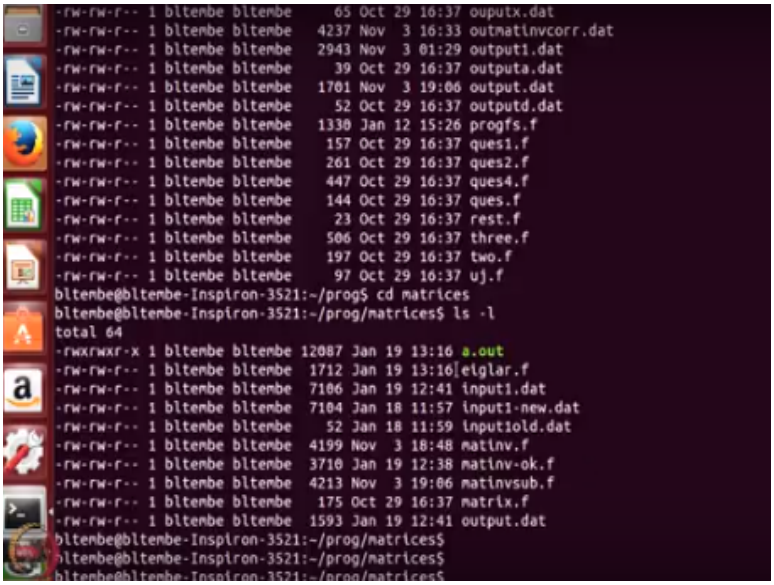
Computational Chemistry & Classical Molecular Dynamics
Prof. B. L. Tembe
Department of Chemistry
Indian Institute of Technology - Bombay

Lecture – 19
Matrix Inversion, Matrix Diagonalization

Hello and welcome to this session. So what we will do in this session is to use the programs for matrix inversion and matrix diagonalization and illustrate how you can invert and diagonalize matrices as well as the diagonalization process is concerned will only get the largest eigen value inversion we will do for any matrix any size n/n so what you will see now I am in my program directory. in my program directory I have created many sub directories.

(Refer Slide Time: 00:47)



To illustrate different task that I am going to do so let me do this ls. When I do ls it will give the list of all the programs and which are the directories can you see those directories which this light blue color, so see this light blue color will be one directory interp, okay so let me so what I have done here I have listed all the files in my directory prog. So, these are all my directories and you will see that whichever are directories will come with the different color.

So, there is the line of separating system I have directory interp where I have discussed the programs are interpolation, I have directory matrices in which I have put all the programs and all the executable programs like a. out bxz they are all written in a yellow color. So, let us do the

same thing I will do ls-l okay. So, when I do ls-l you will see that directories come with this blue color interp matrices and the executable files will come in this yellow color.

So, you will see this in that you may see other yellow or bright color bright green color okay. So, now I want to go to my directory matrices in which all the programs are there let me list all of them okay so my programs there are mat invert. f that is the program to invert matrices. Now I use several inversions of that mat inverse and the one which have worked best, and I have called it matinv dash ok. f mat inversion program.

And the other program is eiglar.f. This is the program to get the largest eigen value of that matrix okay and for all these I will need input so that I have called input1.dat and whatever output is there it will write as output.dat output will be the output of the program. So, now let me edit this matinv. f to find what that program is how do I edit pi, so this is mat inv dash ok.f.

(Refer Slide Time: 03:12)

So, this is my program okay so some things you are not able to see okay so to do that what I shall do so now you are able to see the whole thing on your screen. So, I have inserted those extra lines here initially I shall delete before compiling, okay. So, we want to understand this particular program matinv. f I have started in one lecture before the last it is an inversion program. So, I want to do it for a 4/4 case.

First line is the comment card matrix inversion program by gauss elimination okay, so I need several matrices there called it a 4,4 b 4,4 aold 4,4 c4,4 this dimension statement and remember the dimension statement can also be returned using real okay. So I did not say dimension here I just say real a 4,4 so this gives the array a the a which is the variable it will now a matrix 4/4 matrix.

So, these are my arrays I need in my program so now there is the parameter N I can define parameter to this parameter statement. The advantage of that is you do not have to change N in this rest of this programs. Suppose it is a 4/4 case so instead of writing 4 everywhere if I just say n=4 it takes n to the 4 okay see there will be many read and write statement where N is used. So, if I use this parameter, I can write my entire program with n as a variable.

