Some Geometrical Properties of a NxN System of Linear Equations

Dr. V. S. Patwardhan[1]

2 August 2022

Abstract

The NxN system of linear equations, $\mathbf{A_0x} = \mathbf{b_0}$, represents N intersecting hyperplanes, whose intersection is the solution to the system of equations. The solution is often obtained iteratively by minimising an appropriate quadratic form using well known optimisation techniques. These methods do not consider the geometrical nature of the intersecting hyperplanes explicitly. Investigation of the underlying geometry reveals some interesting properties and observations, which can be useful in developing a different iterative approach. This paper investigates some geometrical ideas which include (1) flatness, a newly defined parameter which characterises the local geometry, (2) flatness values along random directions and along eigenvectors for the largest and smallest eigenvalues, (3) the change in flatness values during steepest descent approach, (4) calculation of straight lines which pass in close vicinity of the solution, and (5) the effect of adding a new consistent equation to the system of equations, on the eigenvectors for the largest and smallest eigenvalues.

1. Background

Solving a system of linear algebraic equations is a classical problem which has many practical applications in areas such as engineering and science. The NxN system of linear equations can be written in a matrix form as $\mathbf{A_0x} = \mathbf{b_0}$, where $\mathbf{A_0}$ is a NxN matrix, $\mathbf{b_0}$ is an N-vector, and $\mathbf{x}$ is the solution vector to be determined. Each row of $\mathbf{A_0}$ is assumed to be normalized. Each row of $\mathbf{A_0}$ represents a hyperplane passing through the solution $\mathbf{s}$, and the system of equations represents N intersecting hyperplanes. There are many direct methods such as gaussian elimination and others which give an exact solution. However, it is often sufficient to get an approximate solution in practical applications. Iterative methods essentially start with a guess solution and improve it iteratively to get closer to the solution within acceptable accuracy. A quick summary of these direct as well as iterative methods can be found in standard books on linear algebra [for example, J. E. Gentle, 2007]. Iterative methods such as the steepest descent method and the conjugate gradient method are used when the matrix is symmetric and positive definite. (It is well known that $\mathbf{A_0x} = \mathbf{b_0}$ can be put in the form $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A} = \mathbf{A_0}^T\mathbf{A_0}$ and $\mathbf{b} = \mathbf{A_0}^T\mathbf{b_0}$. This gives a symmetric positive definite matrix $\mathbf{A}$ for any $\mathbf{A_0}$.) These are based on minimizing an appropriately defined quadratic function, using optimization techniques. Details of these methods, including convergence analysis, are available in standard books and reports [for example, J. R. Shewchuk, 1994]. Variations of the steepest descent method are available which use the momentum concept to achieve faster convergence [Y. Nesterov, 1983; I. Sutskever et. Al., 2013]. None of these methods consider geometrical aspects of the intersecting hyperplanes explicitly.

---

[1] Independent researcher. Formerly, Scientist G, National Chemical Laboratory, Pune 411008, India.
Email: vspatw@gmail.com , URL : https://www.vspatwardhan.com

It is well known that the steepest descent method leads to a fast approach to the solution (i.e., gives rapid reduction in residuals, or distance from solution) in the first few steps and slows down substantially in the following steps. There are two possible ways of bypassing this slow down [Patwardhan, 2022] and making the steepest descent method much faster, which include random movement of the point between iterations, and possible matrix transformations between iterations. It was shown that these approaches can increase the speed of convergence of the steepest descent method by several orders of magnitude.

In this paper, some interesting geometrical properties of the intersecting hyperplanes are investigated, which can be useful in developing fast algorithms based on the steepest descent method alone.

2. Geometrical properties of the intersecting hyperplanes

A hyperplane in N dimensions divides the N-dimensional space into two half spaces. If we consider all the N hyperplanes defined by the rows of the $\mathbf{A_0}$ matrix, they divide the N-dimensional space into 2^N parts, each of which can be termed as a hyper-orthant (referred to just as an orthant in this paper for convenience). Each orthant is a cone with its apex at the solution point $\mathbf{s}$ which satisfies $\mathbf{A_0 s} = \mathbf{b_0}$. In general, some orthants are sharp and pointed, while others are wider and somewhat flat. Let us consider a point P which is at a distance d_{soln} from the solution, and the corresponding vector $\mathbf{p}$ which is a vector of the coordinates of P. The closeness of P to the N hyperplanes defined by $\mathbf{A_0 x} = \mathbf{b_0}$, can be expressed in terms of $\sigma_{Res,0}$ (i.e., the root mean square of the residuals for the point P, defined as $\mathbf{r_0} = \mathbf{b_0} - \mathbf{A_0 p}$). The residuals $\mathbf{r_0}$ represent actual distances of P from the N hyperplanes since the rows of $\mathbf{A_0}$ are assumed to be normalized. It is obvious that if we move away from the solution along direction SP, then both d_{soln} and $\sigma_{Res,0}$ increase linearly, but their ratio remains unchanged. We define a "flatness index" for the point P as $\phi_0 = \sigma_{Res,0} / d_{soln}$ which is constant along the line SP. A low value of ϕ_0 implies that all hyperplanes are close to P in general, and a high value of ϕ_0 implies that all hyperplanes are far away from P in general. Let us consider two limiting cases. If all the hyperplanes are very close to P, then $\sigma_{Res,0}$ can be very low, giving $\phi_0 = 0$. On the other hand, if all the hyperplanes are almost orthogonal to the line SP, then the absolute value of every residual would be close to d_{soln}. In the extreme case, every residual would be equal to d_{soln}, giving $\sigma_{Res,0} = d_{soln}$, and $\phi_0 = 1$. Therefore, ϕ_0 satisfies $0 \le \phi_0 \le 1$. These are extreme values covering all instances of matrices in the N-dimensional space. For a given matrix, the range of ϕ_0 values would be a subset of this range.

We can define a similar flatness index $\phi = \sigma_{Res} / d_{soln}$ for the matrix $\mathbf{A}$ where σ_{Res} is the root mean square of the residuals for the point P, defined as $\mathbf{r} = \mathbf{b} - \mathbf{Ap}$. It is easy to show that $\mathbf{r} = \mathbf{A_0}^T \mathbf{r_0}$. The lower limit for ϕ is therefore 0. However, rows of $\mathbf{A}$ are not in a normalized form, and $\mathbf{r}$ does not represent the actual distance from rows of $\mathbf{A}$. Therefore, the upper limit on ϕ is not equal to 1.

