Madhu et al. [9] improved double step Newton's method and also developed its multi-step version are given below

$$\begin{aligned} \mathbf{x}^{(n+1)} &= y(\mathbf{x}^{(n)}) - H_1(\mathbf{x}^{(n)})[F'(\mathbf{x}^{(n)})]^{-1}F(y(\mathbf{x}^{(n)})), \\ H_1(\mathbf{x}^{(n)}) &= 2I - \tau(\mathbf{x}^{(n)}) + \frac{5}{4}(\tau(\mathbf{x}^{(n)}) - I)^2, \tau(\mathbf{x}^{(n)}) = [F'(\mathbf{x}^{(n)})]^{-1}F'(y(\mathbf{x}^{(n)})). \end{aligned} \quad (8)$$

Literature Survey

Yang [10] implemented an algebraic high compatible positioning via non iterative method employing direct solution of the double-difference pseudorange equations for solving the GPS double-difference pseudorange equations if two or more GPS receivers operate simultaneously. Pachter et al. [11] improved the performance of estimation under high geometric dilution of precision (GDOP) conditions. In this work, the stochastic modeling algorithm for the GPS pseudorange equations is derived and compared with the conventional ILS algorithm. In order to achieve optimality when the trilateration system of equations becomes over-determined, Li et al. [12] proposed an algorithm which uses the direct linearization technique to reduce the computation time overhead.

Similarly, to investigate land-based radio positioning, Kwang-Soob et al. [13], mathematically derived two-dimensional positioning based on GPS Pseudorange linearized state equation in which the geometry model with respect to triangles are formed using unit-vectors. GPS navigation solution using iterative least absolute deviation approach has been described by Jwoa et al. [14] based on the Least Absolute Deviation (LAD) criterion for estimating navigation solutions since the least square technique is very much sensitive to accuracy and the performance for mitigating GPS multipath errors. In Awange et al. [15], an alternative closed form GPS pseudo-ranging four-point problem P4P in matrix form using multi polynomial resultant and Groebner basis has been implemented in algebraic software such as MATHEMATICA and MAPLE to solve the nonlinear GPS pseudo-ranging four-point equations. In general, the positioning module of the GNSS receiver use conventional Taylor series and Bancroft algorithms for solving non-linear pseudorange equations. Elnaggar [16] presented a modified Taylor series method which linearizes the pseudorange equation by considering four satellite coordinates and pseudorange values for improving the positioning. On the other hand, Abad et al. [3] described a solution for GPS pseudorange equations by solving different iterative methods for four visible satellites.

In this work, we have considered four or more than four visible satellites scenario for linearizing the pseudorange equations for obtaining better GDOP value and one can notice that the computational order calculation is not matching with the theoretical order when we increased the number of satellites more than four, most of the iterative methods does not match with the theoretical order but the new multistep iterative method described here approximately matches with the order and still preserves the accuracy. Initially the set of nonlinear problems are solved using various iterative methods and then the GNSS pseudorange nonlinear equations are solved in the next section. The input GNSS signal simulated from OROLIA simulator has been given as an input to the software based GNSS receiver to find the user position. The proposed iterative technique used in the position calculation module of the GNSS receiver is depicted in Figure 1.

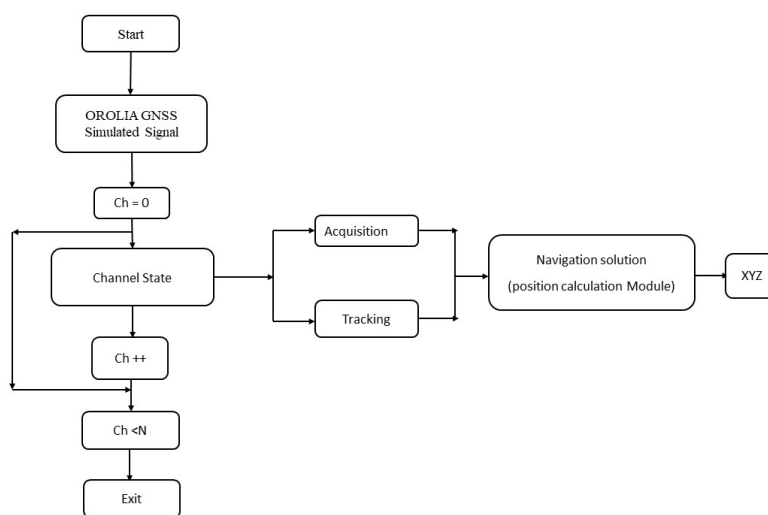


Figure 1. Flowchart of GNSS of positioning.

The input signal is acquired and the number of visible satellites are passed on to the tracking module to lock the code and carrier phase. Further, in the position calculation module the iterative methods are used to find the user position. The conventional way of solving GPS pseudorange equations involve Taylor series and iterative methods which typically take more than 6 iterations if the number of visible satellites are considered to be 4 but in the proposed work, the user position is computed within 3 iterations that significantly reduce the computation time in the position calculation module of the GPS receiver.

The content of the paper is formulated as follows: a new method of 5th order and their multi-step scheme with $5 + 3r$ order, and its convergence analysis are presented in Section 2. In Section 3, numerical test problems are given and it is theoretical convergence order is analyzed and the results are validated through MATLAB simulation. The efficiency index and computational efficiency index are given in Section 4. The GNSS application problem is carried out for new methods and few existing methods are described in Section 5. Test problems and application problem results are discussed in Section 6. The final section is concluded with extending this work on solving position calculation of GNSS multi-constellation receiver.

2. Mathematical Description of New Method and Its Convergence Analysis

2.1. A Two-Step Fifth-Order Method

We present here a new fifth order method ($5^{th}PM$), derived from combining of Traub's method (2) and weight functions. It is given as follows

$$\begin{aligned} \mathbf{x}^{(n+1)} &= \mathbf{y}(\mathbf{x}^{(n)}) - H_1(\mathbf{x}^{(n)})[F'(\mathbf{x}^{(n)})]^{-1}F(\mathbf{x}^{(n)}), \\ H_1(\mathbf{x}^{(n)}) &= \tau(\mathbf{x}^{(n)}) + \frac{1}{4}(\tau(\mathbf{x}^{(n)}) - I)^2, \text{ where } \tau(\mathbf{x}^{(n)}) = [F'(\mathbf{y}(\mathbf{x}^{(n)}))]^{-1}F'(\mathbf{x}^{(n)}), \end{aligned} \quad (9)$$

$\mathbf{y}(\mathbf{x}^{(n)})$ is given in Equation (1), and I is the identity matrix by $n \times n$.

2.2. A Multi-Step $(5 + 3r)^{th}$ -Order Method

Further, we improved the 5th order method to $5 + 3r$ order method by additional one function per each step. It is shown below

$$\begin{aligned} \mathbf{x}^{(n+1)} &= (5 + 3r)^{th}PM(\mathbf{x}^{(n)}) = \psi_r(\mathbf{x}^{(n)}), \\ \psi_m(\mathbf{x}^{(n)}) &= \psi_{m-1}(\mathbf{x}^{(n)}) - H_2(\mathbf{x}^{(n)})[F'(\mathbf{x}^{(n)})]^{-1}F(\psi_{m-1}(\mathbf{x}^{(n)})), \\ H_2(\mathbf{x}^{(n)}) &= \tau(\mathbf{x}^{(n)}) + \frac{1}{2}(\tau(\mathbf{x}^{(n)}) - I)^2, \\ \psi_0(\mathbf{x}^{(n)}) &= 5^{th}PM(\mathbf{x}^{(n)}), \quad m = 1(1)r. \end{aligned} \quad (10)$$

Note that, the spacial case $r = 0$ is the $5^{th}PM$ method is given in (9). It is noted that the $5^{th}PM$ is Cordero et al. [17] they used three arbitrary parameters for obtaining the method. However, we have used the Taylor series technique to propose the $5^{th}PM$ method (9). We can say that the method $5^{th}PM$ is reconstructed of Cordero et al. [17] method.

