Finite Element Method and Computational Structural Dynamics

Prof. Manish Shrikhande
Department of Earthquake Engineering
Indian Institute of Technology, Roorkee

Lecture - 49
The Algebraic Eigenvalue Problem - III

Hello friends. So, we have seen in our last lecture some of the basic tools that are used for computation of algebraic eigenvalue problem, large scale algebraic eigenvalue problem, for small scale the definition is good enough determinant of characteristic polynomial, roots of the characteristic polynomial can give the eigenvalues, but that definition of eigenvalues is not really a sound numerical algorithm when we are dealing with large scale eigenvalue problems.

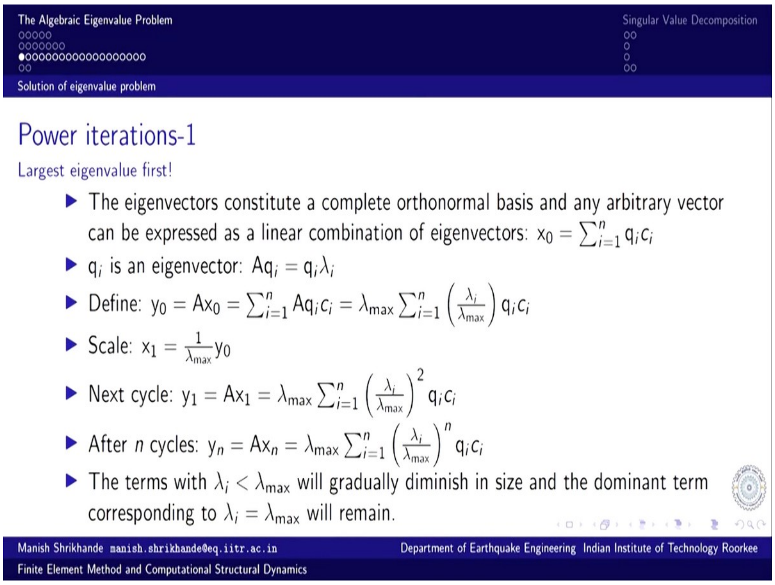
And for large scale eigenvalue problems we need to look at more efficient data structures and more efficient numerical and stable numerical algorithms and we saw some of the basic tools of the trade starting from similarity transform, orthogonal similarity transform.

So, they provide us the basic template for developings eigenvalue solvers by series of orthogonal similarity transform, the matrix can be nudged towards a diagonal matrix or to desired form from which eigenvalues and eigenvectors can be extracted and suitable matrices suitable orthogonal transform matrices, they are we saw two examples of that givens rotations that will target one element one of diagonal element at a time.

And gradually shift the complete spectral mass towards the diagonal of the matrix or a second one is of course, householder transformation where we look at all, but the first element of the vector to be annihilated and that results in n mass zeros enforced on the matrix structure and that can lead to substantial savings in the computations and leads to very efficient algorithms for eigenvalue solution.

So, let us now start with proper algorithms for eigenvalue solution and we start with our what we started with the power iterations. So, we discussed all the eigenvectors constitute orthonormal basis and therefore, any arbitrary vector can be expressed as a linear combination of eigenvectors.

(Refer Slide Time: 03:01)



The Algebraic Eigenvalue Problem

Singular Value Decomposition

Solution of eigenvalue problem

Power iterations-1

Largest eigenvalue first!

- The eigenvectors constitute a complete orthonormal basis and any arbitrary vector can be expressed as a linear combination of eigenvectors: $x_0 = \sum_{i=1}^{n} q_i c_i$
- q_i is an eigenvector: $Aq_i = q_i \lambda_i$
- Define: $y_0 = Ax_0 = \sum_{i=1}^{n} Aq_i c_i = \lambda_{max} \sum_{i=1}^{n} \left(\frac{\lambda_i}{\lambda_{max}}\right) q_i c_i$
- Scale: $x_1 = \frac{1}{\lambda_{max}} y_0$
- Next cycle: $y_1 = Ax_1 = \lambda_{max} \sum_{i=1}^{n} \left(\frac{\lambda_i}{\lambda_{max}}\right)^2 q_i c_i$
- After n cycles: $y_n = Ax_n = \lambda_{max} \sum_{i=1}^{n} \left(\frac{\lambda_i}{\lambda_{max}}\right)^n q_i c_i$
- The terms with $\lambda_i < \lambda_{max}$ will gradually diminish in size and the dominant term corresponding to $\lambda_i = \lambda_{max}$ will remain.

Manish Shrikhande manish.shrikhande@eq.iitr.ac.in Department of Earthquake Engineering Indian Institute of Technology Roorkee

Finite Element Method and Computational Structural Dynamics

So, any trial vector x_0 can be written as a linear combination of eigenvectors let us say q_i. So, $q_1 c_1 + q_2 c_2 + q_3 c_3$ etc. These are the generalized coordinates which need to be determined for a given q for a given x_0 and for given set of a orthogonal vectors $q_1 q_2 q_3$ etc.. So, q_i is an eigenvector, $A q_i = q_i \lambda_i$.

