СТРУЧНИ ЧЛАНЦИ
ПРОФЕССИОНАЛЬНЫЕ СТАТЬИ
PROFESSIONAL PAPERS

THE METHOD OF "EXTERNAL SPIRAL" FOR SOLVING A LARGE SYSTEM OF LINEAR EQUATIONS

Aleksa S. Srdanov[a], *Radiša* R. Stefanović[b], *Nada* V. Ratković Kovačević[c], *Aleksandra* M. Jovanović[d], *Dragan* M. Milovanović[e]

[a] Technical College of Vocational Studies, Požarevac, Republic of Serbia, e-mail: aleksa.srdanov@vts-pozarevac.edu.rs, ORCID iD: http://orcid.org/0000-0001-6042-0750

[b] Technical College of Vocational Studies, Požarevac; University of Defence in Belgrade, Military Academy, Belgrade; Republic of Serbia, e-mail: radisastefanovic@yahoo.com, ORCID iD: http://orcid.org/0000-0001-5497-3826

[c] Technical College of Vocational Studies, Belgrade, Republic of Serbia, e-mail: nratkovicmf@gmail.com, ORCID iD: http://orcid.org/0000-0001-6398-4391

[d] Technical College of Vocational Studies, Požarevac, Republic of Serbia, e-mail: aleksandra.jovanovic@vts-pozarevac.edu.rs, ORCID iD: http://orcid.org/0000-0003-1621-1120

[e] Technical College of Vocational Studies, Požarevac, Republic of Serbia, e-mail: dragan.milovanovic@vts-pozarevac.edu.rs, ORCID iD: http://orcid.org/0000-0002-0320-2957

http://dx.doi.org/10.5937/vojtehg66-14625

FIELD: Mathematics, Computer Sciences
ARTICLE TYPE: Professional Paper
ARTICLE LANGUAGE: English

Abstract:

Solving a linear system of n × n equations can be very difficult for the computer, especially if one needs the exact solution, even when the number n - of equations and of unknown variables is relatively small (a few thousands). All existing methods have to overcome at least one of the following problems: 1. Computational complexity, which is expressed with the number of arithmetic operations required in order to determine a

solution; 2. The possibility of overflow and underflow problems; 3. Causing variations in the values of some coefficients in the initial system, which may be leading to instability of the solution; 4. Requiring additional conditions for convergence; 5. In cases of a large number of equations and unknown variables it is often required that the systems matrix be: either sparse, or symmetrical, or diagonal, etc. This paper presents a method for solving a system of linear equations of arbitrary order (any number of equations and unknown variables) to which the problems listed above do not reflect.

Key words: system of linear equations, method of "external spiral", hyperplane.

Introduction

If we perceive mathematics as a science oriented primarily towards a man as a subject of its application, the problem of solving large systems of linear equations is not a mathematical one. It is essentially the problem steming from computer science since the very forming of such a system is impossible without the help of computers. Just to write down thousands of equations with thousands of unknowns, a man would have to spend a lifetime. Let us suppose you need to solve the full system of linear equations having a very large number of equations and unknowns, e.g. *n ~ 100,000 or more*. In order to solve such a system, it is necessary to devise a method which: 1) requires execution of the least possible number of operations; 2) does not require exhaustive memory usage; 3) does not produce unexpected overflow and underflow effects; 4) does not change the coefficients of the initial system; 5) can be applied for an arbitrary system scale – any number of equations; 6) does not insist that the systems matrix has any particular additional feature, and 7) unconditionally and quickly converges starting from an arbitrary initial point. The existing methods do not meet at least one of the issues listed above. The difficulties stated in abstract from 1. to 5., as well as the requirements listed above from 1) to 5) are well-known and described, e. g. in (Boht, 1978), (Higham, 2002), (Stoer & Bulirsch, 2002), and some in connection to solving large systems of linear differential equations in modelling complex systems, such as (Randall, 2015) or in other real-world applications (Gajić et al, 2008).

An exact solution to the system of $n \times n$ has n components. Each of these components functionally depends on every individual coefficient of the system. The number of coefficients is of order n^2. This implies that a minimum number of operations required to obtain the exact solution of the system of $n \times n$ is proportional to the number n^3, in general. If we

want to reach a solution using fewer operations, it is necessary to seek the approximate methods or approximate solution.