It is interesting to see the range covered by ϕ values for a given matrix. Let S be the solution point, which gives $\mathbf{As} = \mathbf{b}$. Let $\mathbf{v}$ be a unit vector in the SP direction, so we can write $\mathbf{p} = \mathbf{s} + d_{soln}\mathbf{v}$. The residuals at P are given by $\mathbf{r} = \mathbf{b} - \mathbf{Ap} = - d_{soln}\mathbf{Av}$. The sum of squares of residuals at P can be written as $SS_{res} = \mathbf{r}^T\mathbf{r} = (d_{soln})^2(\mathbf{Av})^T\mathbf{Av}$. Let ($\mathbf{v}_i$, i = 1 to N) be the unit length eigenvectors of matrix $\mathbf{A}$, and (λ_i, i = 1 to N) be the corresponding eigenvalues. Since A is symmetric and positive definite, all the λ_i, values are positive, and all the eigenvectors $\mathbf{v}_i$ are mutually orthogonal. The vector $\mathbf{v}$ therefore can be written as

$$\mathbf{v} = \sum_{i=1}^{N} \alpha_i \mathbf{v}_i \tag{1}$$

where α_i is the length of the component of **v** in the $\mathbf{v}_i$ direction. Since all the **v**'s are unit vectors, we have $\sum_i \alpha_i{}^2 = 1$. **Av** can now be written as

$$\mathbf{Av} = \sum_{i=1}^{N} \alpha_i \mathbf{A}\mathbf{v}_i = \sum_{i=1}^{N} \alpha_i \lambda_i \mathbf{v}_i \tag{2}$$

As mentioned above, SS_{Res} can be written as

$$SS_{Res} = d_{soln}^2 (\mathbf{Av}).(\mathbf{Av}) = d_{soln}^2 \sum_{i=1}^{N} \alpha_i^2 \lambda_i^2 \tag{3}$$

and the flatness index ϕ can be written as

$$\varphi = \left[\left(\frac{1}{N}\right) \sum_{i=1}^{N} \alpha_i^2 \lambda_i^2\right]^{0.5} \tag{4}$$

The maximum value of the summation in equation (4) is $\lambda_{max}{}^2$, since all $\alpha_i{}^2$ values are positive, and add up to 1. Therefore

$$\varphi_{max} = \lambda_{max}/N^{0.5} \tag{5}$$

This ϕ value corresponds to the case where **v** coincides with the eigenvector corresponding to the largest eigenvalue λ_{max}. Using a similar argument, we get

$$\varphi_{min} = \lambda_{min}/N^{0.5} \tag{6}$$

This is the ϕ value for the case where **v** coincides with the eigenvector corresponding to the smallest eigenvalue λ_{min}. The ratio ϕ_{max}/ϕ_{min} is equal to the condition number of **A**.

3. Illustrative example in two dimensions

Let us illustrate the ideas presented above with an example in two dimensions. Let us consider the matrix $\begin{bmatrix} 3 & 2 \\ 1 & 8 \end{bmatrix}$, which, after normalization, gives $\mathbf{A_0} = \begin{bmatrix} 0.832 & 0.555 \\ 0.124 & 0.992 \end{bmatrix}$ and $\mathbf{A} = \begin{bmatrix} 0.708 & 0.585 \\ 0.585 & 1.292 \end{bmatrix}$. Let $\mathbf{b_0}$ and **b** be equal to zero, so that the solution **s** is at the origin. Calculations give $\lambda_{max} = 1.654$, $\lambda_{min} = 0.346$, $\phi_{max} = 1.169$, and $\phi_{min} = 0.245$.

Figure 1 below shows the calculated results graphically. The blue ellipse shows the contour for one specific value of SS_{Res} . The two grey lines, i.e., lines A1 and A2, show the equations corresponding to the two rows of **A**. The dashed and dotted orange lines show eigenvector directions corresponding to the maximum and minimum eigenvalues respectively, and are seen to be orthogonal, as expected. They also coincide with the axes of the contour ellipse. The range of ϕ values for this matrix is [0.245, 1.169]. The dashed-dotted black lines are just randomly selected directions for illustration. The lines A1 and A2 divide the area into four parts. Line 1 lies in the narrow orthant between lines A1 and A2. It has a flatness index value of ϕ = 0.28. Line 2, on the other hand, lies in the wider orthant defined by lines A1 and A2, and has a higher flatness index value of ϕ = 1.10. These ϕ values lie within the calculated range of ϕ values, i.e., $0.245 \le \phi \le 1.169$. The lower flatness index of 0.28 for Line 1 corresponds to a sharper orthant. The highest ϕ value occurs along the eigenvector corresponding to the highest eigenvalue, and the lowest ϕ value occurs along the eigenvector corresponding to the lowest eigenvalue. It may be noted that the direction along which $\phi = \phi_{max}$ corresponds to the shortest axis of the ellipse and is the direction along which SS_{Res} increases most rapidly.

4. Steepest descent method and the variation of ϕ

The steepest descent method reduces SS_{Res} at each step. It has been shown earlier [Patwardhan, 2022] that the reduction is large in the initial few steps, and then it tapers off drastically. It is interesting to see how ϕ changes as the steepest descent method progresses. For this purpose, random problems of various sizes were generated. The generation of such problems has been described earlier in detail, and involves generation of $\mathbf{A_0}$, $\mathbf{b_0}$, **s,** and a starting point $\mathbf{x_0}$ [Patwardhan, 2022]. Details of the steepest descent method are not repeated here, since they are very well known [for example, J. R. Shewchuk, 1994]. After generating the problem, steepest descent steps are taken starting at $\mathbf{x_0}$ and the flatness index ϕ is calculated at each step.

Figure 2 shows the variation of ϕ at successive steps, for N = 10, for ten different starting points $\mathbf{x_0}$. The largest and smallest eigenvalues, λ_{max} and λ_{min}, were calculated using power iteration with **A** and $\mathbf{A_{inv}}$ (i.e., the inverse of **A**). The **A** matrix had $\phi_{max} = 1.093$ and $\phi_{min} = 2.82 \times 10^{-5}$. Figure 2 shows that the ϕ values at the beginning (i.e., at different starting points $\mathbf{x_0}$) were between 0.3 and 0.85, and they all reduced to less than 0.05 in ten steps. The flatness index ϕ reduced rapidly in the first one or two steps, and then it reduced further only slowly. This is similar to the behaviour of SS_{Res} found earlier.

Figure 3 shows the variation of ϕ at successive steps, for N = 1000, for ten different starting points $\mathbf{x_0}$. The **A** matrix had $\phi_{max} = 0.128$ and $\phi_{min} = 3.61 \times 10^{-8}$. Figure 3 shows that the ϕ values at the beginning (i.e., at different starting points) were between 0.04 and 0.045, and they all reduced to less than 0.002 in ten steps. The flatness index ϕ reduced rapidly in the first one or two steps, and then it reduced further slowly. This is similar to the behaviour of SS_{res} found earlier. The calculated results for N = 10 and 1000 show that the ϕ values are lower, and the variation in the graphs for different $\mathbf{x_0}$ vectors is much smaller for the larger value of N = 1000.