So, ith eigen pair and we can define y_0, a vector as $A x_0$ and this can be expressed if I represent a x_0 as a summation of eigenvectors $q_i c_i$. Then this substitution results in A x_0 = $\sum_i q_i c_i$ and if I take out hypothetical what is the largest eigenvalue that is common factor, if I take that common factor out then each of these terms is weighted by this coefficient $\frac{\lambda_i}{\lambda_{max}}$.

So, λ_i is the eigenvalue corresponding to the corresponding eigenvector ith eigenvector and λ_{max} is the largest of λ is all eigenvalues in magnitude and then we can scale this vector y_0 that we have computed by this factor, let us say y_{max} λ_{max} and that gives us new trial vector x_i and now again I iterate I use this for a multiplications. So, I define y 1 new vector y_i as $A x_i$.

So, next cycle begins and again I take out λ_{max} as a common factor and this term now amplifies by 1 power. So, ith term of this expansion is now weighted by $\frac{\lambda_i}{\lambda_{max}}$ A^2

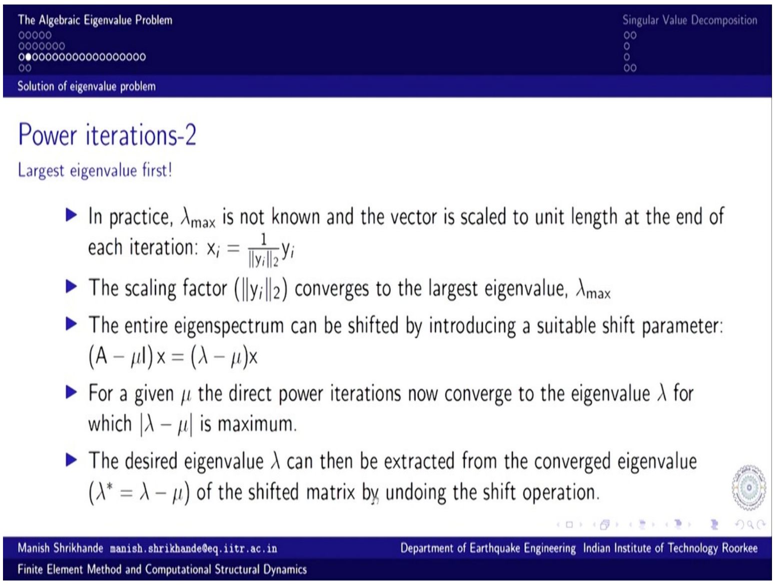
and λ_{max} can be again used to scale and find out what is the next iteration and after n number of cycles we will end up with y_n the vector y_n will be $A x_n$.

And that will again be equal to now each term is scaled by a factor $(\frac{\lambda_i}{\lambda_{max}})^n$ and since each of these terms except for λ_{max} will be less than unity by definition because λ_{max} is numerically maximum value. So, this is this weighting coefficient is always going to be a fractional number.

So, raising it to higher power, n is a positive number and first after sufficiently large number of cycles n number of cycles, these coefficients will tend to diminish and they will approach 0 and only one term will dominate and that will be corresponding to λ_{max} itself, the eigenvector to which λ_{max} the eigenvalue is the maximum of them maximum of all eigenvalues and that vector will dominate and that is how we converge to the dominant eigenvector.

So, the terms with λ_i less than λ_{max} will gradually diminish in size and the dominant term corresponding to $\lambda_i = \lambda_{max}$ will only remain in this expansion in this series.

(Refer Slide Time: 07:00)



The Algebraic Eigenvalue Problem

Singular Value Decomposition

Solution of eigenvalue problem

Power iterations-2

Largest eigenvalue first!

- In practice, λ_{max} is not known and the vector is scaled to unit length at the end of each iteration: $x_i = \frac{1}{\|y_i\|_2} y_i$
- The scaling factor ($\|y_i\|_2$) converges to the largest eigenvalue, λ_{max}
- The entire eigenspectrum can be shifted by introducing a suitable shift parameter: $(A - \mu I)x = (\lambda - \mu)x$
- For a given μ the direct power iterations now converge to the eigenvalue λ for which $|\lambda - \mu|$ is maximum.
- The desired eigenvalue λ can then be extracted from the converged eigenvalue $(\lambda^* = \lambda - \mu)$ of the shifted matrix by undoing the shift operation.