Formulation of the problem

Assume that the computer memory contains a system of linear equations $n \times n$ (where n is much higher than 1000). Let this system has the following form:

$$\begin{aligned} a_{1,1}x_1 + a_{1,2}x_1 + \ldots + a_{1,n}x_1 &= b_1 \\ a_{2,1}x_1 + a_{2,2}x_1 + \ldots + a_{2,n}x_1 &= b_2 \\ a_{n,1}x_1 + a_{n,2}x_1 + \ldots + a_{n,n}x_1 &= b_n \end{aligned} \quad (1)$$

For n greater than a thousand, the system (1) can only be formed inside a computer memory. For a system like this, it is possible to define the procedure for finding the solution, which has a minimal, almost trivial computation, and then, if the solution exists, to assure convergence to it.

For this work the only assumption made is that system (1) has a solution, as well as that when checking whether obtained result is a solution, there is no overflow in the intermediate results in any of equations in (1). If overflow of this kind happened, then it would be impossible to solve the system (1) in such an environment. Although the initial assumption is that system (1) has a solution, the procedure defined here could be utilized to determine whether system (1) is insolvable (impossible) or if it has some degree of freedom. It is important to assume that the solution of (1) exists in order to assure the convergence of the method.

Description of the procedure

In order to reach a solution of system (1), it is not necessary to solve it. To find the solution of system (1), it is suitable to apply a cybernetic approach (method of invariants), similar to the procedure defined in (Srdanov & Stefanović, 2017). Each equation in a system given in (1) carries a considerable amount of information. For example, each equation from system (1) can be interpreted geometrically as one hyperplane within a space whose dimension is at least equal to the number of unknown variables e. g. n. Then, each equation from (1) can reveal the coordinates of the line perpendicular to its respective hyperplane, and so forth. Parts of this information can be utilized in the process of finding solution of system (1). The essence of this idea is as

following: If the solution of system (1) exists, then it is a point of intersection of all hyperplanes defined by the equations of system (1). Starting the calculation of the solution approximation from an arbitrary point and advancing further using orientations and the positions of all hyperplanes makes it possible to reach a common intersection eventually.

At this point, it is necessary to precisely define the term angle between two hyperplanes – to generalize the term angle between two intersecting planes.

The angle between two intersecting three-dimensional spaces, both contained within the common four-dimensional space, is defined analogously to the definition of a dihedral angle between two intersecting planes within the same three-dimensional space. The angle between two three-dimensional spaces can be defined as a dihedral angle contained within a third three-dimensional space which is perpendicular to both of these. This dihedral angle consists of two intersecting planes of the third three-dimensional space with the previous two three-dimensional spaces. All three three-dimensional spaces are contained within the same four-dimensional space. Therefore, the generalized dihedral angle between the two three-dimensional spaces is the dihedral angle located within the third three-dimensional space that is perpendicular to both of the previous spaces (dihedron that makes up the cross-section area of the third space with the previous two spaces). All of these three three-dimensional spaces are within the same four-dimensional space.

Generalizing previous, more same-dimensional dihedral angles having a common point can be called justifiably the clew.

In a completely analogous manner, the angles between two four-dimensional spaces contained within the same five-dimensional space could be defined, etc.

For the first approximation of the unknown solution, an arbitrary point could be taken from the space the solution belongs to. Statistically speaking, the most optimal way to choose the starting point is pseudo-random. The problem connected to this is that the existing algorithms for generating pseudo-random numbers, as arule, provide pseudo-random choices within some range. Therefore, a pseudo-random initial choice should not take precedence over the chance to choose the starting point as always the same. If we choose the initial point in advance, it is preferable that it is the point $O = \underbrace{(0,0,\ldots,0)}_{n}$. A better method for the selection of the starting point is to determine the "centroid of the system (1)". To achieve this, it would be perfect to add up all the columns in

system (1) and all the results. Then, for the initial values of all unknown variables it would be perfect to take the value given by: $x_i^0 = \frac{1}{n} \cdot \frac{\sum_{i=1}^{n} b_i}{\sum_{j=1}^{n} a_{i,j}}$ if the nominator is not zero, and $x_i^0 = \frac{1}{n} \sum_{i=1}^{n} b_i$ otherwise, for all $i = 1, 2, ..., n$.