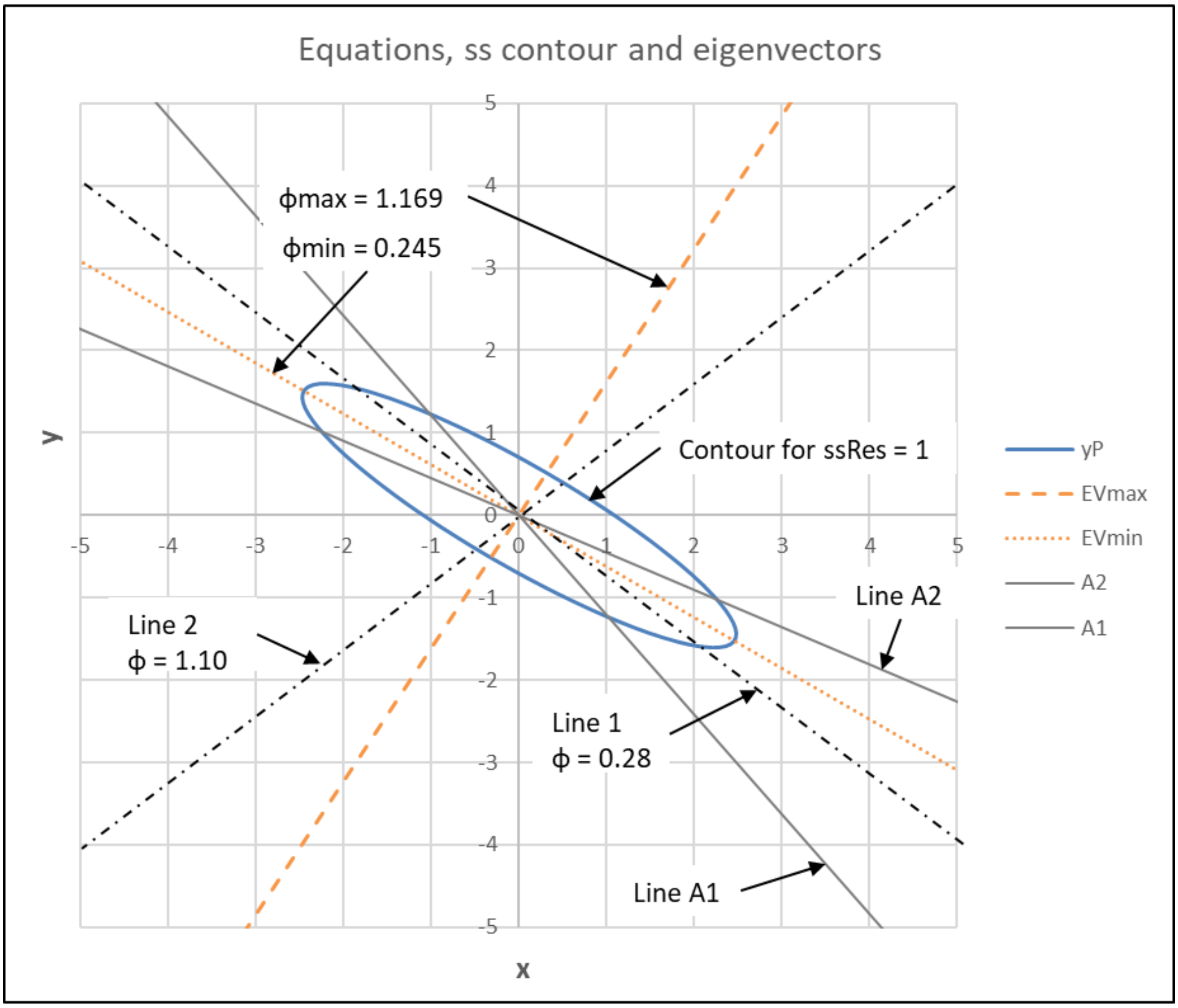

Figure 1. Results obtained for the illustrative example, with values of flatness index

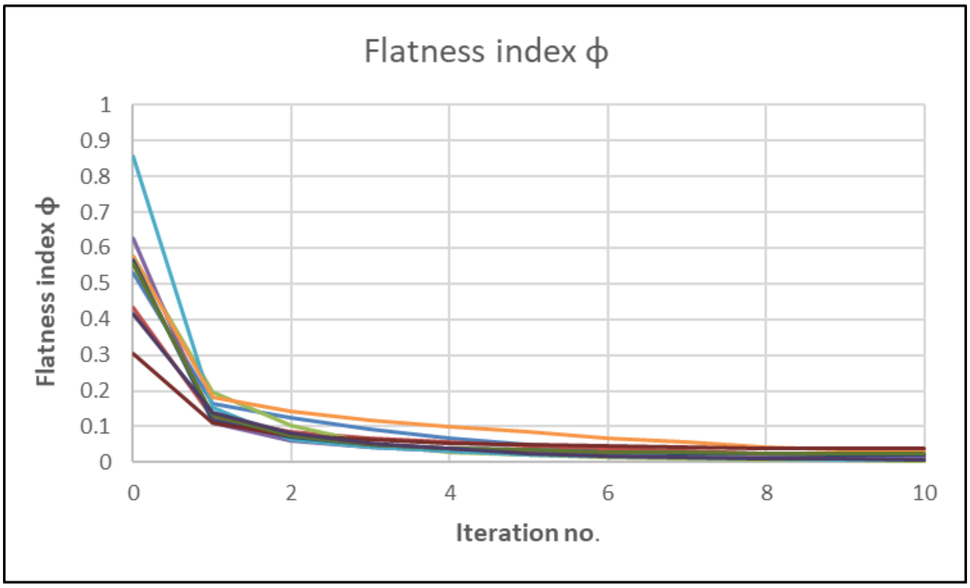

Figure 2. Variation of ϕ with steepest descent method for ten different starting points, for N = 10

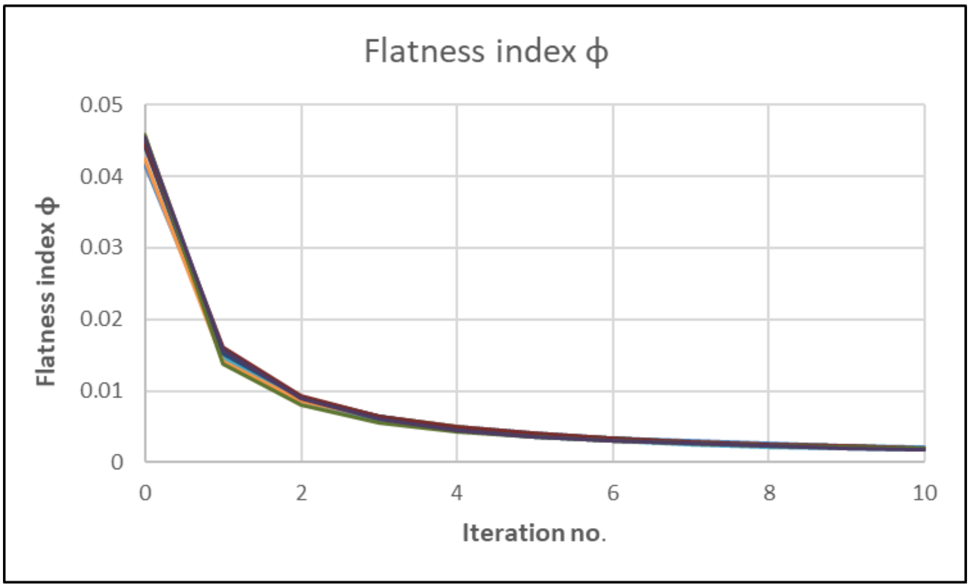

Figure 3. Variation of ϕ with SD steps for ten different starting points, for N = 1000