Manish Shrikhande manish.shrikhande@eq.iitr.ac.in Department of Earthquake Engineering Indian Institute of Technology Roorkee

Finite Element Method and Computational Structural Dynamics

In practice λ_{max} is of course, not known and it is commonly done what is commonly done is vector is scaled to unit length at the end of each iteration and x_i is obtained as scaling of y_i such that it has unit length and that x_i is used for subsequent iterations and the scaling factor that we have here, it converges to the largest eigen largest eigenvalue and we can do this exercise and you can see the convergence happening.

So, this second norm is actually the Euclidean norm. So, sum of squares square root of sum of squares. So, $\sqrt{(y_1^2+y_2^2+y_3^2...)}$. So, that is the distance matrix in Euclidean geometry. So, entire eigen spectrum can be shifted by introducing a suitable shift parameter. So, for example, I can this is as far as until now we are dealing with the original matrix A and we find out that no matter what we do it will end up with the largest eigenvalue and the corresponding eigenvector. So, what to do if we have to find some other eigenvalue.

So, that is done by introducing a shift parameter. So, instead of iterating with a matrix A, I iterate I consider the matrix as a shifted matrix by a parameter μ . So, I consider the matrix as $A-\mu_i$. So, the entire spectrum is shifted by μ . So, now, the eigenvalue for this $A-\mu_i$ is $\lambda-\mu$. So, this iteration with $A-\mu_i$ will converge to the largest value of $\lambda-\mu$.

So, if I can choose mu judiciously, then it is possible to use power iterations for converging to any eigenvalue that we wish and for that purpose this shift parameter Gershgorin radius, then Gershgorin disks, they are very helpful in identifying or choosing this suitable shift parameter which will allow us to converge to different eigenvalues other than the largest eigenvalue.

And of course, if we have computed largest eigenvalue then mu can be made equal to or very close to the largest eigenvalue then $\lambda-\mu$, so $\lambda_{max}-\mu$ will be a small number and then it will be converging to some other eigenvalue. So, for a given mu, the direct power iterations now converge to the eigenvalue %lambda for which %lambda - mu is the maximum.

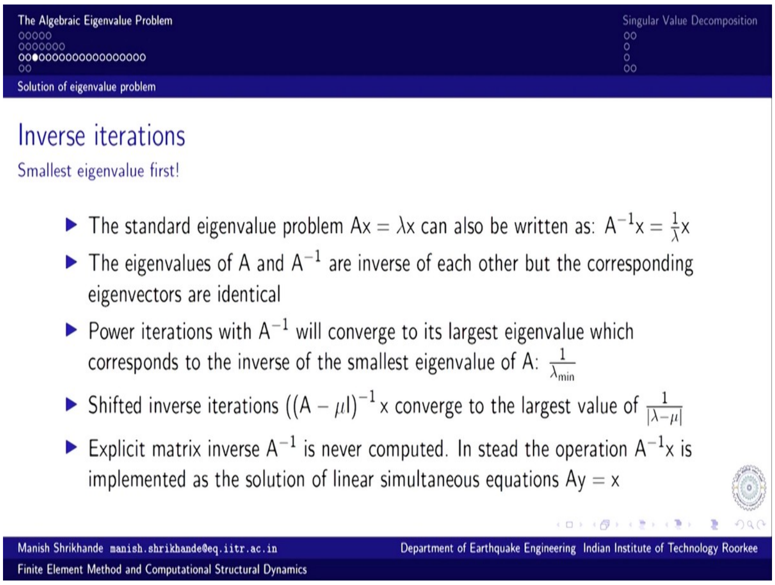
So, in a way it may if mu is closed to λ_{max} then this $A-\mu_i$, might converge to %lambda for which %lambda actual λ will be the smallest eigenvalue and desired eigenvalue %lambda can then be extracted from the computed converged eigenvalue $\lambda^{*}=\lambda-\mu$ of the shifted matrix by undoing the shift operation. So, $\lambda=\lambda^{"}*"+\mu$.

And that once we obtain λ , then we can again go back to $A-\lambda_i$ and find suitable corresponding eigenvector corresponding to this eigenvalue, computed eigenvalue. So, now, we discussed this the power iterations, direct power iterations we just matrix multiplication and a trial vector and multiplying it by matrix repeatedly and scaling at the end of each cycle, it leads to the largest eigenvalue and the corresponding eigenvector.

Now most of the time what we are interested in is the smallest eigenvalue, not really the largest eigenvalue. Sometimes largest eigenvalue is also required that we will see in the next topic that we will discuss the time marching scheme numerical integration of equations in time, but for most of the analysis purposes as far as a linear basis is concerned.

It is the lower eigenvalue spectrum and lower corresponding eigenvectors that are of more primary interest and therefore, this procedure of largest eigenvalue converging to largest eigenvalue is little inconvenient. So to say, so, is it possible to look at a procedure by which we can compute the smallest eigenvalue. So, let us look at rearranging this system of equations.

(Refer Slide Time: 12:37)



The Algebraic Eigenvalue Problem

Singular Value Decomposition

Solution of eigenvalue problem

Inverse iterations

Smallest eigenvalue first!

