

# Explicit Iterative Methods of Second Order and Approximate Inverse Preconditioners for Solving Complex Computational Problems

Anastasia-Dimitra Lipitakis

Department of Informatics and Telematics, Harokopio University, Athens, Greece

Email: [adlipita@hua.gr](mailto:adlipita@hua.gr)

**How to cite this paper:** Lipitakis, A.-D. (2020) Explicit Iterative Methods of Second Order and Approximate Inverse Preconditioners for Solving Complex Computational Problems. *Applied Mathematics*, 11, 307-327. <https://doi.org/10.4236/am.2020.114023>

**Received:** March 3, 2020

**Accepted:** April 19, 2020

**Published:** April 22, 2020

Copyright © 2020 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0). <http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

Explicit Exact and Approximate Inverse Preconditioners for solving complex linear systems are introduced. A class of general iterative methods of second order is presented and the selection of iterative parameters is discussed. The second order iterative methods behave quite similar to first order methods and the development of efficient preconditioners for solving the original linear system is a decisive factor for making the second order iterative methods superior to the first order iterative methods. Adaptive preconditioned Conjugate Gradient methods using explicit approximate preconditioners for solving efficiently large sparse systems of algebraic equations are also presented. The generalized Approximate Inverse Matrix techniques can be efficiently used in conjunction with explicit iterative schemes leading to effective composite semi-direct solution methods for solving large linear systems of algebraic equations.

## Keywords

Approximate Inverse Preconditioners, Iterative Methods, Second Order Iterative Schemes, Exact Inverse Methods, Approximate Inverse, Explicit Preconditioning, Conjugate Gradients, Convergence Analysis

---

## 1. Introduction

During the last decades, considerable research effort has been directed to the solution of complex linear and nonlinear systems of algebraic equation by using a class of iterative methods. This class includes the conjugate gradient method and its hybrid multi-variants. The conjugate gradient method originally introduced by Hestenes and Stiefel [1], was a direct solution method but later on has been extensively used as an iterative method for solving efficiently large sparse linear

and nonlinear systems of complex computational problems [2] [3] [4]. The incomplete factorization methods and various hybrid preconditioning techniques [5] [6] [7] [8] [9] are well known and have been efficiently applied for solving iteratively large sparse linear systems of algebraic equations [5] [10] [11] [12] [13] [14].

Certain theoretical issues on this subject, such as 1) the stability and conditions of correctness of approximate system matrix in the form of product of triangular matrix factors, and 2) the convergence analysis of the iterative methods and the quantitative evaluation of the convergence rate of the iterative schemes, are of particular interest for the choice of the most efficient methods for systems with predetermined properties.

In the framework of this research work and in order to substantiate the main motivation for the efficient usage of the second order iterative method for solving nonlinear systems, the following basic related questions will be considered: 1) are the second order iterative methods comparable (or even superior) to first order iterative methods for solving nonlinear systems? A survey of related research work will be given; 2) are second order iterative schemes preferable than the first order iterative schemes for solving very large complex computational problems? The computational complexity per iterative step will be also examined.

The structure of this research paper is as follows: in Section 2, several advantages of second order iterative methods in comparison with the first order iterative methods for solving nonlinear systems are presented. In Section 3, a class of general iterative methods of second order is described, while in Section 4, certain explicit iterative schemes and approximate inverse preconditioners are introduced. In Section 5, exact and approximate inverse matrix algorithmic techniques are introduced, while in Section 6, some aspects of stability and correctness of incomplete factorization methods are presented. Finally, in Section 7, the convergence analysis and quantitative evaluation of convergence rate of incomplete factorization methods are discussed.

## 2. General Iterative Methods of Second Order: Part I

In recent years, considerable research effort has been focused in the topic of second order iterative methods. In order to substantiate our motivation in our research study, we present a synoptic survey of related computational methods developed on the subject. Several early iterative methods of second order with fixed parameters or variable parameters have been extensively studied [15] [16] [17] [18].

A class of (optimal) second degree iterative methods for accelerating basic linear stationary methods of first degree with real eigenvalues has been presented [18] [19] and has been extended as application of conformal mapping and summation techniques for the case when eigenvalues are contained in elliptical regions in the complex plane [20] [21] [22]. Another similar contribution on op-

timal parameters for linear second degree stationary iterative methods applied to unsymmetric linear systems have been computed by solving the minimax problem used to compute optimal parameters for Chebyshev iteration that is asymptotically equivalent to linear second-degree stationary method [23]. A nonstationary iterative method of second order for solving nonlinear equations without requiring the use of any derivative has been presented. This method for algebraic equations coincides with Newton's method and is more efficiently [24]. Note that Newton's methods and high order iterative schemes (Householder's iterative methods), under some conditions of regularity of the given function and its derivatives, have been used for the numerical treatment of single nonlinear equations [25].

A three-step iterative method for solving nonlinear equations by using Steffensen's method in the third step having eight order convergence has been recently presented. This method requires a small number of calculations and does not require calculation of derivative in the third iterative step. A two-step iterative method for solving nonlinear equations, a modified Noor's method without computing the second derivatives and with fourth order convergence has been presented [26]. A second order iterative method for solving quasi-variational inequalities has been introduced and sufficient conditions for convergence rate have been given [27]. Two iterative methods of order four and five respectively by using modified homotopy perturbation techniques for solving nonlinear equations and the convergence analysis have been presented [28]. An efficient second order iterative method for IR drop analysis in power grid has been presented [29]. In this research study, they consider a first order iterative method

$$x_{n+1}^* = Gx_n + K_1 \quad (1)$$

and the resulting second order iterative method

$$x_{n+1} = x_n + a(x_n - x_{n-1}) + b(x_{n+1}^* - x_n), \quad (2)$$

where  $x_{n+1}^*$  denotes the first-order iteration, and  $a$  and  $b$  are real accelerating parameters effecting the convergence rate. For the consistency and convergence of the second order iterative methods the following statements hold:

Preposition 1: If the 1st-order iterative method converges to the exact solution, then the 2nd-order method will converge to the same solution for any values of  $a \neq 0$  and  $b \neq 0$ .

Preposition 2: The iterative matrix of the 2nd-order method is known. A necessary and sufficient condition that the iterative method converges for all initial conditions is that, if the spectral radius of matrix  $G$  is minimized then the convergence rate is maximized [29].

## 2.1. Some Problems in Solving Very Large Complex Computational Problems

It is known that due to the extremely large sizes of *power grids*, *IR drop analysis* has become a computationally challenging problem in terms of memory usage

and runtime, and *second-order iterative* algorithms that can significantly reduce the runtime have been proposed. Specifically, the main problems include the following:

- 1) Very large-scale simulations (millions of elements in power grids) and run time is slow.
- 2) Memory inefficiency (1 million nodes and trillion elements in matrix)
- 3) Trade-off between runtime and predetermined accuracy
- 4) Power delivery issues (increased complexity of VLSI circuits, increased power (current) consumption, decreasing supply voltage, reduced noise margin, increased gate delay)
- 5) Modelling and analysis of power grid network must be accurate and power grid networks tend to be very large [22].

Typical applications include also a large class of initial-boundary value problems of general form in 3 space dimensions with strong nonlinearities:

$$u_i + \sum_{i=1}^N a_i(x, t, u) u_{xi} + b(x, t, u) = \varepsilon \sum_{i=1}^N u_{xi} u_{xi} \quad (2a)$$

where the positive perturbation parameter tends to zero [30].