Figures 2 and 3 show that each steepest descent step gives a reduction not only in SS_{res} , but in the flatness index ϕ as well. The numerical results which go into Figures 2 and 3 clearly indicate this although they are not presented here, for the sake of brevity. This implies that steepest descent steps may ultimately give $\phi = \phi_{min}$, though it may take a very large number of steps. Thus, steepest descent method would eventually approach the eigenvector corresponding to the smallest eigenvalue and would reach the orthant containing the corresponding eigenvector. Of course, it may not approach it directly, but may follow a tortuous path around this orthant. The details are not shown since they are not relevant here.

5. Lines passing through the solution point

Let us consider a two-dimensional problem for the sake of simplicity. Figure 4 shows the example considered earlier, with some additions. Lines A1 and A2, and the orange lines are the same as in Figure 1. Figure 4 shows an arbitrary line L1 which, along with lines A1 and A2, defines a triangle, which is merely a simplex in two dimensions. The incircle of this simplex is shown as a circle with its center at C1. Line L2 has been drawn such that it is parallel to line L1 and is farther way from the origin. Lines L2, A1, and A2 define another simplex, and its incircle is the circle shown with center at C2. It is trivial to show, using simple geometry, that line C1-C2 passes through the solution (the origin, in this case) which is the intersection of lines A1 and A2.

The observations made with Figure 4 can be generalized to N dimensions. It has been mentioned earlier that the N hyperplanes defined by the rows of the $\mathbf{A_0}$ matrix in the system of equations $\mathbf{A_0x} = \mathbf{b_0}$ (or the $\mathbf{A}$ matrix in $\mathbf{Ax} = \mathbf{b}$) pass through the solution point $\mathbf{s}$ and make 2^N orthants. If we take any single orthant out of these, and generate two simplexes with two parallel hyperplanes placed at different distances from the solution, then the line joining their in-centers would pass through $\mathbf{s}$. This will happen not only with the in-centers, but with any other uniquely defined centers of these simplexes. Many such centers have been used earlier, such as center of mass, the centroid, the center of the smallest sphere which includes the simplex, the analytical center, the weighted projection center, the BI center, and the harmonic center [Moretti, 2003; Patwardhan, 2020]. The calculation of these centers, however, involves different degrees of computational effort. For example, the exact calculation of the in-center of a simplex requires $O(N^3)$ computations. Since calculation of the solution of **Ax** = **b** with direct methods also involves $O(N^3)$ computations, calculating two in-centers and joining them to get the solution, gives no computational advantage. However, even if the two centers are calculated only approximately, they would give a line which passes in close vicinity of the solution. Such a line can be useful in developing a new iterative procedure for solving **Ax** = **b**. Calculation of such a line using the steepest descent method is described below.

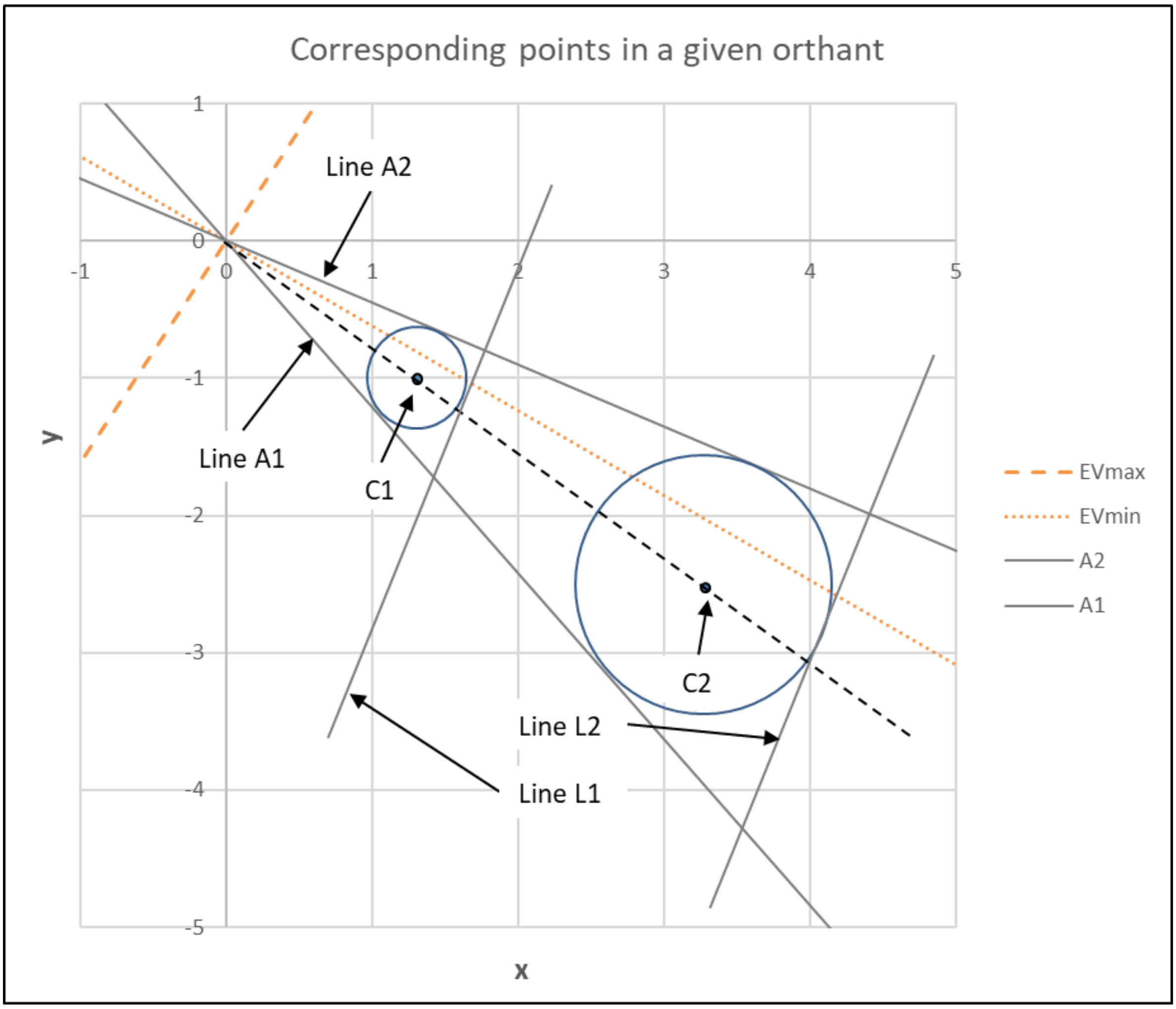

Figure 4. A line connecting centers of two distinct in-circles in an orthant

6. Approximate calculation of a line passing close to the solution point using steepest descent

The idea stated above can be extended further. To get a line passing close to the solution, it is not essential to construct two simplexes and calculate their approximate centers. It would be enough to get two "corresponding points" at different distances from the center (even if they lie in different orthants). It has been mentioned earlier that steepest descent method gives a continuous reduction in SS_{Res} as well as the flatness index ϕ. The following algorithm uses the steepest descent method, starting from two different points, to get two approximate "corresponding points" which give a line passing in close vicinity of the solution.