- The standard eigenvalue problem $Ax = \lambda x$ can also be written as: $A^{-1}x = \frac{1}{\lambda}x$
- The eigenvalues of A and A^{-1} are inverse of each other but the corresponding eigenvectors are identical
- Power iterations with A^{-1} will converge to its largest eigenvalue which corresponds to the inverse of the smallest eigenvalue of A: $\frac{1}{\lambda_{\min}}$
- Shifted inverse iterations $((A-\mu I)^{-1}x$ converge to the largest value of $\frac{1}{|\lambda-\mu|}$
- Explicit matrix inverse A^{-1} is never computed. In stead the operation $A^{-1}x$ is implemented as the solution of linear simultaneous equations $Ay = x$

Manish Shrikhande manish.shrikhande@eq.iitr.ac.in Department of Earthquake Engineering Indian Institute of Technology Roorkee

Finite Element Method and Computational Structural Dynamics

So, $Ax=\lambda x$, it can also be written as $A^{-1}x=\frac{1}{\lambda}x$. So, I pre multiply it by A^{-1} on both sides and then divide by %%lambda on both sides and this equation gets transformed to $A^{-1}x=1$ over %lambda x.

So, that brings us to the basic property of eigenvalues. If λ is an eigenvalue of A then 1 over λ is an eigenvalue of^-1 of A and then $A^{-1}x$ essentially indicates solution of simultaneous equations. So, we do not need to compute $A^{-1}x$, we only need to solve the system of equations for x to be as a right hand side vector. So, that is what we do. So, eigenvalues of A and A^{-1} are inverses of each other, but the corresponding eigenvectors are of course, identical.

So, power iterations with A^{-1} if for some fortuitous reasons we have A^{-1} matrix available, then power iterations with A^{-1} will converge to its largest eigenvalue which so, largest eigenvalue of A^{-1} and that would be corresponding to the inverse of the smallest eigenvalue of A. So, if %lambda min is the smallest eigenvalue of A, then the largest eigenvalue of A^{-1} will be_1 over λ_{min} .

So, essentially if I do power iterations with the same I have set of iterations with A^{-1} , then I converge to the lower end of the eigen spectrum or the lowest eigenvalue and corresponding eigenvector, but as I said rarely if ever we will have this A^{-1} with us and when we say $A^{-1}x$ we always refer to it as a solution of simultaneous equation inverse need not be computed.

And we can further try to converge and accelerate the convergence by again using shifted inverse iteration. So, because the convergence actually depends on the ratio $\frac{\lambda_i}{\lambda_{max}}$. So, if this ratio is made very small then the convergence is going to be very very rapid and to that effect the shift parameter can be used to great help to converge.

And convergence I mean iterations with $A-\mu_i^{-1}$ iterations, they would converge to the largest value of $\lambda-\mu$ and as I said explicit matrix inverse is never computed instead the operation A^{-1} x is implemented as the solution of linear simultaneous equation as Ay = x and once we know once we have the root algorithms implemented for solution of simultaneous equations, these equations can be solved repeatedly and with each iteration at the end of each iteration they can be scaled and again started with new right hand side vector.

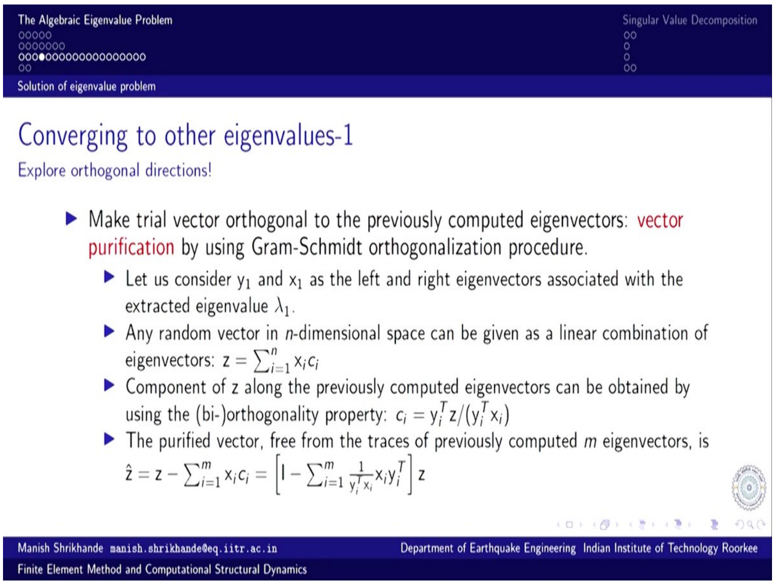
So, that is the process for converging to lower eigenvalues or lowest eigenvalue and rest of the things remain same as a power iteration sequentially we compute at each end of each cycle we scale the vector computed vector to unit length and start again with the new vector. So, in case of the inverse iteration, we start with new vector as the new right hand side to be solved for unknowns of equation Ay = x and eventually it will converge to the smallest eigenvalue of matrix A and inverse of the smallest eigenvalue of matrix A from which smallest eigenvalue can be calculated and corresponding eigenvector.