The discrete analogues of Equations (1) (2) lead to the solution of the general linear system

$$Au = s \quad (2b)$$

where the coefficient matrix  $A$  is a large sparse unsymmetric real ( $n \times n$ ) matrix of irregular structure.

## 2.2. Computational Complexity per Iterative Step

Computational complexity of algorithms is an important subject in which considerable research has been focused in the last decades [6] [31] [32] [33] [34].

An interesting topic in the framework of comparison of the number of flops per iteration to be performed for several classes of solution methods, *i.e.* 1) direct second order methods (based on direct matrix decomposition), 2) iterative first order methods and 3) iterative second order methods, for the cases of dense and sparse problems reveals the following:

- 1) In first order iterative methods, the number of flops per iteration is generally much smaller than that of second order methods, while the former generally need many iterations to converge (maybe few hundreds up to few thousands) and also have difficulties to obtain highly accurate optimal solutions.
- 2) The second order iterative methods usually converge quickly (few tens iterations) to highly accurate solutions.
- 3) The first order iterative methods overall are considered to be much superior to second order methods based on direct factorization in solving large scale SDP problems. However, the latter methods in solving linear systems may be prohibitively expensive (even impossible) due to high number of flops required and the high amount of memory space needed for storing the coefficient matrix.
- 4) The computational complexity of second order iterative methods has the

following characteristics:

- a) There is no need for computing and/or store the coefficient matrix
- b) The computational work and storage requirements of inner iterations are very comparable to that of iterations of log-barrier method
- c) The total number of inner iterations performed by iterative second order method depends on the choice of preconditioner used in the iterative algorithm of the original system.

Note that a logarithmic barrier term, which emphasizes the improvement of poor quality elements, solves the constrained optimization problem using the gradient of the objective function [35] [36].

Conclusively, the second order iterative methods, as far as the inner iterations are concerned, behave quite similar to first order methods. Furthermore, the development of efficient preconditioners for solving the original linear system is a decisive factor for making the second order iterative methods superior to the first order iterative methods.

### 3. General Iterative Methods of Second Order: Part II

In this section, a class of iterative methods of second order for solving large sparse linear systems of the form  $Au = b$  is presented and explicit preconditioned methods for approximating the solution of complex computational problems are given. Apart of the extensive research work that has been done for solving linear systems by using second order iterative methods as indicated in section 2, it is worthwhile to mention that the efficient usage of second order iterative schemes for solving nonlinear systems has been reported by Lipitakis (1990) and these iterative schemes are originated from the E-algorithm, a modified version of the well-known algorithm of Euclid, written in the form of a general second order iterative scheme [37]. This general iterative scheme is synoptically described in the following:

Let us consider a class of non-stationary iterative methods of second order the form:

$$u_{i+1} = \pi_i u_i + \tau_i u_{i-1} + \delta_i, \quad i \geq 0 \quad (3)$$

where  $\pi_i$ ,  $\tau_i$  are real parameters, the so-called E-parameters [37], and  $\delta_i$  is a “correction” term. The iterative scheme is known as the E-iterative method [37].

These iterative schemes with appropriate selection of the parameters can be used in conjunction with various explicit preconditioned methods, such as explicit Richardson, Chebychev, fractional step iterative method, Grank-Nickolson, multiple explicit Jacobi, explicit preconditioned Conjugate Gradients, etc. for solving complex computational problems, such as 3D-initial boundary value problems. It is known that the corresponding approximate factorization algorithms and the approximate inverse algorithms are tediously complicated, especially in the three-dimension space with complicated boundary conditions in irregular domains.

Let us consider an unsymmetric large sparse linear system  $Au = b$ . Then, the choice of E-parameters:

$$\pi_i = [I - H^{-1}A], \tau_i = 0, \delta_i = H^{-1}b \tag{4}$$

where the non-singular matrix  $H$  is chosen such that the computational work required for solving the linear system  $Hu = s$  (where  $s$  is given) is considerably small compared to that required to solve the system  $Au = b$ , leading to the iterative procedure,

$$u_{i+1} = [I - H^{-1}A]u_i + H^{-1}b, i \geq 0 \tag{5}$$

Note that several choices of matrix  $H$  lead to well-known iterative methods, *i.e.* Richardson, Block Jacobi, Gauss-Seidel, SOR, SSOR, etc.

Consider the unsymmetric linear system  $Au = b$ , a family of ellipses of centred with foci  $d \pm c$  and assuming that the E-parameters are selected as

$$\pi_i = [1 - \beta_i - a_i A], \tau_i = -\beta_i, \delta_i = a_i b, i \geq 0 \tag{6}$$

where the acceleration parameters  $a_i, \beta_i$  are defined as

$$a_i = [d - (c/2)^2 a_{i-1}]^{-1}, \beta_i = da_i - 1, i \geq 1 \tag{7}$$

with

$$a_1 = 2d / (2d^2 - c^2), \beta_1 = da_0 - 1, a_0 = 1/d, \beta_0 = 0 \tag{8}$$

The iterative method (3) then can be equivalently written as a second order non-stationary iterative scheme (known as Chebychev iterative method):

$$u_{i+1} = (1 + \beta_i - a_i A)u_i - \beta_i u_{i-1} + a_i b, i \geq 1 \tag{9}$$

or equivalently

$$u_{i+1} = u_i + a_i r_i + \beta_i (u_i - u_{i-1}), i \geq 1 \tag{10}$$

where  $r$  is the residual factor  $r = b - Au$ , and  $u_0, u_1$  are arbitrary chosen initial values.

The simplified choice of E-parameters

$$\pi_i = [1 - \beta - aA], \tau_i = -\beta, \delta_i = ab, i \geq 0 \tag{11}$$

leads to second-order iterative scheme (known as the second-order Richardson's method):

$$u_{i+1} = [u_i + ar_i + \beta(u_i - u_{i-1})], i \geq 1 \tag{12}$$

where the parameters  $\alpha, \beta$  are the asymptotic values of  $a_i, \beta_i$  [19].

Let us consider the approximate factorization  $A \approx L_s U_s$ , where  $L_s$  and  $U_s$  are sparse decomposition factors [37] [38] and let us assume that  $M$  a non-singular real  $(n \times n)$  matrix, is the inverse of  $(L_s U_s)$ , *i.e.*  $M_r = (L_r U_r)$ ,  $r \in [1, m-1]$ , where  $r$  is the "fill-in" parameter, *i.e.* the number of outermost-off diagonal entries which are retained in semi-bandwidth  $m$ . Then, the explicit iterative methods based on the generalized approximate inverse matrix techniques

[39], can be obtained by determining the element of  $A^{-1}$  without inverting the decomposition factors  $L_r, U_r$ .

Note that the effectiveness of explicit iterative methods is mainly related to the fact that the exact inverse of the sparse matrix  $A$ , although is full, exhibits a similar “fuzzy” structure as  $A$ , *i.e.* the largest elements are clustered around the principal diagonal and main diagonals [13].

The selection of the E-parameters

$$\begin{aligned} \pi_i &= [1 + b - aM_r A], \\ \tau_i &= -\beta, \delta_i = aM_r b, i \geq 0 \end{aligned} \tag{13}$$

and

$$\begin{aligned} \pi_i &= [1 + \beta_i - a_i M_r A], \\ \tau_i &= -\beta_i, \delta_i = a_i M_r b, i \geq 0 \end{aligned} \tag{14}$$

where  $M_r$  is an approximate form of the inverse of  $A$  and  $\alpha, \beta$  and  $a_i, \beta_i$  are preconditioned parameters and sequences of parameters respectively, leads to the following explicit iterative schemes:

$$u_{i+1} = (1 + \beta - \alpha M_r A)u_i - \beta u_{i-1} + a M_r b, i \geq 1 \tag{15}$$

$$u_{i+1} = (1 + \beta - \alpha_i M_r A)u_i - \beta_i u_{i-1} + a_i M_r b, i \geq 1 \tag{16}$$

which are known as the Explicit second order Richardson and Explicit Chebychev methods respectively [13].