And suppose instead of 4 I want to do 5/5 I just take this n to be 5 so this defining parameters in a program is a effective way of programming. So, I have defined my dimension statement, I have defined my parameter, I open unit=11 file=input 1.dat my input 1 has input for this inversion program the results I want to write it as output.dat, okay. Then I need some small number because remember we are going to divide by the diagonal element of that matrix.

If that diagonal element is very small, I cannot divide by a small number so what is small 0 is not necessity for best way because the computer it has only 8 digits after a decimal point for a single precision. So, a number which is 10-8 I am going to take it as very small number, so I do not want to divide by any number smaller than .00000001 so this my small number. So, now I come to this write statement so write 12, star it does not do anything just writes a blank line.

Okay so but see this is the comment card c write 12 it is just a blank line nothing is done this write statement write 12, star the original matrix is okay. Okay it tells me that on the screen it will write the original matrix is okay, so the read statement is read 11, star aij j going from 1 to n I going from 1 to n these are statement where I read all the elements of that matrix too implicit loops are there J going from 1 to n and I going from 1 to n.

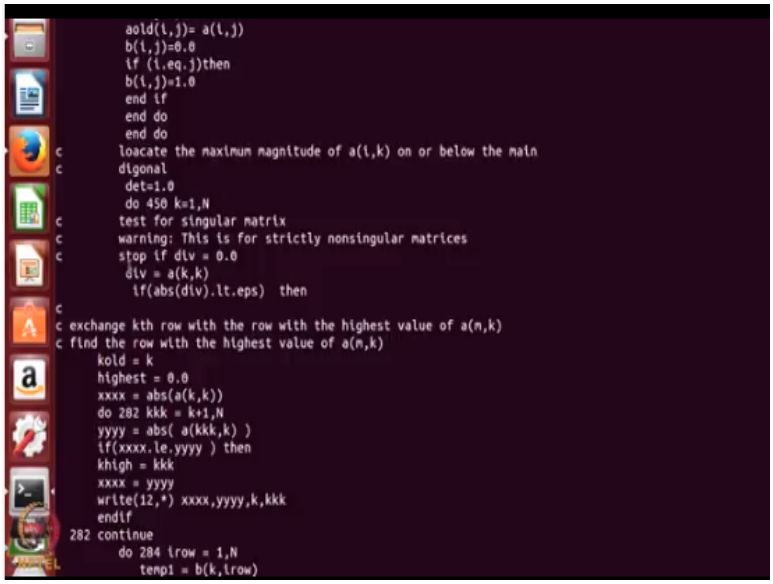
So I am going to read all the rows first and first row is read because I is 1 then j goes from 1 to 4 ai1 ai2 ai3 ai4 first row is read first then second row third row 4th row then I also write that matrix in file 12. Remember that I had mentioned to you that whatever you read you write it so that you know what calculations are being done okay. So then what I want to do original value of whatever original value of a was there I will save it in aold.

Because the moment I change the values of a remember in our operations by elementary operations the values of a go on changing during the process. So the original values I want to save in variable called aold. So my these loops here now what the next loop does it writes all the value of aij*aold ij it also now the next task is I want to define matrix b which is an identity matrix.

Remember whatever elementary operations I am going to use I am going to operate with the elementary operations on matrix a as well as a unit matrix. So, my b is the unit matrix, how do I define a unit matrix look at this loop I said all bij=0 except when i=j if i=j then bij=1 all the rest it is 0 okay. So, this if statement I use to make all bij values 1 only when i= j okay so that completes my definition of b as an identity matrix, so what is my next task now?

So, I want know start dividing by the diagonal element okay.

(Refer Slide Time: 08:46)



So, first elementary operation has started I want k going from 1 to n so my operation has started now okay. So, I will just see whether k=1 means and now going to divide by akk the diagonal value of the first row so I check if I call akk as div div is defined as akk and if the absolute value of div is < epsilon the number epsilon I have already taken to be 10-8 if this is less than that then.

I do not want to divide I want to exchange the ith row and the row corresponding to the largest value of elements in that first column okay. So, that is what I am doing so with these lines I am going to exchange the largest value in that particular column okay that is what I am going in this particular loop okay where is my loop okay kold=k highest=0 absolute value in that particular column I find the largest value whichever is the largest value I exchanged that particular row.

See I am exchanging that particular row with the first row, so this is needed only when my diagonal element is 0 in our case 0 means < 10-8.