So, essentially what we have been able to do so far is calculate largest eigenvalue and calculate smallest eigenvalue by inverse iteration and if we are lucky and we are more um adventurous, then we can play with a shift parameters and hope to converge to some of the intermediate eigenvalues, but it is really too much of a tedious trial and error work by playing with shift parameters to compute the complete eigen spectrum.

There should be a more structured method for arriving at eigenvalues other than the largest and the smallest eigenvalue and to that we make use of the orthogonality property of eigenvectors. As we said eigenvectors they are mutually orthogonal to each other I mean we are dealing with symmetric matrix if A is symmetric then the eigenvectors are mutually orthogonal.

If A is not symmetric then left eigenvectors are orthogonal to left right eigenvectors. So, there is a biorthogonality that exists. So, whichever way, we do have some kind of orthogonal orthogonality available and that can be used to make the solutions converge to other eigenvalues other than largest and the smallest. Let us see how. So, converging to the other eigenvalues and we this captures the basic essence explore orthogonal directions.

(Refer Slide Time: 18:53)



The Algebraic Eigenvalue Problem

Singular Value Decomposition

Solution of eigenvalue problem

Converging to other eigenvalues-1

Explore orthogonal directions!

- Make trial vector orthogonal to the previously computed eigenvectors: vector purification by using Gram-Schmidt orthogonalization procedure.
  - Let us consider y_1 and x_1 as the left and right eigenvectors associated with the extracted eigenvalue λ_1.
  - Any random vector in n-dimensional space can be given as a linear combination of eigenvectors: $z = \sum_{i=1}^{n} x_i c_i$
  - Component of z along the previously computed eigenvectors can be obtained by using the (bi-)orthogonality property: $c_i = y_i^T z / (y_i^T x_i)$
  - The purified vector, free from the traces of previously computed m eigenvectors, is $\hat{z} = z - \sum_{i=1}^{m} x_i c_i = \left[I - \sum_{i=1}^{m} \frac{1}{y_i^T x_i} x_i y_i^T \right] z$

Manish Shrikhande manish.shrikhande@eq.iitr.ac.in Department of Earthquake Engineering Indian Institute of Technology Roorkee

Finite Element Method and Computational Structural Dynamics

So, earlier we found one direction by power iterations and inverse iteration, we found one direction that gradually converges to the largest dominant eigenvalues eigen and eigenvector corresponding to the dominant eigenvalues now; obviously, next eigenvector is not going to be in this direction, it will be orthogonal to this, now orthogonal can be any way.

But that is sure that what has been computed, new eigenvector will not be aligned in this direction. So, that is what we do by a process called vector purification and we invoke gram Schmidt orthogonalization procedure to enforce that the vector trial vector that we do that we use that is orthogonal to the previously computed eigenvectors.

So, if the trial vector that we use for iteration if that is orthogonal to the previously computed eigenvectors then of course, it will not the process will not converge to the previously computed eigenvectors. So, in theory if we do it once at the start of iteration that should work, but in practice because of a finite precision arithmetic of floating point operations, there are round of errors and the orthogonality may be lost during the iteration cycles because of floating point of operations. So, this orthogonalization is required at every cycle.

So, it is not just once and then forget. The vector trial vector needs to be made orthogonal with respect to previously computed eigenvectors through the gram Schmidt orthogonalization procedure at the beginning of each iteration. So, how do we do this?

So, let us consider y1 and x1 as left and right eigenvectors associated with the extracted eigenvalue λ_1 right.

So, if λ_1 is available then yi and xi. So, left and right eigenvectors can be computed. So, any random vector in n dimensional space can be given as a linear combination of the eigenvectors. So, z as an arbitrary vector can be written as $x_i c_i$, summation over i. So, xi are the eigenvectors and ci they are the let us consider them I mean they are generalized coordinates for making up any vector z in terms of xi.

Component of z along previously computed eigenvectors can be obtained by using biorthogonality property. So, if we have computed let us say yi and xi are known, then we can compute we can pre multiply this by yi. So, $y_i^T z$ and if we look at it then yi would be orthogonal to all eigenvectors except the first eigenvector, xi and yi and xi they are of I mean they would be of unit length.

So, they can be scaled to have unit I mean such that $y_1^T x_i$ are of unit or unit length or that is of unity or whatever number that is, but $y_i^T x_j$ where i is not equal to j that is 0, that is the biorthogonality property. So, only one term will remain corresponding to if we choose i^th^ vector $y_i^T z$ then only i^th^ term in this entire expansion will remain and rest of the terms will vanish by virtue of orthogonality.