In these explicit iterative schemes several approximate forms of the inverse can be effectively used by retaining only  $\delta_U$  and  $\delta_L$  ( $\delta_U \in [2, n-1], \delta_L \in [1, n]$ ) diagonals in the upper and lower triangular parts of  $M_r$ , the remaining elements being just not computed at all [13].

## 4. Explicit Iterative Schemes and Approximate Inverse Preconditioners

### 4.1. A Class of Optimized Approximate Inverse Variants

A class of optimized approximate inverse variants can be obtained by considering a (near) optimized choice of the approximate inverse  $M$  depends on the selection of related parameters, *i.e.* fill-in parameters  $r_1, r_2$ , retention parameters  $\delta'_1, \delta'_2$  and entropy-adaptivity-uncertainty (EAU) parameters [30] (Figure 1). Note that the selection of retention parameter values as multiples of the corresponding semi-bandwidths of the original matrix leads to improved numerical results [40]. Then, the following sub-classes of approximate inverses, depending on the accuracy, storage and computational work requirements, can be derived

sub-class I	sub-class II	sub-class III	sub-class IV
$A^{-1} \equiv M \rightarrow M_{r_1=m-1, r_2=p-1}^{\delta'_1, \delta'_2}$	$\rightarrow M_{r_1=m-1, r_2=p-1}^S \delta'_1, \delta'_2$	$\rightarrow M_{r_1, r_2}^{\delta'_1, \delta'_2}$	$\rightarrow M_{r_1, r_2}^* \delta'_1, \delta'_2 \equiv M^* (17)$

Figure 1. Certain subclasses of approximate inverse matrices.

where  $M_{\eta=m-1, r_2=p-1}^{\delta l_1, \delta l_2}$  of sub-class I is a banded form of the exact inverse retaining  $\delta l_1, \delta l_2$  elements along each row and column respectively, while its elements are equal to the corresponding elements of the exact inverse. The term  $M_{\eta=m-1, r_2=p-1}^S \delta l_1, \delta l_2$  of sub-class II is a banded form of  $M$ , retaining only  $\delta l_1, \delta l_2$  elements along each row and column during the computational procedure of the approximate inverse and under certain hypotheses can be considered as a good approximation of the original inverse, while the entries of the approximate inverse in sub-class III have been retained after computing  $M^*$  ( $r_1 < m-1$  and  $r_2 < p-1$ ) and are less accurate than the corresponding entries of  $M_{\eta, r_2}^* \delta l_1, \delta l_2$ . Finally, in sub-class IV the elements of the approximate inverse can be computed.

Note that the generalized Approximate Inverse Matrix (AIM) techniques can be efficiently used in conjunction with explicit iterative schemes leading to effective composite semi-direct solution methods solving large linear systems of algebraic equations on multiprocessor systems.

## 4.2. Explicit Iterative Schemes

Analogous relationships can be derived for various explicit CG-type methods as indicated in the following. The generalized alternating group explicit (AGE) iterative methods, based on the known ADI methods [12] [17] [41] [42], can be derived as follows: consider the splitting

$$A = D + G_1 + G_2 \quad (18)$$

where  $D$  is a non-negative diagonal matrix and  $D, G_1, G_2$  satisfy the conditions

- 1) The matrices  $(G_i + \theta D + \rho I), i=1,2$  are non-singular for any  $\theta \geq 0, \rho > 0,$
- 2) The system

$$x = G_1^{-1}v \quad \text{and} \quad y = G_2^{-1}z \quad (19)$$

can be easily solved explicitly for any vectors  $v$  and  $z$ . Note that this statement holds only to certain model problems. Then, the following generalized AGE scheme can be obtained

$$(G_i + p_{i+1}I)u_{i+1/2} = (p_{i+1}I - G_2)u_i + b \quad (20)$$

$$(G_i + p_{i+1}I)u_{i+1} = (G_2 - (I - \omega)p_{i+1}I)u_i + (2 - \omega)p_{i+1}u_{i+1/2} \quad (21)$$

where  $\omega$  is a non-negative acceleration parameter [1].

In an analogous manner can be derived the explicit preconditioned methods 1) Richardson + AGE method and 2) Chebychev semi-iterative method + AGE method [43]. Note that the AGE method which is based on the combination of certain elementary first order difference processes permitting the reduction of a given complicated problem into a sequence of simpler problems, can be considered as a fractional (splitting-up) method [44] [45].

The multiple explicit Jacobi ( $\mu$ -EJ) method and its several parametric forms, provided that their spectral radius  $\rho$  satisfies the corresponding convergence

condition, in combination with the Lanczos economized Chebychev polynomials of degree  $\mu$ , has proved to be effective for solving elliptic boundary-value problems in parallel processors [39]. The multiple explicit Jacobi method in conjunction to economized Chebychev polynomial and Neumann series of certain degree can be effectively applied for solving explicitly large sparse linear systems resulting from the discretization of initial boundary-value problems [39].

### 4.3. Explicit Preconditioned Conjugate Gradients Method of Second Order

Let us assume that the E-parameters are selected as follows:

$$\pi_i = p_{i+2} (1 + \gamma_{i+2} M_\mu A), \tau_i = 1 - p_{i+1}, \delta_i = p_{i+1} \gamma_{i+1} M_\mu b, i \geq 0 \quad (22)$$

where

$$p_{i+1} = 1 + (a_i / a_{i-1}) \beta_i, \gamma_{i+1} = \alpha_i / p_{i+1}, i = 0, 1, 2, \dots, k-1 \quad (23)$$

with  $k$  the smallest integer such that  $r_k = b - Au_k = 0$ ,  $p_i = 1$  and  $a_i, \beta_i$  are the scalar quantities as defined in the original CG paper [1].

The iterative scheme (21) then becomes

$$u_{i+1} = p_{i+1} (1 + \gamma_{i+1} M_\mu A) u_i + (1 - p_{i+1}) u_{i-1} + p_{i+1} \gamma_{i+1} M_\mu b, i \geq 1 \quad (24)$$

or equivalently the second order explicit preconditioned CG scheme can be obtained as

$$u_{i+1} = u_{i-1} + p_{i+1} (\gamma_{i+1} M_\mu r_i + u_i - u_{i-1}), i \geq 1 \quad (25)$$

In the following, the selection of E-parameters for various explicit preconditioned methods is presented.

The proposed explicit iterative methods can be used for solving large sparse linear systems and the E-iterative schemes can generate useful explicit iterative schemes of higher order with suitable selection of E-parameters for solving a wide class of very large sparse linear systems in multiprocessor systems.

Typical applications include a large class of initial-boundary value problems of general form in three space dimensions with strong nonlinearities

$$u_t + \sum_{i=1}^N a_i(x, t, u) u_{x_i} + b(x, t, u) = \varepsilon \sum_{i=1}^N u_{x_i} u_{x_i} \quad (25a)$$

where the positive perturbation parameter tends to zero [46].

The discrete analogues of Equation (25a) lead to the solution of the general linear system

$$Au = s \quad (25b)$$

where the coefficient matrix  $A$  is a large sparse unsymmetric real  $(n \times n)$  matrix of irregular structure.