2.3. Convergence Analysis

Theorem 1. Let F be sufficiently differentiable in an open convex set D , $\mathbf{x}^{(*)} \in D$, the roots of the problem $F(\mathbf{x}) = 0$. $F'(\mathbf{x})$ is continuing function and nonsingular in $\mathbf{x}^{(*)}$. Then the method (9) converging to the $\mathbf{x}^{(*)}$ with convergence order at least five,

$$\begin{aligned} \epsilon^{(n+1)} &= 5^{th}PM(\mathbf{x}^{(n)}) - \mathbf{x}^{(*)} = M_1\epsilon^{(n)5} + O(\epsilon^{(n)6}), \quad \epsilon^{(n)} = \mathbf{x}^{(n)} - \mathbf{x}^{(*)} \\ M_1 &= 2C_2^4 - C_2C_3C_2, \text{ where } C_h = (1/h!)[F'(\mathbf{x}^{(*)})]^{-1}F^{(h)}(\mathbf{x}^{(*)}), h \geq 2(1)n. \end{aligned} \quad (11)$$

Proof. Expanding F and F' around $\mathbf{x}^{(*)}$ at the point $\mathbf{x}^{(n)}$, we obtain

$$F(\mathbf{x}^{(n)}) = F'(\mathbf{x}^{(*)}) \left[\epsilon^{(n)} + C_2\epsilon^{(n)2} + C_3\epsilon^{(n)3} + C_4\epsilon^{(n)4} + C_5\epsilon^{(n)5} \right] + O(\epsilon^{(n)6}), \quad (12)$$

and

$$F'(\mathbf{x}^{(n)}) = F'(\mathbf{x}^{(*)}) \left[I + 2C_2\epsilon^{(n)} + 3C_3\epsilon^{(n)2} + 4C_4\epsilon^{(n)3} + 5C_5\epsilon^{(n)4} \right] + O(\epsilon^{(n)5}). \quad (13)$$

We have

$$[F'(\mathbf{x}^{(n)})]^{-1} = [F'(\mathbf{x}^{(*)})]^{-1} \left[I + X_1 \epsilon^{(n)} + X_2 \epsilon^{(n)2} + X_3 \epsilon^{(n)3} + X_4 \epsilon^{(n)4} \right] + O(\epsilon^{(n)5}), \quad (14)$$

here $X_1 = -2C_2$, $X_2 = -3C_3 + 4C_2^2$, $X_3 = +6C_2C_3 - 8C_2^3 - 4C_4 + 6C_3C_2$ and $X_4 = -5C_5 + 9C_2^2C_3 + 8C_2C_4 + 8C_4C_2 + 16C_2^4 - 12C_2^2C_3 - 12C_3C_2^2 - 12C_2C_3C_2$.

Then

$$\begin{aligned} [F'(\mathbf{x}^{(n)})]^{-1}F(\mathbf{x}^{(n)}) &= \epsilon^{(n)} - C_2\epsilon^{(n)2} + (2C_2^2 - 2C_3)\epsilon^{(n)3} + (-3C_4 - 4C_2^3 + 4C_2C_3 + 3C_3C_2)\epsilon^{(n)4} \\ &+ (6C_3^2 + 8C_2^4 - 8C_2^2C_3 - 6C_2C_3C_2 - 6C_3C_2^2 + 6C_2C_4 + 4C_4C_2 - 4C_5)\epsilon^{(n)5} + O(\epsilon^{(n)6}). \end{aligned} \quad (15)$$

In addition, we have

$$\begin{aligned} y(\mathbf{x}^{(n)}) &= \mathbf{x}^{(*)} + C_2\epsilon^{(n)2} + (-2C_2^2 + 2C_3)\epsilon^{(n)3} + (3C_4 - 4C_2C_3 + 4C_2^3 - 3C_3C_2)\epsilon^{(n)4} \\ &+ (-6C_3^2 - 8C_2^4 + 8C_2^2C_3 + 6C_2C_3C_2 + 6C_3C_2^2 - 6C_2C_4 - 4C_4C_2 + 4C_5)\epsilon^{(n)5}. \end{aligned} \quad (16)$$

Expanding F and F' around $\mathbf{x}^{(*)}$ at $y(\mathbf{x}^{(n)})$ in Taylor series respectively given below

$$\begin{aligned} F(y(\mathbf{x}^{(n)})) &= F'(\mathbf{x}^{(*)}) \left[(y(\mathbf{x}^{(n)}) - \mathbf{x}^{(*)}) + C_2(y(\mathbf{x}^{(n)}) - \mathbf{x}^{(*)})^2 + C_3(y(\mathbf{x}^{(n)}) - \mathbf{x}^{(*)})^3 + \dots \right] \\ &= F'(\mathbf{x}^{(*)}) \left[C_2\epsilon^{(n)2} + 2(-C_2^2 + C_3)\epsilon^{(n)3} + (3C_4 + 5C_2^3 - 4C_2C_3 - 3C_3C_2)\epsilon^{(n)4} \right. \\ &\quad \left. + (-6C_3^2 - 12C_2^4 + 10C_2^2C_3 + 8C_2C_3C_2 + 6C_3C_2^2 - 6C_2C_4 - 4C_4C_2 + 4C_5)\epsilon^{(n)5} \right], \end{aligned} \quad (17)$$

$$F'(y(\mathbf{x}^{(n)})) = F'(\mathbf{x}^{(*)}) \left[I + P_1\epsilon^{(n)2} + P_2\epsilon^{(n)3} + P_3\epsilon^{(n)4} \right] + O(\epsilon^{(n)5}), \quad (18)$$

where

$$P_1 = 2C_2^2, P_2 = 4C_2C_3 - 4C_2^3, P_3 = 8C_2^4 + 6C_2C_4 - 8C_2^2C_3 + 3C_3C_2^2 - 6C_2C_3C_2.$$

Using (14) and (18), we have

$$\begin{aligned} [F'(y(\mathbf{x}^{(n)}))]^{-1}F'(\mathbf{x}^{(n)}) &= I + 2C_2\epsilon^{(n)} + (-2C_2^2 + 3C_3)\epsilon^{(n)2} + (-4C_2C_3 + 4C_4)\epsilon^{(n)3} \\ &+ (5C_5 + 2C_2^2C_3 - 2C_2C_3C_2 - 3C_3C_2^2 + 4C_2^4 - 6C_2C_4)\epsilon^{(n)4} + O(\epsilon^{(n)5}). \end{aligned} \quad (19)$$

Then

$$H_1(\mathbf{x}^{(n)}) = I + 2C_2\epsilon^{(n)} - (C_2^2 - 3C_3)\epsilon^{(n)2} + (-C_2C_3 - 2C_2^3 + 4C_4)\epsilon^{(n)3} + O(\epsilon^{(n)4}). \quad (20)$$