So, $y_i^T x_i$ that will go in the denominator and that is what gives us the component to what extent x_i is present in making up the vector trial vector z. So, the component of x_i in z is c_i and that is given by this relation, the projection and once we have this then what how do I make the trial vector orthogonal with respect to y_i, I just or orthogonal with respect to x_i, I just a subtract the component of x_i from z.

So, x_i is present in z to the extent of $c_i x_i$. c_i is the weighting coefficient. So, if I subtract c_i x_i from z then this component of x i is completely removed and the resulting vector will not have any component along x i and that is it is orthogonal with respect to x_i. So, the purified vector free from the traces of previously computed m eigen vectors is so, if I have computed m number of eigenvectors. Then I can compute similar components of all the m eigenvectors and subtract it from this trial vector z.

And this new trial vector purified trial vector let us call it z hat and that can be used for iteration. Now this can be arranged in a matrix form and I can actually take this I mean because c i we have c i here and z is a part of c i. So, this can be z can be taken out and this can be looked at as a matrix.

So, i - sigma i raised to i from 1 to m and thats a again individual for each ith eigenvector set, we have a matrix that accounts for orthogonality with respect to ith eigenvector. So, this matrix is called Sweeping matrix and when this is multiplied with any vector trial vector then effectively what it does is it implements this process of computing the component and then subtracting the traces.

So, this matrix multiplication will take care of these both operations, computing the component and extent of representation and then removing the traces of that vector from the original vector and thereby making it orthogonal. So, this is called the Sweeping matrix and gradually we start with one vector and then after computing second vector we add one more term to this and so on.

So, we can use this Sweeping matrix for a gradually improve and trial iterations can be um purified vector can be used by using the sweeping matrix and at the beginning of each iteration the trial iteration can be again purified by multiplying with Sweeping matrix and then the iterations would converge to eigenvalues largest or smallest depending on whether we are going for forward iteration or inverse iteration.

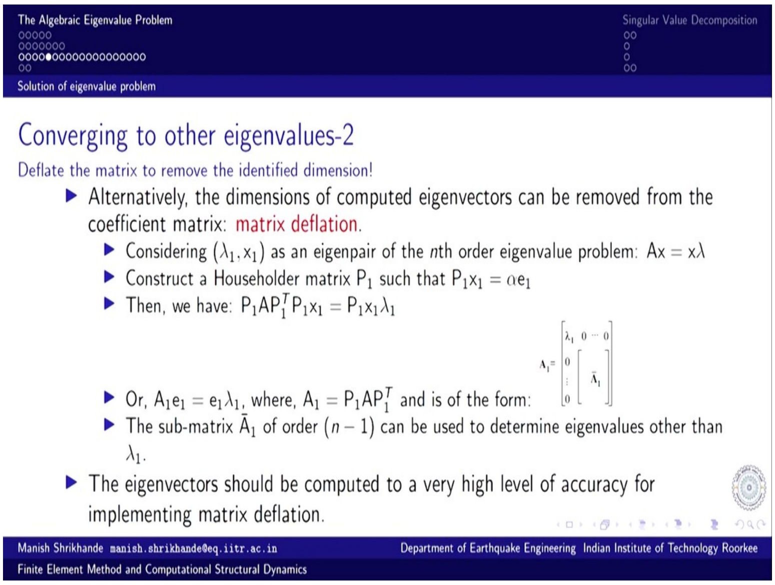
Other than this m number of eigenvalues and eigenvectors that have already been computed. So, this is what we call as a vector purification and this can be implemented very easily although it is not very efficient procedure, numerically efficient procedure. There are more efficient numerical procedures for solving eigenvalue problems, but this is the arguably simplest one to implement on a computer.

All that it requires is a series of matrix multiplication matrix vector multiplication that is all. So, this can be done very easily and coded very easily and it can be done for a again as a finishing operation. So, these power iterations can be done as a finishing operation for a some of the eigenvalues estimates and eigenvectors that are computed from other methods.

So, once we have an idea of a eigenvalues and eigenvectors, but we need to refine them, then we can use inverse iteration with shift and that will converge very very quickly and just one or two iterations of inverse iteration which shift parameter that will produce beautiful and very accurate eigenvalues and eigenvectors.

So, another way is to converge to other eigenvalues is not looking at the eigenvector. or the trial vector or removing the traces of computed eigenvectors from the trial vector rather we can modify the matrix itself, such that the iterations with the modified matrix do not lead to previously computed eigenvectors. So, we kind of remove the traces of previously computed eigenvectors from the original matrix. So, that is called matrix deflation.

(Refer Slide Time: 29:09)



The Algebraic Eigenvalue Problem

Singular Value Decomposition

Solution of eigenvalue problem

Converging to other eigenvalues-2

Deflate the matrix to remove the identified dimension!