### 4.4. Selection of Explicit Iterative Solver and Algorithmic Implementation

In this section an iterative solver of second order for solving linear and non-

linear systems of irregular structure is presented in pseudo algorithmic form:

**Algorithm SOIS-1 (EXSOM, SEIS, U-1, U0, NMAX,  $\Pi$ , R,  $\Delta$ , U)**

**Purpose:** This algorithm describes the general selection of an iterative solver of second order for solving linear and nonlinear nonsymmetric systems of irregular structure and the selection of corresponding parameters.

**Input:** the computational module EXSOM selecting the explicit iterative solver SEIS and corresponding parameters, the initial values  $u_{-1}$ ,  $u_0$ , max number of iterations NMAX.

**Output:** final explicit iterative solver SEIS and approximate iterative solution U.

**Computational Procedure:**

step 1: Call the explicit iterative module EXSOM and determine the explicit iterative solver to be used

step 2: Select initial vectors  $u_{-1}$  and  $u_0$  and set the max number of iterations NMAX

step 3: Select the corresponding E-parameters  $\pi$ ,  $r$  and correction term  $\delta$

step 4: For  $i = 0, 1, 2, \dots$

step 5: Compute approximate vector  $u_{i+1}$  as  $u_{i+1} = \pi_i u_i + r_i u_{i-1} + \delta_i$

// the E-parameters  $\pi$ ,  $r$  and correction term  $\delta$  are selected from the corresponding explicit iterative scheme of computational module EXSOM //

step 6: If the convergence criterion is satisfied and the max number of iteration NMAX is hold,

then go to step 7,

else go to step 4 and

continue

step 7: Print the explicit iterative solver SEIS

step 8: Form approximate solution  $u$

The explicit computational module EXSOM can be used for solving an explicit iterative solver as follows:

**module EXSOM (ERI, ECH, GAGE, RAGE, CHAGE, GFSI, MEJ, EPCGSO, SEIS)**

**Purpose:** this module determines the explicit iterative solver that will be used for solving the given linear nonsymmetric system of irregular structure

**Input:** the explicit iterative solvers ERI, ECH, GAGE, RAGE, CHAGE, GFSI, MEJ, EPCGSO, IS integer (=1, 2, 3, 4, 5, 6, 7, 8) is the number of explicit iterative solver to be used

**Output:** The selected explicit iterative solver SEIS = IS

**Computational Procedure:**

step 1: Consider one of the following available explicit iterative solvers and set the corresponding value of IS

step 1.1: If IS==1,

then the explicit Richardson method ERI is to be used;

return

```

step 1.2: If else IS==2,
then the explicit Chebychev ECH is to be used
return
step 1.3: If else IS==3,
then the generalized AGE method GAGE is to be used;
return
step 1.4: If else IS==4,
then the Richardson + AGE method RAGE is to be used;
return
step 1.5: If else IS==5,
then the Chebychev + AGE method CHAGE is to be used;
return
step 1.6: If else IS==6,
then the general fractional step iteration GFSI is to be used;
return
step 1.7: If else IS==7,
then the multiple explicit Jacobi MEJ is to be used;
return
step 1.8: If else IS==8,
then the explicit preconditioned CG of second order is to be used
return
step 2: return the selected explicit iterative solver (SEIS)

```

Additional explicit iterative solvers can be used in module EXSOM for solving the given linear system. The algorithm SOIS-1 can be used as a general explicit iterative solver of second order for solving linear and nonlinear non-symmetric systems of irregular structure by selecting the corresponding parameters.

## 5. The Exact and Approximate Inverse Matrix Algorithmic Techniques

### 5.1. On the Selection of Approximate Inverse Preconditioners

The selection of an efficient approximate inverse preconditioner for solving explicitly complex computational problems is an interesting research topic of critical importance. Let us assume that a non-singular large sparse unsymmetric matrix of irregular structure can be factorized as  $A = LU$  (**Figure 2**), where  $L$  and  $U$  are triangular matrices and  $A \approx L_s U_s$  is an approximate factorization with  $L_s$  and  $U_s$  the corresponding sparse decomposition factors (**Figure 3** and **Figure 4**). The elements of the decomposition factors can be efficiently computed [39]. Let us assume that  $M_s^*$  is an approximate inverse of  $A$ , i.e.  $A^{-1} \approx M_s^*$ .

In the following, a class of adaptive exact and approximate inverse solvers based on exact/approximate  $LU$  decompositions and approximate inverse methods is described. Let us assume the general linear system

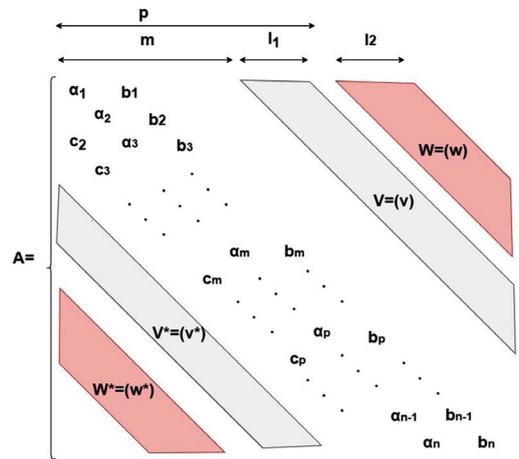


Figure 2. Structure of the unsymmetric coefficient matrix  $A$ .

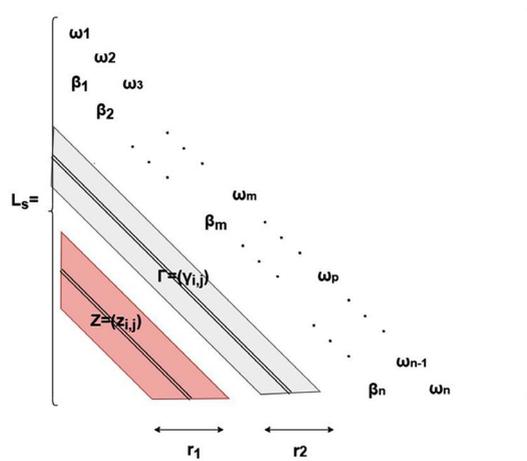


Figure 3. Structure of the lower triangular factor  $L_s$ .

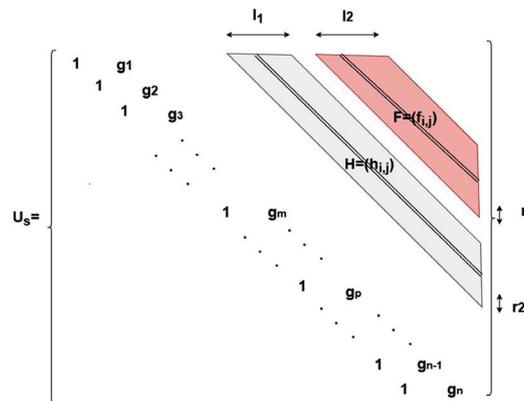


Figure 4. Structure of the upper triangular factor  $U_s$ .

$$Au = s \tag{25b}$$

where the coefficient matrix  $A$  is a large sparse real  $(n \times n)$  matrix of irregular structure.

The structure of  $A$  is shown in the following diagram:

Note that such regular matrix structures occur only for certain model type problems.