To determine the subsequent approximation of the solution of system (1) it is required to extract from (1) the two equations simultaneously. From now on, the first equation of these two will be referred to as the first plane, and the second equation - the second plane. Each equation is a hyperplane. The two hyperplanes within the space of the same dimension can have one of the following relationships - that these two hyperplanes: intersect, do not intersect or are overlaping each other. If hyperplanes do not belong to the same dimensional space, very different mutual relations are possible, which is not relevant from the point of view of this article.

Let us suppose that the two hyperplanes intersect. Then we can examine in detail the following cases. Dihedral angle that these two are forming is: acute, right or obtuse. Besides this, a position that the approximation of the solution has reached, namely the point O, is important as well (the position of O compared to these hyperplanes). The point O may: belong only to one hyperplane, to both hyperplanes or to neither one of these two hyperplanes. The idea of the method is that we should reach some of the points within the intersection of the first and the second hyperplane ariving from the initial point O. This would complete one semi-iteration. Then we eliminate the first hyperplane, the second hyperplane is declared to be the first, and for the second hyperplane - the following equation from system (1) is proclaimed. From the previously attained point O, we are descending into one of the points belonging to the intersection of the newly examined two hyperplanes. When the last of all equations in system (1) has been examined, a full iteration is completed.

The procedure of selecting one of the points from the intersection differs for each of the three above mentioned cases. It can be described as follows. Through the point O, we place two straight lines whose directions are determined by the perpendicular vectors of the first and the second hyperplane. We distinguish between the three cases: a) to c).

a) The point O already belongs to the intersection of the hyperplanes in concern. In this case, we should proceed with calculation retaining the previous point O.

b) The point O belongs to only one of the two hyperplanes. Then we should determine the point of penetration the intersection of the two hyperplanes with the line positioned perpendicular to the other hyperplane. Through this point of penetration, we set the hyperline and determine its intersection with the other hyperplane. The resulting penetration point represents the next choice for the point O.

c) The point O does not belong to any of the noted hyperplanes. Then we determine penetrations through both hyperplanes using the perpendicular hyperlines drawn from O. Thus, in each hyperplane we get two points that define the new pair of hyperlines. The intersection point of these two hyperlines should be the next position for O.

The method defined here differs from the one given in (Srdanov & Stefanović, 2017) as follows. The points we are taking to get closer to a solution, according to the method in (Srdanov & Stefanović, 2017), belong to a spiral located within some of the 2^n clews which are forming up all of the hyperplanes corresponding to all of the equations of system (1). Wherein, if a point is within the dihedron that is larger than a right angle, the next point should be taken in the new clew. This is carried out until the point is taken the clew which has all dihedral angles acute. Then further convergence towards a solution follows a unique internal spiral whose points of intersection with the clew are the bases of the perpendicular lines placed from one point of the dihedron side on the neighboring side (internal spiral). The method described in this paper follows the external spiral - the spiral located on one side of the clew face which passes through the points obtained as the bases of the perpendiculars from a point of the edge of the dihedron to the adjacent edge of the same face. The term the edge of the dihedron is basically the same: the edge is formed from the intersection points of the dihedron sides - dihedral planes.

For the system described with (1), the coordinates of the orthonormal vectors of all hyperplanes are given as the coefficients of the system: $\vec{n}_l = \left(a_{i,1}, a_{i,2}, \ldots, a_{i,n}\right) \;\; i = 1,2,\ldots,n.$ A procedure of finding an intersection of the line going through the given point O and being perpendicular to the second hyperplane requires a minimum of computation and is given as follows. Let the point O has the coordinates $\left(y_1^{'}, y_2^{'}, ..., y_n^{'}\right)$. Let the orthogonal vector of the first plane be