Using (14) and (17), we have

$$\begin{aligned} [F'(\mathbf{x}^{(n)})]^{-1}F(y(\mathbf{x}^{(n)})) &= C_2\epsilon^{(n)2} + (2C_3 - 4C_2^2)\epsilon^{(n)3} + (13C_2^3 - 8C_2C_3 - 6C_3C_2 + 3C_4)\epsilon^{(n)4} \\ &+ (-6C_3^2 - 10C_2^4 + 8C_2^2C_3 + 7C_2C_3C_2 + 6C_3C_2^2 - 6C_2C_4 - 4C_4C_2 + 4C_5)\epsilon^{(n)5} + O(\epsilon^{(n)6}). \end{aligned} \quad (21)$$

Then

$$\begin{aligned} H_1(\mathbf{x}^{(n)})[F'(\mathbf{x}^{(n)})]^{-1}F(y(\mathbf{x}^{(n)})) &= C_2\epsilon^{(n)2} + (2C_3 - 2C_2^2)\epsilon^{(n)3} + (3C_4 + 4C_2^3 - 4C_2C_3 - 3C_3C_2)\epsilon^{(n)4} \\ &+ (-6C_3^2 + 4C_5 - 6C_2C_4 - 4C_4C_2 - 22C_2^4 + \frac{11}{2}C_2C_3C_2 + 8C_2^2C_3 + \frac{39}{2}C_3C_2^2 - 12C_2^2C_3)\epsilon^{(n)5} + O(\epsilon^{(n)6}). \end{aligned} \quad (22)$$

Using (16) and (22) in (9), thus the error is calculated as

$$\epsilon^{(n+1)} = (2C_2^4 - C_2C_3C_2)\epsilon^{(n)5} + O(\epsilon^{(n)6}).$$

Thus, the above equation conform the fifth-order convergence. \square

Theorem 2. Let F be sufficiently differentiable in an open convex set D , $\mathbf{x}^{(*)} \in D$, the roots of the problem $F(x) = 0$. $F'(x)$ is continuing function and nonsingular in $\mathbf{x}^{(*)}$. Then the method (10) converging to the $\mathbf{x}^{(*)}$ with convergence order at least $5 + 3r$.

Proof. A Taylor expansion of $F(\psi_{m-1}(\mathbf{x}^{(n)}))$ around $\mathbf{x}^{(*)}$ yields

$$F(\psi_{m-1}(\mathbf{x}^{(n)})) = F'(\mathbf{x}^{(*)}) \left[(\psi_{m-1}(\mathbf{x}^{(n)}) - \mathbf{x}^{(*)}) + C_2(\psi_{m-1}(\mathbf{x}^{(n)}) - \mathbf{x}^{(*)})^2 + \dots \right] \quad (23)$$

In addition, let

$$H_2(\mathbf{x}^{(n)}) = I + 2C_2\epsilon^{(n)} + 3C_3\epsilon^{(n)2} + (2C_2C_3 - 4C_2^3 + 4C_4)\epsilon^{(n)3} + \dots \quad (24)$$

Using (14) and (24), we have

$$H_2(\mathbf{x}^{(n)})F(\mathbf{x}^{(n)})^{-1} = \left[I + M_2\epsilon^{(n)3} + \dots \right] [F'(\mathbf{x}^{(*)})]^{-1}, \quad M_2 = 2C_2C_3 - 4C_2^3 \quad (25)$$

Using (23) and (25), we obtain

$$\begin{aligned} \psi_m(\mathbf{x}^{(n)}) - \mathbf{x}^{(*)} &= \psi_{m-1}(\mathbf{x}^{(n)}) - \mathbf{x}^{(*)} - H_2(\mathbf{x}^{(n)})F(\mathbf{x}^{(n)})^{-1}F(\psi_{m-1}(\mathbf{x}^{(n)})) \\ &= \psi_{m-1}(\mathbf{x}^{(n)}) - \mathbf{x}^{(*)} - \left[I + M_2\epsilon^{(n)3} + \dots \right] \left[(\psi_{m-1}(\mathbf{x}^{(n)}) - \mathbf{x}^{(*)}) + C_2(\psi_{m-1}(\mathbf{x}^{(n)}) - \mathbf{x}^{(*)})^2 + \dots \right] \\ &= -M_2\epsilon^{(n)3}(\psi_{m-1}(\mathbf{x}^{(n)}) - \mathbf{x}^{(*)}) + \dots \end{aligned} \quad (26)$$

Proceeding by mathematical induction of (26) and using (11), we obtain

$$\psi_r(\mathbf{x}^{(n)}) - \mathbf{x}^{(*)} = M_1(-M_2)^r (\epsilon^{(n)})^{(5+3r)} + O(\epsilon^{(n)})^{(3r+6)}, r \geq 1.$$

Thus, the proposed method conform that $5 + 3r$ order of convergence. \square

3. Numerical Examples

In this section, numerical results are carried out using MATLAB software. The contribute methods are used to approximate the solution of some nonlinear system of equations and they are compared with the results obtained for some existing methods.

$$err_{min} = \|\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}\|_2 < 10^{-100}. \quad (27)$$

In addition, approximated computational order of convergence p_c is used, see ([18])

$$p_c \approx \frac{\log(\|\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}\|_2 / \|\mathbf{x}^{(n)} - \mathbf{x}^{(n-1)}\|_2)}{\log(\|\mathbf{x}^{(n)} - \mathbf{x}^{(n-1)}\|_2 / \|\mathbf{x}^{(n-1)} - \mathbf{x}^{(n-2)}\|_2)}. \quad (28)$$

The following test problems are considered: **Test Problem 1** (TP1) (see [9])

$$F(x) = \begin{cases} x_1 + \exp(x_2) - \cos(x_2) = 0, \\ 3x_1 - x_2 - \sin(x_2) = 0. \end{cases}$$

Here initial points are $x^{(0)} = (1.5, 2)^T$, and the solution is $x^{(*)} = (0, 0)^T$.

Test Problem 2 (TP2) (see [9])

$$F(x) = \begin{cases} x_2x_3 + x_4(x_2 + x_3) = 0, \\ x_1x_3 + x_4(x_1 + x_3) = 0, \\ x_1x_2 + x_4(x_1 + x_2) = 0, \\ x_1x_2 + x_1x_3 + x_2x_3 = 1. \end{cases}$$

Here initial points are $x^{(0)} = (0.5, 0.5, 0.5, -0.2)^T$.

The solution is $x^{(*)} \approx (0.577350, 0.577350, 0.577350, -0.288675)^T$.

Test Problem 3 (TP3) (see [9])

$$F(x) = \begin{cases} x_i x_{i+1} - 1 = 0, & i = 1, 2, \dots, 15, \\ x_{15} x_1 - 1 = 0. \end{cases}$$

Here $x^{(0)} = (1.5, 1.5, 1.5, \dots, 1.5)^T$, and $x^{(*)} = (1, 1, 1, \dots, 1)^T$.

Table 1, shows the results for the above test problems, where M represents the number iterations required for convergence. Here the conventional method $2^{nd}NR$ is converging with 10, 8, and 9 iterations for $TP1$, $TP2$ and $TP3$ respectively, whereas the proposed method $11^{th}PM, r = 2$ is converging with 4, 3, and 4 iterations for $TP1$, $TP2$ and $TP3$ respectively. All other compared methods are converging slower than the new $11^{th}PM, r = 2$ method. Hence, we conclude that $11^{th}PM, r = 2$ method is the efficient one compared with other methods.