Let us consider, the factorization  $A = LU$  and an approximate factorization of the coefficient matrix  $A$ ,

$$A \approx L_s U_s \quad (27)$$

where  $L_s$  and  $U_s$ , are lower and upper sparse triangular matrices of irregular structures of semi-bandwidths  $m$  and  $p$  retaining  $r_1$  and  $r_2$  fill-in terms respectively. The decomposition factors  $L_s$  and  $U_s$  are banded matrices with  $l_1$  and  $l_2$  the numbers of diagonals retained in semi-bandwidths  $m$  and  $p$  respectively, of the following form:

The elements of the decomposition factors  $L_s$  and  $U_s$  can be obtained from the algorithmic procedure FELUBOT [39].

A class of exact and approximate inverse matrix techniques can be considered containing several sub-classes of approximate inverses according to memory requirements, computational work, accuracy, as indicated in the following scheme:

Let us assume that  $M_{r_1, r_2}$ , a non-singular  $(n \times n)$  matrix, is an approximate inverse of  $A$ , i.e.  $M_{r_1, r_2} = \{M_{i,j}\}^{r_1, r_2}$ ,  $i, j \in [1, n]$ . Note that if  $r_1 = m - 1$  and  $r_2 = p - 1$  non-zero elements have been retained in the corresponding decomposition factors, then  $M_{r_1, r_2} = M$ , where  $M$  is the exact inverse of  $A$ . The elements of  $M$  can be determined by solving recursively the systems

$$ML = U^{-1} \text{ and } UM = L^{-1} \quad (28)$$

having main disadvantages, i.e. high storage requirements and computational work involved particularly in the case of solving very large unsymmetric linear systems.

A class of approximate inverses  $M^{\delta l_1, \delta l_2}$  can be obtained by retaining only  $\delta l_1$  and  $\delta l_2$  diagonals in the lower and upper triangular parts of inverse respectively, the remaining elements being just not computed at all. Optimized forms of this algorithm are particularly effective for solving banded sparse FE systems of very large order, i.e.  $[\delta l_1 + \delta l_2] > n/2$  or in the case of narrow-banded sparse FE systems of very large order, i.e.  $[\delta l_1 + \delta l_2] \ll n/2$ . Then an explicit approximate inverse preconditioner can be described in the following adaptive algorithmic form.

## 5.2. The Exact and Approximate Inverse Preconditioners

The elements of the exact and approximate inverse of a given unsymmetric and irregular structure can be obtained as follows and the explicit banded approximate inverse algorithm can be described by the following algorithmic procedure in pseudocode form:

**Algorithm EBAIM-1** ( $A, n, \epsilon EM, r_1, r_2, m, p, l_1, l_2, \delta l_1, \delta l_2, M$ )

**Purpose:** This algorithm computes the elements of the exact inverse of a given real unsymmetric  $(n \times n)$  matrix of irregular structure arising in FE/FD

discretization of elliptic and parabolic boundary value in three space dimensions. The algorithm can also compute approximate inverse matrices of the given coefficient matrix.

**Input:** given matrix  $A$ ;  $n$  order of  $A$ ; submatrices  $F, H, \Gamma, Z$ ; parameter  $\varepsilon EM$  (indicating the exact inverse or approximate inverse matrix),  $s, m, p, l_1$  and  $l_2$  numbers of diagonals retained in semi-bandwidths  $m$  and  $p$  respectively,  $\delta l_1$  and  $\delta l_2$  widths of bands in  $A$

**Output:** elements  $\mu_{i,j}$  of the exact inverse  $M$

**Computational Procedure:**

Step 0: Read the value of adaptive parameter  $\varepsilon EM$

//for the appropriate value of adaptive parameter  $\varepsilon EM$  the algorithm computes the exact inverse matrix or approximate inverse matrices//

Call module exactmode-1( $\varepsilon EM$ )

//If the module exactmode-1is activated then the algorithm EBAIM-1 computes the exact inverse of a given unsymmetric matrix of irregular structure using an exact LU factorization, otherwise the algorithm can be used for computing an approximate inverse matrix of the given coefficient matrix//

step 1: Let  $rl1 = r_1 + l_1$ ;  $rl2 = r_2 + l_2$ ;  $rl11 = rl1 - 1$ ;  $rl21 = rl2 - 1$ ;  $mr1 = m - r_1$ ;  $ml1 = m + l_1$ ;  $pr2 = p - r_2$ ;  $pl2 = p + l_2$ ;  $nmr1 = n - m + r_1$ ;  $npr2 = n - p + r_2$

step 2: For  $i = n:1$

step 3: For  $j = i : \max(1, i - \delta l_1 + 1)$

step 4: If  $j > nmr1$  then

step 5: If  $i = j$  then

step 6: If  $i = n$  then

step 7:  $\mu_{n,n} = 1$

step 8: else

step 9:  $\mu_{i,j} = 1 - g_j \mu_{i,j+1}$

step 10:  $\mu_{i,j} = \omega_n - \beta_j \mu_{i,j+1}$

step 11: else

step 12:  $\mu_{i,j} = -g_j \mu_{i,j+1}$

step 13:  $\mu_{i,j} = -\beta_j \mu_{i,j+1}$

step 14: else

step 15: If  $j > npr2$  and  $j \leq nmr1$  then

step 16: If  $i = j$  then

step17:  $\mu_{i,j} = 1 - g_j \mu_{i,j+1} - \sum_{k=0}^{nmr1-j} h_{rl11-k, j+k+1-r1} \mu_{i, j+mr1+k}$

step18:  $\mu_{i,j} = \omega_1 - \beta_j \mu_{i,j+1} - \sum_{k=0}^{nmr1-j} \gamma_{rl11-k, j+k+1-r1} \mu_{i, j+mr1+k}$

step 19: else

step 20:  $\mu_{i,j} = -g_j \mu_{i,j+1} - \sum_{k=0}^{nmr1-j} h_{rl11-k, j+k+1-r1} \mu_{i, j+mr1+k}$

step 21:  $\mu_{i,j} = -\beta_j \mu_{i,j+1} - \sum_{k=0}^{nmr1-j} \gamma_{rl11-k, j+k+1-r1} \mu_{i, j+mr1+k}$

step 22: else

step 23: If  $j \geq rl_1$  and  $j \leq npr2$  then

step 24: If  $i = j$  then

step 25:

$$\mu_{i,j} = 1 - g_j \mu_{i,j+1} - \sum_{k=0}^{nmr1-j} h_{rl11-k,j+k+1-r1} \mu_{i,j+mr1+k} - \sum_{k=0}^{npr2-j} f_{rl12-k,j+k+1-r2} \mu_{i,j+pr2+k}$$

Step 26:

$$\mu_{i,j} = \omega_1 - \beta_j \mu_{i,j+1} - \sum_{k=0}^{nmr1-j} \gamma_{rl11-k,j+k+1-r1} \mu_{i,j+mr1+k} - \sum_{k=0}^{npr2-j} z_{rl12-k,j+k+1-r2} \mu_{i,j+pr2+k}$$

Step 27: else

Step 28:

$$\mu_{i,j} = -g_j \mu_{i,j+1} - \sum_{k=0}^{nmr1-j} h_{rl11-k,j+k+1-r1} \mu_{i,j+mr1+k} - \sum_{k=0}^{npr2-j} f_{rl12-k,j+k+1-r2} \mu_{i,j+pr2+k}$$

Step 29:

$$\mu_{i,j} = -\beta_j \mu_{i,j+1} - \sum_{k=0}^{nmr1-j} \gamma_{rl11-k,j+k+1-r1} \mu_{i,j+mr1+k} - \sum_{k=0}^{npr2-j} z_{rl12-k,j+k+1-r2} \mu_{i,j+pr2+k}$$