$\vec{n}_l = \left(a_{i,1}, a_{i,2}, \ldots, a_{i,n}\right)$, and of the second $\overrightarrow{n_{i+1}} = \left(a_{i+1,1}, a_{i+1,2}, ..., a_{i+1,n}\right)$, where $1 \le i \le n-1$. In order to determine the penetration point for the first perpendicular line through the second plane, i. e. the point $X_1 = \left(x_1^{'}, x_2^{'}, ..., x_n^{'}\right)$, we determine firstly $t = \dfrac{b_{i+1} - \left(a_{i+1,1} \cdot y_1^{'} + a_{i+1,2} \cdot y_2^{'} ... + a_{i+1,n} \cdot y_n^{'}\right)}{\overrightarrow{n_i} \cdot \overrightarrow{n_{i+1}}}$, and then $x_j^{'} = a_{i,j} \cdot t + y_j^{'}$, where $j = 1,2,..., n$. To determine the penetration point of another perpendicular line, i. e. the point $X_2 = \left(x_1^{"}, x_2^{"}, ..., x_n^{"}\right)$, we determine at first $t = \dfrac{b_{i+1} - \left(a_{i+1,1} \cdot y_1^{'} + a_{i+1,2} \cdot y_2^{'} ... + a_{i+1,n} \cdot y_n^{'}\right)}{\left\|\overrightarrow{n_{i+1}}\right\|^2}$, and then $x_j^{"} = a_{i+1,j} \cdot t + y_j^{'}$, where $j = 1,2,..., n$. Now the line is placed through the points X_1 and X_2 and its intersection with the first plane is determined. For this, it is necessary to calculate $t = \dfrac{b_i - \left(a_{i,1} \cdot x_1^{'} + a_{i,2} \cdot x_2^{'} ... + a_{i,n} \cdot x_n^{'}\right)}{\overrightarrow{n_i} \cdot \overrightarrow{\left(x_i^{'} - x_i^{"}\right)}}$ prior to calculating the coordinates of the new O point using the formulas $y_j^{'} = \left(x_j^{'} - x_j^{"}\right) \cdot t + x_j^{'}$, $j = 1,2,..., n$. When a full circle is taken, from $i = 1$ to $i = n-1$, the next iteration can begin.

Only in the first intermediate step, the point O can be placed outside both of the two planes observed. Each subsequent choice of the point O would belong to at least the first plane. During computation of intermediate steps within the same iteration, the following cases should be considered for in the program. a) The two planes that have been reached are mutually perpendicular. Then the line perpendicular to the second plane should be placed through the point O, and through its intersection with the second plane, a line should be placed that is perpendicular to the first plane. Then the intersection of this line with the first plane should be determined. b) The two planes reached are two parallel hyperplanes. Then, there are two passing lines through the point O that are the same (identical) and these lines penetrate both hyperplanes in the same points. The system is then impossible and the process should be stopped. c) Similarly to the above, only the

Srdanov, A. et al, The method of "external spiral" for solving large system of linear equations, pp.399-414

penetration points in the hyperplanes observed are coinciding with the point O. If this is the case, the process should be stopped because system (1) is indetermined.

When one pass through all of the equations is made this way, one full iterative cycle is completed. The next iterative cycle will begin with placing the line perpendicular to the first hyperplane through the O point reached in the last hyperplane.

The Problem of Accuracy and Completion of Iterating Process

After a full circle and one iteration completed, checks should be performed whether the required accuracy is reached. The problem of accuracy and completion of repeating the iterative method is possible to display in ranked levels. The user should choose the one level offering balance between the number of iterations required and acceptable accuracy. The method to be applied to a large number of equations and unknown variables should require the least possible necessary additional computation in every step, since in total there is a considerable amount of computation already. Another goal is that the solution obtained has the least possible relative deviation from the exact one, when each unknown variable is considered. The highest level of accuracy for the "internal spiral" is achieved by checking out whether the sum of the distances from the last point obtained to all of the planes is small enough. For the method according to the "external spiral", this criterion may be the sum of all distances between two consecutive points obtained over a full iterative cycle (the length of an arc of the spiral). In the first case, the number of operations is proportional to the number $2n^2+n$, and in the latter case - to the number n^2. These criteria for checking the accuracy both require relatively large number of operations. The next level of accuracy in both methods may be based on the length of the distance between the points of two consecutive iterations. In both methods, this criterion requires $2n$ operations. The number of operations is considerably lower; however, the resulting conclusion is in accordance with that. Using such a criterion in cases where the clew has all dihedral angles very small, it is possible to be still relatively far from the solution and to erroneously assume that the very high accuracy is already reached. The third level of accuracy may include checking out the distance of two consecutive semi-iterations or something of a kind. The number of necessary operations will not be decreased significantly; however, the end result may be improved considerably.