Let P1 be a point selected at random. Let Q1 be the point obtained by applying a given number of steepest descent steps to P1. Let P2 be a point taken in a random direction from Q1 at a selected distance. Let Q2 be the point obtained by applying the same number of steepest descent steps to P2. It is interesting to see if the line Q2-Q1 drawn from Q2 passes through the vicinity of the solution. The detailed algorithm is given below:

Algorithm 1: Given: N, $\mathbf{A_0}$, $\mathbf{b_0}$, and **s**

1 Make $\mathbf{A}$ (= $\mathbf{A_0}^T\mathbf{A_0}$) and $\mathbf{b}$ (= $\mathbf{A_0}^T\mathbf{b_0}$)
2 Get a random vector $\mathbf{v_1}$ with elements N(0,1)
3 Normalize $\mathbf{v_1}$
4 Set P1 = the point whose coordinates are given by the elements of $\mathbf{v_1}$
5 Get Q1 by applying 20 steepest descent steps to P1, using **A** and **b**
6 d = distance P1-Q1
7 Get another random vector $\mathbf{v_2}$ with elements N(0,1)
8 Normalize $\mathbf{v_2}$
9 Set P2 = Q1 + 10 d $\mathbf{v_2}$
10 Get Q2 by applying 20 steepest descent steps to P2, using **A** and **b**
11 Get θ_1, the acute angle between lines Q2-Q1 and Q2-S

The steps 2-4 ensure that P1 is not too close to the solution. This is so, because the randomly generated $\mathbf{v_1}$ would be nearly orthogonal to the line P1-S for large N. Thus, P1 would be at least unit distance from the solution.

The results obtained with Algorithm 1 are shown in Figure 5. This figure shows results obtained for N = 10 to 1000, and each point is an average of ten starting points P1. Figure 5 shows that θ_1, the acute angle between lines Q2-Q1 and Q2-S is just a few degrees. So, direction Q2-Q1 can be used as a good approximation for the direction Q2-S. It was also observed in all these cases that the angle between the line Q2-Q1 and the eigenvector corresponding to the largest eigenvalue was greater than 89 degrees, which implies that these two directions are almost orthogonal.

It is interesting to see the angles made by the line Q2-Q1 with the hyperplanes corresponding to the N linear equations. Let $\theta_{A0,i}$ and $\theta_{A,i}$ be the acute angles made by the line Q2-Q1 with the i^{th} row of $\mathbf{A_0}$ and **A** respectively. The angles of course, are different for different rows. The range covered by these values over i = (1, N) is shown in Table 1 below. The table also shows similar angles made by a randomly selected direction (with N(0,1) elements) with rows of matrices $\mathbf{A_0}$ and **A**. It is well known that any two randomly generated directions in N dimensions are almost orthogonal for large N. However, we can make some interesting observations from Table 1: (1) the range covered by the

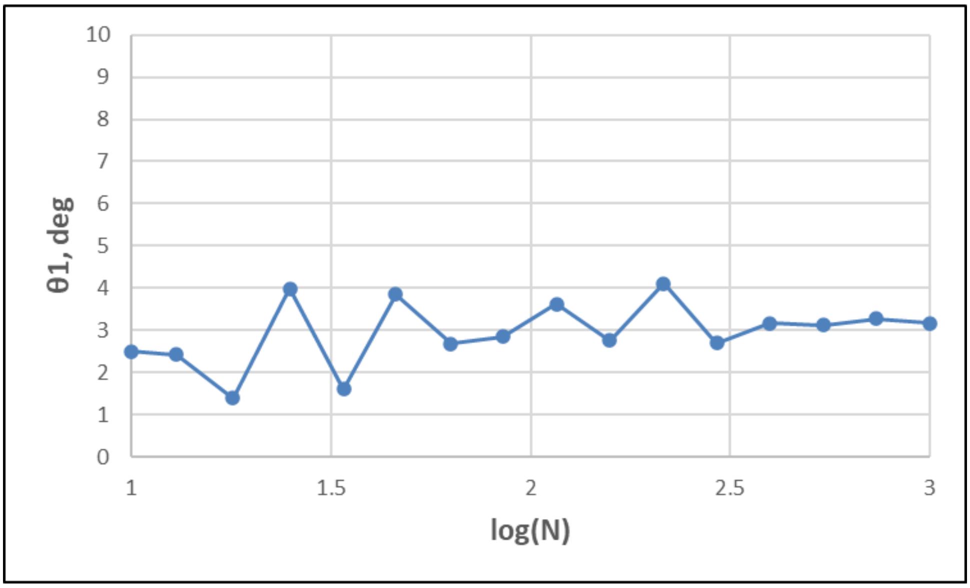

Figure 5. The acute angle between lines Q2-Q1 and Q2-S (N = 10 and 1000)

N	10		1000	
Direction	*Q2-Q1*	*Random direction*	*Q2-Q1*	*Random direction*
$\theta_{Ao,min}$	85.3	59.9	89.0	83.5
$\theta_{Ao,avg}$	87.9	74.5	89.8	88.6
$\theta_{Ao,max}$	89.7	86.5	90.0	90.0
$\theta_{A,min}$	88.4	65.4	89.7	83.8
$\theta_{A,avg}$	89.4	77.8	89.9	88.5
$\theta_{A,max}$	89.9	89.1	90.0	90.0

Table 1. Angles (degrees) made by Q2-Q1 direction and random directions with $\mathbf{A_0}$ and **A** rows

angles between a randomly generated direction and the N hyperplanes of $\mathbf{A_0}$ is 59.9-86.5 degrees, while that for **A** is 65.4-89.1 degrees for N=10. The higher values for **A** indicate that the hyperplanes of **A** rows have a sharper configuration than those of $\mathbf{A_0}$. A similar observation holds for N = 1000 as well. (2) The angles for N = 1000 are higher than those for N = 10. (3) The angles made by the Q2-Q1 direction are much larger than those made by a random direction for N = 10. This indicates the favourable effect of the steepest descent method in aligning the Q2 and Q1 points so that the Q2-Q1 line makes larger angles with the **A** as well as $\mathbf{A_0}$ rows. This observation also holds true for N = 1000. The good alignment of Q2 and Q1 is also obvious from Figure 5, which shows that the acute angle between lines Q2-Q1 and Q2-S is just a few degrees.