Step 30: else

Step 31: If  $i = j$  then

Step 32: If  $i = 1$  then

$$\text{Step 33: } \mu_{1,1} = 1 - g_1 \mu_{1,2} - \sum_{k=1}^{l1} h_{1,k} \mu_{1,m+k-1} - \sum_{k=1}^{l2} f_{1,k} \mu_{1,p+k-1}$$

$$\text{Step 34: } \mu_{1,1} = \omega_1 - \beta_1 \mu_{1,2} - \sum_{k=1}^{l1} \gamma_{1,k} \mu_{1,m+k-1} - \sum_{k=1}^{l2} z_{1,k} \mu_{1,p+k-1}$$

Step 35: else

$$\text{Step 36: } \mu_{i,j} = 1 - g_j \mu_{i,j+1} - \sum_{k=j+1-r1}^{l1} h_{j,k} \mu_{i,m+k-1} - \sum_{k=1}^{j-1} h_{j-k,l1+k} \mu_{i,ml1+k-1}$$

$$\mu_{i,j} = \omega_1 - \beta_j \mu_{i,j+1} - \sum_{k=j+1-r1}^{l1} \gamma_{j,k} \mu_{i,m+k-1} - \sum_{k=1}^{j-1} \gamma_{j-k,l1+k} \mu_{i,ml1+k-1}$$

Step 37:

$$- \sum_{k=1}^{j-1} z_{j-k,l2+k} \mu_{i,pl2+k-1} - \sum_{k=j+1-r2}^{l2} z_{j,k} \mu_{i,p+k-1}$$

Step 38: else

$$\mu_{i,j} = -g_j \mu_{i,j+1} - \sum_{k=j+1-r1}^{l1} h_{j,k} \mu_{i,m+k-1} - \sum_{k=1}^{j-1} h_{j-k,l1+k} \mu_{i,ml1+k-1}$$

Step 39:

$$- \sum_{k=1}^{j-1} z_{j-k,l2+k} \mu_{i,pl2+k-1} - \sum_{k=j+1-r2}^{l2} z_{j,k} \mu_{i,p+k-1}$$

$$\mu_{i,j} = -\beta_j \mu_{i,j+1} - \sum_{k=i+1-r1}^{l1} \gamma_{j,k} \mu_{i,m+k-1} - \sum_{k=1}^{j-1} \gamma_{j-k,l1+k} \mu_{i,ml1+k-1}$$

Step 40:

$$- \sum_{k=1}^{j-1} z_{j-k,l2+k} \mu_{i,pl2+k-1} - \sum_{k=j+1-r2}^{l2} z_{j,k} \mu_{i,p+k-1}$$

Step 41: For  $j = i - 1$  to  $\max[1, i - \delta l + 1]$

Step 42:  $\mu_{i,i} = \mu_{i,j}$

Step 43: Print out the inverse elements  $\mu_{i,i}$

Step 44: End

Note that if the computational module exactmode-1 is activated, as indi-

cated below, then the algorithm EBAIM-1 computes the elements of the exact inverse of a given unsymmetric matrix of irregular structure using an exact LU factorization.

**Module exactmode-1 ( $\varepsilon$ EM)**

/If this module is activated ( $\varepsilon$ EM = 1), then the algorithm EBAIM-1 computes the exact inverse of a given unsymmetric matrix of irregular structure using an exact LU factorization, otherwise the algorithm can be used for computing an approximate inverse matrix of the given coefficient matrix/

```

If  $\varepsilon$ EM = 1 then
Set  $r_1 = p - 1$ ;  $r_2 = m - 1$ 
Set  $l_1 = n - 2$ ;  $l_2 = n - p$ 
Set  $\delta l_1 = n - 2$ ;  $\delta l_2 = n - 2$ 
else
exit
end module exactmode-1

```

The computational work of the EBAIM-1 algorithm is  $\approx O[n(\delta l_1 + \delta l_2)(r_1 + r_2 + l_1 + l_2 + 2)]$  multiplications, while the memory requirements are  $(n \times n)$  words. In the case of very large systems, the memory requirements could be prohibitively high and the usage of efficient memory requirements of approximate inverse matrices is desirable.

### 5.3. An adaptive Preconditioned Conjugate Gradient Method Using the Explicit Approximate Preconditioner

The preconditioned PCG method can solve the problem  $\min \|b - AR^{-1}x\|$ , where  $R$  is the sparse, non-singular  $QR$  factor, while the preconditioned CGLS method can solve the equations:

$$M = R^T R, \quad R^{-T} A^T A R^{-1} u = R^{-T} A^T b \quad \text{and} \quad u = Rx. \quad (29)$$

Note that the factor  $Q$  cannot be stored, while the only additional computational work is solving the two equations  $R^T w = v$  and  $Rz = w$ . All the factorization processes are numerically stable. It should be pointed out that the sparse preconditioner  $M^*$  used in the modified PCG method is the approximate inverse of factor  $R$ , which using is closely related with the so-called mrTIGO method.

In order to compute efficiently the solution of the linear system  $Ax = b$ , a modified Preconditioned Conjugate Gradient (mPCG) method in conjunction to the modified rTIGO method [47] is applied in the following algorithmic form.

**Algorithm mPCG ( $A, b, \text{tol}, x_0, M^*, x$ )**

**Purpose:** a modified PCG method is used for solving a given system of linear equations

**Input:**  $A$  is a symmetric and positive definite coefficient matrix,  $b$  is the right hand side vector,  $\text{tol}$  is the predetermined tolerance,  $x_0$  is the initial guess,  $M^*$  is the required preconditioner

**Output:**  $x$  the solution vector

**Computational Procedure:**

Step 1: Given  $x_0$  and preconditioner  $M$

Step 2: Set  $r_0 = Ax_0 - b$

Step 3: Solve  $My_0 = r_0$

Step 4: Set  $p_0 = -y_0, k = 0$

Step 5: While  $r_k \neq 0$

Step 6: Compute a step length  $a_k = (r_k^T y_k) / (p_k^T A p_k)$

Step 7: Update the approximate solution  $x_{k+1} = x_k + a_k p_k$

Step 8: Update the residual  $r_{k+1} = r_k + a_k A p_k$

Step 9: Solve  $M^* y_{k+1} = r_{k+1}$

Step 10: Compute a gradient correction factor  $\beta_{k+1} = (r_{k+1}^T y_{k+1}) / (r_k^T y_k)$

Step 11: Set the new search direction  $p_{k+1} = -y_{k+1} + \beta_{k+1} p_k$

Step 12:  $\kappa = \kappa + 1$

Step 13: End (while)

This algorithm requires the additional work that is needed to solve the linear system

$$M^* \tilde{y}_n = r_n \quad (29)$$

once per iteration. Therefore, the preconditioner  $M^*$  should be chosen such that can be done easily and efficiently.

The preconditioner  $M^* = G$  that results in a minimal memory use. The storage requirement was the vectors  $r, x, y, p$  and the upper triangular matrix  $G$ , in the data implementation. The convergence rate of preconditioned CG is independent of the order of equations and the matrix vector products are orthogonal and independent. The preconditioned CG method is not self-correcting and the numerical errors accumulate every round. Therefore, to minimize the numerical errors in the pCG, it was used double precision variables at the cost of memory use. The explicit pCG method of second order can be alternatively used in conjunction with the explicit approximate inverse  $M\mu^*$  for solving complex computational problems with the appropriate selection of the E-parameters of **Table 1**. Note that the topics of stability and correctness of incomplete factorization methods have been discussed in a recent research work [48].