Apart from selecting any of the criteria above, regardless of the criterion rank, it is possible to reduce significantly the number of operations in the following manner. It is not necessary to check out the accuracy attained after each iterative cycle, but after ten or one hundred of full iterations.

Insolvable (Impossible) system and indetermined system

If the given system has no solution (impossible or insolvable system), the procedure described here can detect that. In terms of Geometry (geometrically speaking), there is no solution if any two of the hyperplanes are parallel. The method presented here can detect this if it happens at any position that the two adjacent hyperplanes have perpendicular lines of the same direction. Having in mind the process of solving equations, as soon as something like that is established, the procedure should be stopped. If paralel hyperplanes are not adjacent, then this can also be determined in advance, prior to solving - that the system is impossible or insolvable, by checking out whether any two vectors are of the same direction. The number of required operations is proportional to the number n^2.

The method proposed here provides a much simpler way to determine whether the system is impossible (insolvable) when compared to the method given in (Srdanov & Stefanović, 2017).

If the system is indetermined, then its uncertainty can be numerically evaluated with respect to the degrees of freedom. In the case observed, the system may have from 1 to 99999 degrees of freedom. If there is one degree of freedom, then all of the hyperplanes have a common line; If there are two degrees of freedom, then all hyperplanes have a common plane; If there are three degrees of freedom, then all hyperplanes have common three-dimensional space within the space of one hundred thousands dimensions, etc; If there are 99999 degrees of freedom, then all equations of the system represent one and the same hyperplane within the space of one hundred thousands dimensions.

In the algorithm proposed, the simplest way is just to establish that the system is indetermined because to find out more than that would require many more checks, which is not of great importance for this paper.

The method proposed here does not always detect that the system is indetermined and in cases when it is and not detected, the program will report the solution. If the program is run again only this time from a

different starting point, the method will again provide a solution, however different from the previous one. This way, it is possible to always accurately distinguish whether the obtained solution is unique or the system is indetermined.

Convergence and speed

We will present an outline of proof, while the rigorous proof differs only in detail. Let there be a system of *n* x *n* linear equations, where $n \in N$, which has a solution $X\left(y_1^*, y_2^*, ..., y_n^*\right)$. Let us assume that we have started solving that system using the method proposed here and starting from an arbitrary point of the first hyperplane, $O_1\left(y_1^0, y_2^0, ..., y_n^0\right)$. Then the next point in the first iteration is denoted by $O_2\left(y_1^1, y_2^1, ..., y_n^1\right)$. In (Srdanov & Stefanović, 2017), the subsequent iteration next point was the orthonormal projection of the point O_1 to the first hyperplane, denoted *M*. In this paper, the next iteration is obtained as the cross-section of the plane perpendicular to both of the hyperplanes with the "dihedral edge" and this is the point O_2. Now the angles $\angle O_1MX$ and $\angle MO_2X$ are both right angles. It is obvious that $O_1X > MX > O_2X$ (as the hypotenuse is of greater length than both catheti). It should be noted here that the reason why this method is faster than the method suggested in (Srdanov & Stefanović, 2017) is because here one point is approaching along two different catheti of the two right-angled triangles. In practice this means twice faster than the method in (Srdanov & Stefanović, 2017).

If we denote with $d_j^1, \quad j = 1,2,...,n$ the distances between the solution and consequtive points of the first iteration, then after the first step of the first iteration is applied, it holds $d_1^1 = d \cdot \cos\left(\angle P_1^1 X P_2^1\right) < d$. A perpendicular line always enters acute angle, which may be zero only if the system has no solutions (parallel hyperplanes). This way we obtain a sequence $d_n^1 < d_{n-1}^1 < ... < d_1^1 < d$. As the procedure is extended in an analogous manner during the following iterations, it is to conclude that the method always converges provided that the system has a solution.

If all angles are right, one single complete iteration is sufficient to reach the exact solution.