7. Effect of adding a new, consistent equation to the $\mathbf{A_0x} = \mathbf{b_0}$ system, on eigenvalues and eigenvectors of A

If a new equation is added to the system of N x N equations (i.e., $\mathbf{A_0x} = \mathbf{b_0}$), the system size becomes (N+1)xN. Let the new equation be $\mathbf{v}^T\mathbf{x} = w$. Let us assume that **v** is normalized, and that the solution **s** satisfies this equation, so that the (N+1)xN system is consistent and has the same unique solution **s**. We can make a new NxN system $\mathbf{A_1x} = \mathbf{b_1}$ (where $\mathbf{A_1}$ is symmetric and positive definite) having the same solution **s** by defining

$$\mathbf{A_1} = (\mathbf{A_0}^T\mathbf{A_0} + \mathbf{vv}^T), \text{ and } \mathbf{b_1} = \mathbf{A_0}^T\mathbf{b_0} + w\mathbf{v} \tag{7}$$

This new $\mathbf{A_1}$ matrix will have eigenvalues and eigenvectors which will, in general, be different from those of **A**. We can extend this procedure by adding the equation $\mathbf{v}^T\mathbf{x} = w$ not once, but k times to the original system of equations (resulting in a (N+k)xN system), to get $\mathbf{A_kx} = \mathbf{b_k}$ where

$$\mathbf{A_k} = (\mathbf{A_0}^T\mathbf{A_0} + k\mathbf{vv}^T) \text{ and } \mathbf{b_k} = \mathbf{A_0}^T\mathbf{b_0} + kw\mathbf{v} \tag{8}$$

It is interesting to see what happens to eigenvalues and eigenvectors as k increases. The eigenvalues of $\mathbf{A_k}$ are defined by the usual equation $\mathbf{A_kx} = \lambda\mathbf{x}$. Substituting for $\mathbf{A_k}$ from equation (8), and using the condition k >> 1, it is trivial to show that the equation $\mathbf{A_kx} = \lambda\mathbf{x}$ Is satisfied by $\lambda = k$ and $\mathbf{x} = \mathbf{v}$. In other words, any arbitrary unit vector **v** can be used to transform the system $\mathbf{A_0x} = \mathbf{b_0}$ to another system $\mathbf{A_kx} = \mathbf{b_k}$ using equation (8) and a large enough k, such that (i) **v** is the eigenvector of $\mathbf{A_k}$ corresponding to the largest eigenvalue (which is equal to k, for large k), and (ii) both the systems have the same solution **s**. From a geometrical viewpoint, the SS_{res} contours change orientation as k increases, while the solution **s** remains unchanged.

Let us now see the effect of k on the eigenvalues and eigenvectors of $\mathbf{A_k}$. Computations were carried out for N = 10 and for N = 1000 by generating random problems as described before. The largest and smallest eigenvalues, λ_{max} and λ_{min}, were calculated using power iteration with $\mathbf{A_k}$ and $\mathbf{A_{k,inv}}$ (i.e., the inverse of $\mathbf{A_k}$). The corresponding eigenvectors, $\mathbf{e}_{max}$ and $\mathbf{e}_{min}$, were also calculated. The direction Q2-Q1 was calculated as described above and was used as **v**. Computations were carried out for various values of k, to calculate $\lambda_{max,k}$, $\lambda_{min,k}$, $\mathbf{e}_{max,k}$ and $\mathbf{e}_{min,k}$. Table 2 shows the acute angles θ (degrees) between different vectors based on these calculations.

Variation in λ_{max} : It is seen from Table 2 that the maximum eigenvalue, λ_{max} , starts with 3.45 for the **A** matrix, and is almost equal to k for k >= 4 or so for N = 10. A similar observation can be made for N = 1000 as well.

Variation in $\mathbf{e}_{max,k}$: For k = 0, the eigenvector corresponding to the largest eigenvalue, $\mathbf{e}_{max,k}$, is almost orthogonal to **v**. However, it changes direction as k increases, and for k = 8, it almost coincides with **v**. This is consistent with the analysis given above. $\Theta(\mathbf{e}_{max,k}, \mathbf{e}_{max})$ goes from 0 to almost 90, indicating that $\mathbf{e}_{max,k}$ rotates almost through 90 degrees as k increases. Similar observations can be made for N = 1000 as well.

Variation in $\mathbf{e}_{min,k}$: It is seen from Table 2 that for N = 10, $\mathbf{e}_{min,k}$ was at 29 degrees with **v** for k = 0, and became almost orthogonal to **v** for large k. It also shows that $\mathbf{e}_{min,k}$ rotated through 62 degrees or so as k changed from 0 to 8. Similar observations can be made for N = 1000 as well.

N	k	λ_{max}	Θ ($\mathbf{e}_{max,k}$,$\mathbf{v}$)	Θ ($\mathbf{e}_{max,k}$,$\mathbf{e}_{max}$)	Θ ($\mathbf{e}_{min,k}$.$\mathbf{v}$)	Θ ($\mathbf{e}_{min,k}$, $\mathbf{e}_{min}$)
10	0	3.45	89.3	0.0	29.0	0.0
	1	3.46	89.1	0.0	88.7	61.4
	2	3.46	88.4	0.9	89.4	62.1
	3	3.46	85.7	3.7	89.6	62.3
	4	4.02	3.3	86.1	89.7	62.4
	5	5.02	1.3	88.1	89.7	62.5
	6	6.02	0.9	88.5	89.8	62.5
	7	7.02	0.0	88.7	89.8	62.5
	8	8.02	0.0	88.8	89.8	62.6
1000	0	4.05	89.4	0.0	80.3	0.0
	1	4.05	89.2	0.0	90.0	72.3
	2	4.05	88.8	0.0	90.0	72.3
	3	4.05	87.7	1.8	90.0	72.3
	4	4.08	58.6	31.3	90.0	72.3
	5	5.03	2.3	87.3	90.0	72.3
	6	6.03	1.2	88.3	90.0	72.3
	7	7.03	0.9	88.7	90.0	72.3
	8	8.03	0.0	88.9	90.0	72.3

Table 2. Variation of λ_{max} and eigenvector directions with k, for N = 10 and 1000 (θ in degrees)

Table 2 also shows that the vector, $\mathbf{e}_{max,k}$ changes very rapidly in the range k = 3 to 5, and changes very slowly outside this range, for N = 10 as well as N = 1000. On the other hand, the vector, $\mathbf{e}_{min,k}$ changes rapidly at much lower values of k, and reaches the asymptotic value at k = 2 itself, both for N = 10 and N = 1000. Figures 6, 7 and 8 show the variation of these quantities graphically.