The presented second order iterative schemes can be efficiently used for solving 2d and 3d initial and boundary value problems.

## 6. Conclusion

A class of general iterative methods of second order is described. Explicit adaptive iterative schemes and exact/approximate inverse preconditioners have been introduced and the selection of iterative parameters has been discussed. Exact and Approximate Inverse Matrix Algorithmic Techniques have been presented. Exact and Approximate Inverse Preconditioners have been described in adaptive algorithmic form. Explicit preconditioned Conjugate Gradients methods of

**Table 1.** Certain cases of the E-iterative scheme which are no more than various explicit preconditioned methods.

E-parameters			Explicit Preconditioned Methods
$\pi_i$	$\tau_i$	$\delta_i$	
$1 + \beta - \alpha M_r A$	$-\beta$	$\alpha M_r b$	Explicit Richardson (Lipitakis and Evans, 1987)
$1 + \beta_i - \alpha_i M_r A$	$-\beta_i$	$\alpha_i M_r b$	Explicit Chebychev (Lipitakis and Evans, 1987)
$(\rho_i I - G_1)^{-1} (\rho_i I - G_2)^{-1}$	0	$2\rho_i (\rho_i I + G_i)^{-1}$	Generalized AGE method (Lipitakis and Evans, 1981)
$1 + \omega_{i+1} (\rho_i I + G_1)^{-1} (\rho_i I + G_2)^{-1} A$	0	$-\omega_{i+1} (\rho_i I + G_1)^{-1} (\rho_i I + G_2)^{-1} b$	Richardson + AGE method (Evans, 1985)
$\omega_i H_i$	$1 - \omega_i$	$2\omega_i \rho_i (\rho_i I + G_1)^{-1} (\rho_i I + G_2)^{-1} b$	Chebychev + AGE method (Evans, 1985)
$(I + \theta G_1)^{-1} (I + \theta G_2)^{-1} \{(I + \theta G_1)(I + \theta G_2) - \eta A\}$	0	$\eta b$	General fractional step iteration (Marchuk, 1975)
$\prod_{i=0}^{v-1} (I - \alpha M_\mu A)$	0	$\alpha_i M_\mu b$	Multiple explicit Jacobi (Lipitakis, 1984)
$\rho_{i+1} (1 + \gamma_{i+1} M_\mu A)$	$1 - \rho_{i+1}$	$\rho_{i+1} \gamma_{i+1} M_\mu b$	Explicit Preconditioned Conjugate Gradients of second order (Lipitakis <i>et al.</i> , 2017)

second order are given. An Adaptive Preconditioned Conjugate Gradient Method using explicit approximate preconditioners has been also presented. Future research work will be focused on the implementation of the presented methods to parallel computer environments and related applications.

### Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

### References

- [1] Hestenes, M.R. and Stiefel, E. (1952) Methods of Conjugate Gradients for Solving Linear Systems. *Journal of Research National Bureau Standards*, **49**, 409-436. <https://doi.org/10.6028/jres.049.044>
- [2] Evans, D.J. and Lipitakis, E.A. (1980) A Normalized Implicit Conjugate Gradient Method for the Solution of Large Sparse Systems of Linear Equations. *Computer Methods in Applied Mechanics and Engineering*, **23**, 1-19. [https://doi.org/10.1016/0045-7825\(80\)90075-4](https://doi.org/10.1016/0045-7825(80)90075-4)
- [3] Saad, Y. and Schultz, M.H. (1985) Conjugate Gradient-Like Algorithms for Solving Nonsymmetric Linear Systems. *Mathematics of Computation*, **44**, 417-424. <https://doi.org/10.1090/S0025-5718-1985-0777273-9>
- [4] Lipitakis, E.A. and Gravvanis, G.A. (1992) A Class of Explicit Preconditioned Conjugate Gradient Methods for Solving Large Finite Element Systems. *International Journal of Computer Mathematics*, **44**, 189-206. <https://doi.org/10.1080/00207169208804104>
- [5] Gustafsson, I. (1983) Modified Incomplete Cholesky (MIC) Methods. In: Evans, D.J., Ed., *Preconditioning Methods, Theory and Applications*, Gordon and Breach Science Publishers, New York, 265-293.
- [6] van Leeuwen, J. (1990) Handbook of Theoretical Computer Science (Vol. A) Algorithms and Complexity. MIT Press, Cambridge.

- <https://doi.org/10.1016/B978-0-444-88071-0.50015-1>
- [7] Yeremin, A.Y., Kolotilina, L.Y. and Nikishin, A.A. (2000) Factorized Sparse Approximate Inverse Preconditionings III. Iterative Construction of Preconditioners. Computational Methods and Algorithms. Part XIII, Zap. Nauchn. Sem. POMI 248, POMI, St. Petersburg, 1998, 17-48. *Journal of Mathematical Sciences (New York)*, **101**, 3237-3254. <https://doi.org/10.1007/BF02672769>
- [8] Benzi, M. (2002) Preconditioning Techniques for Large Linear Systems: A Survey. *Journal of Computational Physics*, **182**, 418-477. <https://doi.org/10.1006/jcph.2002.7176>
- [9] Benzi, M., Meyer, C.D. and Tuma, M. (1996) A Sparse Approximate Inverse Preconditioner for the Conjugate Gradient Method. *SIAM Journal on Scientific Computing*, **17**, 1135-1149. <https://doi.org/10.1137/S1064827594271421>
- [10] Meijerink, J. and van der Vorst, H.A. (1981) Guidelines for the Usage of Incomplete Decompositions in Solving Sets of Linear Equations as They Occur in Practical Problems. *Journal of Computational Physics*, **44**, 134-155. [https://doi.org/10.1016/0021-9991\(81\)90041-3](https://doi.org/10.1016/0021-9991(81)90041-3)
- [11] Evans, D.J. and Forrington, C.V.D. (1963) An Iterative Process for Optimizing Symmetric Successive Over-Relaxation. *The Computer Journal*, **6**, 271-273. <https://doi.org/10.1093/comjnl/6.3.271>
- [12] Manteuffel, T.A. (1980) An Incomplete Factorization Technique for Positive Definite Linear Systems. *Mathematics of Computation*, **34**, 473-497. <https://doi.org/10.1090/S0025-5718-1980-0559197-0>
- [13] Lipitakis, E.A. and Evans, D.J. (1987) Explicit Semi-Direct Methods Based on Approximate Inverse Matrix Techniques for Solving BV Problems on Parallel Processors. *Mathematics and Computers in Simulation*, **29**, 1-17. [https://doi.org/10.1016/0378-4754\(87\)90062-0](https://doi.org/10.1016/0378-4754(87)90062-0)
- [14] Saad, Y. (2004) Iterative Methods for Sparse Linear Systems. SIAM, Philadelphia. <https://doi.org/10.1137/1.9780898718003>
- [15] Varga, R.S. (1962) Matrix Iterative Analysis. Prentice-Hall, Upper Saddle River.
- [16] Golub, G.H. (1961) Semi-Iterative Methods, Successive Over-Relaxation Iterative Methods, and Second-Order Richardson Iterative Methods. I, II. *Numerische Mathematik*, **3**, 147-168. <https://doi.org/10.1007/BF01386014>
- [17] Peaceman, D.W. and Rachford, H.H. (1955) The Numerical Solution of Parabolic and Elliptic Differential Equations. *Journal of the Society for Industrial and Applied Mathematics*, **3**, 28-41. <https://doi.org/10.1137/0103003>
- [18] Young, D. (1972) Second-Degree Iterative Methods for the Solution of Large Linear Systems. *Journal of Approximation Theory*, **5**, 137-148. [https://doi.org/10.1016/0021-9045\(72\)90036-6](https://doi.org/10.1016/0021-9045(72)90036-6)
- [19] Young, D.M. (1971) Iterative Solution of Large Linear Systems. Academic Press, New York.
- [20] Fadeev, D.K. and Fadeeva, V.N. (1963) Computational Methods in Linear Algebra. W.H. Freeman and Company, San Francisco.
- [21] Niethammer, W., de Pillis, J. and Varga, R.S. (1984) Block Iterative Methods Applied to Sparse Least-Squares Problems. *Linear Algebra and Its Applications*, **58**, 327-341. [https://doi.org/10.1016/0024-3795\(84\)90218-0](https://doi.org/10.1016/0024-3795(84)90218-0)
- [22] Kincaid, D.R. (1972) On Complex Second-Degree Iterative Methods. *SIAM Journal on Numerical Analysis*, **11**, 211-218. <https://doi.org/10.1137/0711020>
- [23] Kang, S.M., Rafiq, A. and Kwun, Y.C. (2013) A New Second-Order Iteration Me-