Table 1. Numerical results on different iterative methods.

Methods	TP1			TP2			TP3		
	M	err_{min}	p_c	M	err_{min}	p_c	M	err_{min}	p_c
$2^{nd}NR$	10	1.03×10^{-103}	1.99	8	3.92×10^{-145}	2.00	9	8.96×10^{-179}	1.99
$3^{rd}TM$	7	9.65×10^{-104}	2.99	6	8.87×10^{-236}	3.01	6	5.79×10^{-142}	2.99
$4^{th}NR$	6	5.38×10^{-207}	3.99	5	2.98×10^{-291}	4.03	5	8.96×10^{-179}	3.99
$4^{th}BCST$	6	5.05×10^{-139}	3.99	5	3.49×10^{-238}	4.03	5	1.61×10^{-142}	3.99
$4^{th}ACT$	6	2.80×10^{-309}	3.99	5	3.86×10^{-283}	4.03	5	2.03×10^{-203}	3.99
$4^{th}SGS$	6	2.22×10^{-170}	3.99	5	8.89×10^{-257}	4.03	5	6.08×10^{-155}	3.99
$5^{th}PM, r = 0$	5	2.41×10^{-114}	4.85	4	1.46×10^{-131}	5.30	5	0.00×10^{-0}	4.99
$8^{th}PM, r = 1$	5	0.00×10^{-0}	7.99	4	0.00×10^{-0}	7.99	4	1.60×10^{-301}	7.99
$11^{th}PM, r = 2$	4	2.37×10^{-191}	10.96	3	1.89×10^{-118}	11.65	4	0.00×10^{-0}	10.99

4. Efficiency of the Methods

We utilize the efficiency index (EI), $EI = p^{\frac{1}{d}}$ ([1]), p represent that the order, d represent the total functional evaluations. This is the foremost utilized record, and another one we utilize is the computational efficiency index (CE) characterized as $CE = p^{1/(d+op)}$ ([5]), where op is the operations cost per cycle. We review that the number of items and remainders that to be solved using LU factorization ($\frac{1}{3}n^3 + mn^2 - \frac{1}{3}n$), where n is the system size for m linear systems with the same matrix of the coefficient.

Table 2 gives the result of EI and CE expression for the methods discussed above. Figure 2 shows the execution of diverse methods with regard to EI and CE . It is identified that the new methods perform better than other methods for $n \geq 2$. Figure 3 displays the performance of the proposed $(5 + 3r)PM$ method with respect to EI and CE where $r \geq 1$. When the size of system n and step size r increase then the EI and CE decrease respectively, we noted here, that new methods yield good EI and CE with $n \leq 6$ and $r \leq 3$.

Table 2. Comparison of EI and CE.

Method	EI	CE
2 nd NR	$2 \frac{1}{n+n^2}$	$2 \frac{1}{\frac{1}{3}n^3+2n^2+\frac{2}{3}n}$
3 rd TM	$3 \frac{1}{2n+n^2}$	$3 \frac{1}{\frac{1}{3}n^3+3n^2+\frac{5}{3}n}$
4 th NR	$4 \frac{1}{2n+2n^2}$	$4 \frac{1}{\frac{2}{3}n^3+4n^2+\frac{4}{3}n}$
4 th BCST	$4 \frac{1}{n+2n^2}$	$4 \frac{1}{\frac{2}{3}n^3+5n^2+\frac{1}{3}n}$
4 th ACT	$4 \frac{1}{2n+2n^2}$	$4 \frac{1}{\frac{2}{3}n^3+5n^2+\frac{4}{3}n}$
4 th SGS	$4 \frac{1}{n+2n^2}$	$4 \frac{1}{\frac{2}{3}n^3+5n^2+\frac{4}{3}n}$
5 th PM	$5 \frac{1}{2n+2n^2}$	$5 \frac{1}{\frac{2}{3}n^3+5n^2+\frac{4}{3}n}$
8 th PM	$8 \frac{1}{3n+2n^2}$	$8 \frac{1}{\frac{2}{3}n^3+6n^2+\frac{7}{3}n}$
11 th PM	$11 \frac{1}{4n+2n^2}$	$11 \frac{1}{\frac{2}{3}n^3+7n^2+\frac{10}{3}n}$

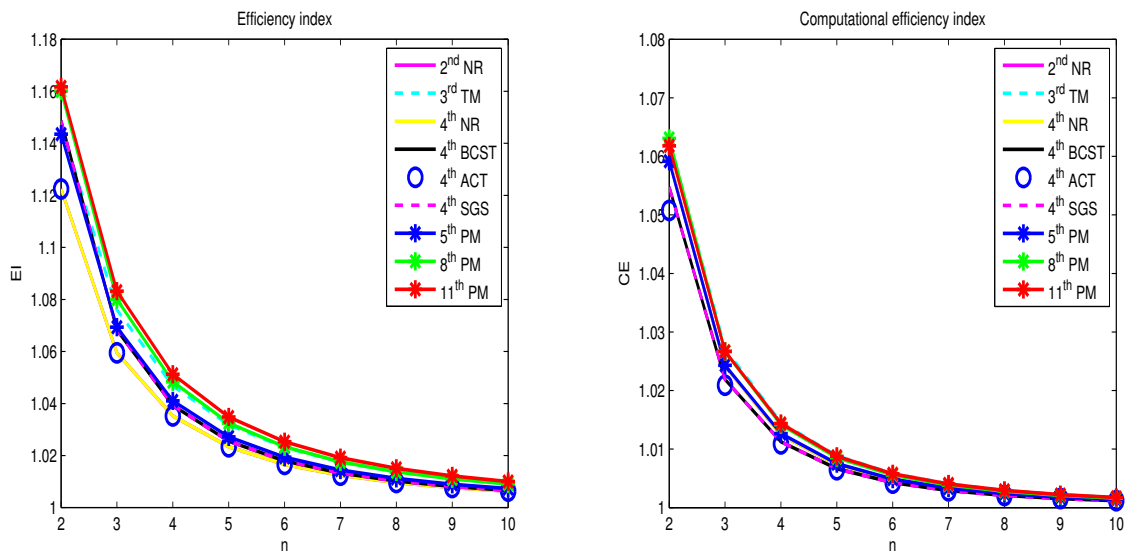


Figure 2. Comparison of EI and CE.