The number of required operations

Once the lengths of the vectors perpendicular to the hyperplanes corresponding to the equations of the system are calculated, it is not necessary to re-calculate these again, and this requires $2n^2$ operations. The program does not always passes through the same path during its execution and the various branches require a different number of operations. The highest number of operations is needed when the point O does not belong to any hyperplane, and these intersect at a sharp angle - then one semi-iteration requires $6n^2+3n$ necessary operations. In any other case one semi-iteration requires $4n^2+2n$ operations. To determine the accuracy, the $2n^2$ operations are needed at most. It can be estimated that to complete one iteration, the number of operations required is proportional to n^2. A solution can be reached after m steps. To conclude, it can be stated that the number of operations required by this method is of order n^2.

An example

Let us assume that a following system of linear equations is given:

$$43x_1-11x_2+13x_3-17x_4+19x_5-23x_6+29x_7-31x_8+37x_9-41x_{10} = -496$$
$$41x_1-43x_2+11x_3-13x_4+17x_5-19x_6+23x_7-29x_8+31x_9-37x_{10} = -1008$$
$$37x_1-41x_2+43x_3-11x_4+13x_5-17x_6+19x_7-23x_8+29x_9-31x_{10} = -204$$
$$31x_1-37x_2+41x_3-43x_4+11x_5-13x_6+17x_7-19x_8+23x_9-29x_{10} = -864$$
$$29x_1-31x_2+37x_3-41x_4+43x_5-11x_6+13x_7-17x_8+19x_9-23x_{10} = 0$$
$$23x_1-29x_2+31x_3-37x_4+41x_5-43x_6+11x_7-13x_8+17x_9-19x_{10} = -864$$
$$19x_1-23x_2+29x_3-31x_4+37x_5-41x_6+43x_7-11x_8+13x_9-17x_{10} = 204$$
$$17x_1-19x_2+23x_3-29x_4+31x_5-37x_6+41x_7-43x_8+11x_9-13x_{10} = -1008$$
$$13x_1-17x_2+19x_3-23x_4+29x_5-31x_6+37x_7-41x_8+43x_9-11x_{10} = 496$$
$$11x_1-13x_2+17x_3-19x_4+23x_5-29x_6+31x_7-37x_8+41x_9-43x_{10} = -1008$$

The exact solution of this system is:

x1=11; x2=13; x3=17; x4=19; x5=23; x6=29; x7=31; x8=37; x9=41; x10=43.

In the paper (Srdanov & Stefanović, 2017) the same example has been tested. Then the following report has been received:

The solution is reached in 205 semi-iterative steps. (20 complete iterations) 10.9999 12.9998 16.9998 18.9999 23 29.0001 31.0002 37.0002 41.0001 43.

The method derived here is considerably improved compared to the method in (Srdanov& Stefanović, 2017).

The program developed in accordance with the instructions given in this paper is outlined by the following pseudo-code:

```
int main() {
  double t, br1, im1, Eps = .00001, Accuracy= 1.0;
  int i, j, k, m = -1, m1, check_out = 0, flagX1;
  UploadSystem(A,B);
  for(m = 0; m < 99; m++)
     for(i = 0; i < 100; i++) {Modul1[m] = += A[m][i]*A[m][i];
Modul2[m] += A[m][i]*A[m+1][i];}
  for(i = 0; i < 100; i++) {XD[i] = 0.0; Modul1[m] += A[m][i]*A[m][i];
Modul2[m] += A[0][i]*A[m][i];}
  while (Accuracy > Eps) {
// at first descend to intersection, if possible
     m++;
     if (m == 100) {m = 0; check_out++;}
     m1 = m+1;
     if (m == 99) m1 = 0;
     for(k = 0; k < 100; k++) X0[k] = XD[k];
// perpendicular line to the first hyperplane and intersection with
// the second hyperplane
     if (Modul2 != 0) {
          br1 = 0;
          for(k = 0; k < 100; k++) br1 += A[m1][k]*X0[k];
          t = (B[m1] - br1)/Modul2[m1];
          if (t != 0) for(k = 0; k < 100; k++) X1[k] = A[m][k]*t + X0[k];
          else FlagX1 = 1;}
     else
          {if(t == 0) cout << "THE SYSTEM IS UNDEFINED";
          else cout << "THE SYSTEM IS IMPOSSIBLE "; return 1;}
// the perpendicular line to the second hyperplane and intersection
// with the second hyperplane
      if (Modul1 != 0) {
          br1 = 0;
          for(k = 0; k < 100; k++) br1 += A[m1][k]*X0[k];
          t = (B[m1] - br1)/Modul1[i];
          if (t != 0) for(k = 0; k < 100; k++) X2[k] = A[m1][k]*t + X0[k];
          else flagX2 = 1;}
       else
```

```
                {if(t == 0) cout << "THE SYSTEM IS UNDEFINED";
                 else cout << "THE SYSTEM IS IMPOSSIBLE "; return 2;}
      // forming the line through intersections obtained
             for(k = 0; k < 100; k++) X3[k] = X2[k] - X1[k];
      br1 = 0; im1 = 0;
      for(k = 0; k < 100; k++) {
                    br1 += A[m][k]*X1[k];
                    im1 += A[m][k]*X3[k];   }
      // intersection of the line obtained with the first hyperplane
             t = (B[m] - br1)/im1;
      if (t == 0) {
                   cout << "THE SYSTEM IS IMPOSSIBLE ";
                else
                    if((br1 == B[m]) && (im1 == 0))
                    cout << " THE SYSTEM IS UNDEFINED";
                return 3; }
      else
          for(k = 0; k < 100; k++) XD[k] = X3[k]*t + X1[k];
          which_one += .1;
      // Assesing the accuracy reached
      if (check_out == 2) {
          Accuracy = 0.0;
          for(i = 0; i < 100; i++) Accuracy += abs(XD[i] - XL[i]);
          check_out = 0;
          for(k = 0; k < 100; k++) XL[k] = XD[k];}
                for(i = 0; i < 100; i++) X0[i] = XD[i]; }
            cout << "Finished in k = "<< which_one << " semi-
iterations " << "  The solution is : \n";
      for(k = 0; k < 100; k++) cout << XD[k]<< "  ";
      return 0;
      }
```