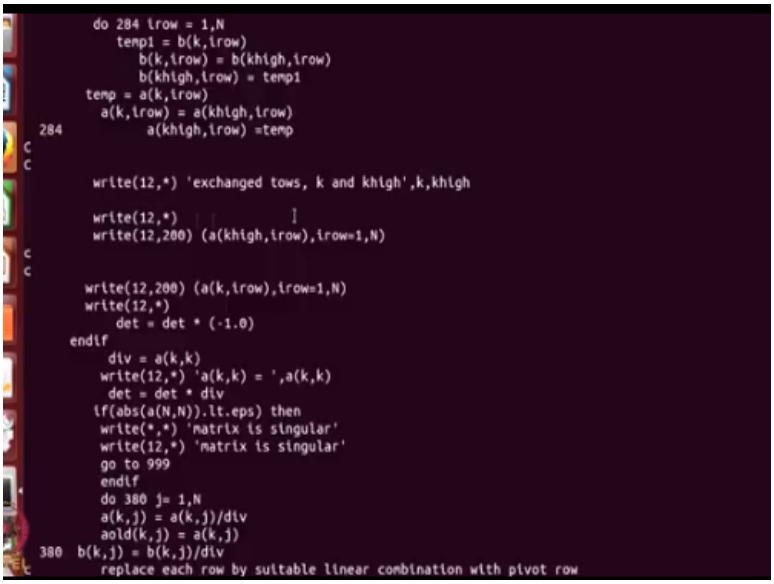
(Refer Slide Time: 10:15)

```
          if(abs(div).lt.eps)  then
c
c exchange kth row with the row with the highest value of a(m,k)
c find the row with the highest value of a(m,k)
      kold = k
      highest = 0.0
      xxxx = abs(a(k,k))
      do 282 kkk = k+1,N
      yyyy = abs( a(kkk,k) )
      if(xxxx.le.yyyy ) then
      khigh = kkk
      xxxx = yyyy
      write(12,*) xxxx,yyyy,k,kkk
      endif
282 continue
        do 284 irow = 1,N
           temp1 = b(k,irow)
              b(k,irow) = b(khigh,irow)
              b(khigh,irow) = temp1
        temp = a(k,irow)
          a(k,irow) = a(khigh,irow)
284           a(khigh,irow) =temp
c
c
        write(12,*) 'exchanged tows, k and khigh',k,khigh

        write(12,*)
        write(12,200) (a(khigh,irow),irow=1,N)


       write(12,200) (a(k,irow),irow=1,N)
       write(12,*)
```

So, I have done this exchange procedure okay so now 282 what does this 282 do that k goes from this kkk goes to k+1 to n because its already the loop which is having the index k. So, the remaining lower values in that particular column. I am exchanging with the largest value in that particular column.

So, all this is just to exchange this 282 loop is to exchange my present row with the row in which that particular column that value of matrix element is the largest. So, this exchanging I am doing here okay so next then now I have to divide I have to divide now so whatever exchanging I did for row a I should do that exchanging for row b as well. Remember a and b are identical so whatever I do on row a I do exactly on row b okay.

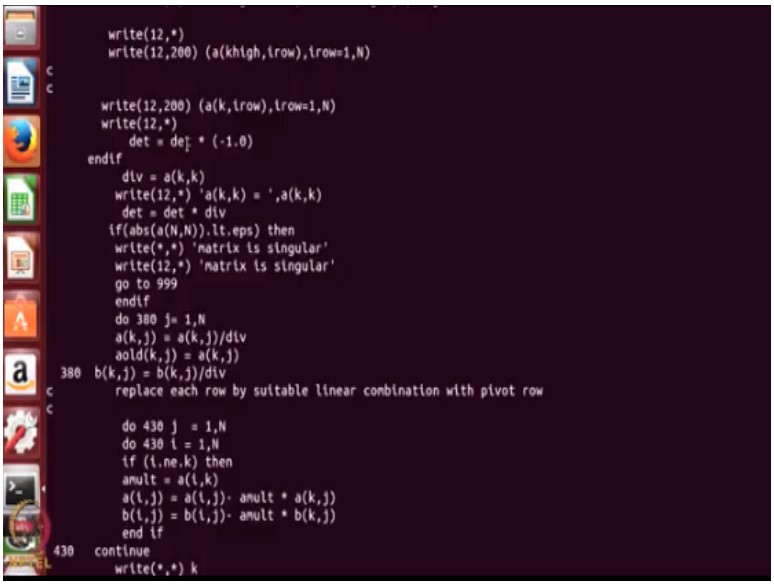
(Refer Slide Time: 11:16)



So, when I do this exchange it is writing on the screen exchange rows k and ki it exchanges those two rows and i write on the screen. So, now I have start the dividend procedure okay so what is the before that I have a determinant which I have initialized to 1 whenever I exchange those the determinant changes sign, so I am putting determinant = determinant*-1 okay. Now my div akk okay.