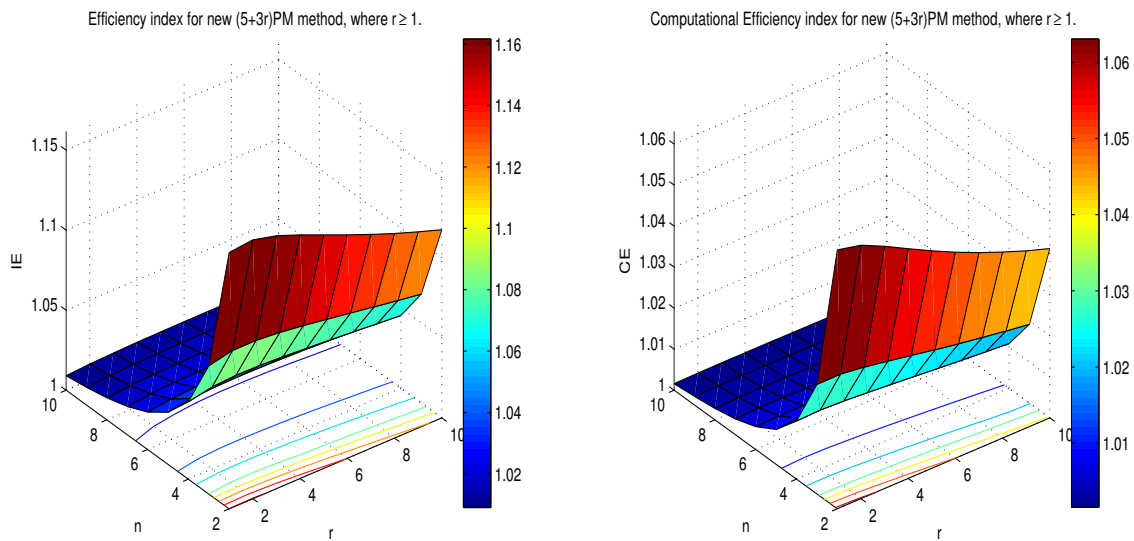


Figure 3. Comparison of EI and CE for (5 + 3r)PM method, where r ≥ 1.

5. Applications on Global Positioning System (GPS)

5.1. Basics on GPS

The worldwide coverage of all satellites has been provided by the space-based navigation systems like GPS around the clock to the users. There are minimum of 24 satellites positioned in a circular orbital constellation with an approximate altitude of 20,000 km above from the earth surface. To solve the user position from the satellite constellation, the nonlinear pseudorange equations need to be solved with higher precision. The methods like linearization and point iteration technique are used to solve the pseudorange equations. Actually, the solutions for these nonlinear equations are in the Cartesian coordinate system, especially in the ECEF format. However, the Earth is not in the shape of a perfect sphere, therefore, once the position of the user is calculated, the coordinates have to be changed into a spherical system that is suitable for latitude, longitude, and altitude form at to integrate the user position in to a MAP application. The position of an unknown point (user position) in the space can be determined by measuring the distance from the known position in space i.e., satellite position to the unknown user position.

In the two-dimensional case of user position determination, as shown in Figure 4, the three distances x_{sat1} , x_{sat2} , x_{sat3} are required from the three satellites (s_1 , s_2 and s_3) center point of the trace of a circle. Two possible solutions can be obtained from two satellites with two distances that create at two points the two circles together will intersect.

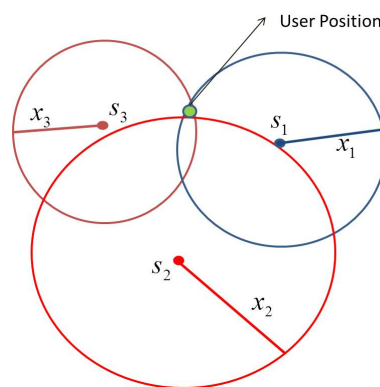


Figure 4. Two dimensional user position.

However, from this the user position cannot be determined uniquely. Therefore, one can use four satellites with four distances to proceed with a three-dimensional case. As shown in Figure 5, in a three-dimensional case, the centred point in an earth sphere forms an equal-distance trace. The transmitted ephemeris data of the satellite information gives the orbital information from this, one can determine the distance of the satellite easily. For more information refer to the literatures [19].

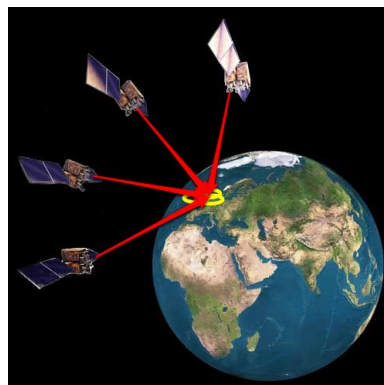


Figure 5. Three dimensional user position ([20]).

5.2. Measurement of Pseudorange

The common bias is known as pseudorange which is measured by a receiver from the ranges to GPS satellites. The distance between the user and the satellite geometric range multiplied by the velocity of light (c) is known as pseudorange (ρ_i) that is the difference in time required between the transmitted and the received signal which is calculated on the L1 frequency signal is written as (see [19])

$$\rho_i = R_i + c.\Delta t - d_{sat} + d_{iono} + d_{tropo} + d_{rel} + d_{ins} \quad (29)$$

where R_i —Geometric range,

$c.\Delta t$ —Unknown distance caused by the receiver clock offset,

d_{sat} —Advance of the satellite clock with respect to system time,

d_{iono} —Ionospheric delay,

d_{tropo} —Tropospheric delay,

d_{rel} —Relativistic delay, and

d_{ins} —Instrumental delay.

The various correction models need to be used to minimize the errors, however, in this case, we assumed that the errors are negligible to some extent. In this case, the pseudorange equations can be given as

$$\rho_i = R_i + c.\Delta t. \quad (30)$$

The geometric range can be written as

$$R_i = \sqrt{(x_{sati} - x_{user})^2 + (y_{sati} - y_{user})^2 + (z_{sati} - z_{user})^2}. \quad (31)$$

where $\rho_i, x_{sati}, y_{sati}, z_{sati}$ are known and $x_{user}, y_{user}, z_{user}, \Delta t$ are unknown, a minimum of four satellites coordinates are enough to find out the solution that is the user position values for $i = 1, 2, 3, 4$ as given by Equation (4) (see [19])

$$\begin{aligned} \rho_1 &= \sqrt{(x_{sat1} - x_{user})^2 + (y_{sat1} - y_{user})^2 + (z_{sat1} - z_{user})^2} + c\Delta t, \\ \rho_2 &= \sqrt{(x_{sat2} - x_{user})^2 + (y_{sat2} - y_{user})^2 + (z_{sat2} - z_{user})^2} + c\Delta t, \\ \rho_3 &= \sqrt{(x_{sat3} - x_{user})^2 + (y_{sat3} - y_{user})^2 + (z_{sat3} - z_{user})^2} + c\Delta t, \\ \rho_4 &= \sqrt{(x_{sat4} - x_{user})^2 + (y_{sat4} - y_{user})^2 + (z_{sat4} - z_{user})^2} + c\Delta t, \\ &\vdots \\ \rho_n &= \sqrt{(x_{satn} - x_{user})^2 + (y_{satn} - y_{user})^2 + (z_{satn} - z_{user})^2} + c\Delta t. \end{aligned} \quad (32)$$

5.3. Solving Nonlinear Pseudorange Equations

In simplified form, the pseudorange equations can be written in general for solving the above system of equations as

$$\rho_i = \sqrt{(x_{sati} - x_{user})^2 + (y_{sati} - y_{user})^2 + (z_{sati} - z_{user})^2} + b_{user}, \quad i = 1(1)n \quad (33)$$

where b_{user} is the user clock bias error expressed in distance, by differentiating the above equation, we have

$$\delta\rho_i = \frac{(x_{sati} - x_{user})\delta x_{user} + (y_{sati} - y_{user})\delta y_{user} + (z_{sati} - z_{user})\delta z_{user}}{\rho_i - b_{user}} + \delta b_{user}. \quad (34)$$