- Alternatively, the dimensions of computed eigenvectors can be removed from the coefficient matrix: matrix deflation.
  - Considering (λ_1, x_1) as an eigenpair of the nth order eigenvalue problem: $Ax = x\lambda$
  - Construct a Householder matrix P_1 such that $P_1 x_1 = \alpha e_1$
  - Then, we have: $P_1 A P_1^T P_1 x_1 = P_1 x_1 \lambda_1$
  - Or, $A_1 e_1 = e_1 \lambda_1$, where, $A_1 = P_1 A P_1^T$ and is of the form: $A_1 = \begin{bmatrix} \lambda_1 & 0 \cdots 0 \\ 0 & \\ \vdots & \left[\ \bar{A}_1\ \right] \\ 0 & \end{bmatrix}$
  - The sub-matrix $\bar{A}_1$ of order $(n-1)$ can be used to determine eigenvalues other than λ_1.
- The eigenvectors should be computed to a very high level of accuracy for implementing matrix deflation.

Manish Shrikhande manish.shrikhande@eq.iitr.ac.in Department of Earthquake Engineering Indian Institute of Technology Roorkee

Finite Element Method and Computational Structural Dynamics

So, if λ_1 and x_1 is an eigenpair of the nth order eigenvalue problem let us say $A x = x\lambda$, then we consider a householder matrix P_1 such that $P_1 x_1 = \alpha e_1$. So, alpha is a coefficient and e_1 is euclidean unit vector first euclidean unit vector 1 0 0 0 0 so on. So, the idea is this reflection matrix P_1 will map x_1 onto the first euclidean unit vector.

So, if I do this then using similarity transform, we can have $P_1 A P_1^T$ and $P_1 x 1$ because if you look at it householder transforms householder transformation matrices they are orthogonal matrices. So, $P^T P$ is identity matrix. So, this really does not make any difference. So, $P_1^T P$ is essentially identity matrix.

So, all that we have done is pre multiply here by P_1 and pre multiply here by P_1. So, $P_1 Ax = P_1 x \lambda$ and in between we have this $P_1^T P$ that is essentially identity matrix. So, $P_1 A P_1^T$ this is of course, similarity transform because P_1 and P_1^T they are inverses of each other and therefore, the eigenvalues do not change and P_1 x is of course, the eigenvalue of these modified matrix.

Eigen vector of this modified matrix and λ_1 is of course, the eigenvalue original eigenvalue. So, what happens P_1 x is of course, αe_1 that is unit vector euclidean unit vector. So, $a_1 e_1$. So, if I call this as a 1 and $P_1 x_1$ is of course, unit vector e_1 then this = $e_1 \lambda_1$ and A1 is of the form $P_1 A P_1^T$ and this is of this form λ_1 .

And this is all 0s and these are all 0s and there is a sub matrix which is A bar let us call it A bar 1. So, this sub matrix. So, this sub matrix now if you see this sub matrix is decoupled from this. So, first eigenvalue is isolated here λ_1 , now I can iterate with this remaining sub matrix and it will lead to if I do the power iterations it will lead to the largest eigenvalue next largest eigenvalue. Other than λ_1 if λ_1 is the largest eigenvalue.

If I do the inverse iteration and λ_1 is the smallest eigenvalue then inverse iteration will lead to the smallest eigenvalue which will be smallest for $\bar{A}_1$, but larger than 1 λ_1 . So, sub matrix $\bar{A}_1$ is of the order n - 1 and can be used to determine eigenvalues other than λ_1 . So, this is used for calculation of the eigenvalues. Once the eigenvalues are obtained because eigenvalues are identical because of this similarity transform, the eigenvectors for calculation of eigenvectors we will of course, have to go back to original matrix A.

So, eigenvector should be computed to a very high level of accuracy for implementing matrix deflation because this reflection matrix P householder transformation matrix that depends on this eigenvector that has been computed earlier. Now if this eigenvector is an error then this entire process leads to very very poor results. So, the first requirement is of course, that the eigenvectors have to be computed to a very high degree of accuracy for implementing this matrix deflation.

So, generally this matrix deflation is not used in practice. Its very rarely that you will find matrix deflation being discussed in engineering analysis. Nevertheless another technique for eigenvalue computation is what we call as Raleigh Quotient Iteration. Recall that Shifted Inverse Iteration or Shifted Direct Iteration, they will converge to the largest eigenvalue and shifted inverse iteration in particular if I can find an estimate of eigenvalue which is close to an eigenvalue. True eigenvalue then $\lambda-\mu$ the shift parameter appears in the denominator.