So, I am going to divide now and incase all the elements in that particular column are 0 if all the elements are 0.

(Refer Slide Time: 11:56)

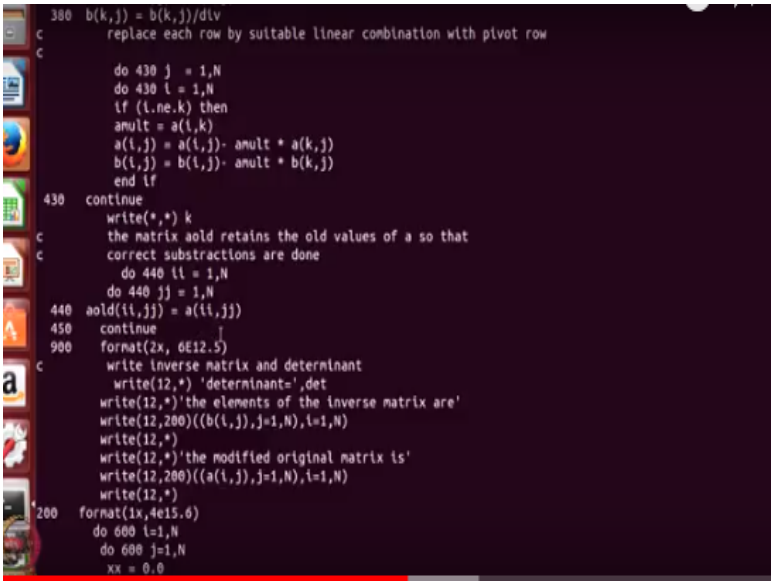


Then one of the eigen values is 0. Okay therefore then I cannot calculate the inverse of a matrix whenever I cannot calculate the inverse then I will print on the screen matrix is singular. Therefore, then I cannot calculate the inverse, remember whenever the determinant of the matrix is 0 okay, I cannot calculate the inverse. So, inversion is only for those component is 0 so whenever it is singular go to 999 it will just end the program.

When you will go to 999 it will end the program if it is not 0 then I shall do these operations. I am going to divide akj okay all the elements of that kth column will be divided by div okay so naturally the diagonal value will be 1 and all the rest will be divided by div okay. So this all the akjs will be divided by div. And this complete the loop and all the bkjs will also be divided vy div okay so what does it do normalizes the diagonal value.

Once I normalize the diagonal value my next job is to replace each of the other rows by linear combination of the ith row and the pivot row. Let us say that the diagonal value was not 0, then I am going to exchange the remaining rows by taking combination of the new row and the old row exactly so that all the remaining values are 0.

(Refer Slide Time: 13:36)

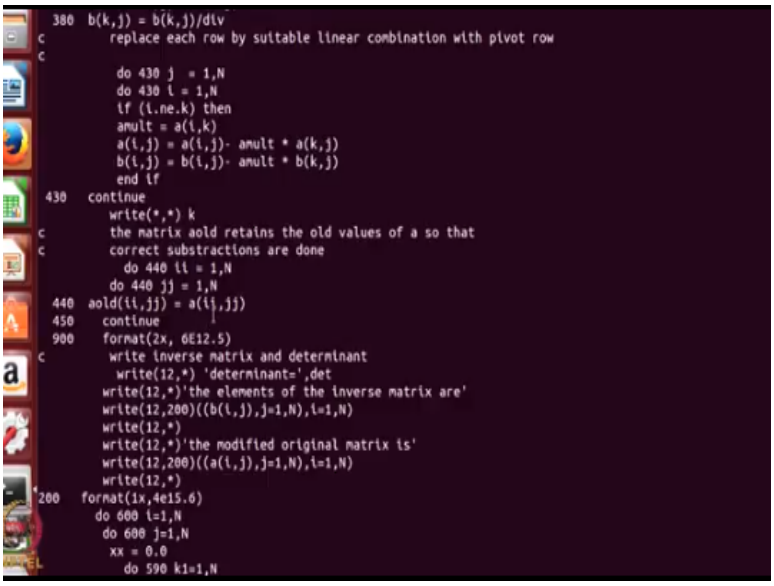


So, remember what I am doing now look at this loop okay. I am going to take the new aij will be old aij-multiplier times the old value of akj. What is this multiplier that particular value in the column? Look at our previous lecture which tells you how to set the all the elements in the particular column to be 0 is exactly the procedure to set all elements in the column other than the diagonal into 0.