The variables x_{user} , y_{user} , z_{user} , b_{user} are termed as known quantities and the initialization values of these variables can be assumed as center of the earth. A new set of values δx_{user} , δy_{user} , δz_{user} , δb_{user} can be calculated from the initial values. The new set of values calculated from the previous step are utilized to add with the present values x_{user} , y_{user} , z_{user} , b_{user} for finding next new set of solutions. To get a desired solution as the final value of δx_{user} , δy_{user} , δz_{user} , δb_{user} , the procedure is continued until the absolute values of δx_{user} , δy_{user} , δz_{user} , δb_{user} are obtained very small within the predefined values. The above procedure is generally known as linearization of iteration method of fixed point. The expression of above equation becomes a set of linear equations that can be written in matrix form as (see [19])

$$\begin{pmatrix} \delta\rho_1 \\ \delta\rho_2 \\ \delta\rho_3 \\ \delta\rho_4 \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & 1 \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & 1 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 1 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & 1 \end{pmatrix} \begin{pmatrix} \delta x_{user} \\ \delta y_{user} \\ \delta z_{user} \\ \delta b_{user} \end{pmatrix}, \quad (35)$$

where

$$\alpha_{i1} = \frac{x_{sati} - x_{user}}{\rho_i - b_{user}}, \alpha_{i2} = \frac{y_{sati} - y_{user}}{\rho_i - b_{user}}, \alpha_{i3} = \frac{z_{sati} - z_{user}}{\rho_i - b_{user}}. \quad (36)$$

The solution of (35) is

$$\begin{pmatrix} \delta x_{user} \\ \delta y_{user} \\ \delta z_{user} \\ \delta b_{user} \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & 1 \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & 1 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 1 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & 1 \end{pmatrix}^{-1} \begin{pmatrix} \delta\rho_1 \\ \delta\rho_2 \\ \delta\rho_3 \\ \delta\rho_4 \end{pmatrix}. \quad (37)$$

This process noticeably does not give the required solutions straightforwardly. In any case, the desired results can be obtained from it. In order to determine the required user position solution, an iterative way could be utilized repetitively to solve this technique. A measure is often used to determine whether the desired result is achieved, and this quantity can be defined as (see [19])

$$\delta v = \sqrt{\delta x_{user}^2 + \delta y_{user}^2 + \delta z_{user}^2 + \delta b_{user}^2}, \quad (38)$$

where δv is lower than a certain predetermined threshold, the iteration will stop. Sometimes, the clock bias b_{user} is not included in (38). In this work, we use the norm $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|_2$ for the stopping criterion because it is stronger than (38). In this GNSS problem, we set the stopping criteria in such a way that the error $err_{min} = \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|_2 < 10^{-10}$ should be less than the predefined value within the number of iterations ('M') required for reaching the minimum residual value.

6. Results and Discussion

The test problems are solved using the above-mentioned iterative methods. The second order Newton method ($2^{nd}NR$) takes longer iteration to converge the error when compared to all methods. The methods like $3^{rd}TM$, $4^{th}NR$, $4^{th}BCST$, $4^{th}ACT$, and $4^{th}SGS$ converge within five or six iterations to solve the test problems with lower bound error. Finally testing with proposed ($5^{th}PM$, $8^{th}PM$, $11^{th}PM$) methods achieved the solution within 4 iterations. So, this could be the optimal choice for solving non-linear equations.

Next, the iterative methods are tested for solving GPS pseudo-range equations. The satellite coordinates and the pseudorange information are collected from the Oroliia GNSS simulator in which the location has been set as ETS, Montreal as shown in Figure 6.

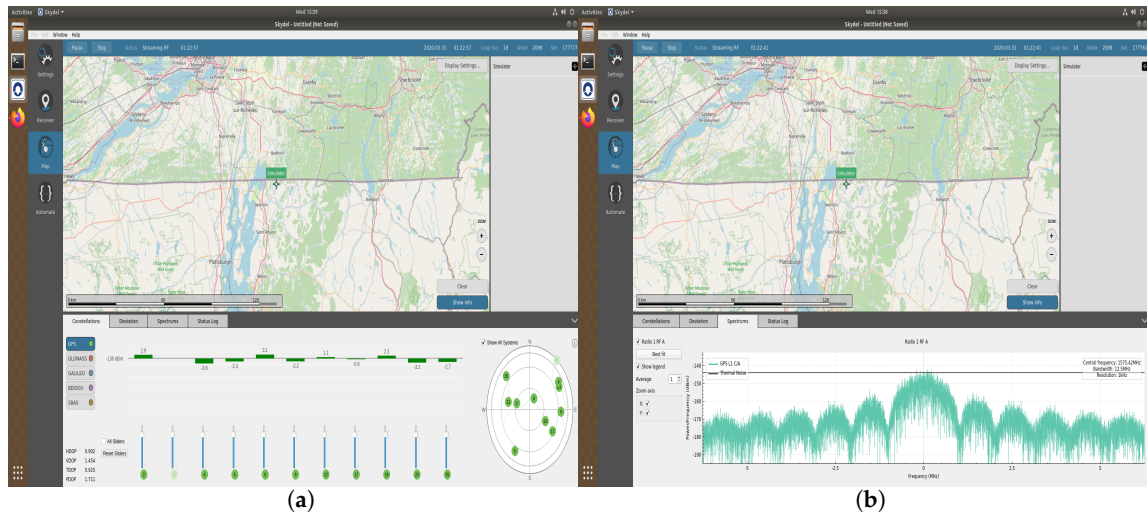


Figure 6. (a) Coordinates used in simulation (Map with skyplot). (b) Coordinates used in simulation (Map with spectrum of GPS signal).

To carry out the problem of finding the user position, the following coordinates of satellites and its pseudorange values are obtained from the Orolia GNSS simulator workbench installed in LASSENA lab is given in Table 3.

Table 3. Coordinates of observed satellite and pseudorange.

X	Y	Z	ρ
5,731,058.70224	14,489,833.1300	21,212,629.60495	20,626,320.6953
14,647,763.94672	6,275,922.21243	21,269,863.31368	22,047,954.0019
2,505,594.09552	20,672,767.43360	16,516,258.99994	21,238,160.1101
21,359,664.8576	12,371,174.84498	10,076,109.65011	252,188.013893
15,520,711.16562	1,047,695.53474	21,458,920.6173	23,317,558.30921
12,374,316.48857	23,533,071.9923	1,008,701.68425	23,125,967.55620
14,253,263.56829	4,103,289.05642	21,789,614.16879	23,747,495.1073
15,759,895.08320	5,849,736.59321	20,805,729.39674	24,788,995.2855

In Tables 4–6 we compare the proposed method with other iterative methods used in GPS receiver, where M represents the number iterations required for convergence. We recall that the coordinates of the center of the Earth and $b_{user} = 0$ gives $\mathbf{x}^{(0)} = (0, 0, 0)^T$ is usually used as starting value. We denote $\mathbf{x}^{(*)} = (1266370.3895, -4279711.6757, 4446203.8082)^T$ as the solution of the nonlinear system which gives the user position on the Earth. Thus, we conclude that 5th PM method is the most efficient method compared to other methods. In general, our proposed scheme converges in lesser iterations than other tested methods with least error and less cpu time. The results indicate that higher-order multi-point iterative methods are simple, fast, and more efficient than other compared methods.

Table 4. Comparison of iterative methods for GPS problem with 4 satellites scenario.