After running that program the following report is received:

The solution is reached in 67 semi-iterative steps (6 completed iterations).

11 13 17 19 23 29 31 37 41 43

It may be noted that three times fewer iterations were performed, and the solution obtained is the exact one.

The algorithm proposed here should be significantly expanded and improved in other ways provided the present method is used. For the example chosen, the speed and accuracy were of greater importance than the other issues that were listed at the beginning of this paper, and are followers of all known methods for solving systems of equations.

Therefore, the program was described with the condensed and simplified algorithm, which can be the core of a more complex and detailed serious algorithm.

Conclusion

From the computational point of view, the method proposed here offers an extremely simple procedure. In addition, the proposed method: 1. Requires the number of operations that is feasible by the computer; 2. It does not produce overflow and underflow effects, except in case when the size of solution causes that (when it is impossible to avoid this by any method); 3. The memory usage is proportional to the number n^2 (where n is the number of unknowns); 4. It does not change the initially given coefficients of the system (therefore, it uniformly converges to a solution for each coordinate with a relative error distributed uniformly over all coordinates); 5. If there is a solution, it always converges to it. This procedure may be initiated from any starting point of space solutions. 6. It is easy to assemble an algorithm and allows the determination in (to distinguish between) cases when the system is either impossible or in determinate. 7. It differs from the method in (Srdanov& Stefanović, 2017) as a process that is able to determine if a system is possible, impossible or indeterminate.

References

Boht, Z. 1978. *Numerčne metode*. Ljubljana: Državna založba Slovenije (in Slovenian).

Gajić, Z., Lim, M-T.,Škatarić, D., Su, W-C. & Kecman, V. 2008. *Optimal Control: Weakly Coupled Systems and Applications (Automation and Control Engineering)*, 1st ed. Boca Raton, FL: CRC Press.

Higham, N. 2002. *Accuracy and Stability of Numerical Algorithms*, 2nd ed. Philadelphia, PA: SIAM - Society for Industrial and Applied Mathematics.

Randall, D.A. 2015. *An Introduction to the Global Circulation of the Atmosphere.* Princeton, NJ: Princeton University Press.

Srdanov, A. & Stefanović, R. 2017. Kako rešiti linearni sitem sa ekstremno mnogo nepoznatih. In: *INFOTEH, Jahorina*, pp.593-596, March (in Serbian).

Stoer, J. & Bulirsch, R. 2002. *Introduction to Numerical Analysis,* 3rd ed. New York: Springer.