So, if μ is close to λ , then that parameter becomes very large. $\frac{1}{(\lambda-\mu)}$ becomes very very large and that will converge very quickly to that particular eigenvalue close to mu and to get an estimate of λ Raleigh quotient is a very very powerful technique. So, for standard eigen value problem Raleigh quotient is a ratio of 2 scalars defined as for any vector x, $\frac{x^T Ax}{x^T x}$ and for generalized eigenvalue problem, it is the ratio of $\frac{x^T Ax}{x^T Bx}$.

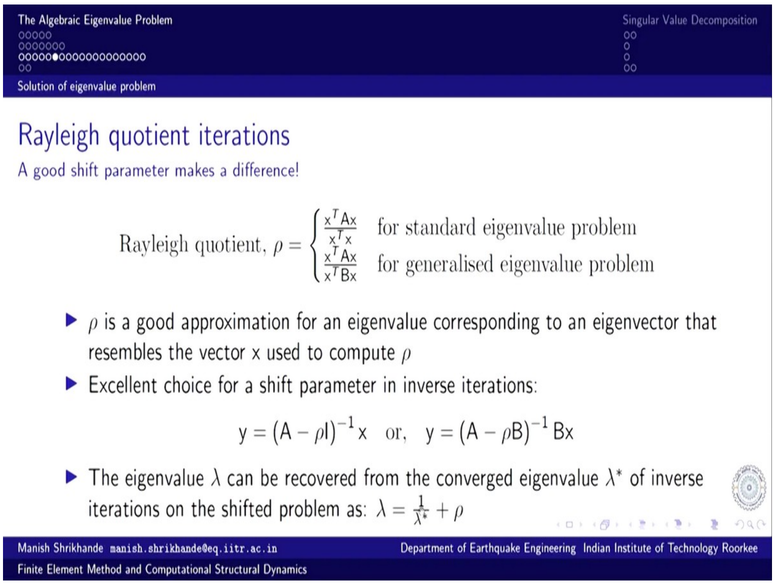
So, A and B are the coefficient matrices of generalized eigenvalue problem. So, for any trial vector x, this Raleigh quotient can be calculated. And generally Raleigh quotient is a good approximation to an eigenvalue and Raleigh quotient converges to eigenvalue faster than x converges to the corresponding eigenvector. So, even if x is a very poor approximation to any eigenvector, Rayleigh quotient will in general be a better approximation or it will be more accurate higher degree of accuracy it has an higher degree of accuracy.

Say its actually second degree accurate, second order accurate to the eigenvalue true eigenvalue. So, it will converge to eigenvalue Rayleigh quotient converges to eigenvalue much faster than the trial vector x converges to the eigenvector.

So, rho I mean the Raleigh quotient is a good approximation for an eigenvalue and if we have some idea of the shape of eigenvector then we can just assume that kind of shape similar shape kind of thing and compute Raleigh quotient and then use this Raleigh quotient in the inverse iteration algorithm and this algorithm will converge very very quickly to the eigenvalue λ that is close to rho that is closest to the Raleigh coefficient rho.

So, it can be done for both direct iterations and I mean standard eigen eigenvalue problem as well as the generalized eigenvalue problem. It only requires different right hand side vectors to be solved for. So, eigenvalue λ can be recovered from the converged eigenvalue λ^* of inverse iterations.

(Refer Slide Time: 37:36)



The Algebraic Eigenvalue Problem

Singular Value Decomposition

Solution of eigenvalue problem

Rayleigh quotient iterations

A good shift parameter makes a difference!

$$\text{Rayleigh quotient, } \rho = \begin{cases} \frac{x^T A x}{x^T x} & \text{for standard eigenvalue problem} \\ \frac{x^T A x}{x^T B x} & \text{for generalised eigenvalue problem} \end{cases}$$

- ρ is a good approximation for an eigenvalue corresponding to an eigenvector that resembles the vector x used to compute ρ
- Excellent choice for a shift parameter in inverse iterations:

$$y = (A - \rho I)^{-1} x \quad \text{or,} \quad y = (A - \rho B)^{-1} Bx$$

- The eigenvalue λ can be recovered from the converged eigenvalue λ^* of inverse iterations on the shifted problem as: $\lambda = \frac{1}{\lambda^*} + \rho$

Manish Shrikhande manish.shrikhande@eq.iitr.ac.in Department of Earthquake Engineering Indian Institute of Technology Roorkee

Finite Element Method and Computational Structural Dynamics

On the shifted problem as $\lambda = \frac{1}{\lambda^*} + \rho$. So, that is the Raleigh quotient iteration and that converges very quickly and that is a very handy tool for computing eigenvalues. And eigenvectors if we have some idea of the shape of eigenvector that we are looking at.

In next lecture we will discuss more powerful methods for computing a set of eigenvalues and eigenvectors. So, far we have been targeting 1 eigenvalue and associated eigenvector at a time, now we will look at methods which can give us a large number of I mean simultaneously some eigenvalues and some associated eigenvectors, in our next lecture.

Thank you.