Method	M	$\ \mathbf{x}^{(k-1)} - \mathbf{x}^{(k-2)}\ _2$	$\ \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\ _2$	$\ \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\ _2$	CPU(sec)	p_c	GDOP
2 nd NR	6	0.2271×10^0	1.9087×10^{-9}	1.3480×10^{-25}	0.695	2.00	
3 rd TM	4	2.6206×10^4	0.0495×10^0	3.3212×10^{-19}	0.774	3.00	
4 th NR	4	2.4777×10^3	1.9087×10^{-9}	2.4095×10^{-32}	0.772	2.09	
4 th BCST	6	21.1858×10^0	3.3216×10^{-5}	8.1651×10^{-17}	1.137	2.00	0.5265
4 th ACT	4	1.9374×10^3	7.1420×10^{-10}	2.7733×10^{-32}	0.903	2.06	
4 th SGS	6	21.3566×10^0	3.3754×10^{-5}	8.4317×10^{-17}	1.157	2.00	
5 th PM	3	6.2946×10^6	3.3716×10^3	1.6324×10^{-12}	0.853	4.23	

Table 5. Comparison of iterative methods for GPS problem with 5 satellites scenario.

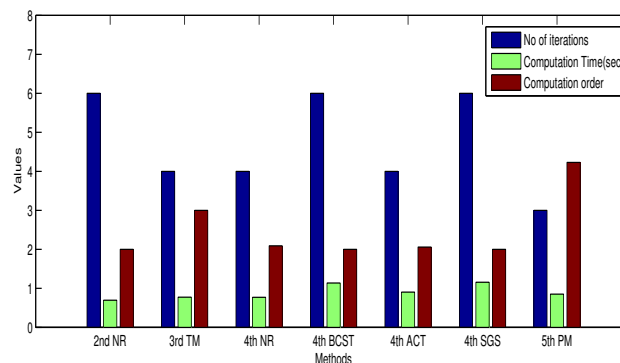
Method	M	$\ \mathbf{x}^{(k-1)} - \mathbf{x}^{(k-2)}\ _2$	$\ \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\ _2$	$\ \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\ _2$	CPU(sec)	p_c	GDOP
2 nd NR	6	0.2577×10^0	2.8677×10^{-6}	1.3444×10^{-11}	0.795	2.00	
3 rd TM	5	0.3489×10^0	2.0554×10^{-6}	9.6836×10^{-12}	0.974	2.96	
4 th NR	4	2.5209×10^3	2.8677×10^{-6}	6.3505×10^{-17}	0.838	2.27	
4 th BCST	7	3.6589×10^{-4}	2.6656×10^{-9}	1.6758×10^{-14}	1.406	2.00	0.4654
4 th ACT	4	1.9676×10^3	1.0730×10^{-7}	2.4238×10^{-18}	1.014	2.12	
4 th SGS	7	2.8345×10^{-4}	1.6174×10^{-9}	7.6224×10^{-15}	1.460	2.00	
5 th PM	3	6.3604×10^6	2.6147×10^3	1.2351×10^{-11}	0.754	4.23	

Table 6. Comparison of iterative methods for GPS problem with 6 satellites scenario.

Method	M	$\ \mathbf{x}^{(k-1)} - \mathbf{x}^{(k-2)}\ _2$	$\ \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\ _2$	$\ \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\ _2$	CPU(sec)	p_c	GDOP
2 nd NR	6	0.4495×10^0	1.4550×10^{-8}	1.4317×10^{-16}	0.866	2.00	
3 rd TM	4	7.8259×10^3	0.0064×10^0	5.6447×10^{-11}	0.957	2.85	
4 th NR	4	2.3459×10^3	1.4550×10^{-8}	9.1394×10^{-26}	0.954	2.23	
4 th BCST	6	54.5307×10^0	4.8575×10^{-4}	5.6948×10^{-12}	1.413	1.99	0.4075
4 th ACT	4	2.0707×10^3	1.0053×10^{-8}	5.6676×10^{-26}	1.107	2.10	
4 th SGS	6	54.7063×10^0	4.8898×10^{-4}	4.2930×10^{-12}	1.428	1.99	
5 th PM	3	6.3604×10^6	2.5886×10^3	3.3911×10^{-11}	0.831	4.09	

In addition, the initial points $\mathbf{x}^{(0)} = (10^6, 10^6, 10^6)^T$ and $\mathbf{x}^{(0)} = (-10^4, -10^4, -10^4)^T$ are used here to find the user position. For the first initial point, the user position is found exterior of the space solution, that is $\mathbf{x}^{(d)} = (785566.9552, 8099956.9218, -19562198.7778)^T$. For the second point, it converges to the user position with six iterations whereas centre of the earth point converges within three iterations. Hence, centre of earth is always efficient one to use as initial point to find the user position.

The available satellites are kept as eight; out of these, initially, we have considered only four satellites for finding the user position. From the comparison results plotted in Figure 7, it has been observed that the 5th order PM method converged quickly and took less computation with fewer iterations when compared to the other methods. Similarly, the error is also computed for all other iterative methods, while finding the solution for test problems, the 5th order, 8th order and 11th order PM methods almost converged within fourth or fifth iterations with lesser computation time; that is the reason we have chosen only 5th PM only and later two higher order methods are not included in solving GPS pseudo range equations.

**Figure 7.** Comparison of various iterative methods.

We increased the satellites count from four to eight. The GDOP value is computed for three cases, among all the methods, 5th order proposed method converges within three iterations with least error

of 1.6324×10^{-12} . The Fuzzy 2 based satellite selection algorithm [21] is used here for selecting the best 'n' combinations of visible satellites. The GDOP value remains same as 0.40563 after increasing the satellites count beyond six. Moreover, for different inputs of all visible satellites, the user position is computed. For instance, from 6 to 8 satellites are considered for computing the user position, the family of iterative methods yield very good performance in terms of lesser iterations and computation time when compared to the traditional Taylor series and Bancraft methods.

The comparison of position fix error for iterative method mentioned in the open source [22] and the proposed method for each iteration is calculated as shown in Tables 7–9 and it has been observed that the error in each iteration for four, five and six satellite scenarios, the proposed method converged quickly within three iterations.

Table 7. Results of error calculation in 5thPM and GPS tool box [22] with four satellites scenario.

Iterative Method (GPS Tool Box)		Proposed Method (5 th PM)	
M	Error	M	Error
1	6.3675×10^6	1	6.2946×10^6
2	2.0978×10^5	2	3.3716×10^3
3	1.2709×10^3	3	1.6324×10^{-12}
4	0.0460		
5	2.8649×10^{-8}		
6	7.7721×10^{-8}		
7	0		

Table 8. Results of error calculation in 5thPM and GPS tool box [22] with five satellites scenario.

Iterative Method (GPS Tool Box)		Proposed Method (5 th PM)	
M	Error	M	Error
1	1.3363×10^6	1	6.3604×10^6
2	6.7220×10^4	2	2.6147×10^3
3	168.6654	3	1.2351×10^{-11}
4	0.0011		
5	1.7198×10^{-8}		
6	1.4639×10^{-8}		
7	1.1727×10^{-8}		
8	1.4233×10^{-8}		
9	2.7819×10^{-8}		
10	0		

Table 9. Results of error calculation in 5thPM and GPS tool box [22] with six satellites scenario.