- thod for Solving Nonlinear Equations. *Abstract and Applied Analysis*, **2013**, Article ID: 487062. <https://doi.org/10.1155/2013/487062>
- [24] Kogan, T., Sapir, L. and Sapir, A. (2007) A Nonstationary Iterative Second-Order Method for Solving Nonlinear Equations. *Applied Mathematics and Computation*, **188**, 75-82. <https://doi.org/10.1016/j.amc.2006.09.092>
- [25] Newton, I. (1711) A Treatise of Algebra Both Historical and Practical, Published by John Wallis; De analysi per aequationes numero terminorum infinitas (Written in 1669, Published in 1711 by William Jones) and in De metodis fluxionum et serierum infinitarum (Written in 1671, Translated and Published as Method of Fluxions in 1736 by John Colson), 1685.
- [26] Khan, W.A., Noor, M.A. and Rauf, A. (2015) Second Derivatives Free Fourth-Order Iterative Method Solving for Nonlinear Equation. *Applied Mathematics*, **5**, 15-20.
- [27] Antipin, A.S., Mijailovic, N. and Jacimovic, M. (2013) A Second-Order Iterative Method for Solving Quasi-Variational Inequalities. *Computational Mathematics and Mathematical Physics*, **53**, 258-264. <https://doi.org/10.1134/S0965542513030032>
- [28] Saqib, M., Iqbal, M., Ahmed, S., Ali, S. and Ismael, T. (2015) New Modification of Fixed Point Iterative Method for Solving Nonlinear Equations. *Applied Mathematics*, **6**, 1857-1863. <https://doi.org/10.4236/am.2015.611163>
- [29] Zhong, Y. and Wong, M.D.F. (2007) Efficient Second-Order Iterative Methods for IR Drop Analysis in Power Grid. Department of Electrical and Computer Engineering University of Illinois at Urbana-Champaign, Champaign. <https://doi.org/10.1109/ASPDAC.2007.358082>
- [30] Lipitakis, A. and Lipitakis, E.A.E.C. (2014) E-Business Performance and Strategy Planning E-Valuation Based on Adaptive Algorithmic Modelling Methods: Critical Factors Affecting E-Valuation and Strategic Management Methodologies. *Universal Journal of Management*, **2**, 81-91.
- [31] Papadimitriou, C. (1994) Computational Complexity. Addison Wesley, Boston.
- [32] Blondel, V.D. and Tsitsiklis, J.N. (2000) A Survey of Computational Complexity Results in Systems and Control. *Automatica*, **36**, 1249-1274. [https://doi.org/10.1016/S0005-1098\(00\)00050-9](https://doi.org/10.1016/S0005-1098(00)00050-9)
- [33] Goldreich, O. (2008) Computational Complexity: A Conceptual Perspective. Cambridge University Press, Cambridge. <https://doi.org/10.1017/CBO9780511804106>
- [34] Arora, S. and Barak, B. (2009) Computational Complexity: A Modern Approach. Cambridge University Press, Cambridge. <https://doi.org/10.1017/CBO9780511804090>
- [35] Nocedal, J. and Wright, S. (2006) Numerical Optimization. 2nd Edition, Springer, New York.
- [36] Sastry, S.P., Shontz, S.M. and Vavasis, S.A. (2014) A Log-Barrier Method for Mesh Quality Improvement and Untangling. *Engineering with Computers*, **30**, 315-329. <https://doi.org/10.1007/s00366-012-0294-6>
- [37] Lipitakis, E.A. (1990) Euclid's Algorithm and Accelerated Iterative Methods. *Special Issue on Computer Mathematics, Bulletin of the Greek Mathematical Society*, **31**, 51-76.
- [38] Lipitakis, E.A. and Evans, D.J. (1981) The Rate of Convergence of an Approximate Matrix Factorization Semi-Direct Method. *Numerische Mathematik*, **36**, 237-251. <https://doi.org/10.1007/BF01396653>
- [39] Lipitakis, E.A. (1984) Generalized Extended to the Limit Sparse Factorization Techniques for Solving Large Unsymmetric Finite Element Systems. *Computing*, **32**,

- 255-270. <https://doi.org/10.1007/BF02243576>
- [40] Lipitakis, E.A. (1984) The Use of Second Degree Normalized Implicit CG Methods for Solving Large Sparse Systems of Linear Equations. *Journal of Computational and Applied Mathematics*, **11**, 39-47.  
[https://doi.org/10.1016/0377-0427\(84\)90030-X](https://doi.org/10.1016/0377-0427(84)90030-X)
- [41] Douglas, J. and Rachford, H.H. (1956) On the Numerical Solution of Heat Conduction Problems in Two or Three Space Variables. *Transactions of the American Mathematical Society*, **82**, 421-439.  
<https://doi.org/10.1090/S0002-9947-1956-0084194-4>
- [42] Axelsson, O. (1994) Iterative Solution Methods. Cambridge University Press, Cambridge. <https://doi.org/10.1017/CBO9780511624100>
- [43] Evans, D.J. (1985) Group Explicit Iterative Methods for Solving Large Linear Systems. *International Journal of Computer Mathematics*, **17**, 81-108.  
<https://doi.org/10.1080/00207168508803452>
- [44] Marchuk, G.I. (1975) Methods of Numerical Mathematics. Springer-Verlag, Berlin.  
<https://doi.org/10.1007/978-1-4615-9988-3>
- [45] Yanenko, N.N. (1971) The Method of Fractional Steps. Springer-Verlag, Berlin.  
<https://doi.org/10.1007/978-3-642-65108-3>
- [46] Howes, F.A. (1986) Multidimensional Initial Boundary Value Problems with Strong Nonlinearities. *Archive for Rational Mechanics and Analysis*, **91**, 153-168.  
<https://doi.org/10.1007/BF00276861>
- [47] Papadopoulos, A.T., Duff, I.S. and Wathen, A.J. (2002) Incomplete Orthogonal Factorization Methods Using Givens Rotations II: Implementation and Results. Tech. Report 02/07, Oxford University Computing Laboratory, Oxford.
- [48] Lipitakis, A.-D. (2020) Stability and Correctness of Incomplete Factorization Methods. ADL-HU-DP-TR47, Depart. of Informatics and Telematics, Harokopio University of Athens, Athens.