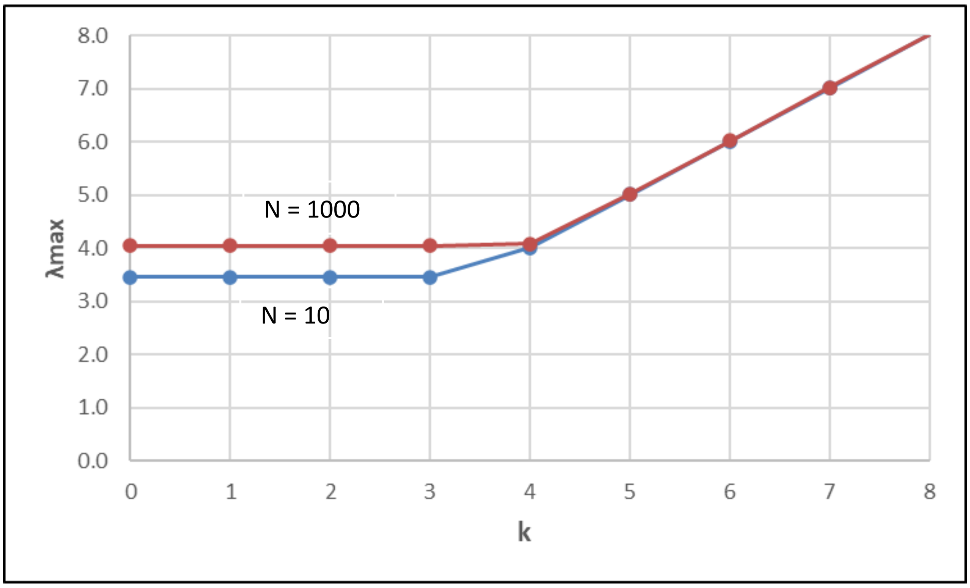

Figure 6. Variation of λ_{max} with k for N = 10 and 1000

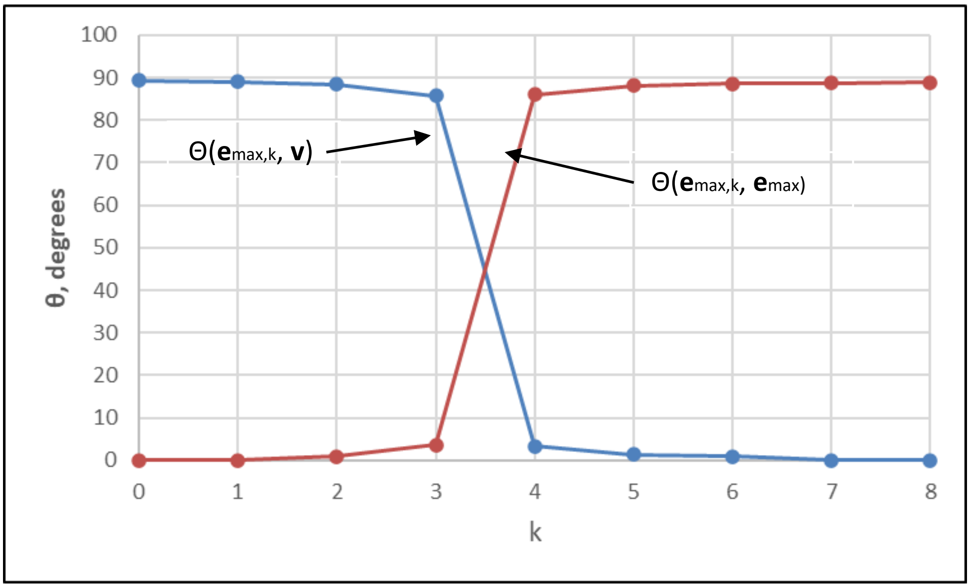

Figure 7. Variation of $\Theta(\mathbf{e}_{max,k}, \mathbf{v})$ and $\Theta(\mathbf{e}_{max,k}, \mathbf{e}_{max})$ for N = 10

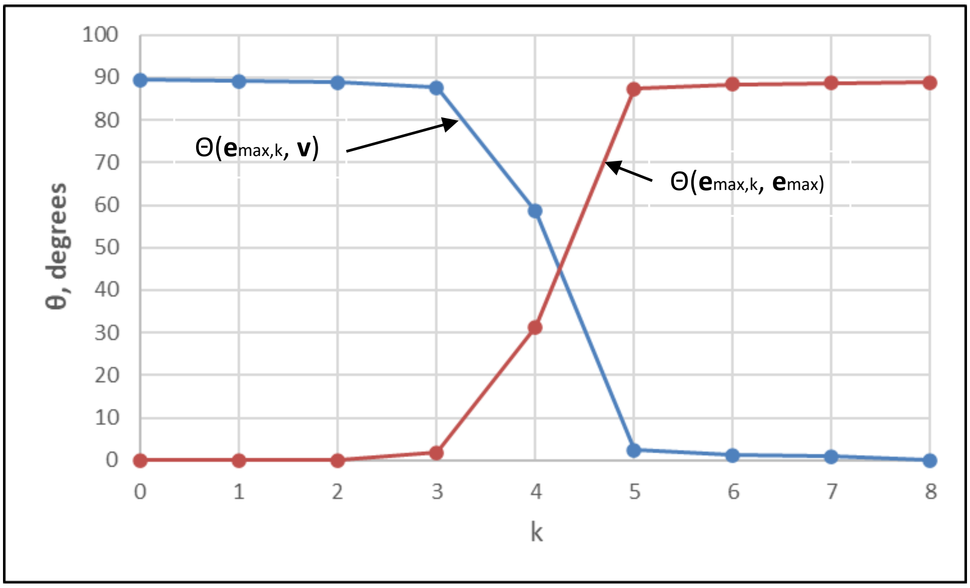

Figure 8. Variation of $\Theta(\mathbf{e}_{max,k}, \mathbf{v})$ and $\Theta(\mathbf{e}_{max,k}, \mathbf{e}_{max})$ for N = 1000

8. The rotation of eigenvectors with increasing k

The rotation of eigenvectors in N dimensions is difficult to visualize. However, rotations can be visualized conveniently in two dimensions. Figure 9 shows graphically how eigenvector directions change due to repeated addition of a row to the matrix $\mathbf{A_0}$. The values used are: $\mathbf{A_0} = \begin{bmatrix} 3 & 4 \\ 1 & 10 \end{bmatrix}$, and $\mathbf{v} = \begin{bmatrix} 1 \\ -.8 \end{bmatrix}$. The figure shows a blue ellipse and lines for the **A** matrix. The red and green ellipses and lines correspond to k = 2 and 4 respectively. The **v** direction is shown in black. As k increases, the dashed lines (which are eigenvectors corresponding to the largest eigenvalue) approach the line **v**.

The dotted lines, which are eigenvectors corresponding to the smallest eigenvalue, are orthogonal to the dashed lines, as expected. (The three ellipses correspond to different values of SS_{Res} which were selected for getting ellipses of suitable sizes for visualization, and their axes are the eigenvector directions. A change in the SS_{Res} value does not change the orientation of the contour ellipses.)