Iterative Method (GPS Tool Box)		Proposed Method (5 th PM)	
M	Error	M	Error
1	1.9700×10^6	1	6.3604×10^6
2	1.1879×10^4	2	2.5886×10^3
3	441.8016	3	3.3911×10^{-11}
4	0.0064		
5	2.0352×10^{-8}		
6	1.0844×10^{-8}		
7	1.9246×10^{-8}		
8	9.4634×10^{-9}		
9	1.7549×10^{-8}		
10	3.0064×10^{-9}		
11	5.3703×10^{-8}		
12	0		

7. Concluding Remarks

In the foregoing article, we have analyzed the order of convergence and numerical results of new two-step fifth and multi-step iterative methods. Consequently, we are getting a better results than other existing discussed methods. The most useful benefit of the new scheme as they don't utilize second order Frechet derivative. For practical applications, it is found that all the iterative methods converge to the user position measured from the center of the earth. Among the methods tested, 5th *PM* method is converged to the user position with a smaller number of iterations and having less CPU time and error for solving both numerical problems and GNSS pseudorange equations. In the future, it is possible implement the position calculation module with multi constellation GNSS receivers that employ GPS, GLONASS, GALELIO, BEIDOU, QZSS and IRNSS satellite navigation systems. At present more than 70 satellites are already in view and once all major four navigation systems (BeiDou + Galileo + GLONASS + GPS) are put in to orbit then more than 120 satellites will be available to the users. At any point of time an average around 30 satellites are visible from most locations around the world. If any one of the navigation systems fails then we rely on other satellite navigation systems. The recently developed GNSS receivers, a maximum of 54 channels can be allotted to different satellite navigation systems. The number of visible satellites can be chosen from the signals of opportunity then one can select more than four visible satellites (mixed GNSS Satellites) for computing better GDOP values and thus by using our proposed higher-order multi-point iterative methods, the user position can be determined more accurately with a lesser number of iterations in lower computation time.

Author Contributions: Conceptualization, K.M. and A.E.; methodology, K.M.; software, K.M. and A.E.; validation, K.M., A.E., R.J.L., and M.A.-a.; formal analysis, A.E.; investigation, K.M. and A.E.; resources, R.J.L. and M.A.-a.; data curation, K.M. and A.E.; writing—original draft preparation, K.M. and A.E.; writing—review and editing, K.M. and A.E.; visualization, R.J.L. and M.A.-a.; supervision, R.J.L. and M.A.-a.; project administration, M.A.-a.; funding acquisition, R.J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: First and last author wish to thank to the Khalifa University internal funding (No: FSU-2019-07), Abu Dhabi, UAE. Second and third author would like to thank LASSENA Laboratory, Department of Electrical Engineering, ETS, Montreal, Canada. In addition, the authors would like to thank the editors and referees for the valuable comments and for the suggestions to improve the readability of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Ostrowski, A.M. *Solutions of Equations and System of Equations*, 1st ed.; Academic Press: New York, NY, USA, 1960.
- Traub, J.F. *Iterative Methods for the Solution of Equations*, 3rd ed.; Prentice-Hall: Upper Saddle River, NJ, USA, 1964.
- Abad, M.F.; Cordero, A.; Torregrosa, J.R. Fourth- and Fifth-Order Methods for Solving Nonlinear Systems of Equations: An Application to the Global Positioning System. *Abstr. Appl. Anal.* **2013**, *2013*, 586708. [[CrossRef](#)]
- Babajee, D.; Madhu, K.; Jayaraman, J. On Some Improved Harmonic Mean Newton-Like Methods for Solving Systems of Nonlinear Equations. *Algorithms* **2015**, *8*, 895–909. [[CrossRef](#)]
- Cordero, A.; Hueso, J.L.; Martinez, E.; Torregrosa, J.R. A modified Newton-Jarratt's composition. *Numer. Algor.* **2010**, *55*, 87–99. [[CrossRef](#)]
- Noor, M.A.; Waseem, M.; Noor, K.I.; Al-Said, E. Variational iteration technique for solving a system of nonlinear equations. *Optim. Lett.* **2013**, *7*, 991–1007. [[CrossRef](#)]
- Sharma, J.R.; Guha, R.K.; Sharma, R. An efficient fourth order weighted-Newton method for systems of nonlinear equations. *Numer. Algor.* **2013**, *62*, 307–323. [[CrossRef](#)]
- Babajee, D.K.R.; Cordero, A.; Soleymani, F.; Torregrosa, J.R. On a Novel Fourth-Order Algorithm for Solving Systems of Nonlinear Equations. *J. Appl. Math.* **2012**, *2012*, 165452. [[CrossRef](#)]

9. Madhu, K.; Babajee, D.; Jayaraman, J. An improvement to double-step Newton method and its multi-step version for solving system of nonlinear equations and its applications. *Numer. Algor.* **2017**, *74*, 593–607. [[CrossRef](#)]
10. Yang, M. Noniterative Method of Solving the GPS Double-Differenced Pseudorange Equations. *J. Surv. Eng.* **2005**, *131*, 130–134. [[CrossRef](#)]
11. Pachter, M.; Nguyen, T.Q. An Efficient GPS Position Determination Algorithm. *J. Inst. Navig.* **2003**, *50*, 131–141. [[CrossRef](#)]
12. Li, W.; Yang, S.H.; Li, D.; Xu, Y.W.; Zhao, W. Design and Analysis of a New GPS Algorithm. In Proceedings of the 2010 IEEE 30th International Conference on Distributed Computing Systems, Genova, Italy, 21–25 June 2010.
13. Ko, K.S.; Choi, C.M. Mathematical Algorithms for Two-Dimensional Positioning Based on GPS Pseudorange Technique. *Int. J. Kim.* **2010**, *8*, 602–607. [[CrossRef](#)]
14. Jwoa, D.; Hsiehb, M.; Leea, Y. GPS navigation solution using the iterative least absolute deviation approach. *Sci. Iran. B* **2015**, *22*, 2103–2111.
15. Awange, J.L.; Grafarend, E.W. Algebraic Solution of GPS Pseudo-Ranging Equations. *GPS Solut.* **2002**, *5*, 20–32. [[CrossRef](#)]
16. El-naggar, A.M. An alternative methodology for the mathematical treatment of GPS positioning. *Alex. Eng. J.* **2011**, *50*, 359–366. [[CrossRef](#)]
17. Cordero, A.; Gomez, E.; Torregrosa, J.R. Efficient High-Order Iterative Methods for Solving Nonlinear Systems and Their Application on Heat Conduction Problems. *Complexity* **2017**, *2017*, 6457532. [[CrossRef](#)]
18. Cordero, A.; Torregrosa, J.R. Variants of Newton’s method using fifth-order quadrature formulas. *Appl. Math. Comp.* **2007**, *190*, 686–698. [[CrossRef](#)]
19. Tsui, J.B.Y. *Fundamentals of Global Positioning System Receivers, a Software Approach*, 2nd ed.; Wiley Interscience: Hoboken, NJ, USA, 2005.
20. Griffin, D. How does the Global Positioning System Work? 2011. Available online: <http://www.pocketgpsworld.com/howgpsworks.php> (accessed on 10 June 2020)
21. Arul-Elango, G.; Murukesh, C.; Rajeswari, K. Type-2 Fuzzy based GPS Satellite Selection algorithm for better Geometrical Dilution of Precision. *Int. J. Comput. Sci. Inf. Secur.* **2016**, *14*, 78–82.
22. Tetewsky, A.K.; Soltz, A. GPS MATLAB Toolbox Review. *GPS World* **1998**, *9*, 50–57.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).