ПРИМЕНЕНИЕ МЕТОДА «ВНЕШНЕЙ СПИРАЛИ» ПРИ РЕШЕНИИ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ С БОЛЬШИМ КОЛИЧЕСТВОМ НЕИЗВЕСТНЫХ

Алекса С. Срданов[а], *Радиша* Р. Стефанович[аб],
Нада В. Раткович Ковачевич[в], *Александра* М. Йованович[а],
Драган М. Милованович[а]

[а] Высшая профессионально-техническая школа,
г. Пожаревац, Республика Сербия
[б] Университет обороны в г. Белград, Военная академия,
г. Белград, Республика Сербия
[в] Высшая профессионально-техническая школа,
г. Белград, Республика Сербия

ОБЛАСТЬ: математика, компьютерные науки
ВИД СТАТЬИ: профессиональная статья
ЯЗЫК СТАТЬИ: английский

Резюме:

Решение систем линейных уравнений n × n может представлять проблему для компьютера, особенно в тех случаях, когда требуется точное решение, и даже в тех случаях, когда количество уравнений и неизвестных относительно невелико (всего несколько тысяч). Все существующие методы сталкиваются с наименее одной из следующего ряда проблем: 1. сложность вычисления, выраженная количеством соответствующих операций, которые необходимо произвести для получения решения; 2. потенциальная возможность неограниченного роста значений результатов, что приводит к проблемам: overflow и underflow; 3. изменение значений некоторых коэффициентов в исходной системе, что приводит к неустойчивости решения; 4. дополнительные требования вследствие конвергенции; 5. в случаях большого количества уравнений и неизвестных необходимо, чтобы матрица системы была или не слишком наполнена, или была симметричной, либо диагональной, и т.д. В данной работе представлены методы решения системы линейных уравнений с произвольным количеством уравнений и неизвестных, на которых не отражаются перечисленные проблемы.

Ключевые слова: линейная система уравнений, метод «внешней спирали», гиперплоскость.

МЕТОДА „СПОЉАШЊЕ СПИРАЛЕ” ЗА РЕШАВАЊЕ ЛИНЕАРНОГ СИСТЕМА СА ВЕЛИКИМ БРОЈЕМ НЕПОЗНАТИХ

Алекса С. Срданов[а], *Радиша* Р. Стефановић[аб],
Нада В. Ратковић Ковачевић[в], *Александра* М. Јовановић[а],
Драган М. Миловановић[а]

[а] Висока техничка школа струковних студија,
Пожаревац, Република Србија
[б] Универзитет одбране у Београду, Војна академија,
Београд, Република Србија,
[а] Висока техничка школа струковних студија,
Београд, Република Србија

Srdanov, A. et al, The method of “external spiral” for solving large system of linear equations, pp.399-414

ОБЛАСТ: математика, рачунарске науке
ВРСТАЧЛАНКА: стручни чланак
ЈЕЗИКЧЛАНКА: енглески

Сажетак:

Решавање линеарног система једначина n × n може бити проблем и за рачунар, поготово ако је потребно тачно решење, чак и када је број једначина и непознат и релативно мали (пар хиљада). Све постојеће методе су оптерећене бар једним од следећих проблема: 1. сложеношћу рачунања израженим кроз број потребних операција које је потребно извршити како би се дошло до решења; 2. потенцијалном могућношћу неограниченог раста величина међу резултата, што узрокује проблеме прекорачења опсега (overflow) и недовољне осетљивости односно прецизности (underflow); 3. променом вредности неких коефицијената у полазном систему, што узрокује нестабилност решења; 4. додатним захтевима, због конвергенције; 5. случајевима великог броја једначина и непознатих који захтевају да матрица система буде: или слабо попуњена, или симетрична, или дијагонална, итд. У овом раду презентује се метода за решавање система линеарних једначина са произвољним бројем једначина и непознатих на коју се наведени проблеми не рефлектују.

Кључне речи: линеарни систем једначина, метод „спољашње спирале", хиперраван.

Paper received on / Дата получения работы / Датум пријема чланка: 23.07.2017.
Manuscript corrections submitted on / Дата получения исправленной версии работы / Датум достављања исправки рукописа: 08.01.2018.
Paper accepted for publishing on / Дата окончательного согласования работы / Датум коначног прихватања чланка за објављивање: 11.01.2018.