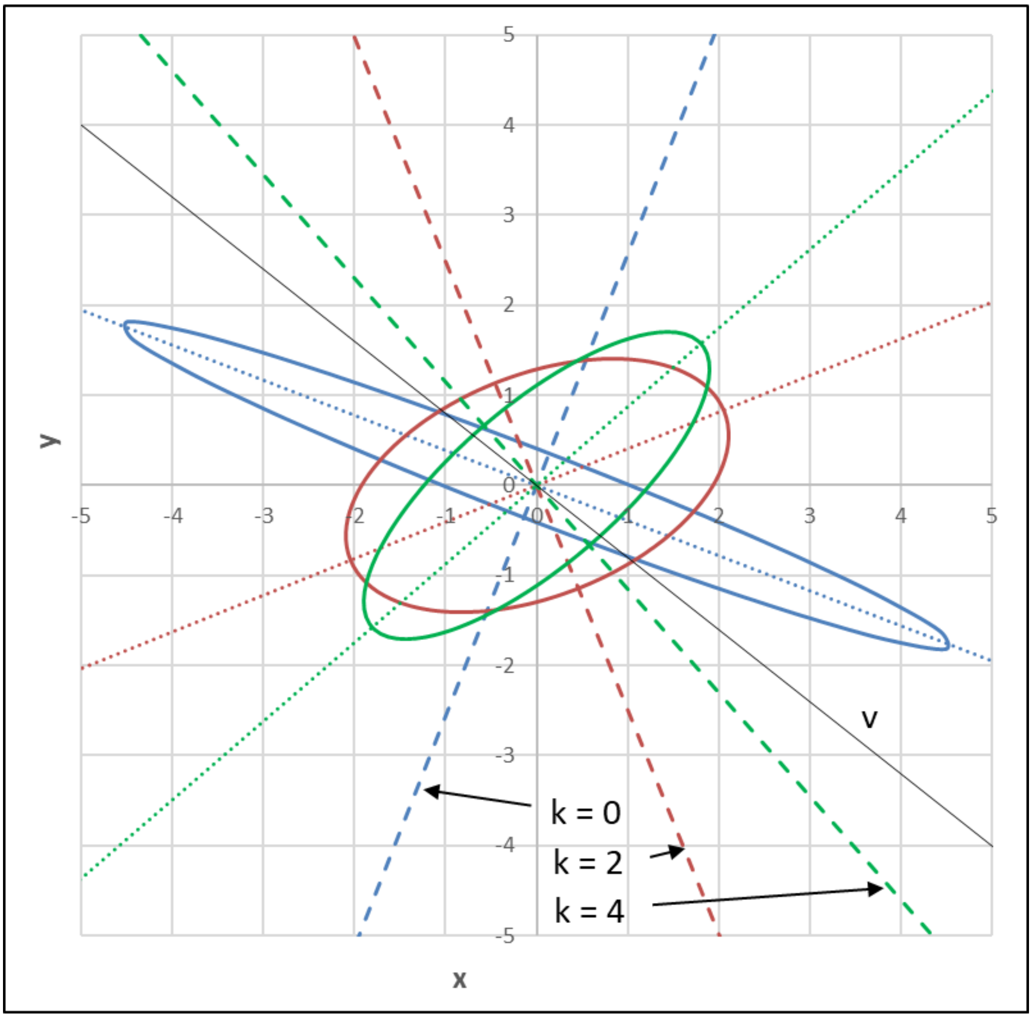

Figure 9. Variation of ss_{Res} contours and eigenvector directions with k, for N = 2

9. Conclusions

The system of linear equations, $\mathbf{A_0x} = \mathbf{b_0}$, represents N intersecting hyperplanes. Some geometrical properties of such a system (assuming the rows of $\mathbf{A_0}$ are normalized) have been investigated in this paper. The conclusions are listed below.

1. A new parameter, termed here as flatness index ϕ_0, has been defined, which characterises the local geometry of the intersecting hyperplanes of $\mathbf{A_0}$ at any point. It has been shown to satisfy the inequality $0 \le \phi_0 \le 1$. For the derived matrix $\mathbf{A}$ (= $\mathbf{A_0}^T\mathbf{A_0}$), the flatness index ϕ satisfies $0 \le \phi$, but the upper limit may be greater than 1. This is so because the residuals $\mathbf{r_0}$ (= $\mathbf{b_0} - \mathbf{A_0x}$) represent actual distances, while the residuals $\mathbf{r}$ (= $\mathbf{b} - \mathbf{Ax}$) do not.

2. Equations (5) and (6) relate the extreme values of ϕ to maximum and minimum eigenvalues.

3. The flatness index ϕ reduces rapidly in the first one or two steps of SD, and then reduces further slowly. Using this property, lines passing through the neighbourhood of the solution can be obtained, which can be useful in iterative calculations. These lines also happen to be almost orthogonal to the eigenvector of **A** corresponding to the largest eigenvalue.

4. It has been shown that if a normalized equation $\mathbf{v}^T\mathbf{x}$ = w, (which passes through the solution of $\mathbf{A_0 x}$ = $\mathbf{b_0}$, i.e., satisfies $\mathbf{v}^T\mathbf{s}$ = w) is repeatedly added k times to the original system of equations, then it becomes an eigenvector direction of $\mathbf{A_k}$, corresponding to the largest eigenvalue λ_{max} of $\mathbf{A_k}$ for k > 5 or so. Moreover, λ_{max} = k for k > 5 or so. The eigenvector direction of $\mathbf{A_k}$, corresponding to the smallest eigenvalue λ_{min} of $\mathbf{A_k}$ also changes significantly with k.

10. References

[1] Gentle, J. E.,
Matrix Algebra - Theory, Computations, and Applications in Statistics
Pub. Springer, New York, USA (2007)

[2] Nesterov, Y.,
A Method of Solving a Convex Programming Problem with Convergence Rate O(1/k^2)
Soviet Mathematics Doklady, **27**, 372-376 (1983)

[3] Sutskever I., Martens J., Dahl G., and Hinton G.,
On the Importance of Initialization and Momentum in Deep Learning
Proceedings of the 30th International Conference on Machine Learning,
PMLR **28(3)**, 1139-1147 (2013)

[4] Patwardhan, V.,
Solution of a NxN System of Linear algebraic Equations: 1 -- The Steepest Descent Method Revisited
https://doi.org/10.48550/arXiv.2206.07482 (2022)

[5] A. C. Moretti,
A weighted projection centering method,
Computational and Applied Mathematics, **22** (2003), pp 19–36

[6] Patwardhan V. S.,
Calculation of the BI Center of a Closed Convex Polytope,
DOI: 10.13140/RG.2.2.32331.95523 (2020)
(Can also be downloaded from www.vspatwardhan.com)

[7] Patwardhan V. S.,
Harmonic Center of a Closed Convex Polytope: Definition, Calculation and Some Properties
arXiv:2004.01447 [math.MG] (2020)