So, this is basically remember we called it Eik of c remember this elementary operation was there Eik of c you multiply that so that all the non-diagonal elements become 0. So, this I will do this for a then I do this for b that ends my loop then once I finish for one particular column then I go to the next column. So, next column I repeat the same procedure whatever I did for the first column I will do for the second column third column 4th column.

So, since this is the 4/4 case. I will repeat all those operations for the 4 columns okay so when I do all those operations okay whatever was my original matrix a that has now become the inverse of a and what a has become sorry a has become the identity matrix. A has become the identity matrix and b has become the inverse of a.

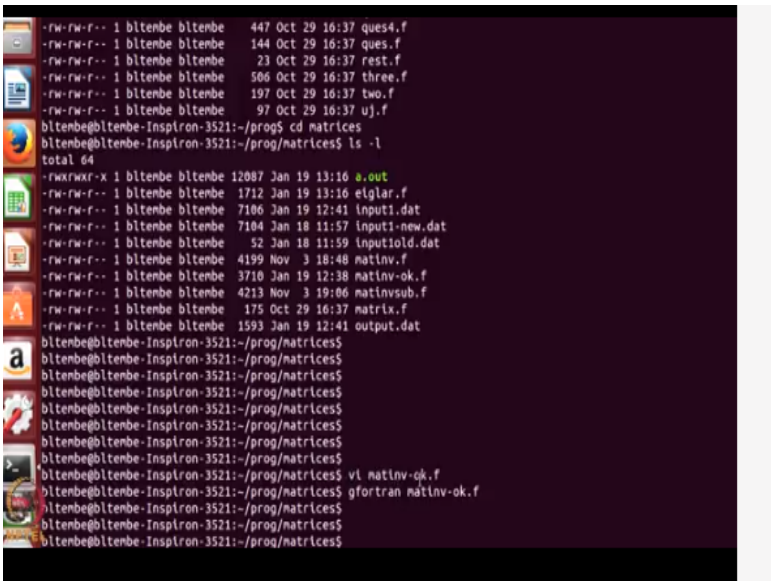
(Refer Slide Time: 15:06)



So, to check that what you have to do I have to multiply old matrix which was aold okay. I multiplied by the matrix b, so I take the product of original a and b and the product of the original and inverse should be identity matrix, okay. So these are the operations now so to summarize what I have done I divided by the diagonal values to make the diagonal value 1 all the off diagonal elements I set to 0.

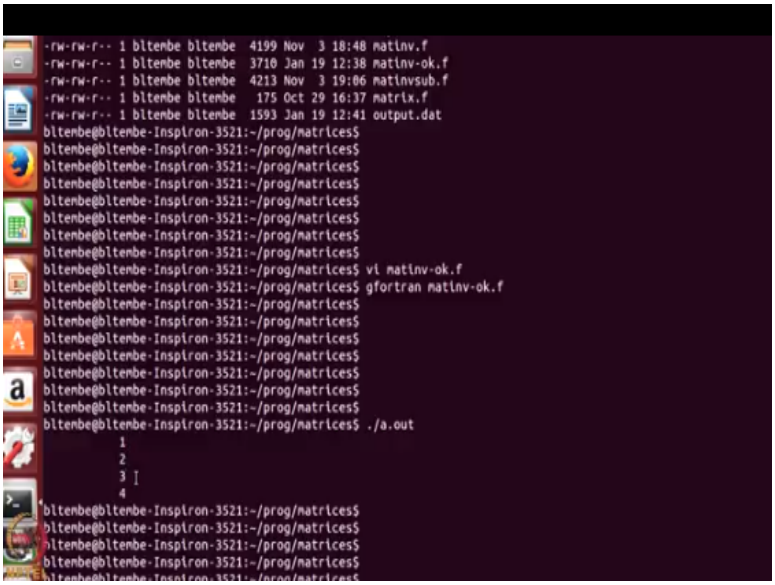
And I did this for every column of matrix a, so the resultant would be a is now a unique matrix and b is an inverse of a. Then I also multiply a and a inverse to check whether it is a identity matrix. So, these are all the remaining programs now I will compile these programs. Before I compile these programs initially, I had some blank statements so what I shall do I shall remove all the blank statements, okay.

(Refer Slide Time: 16:12)



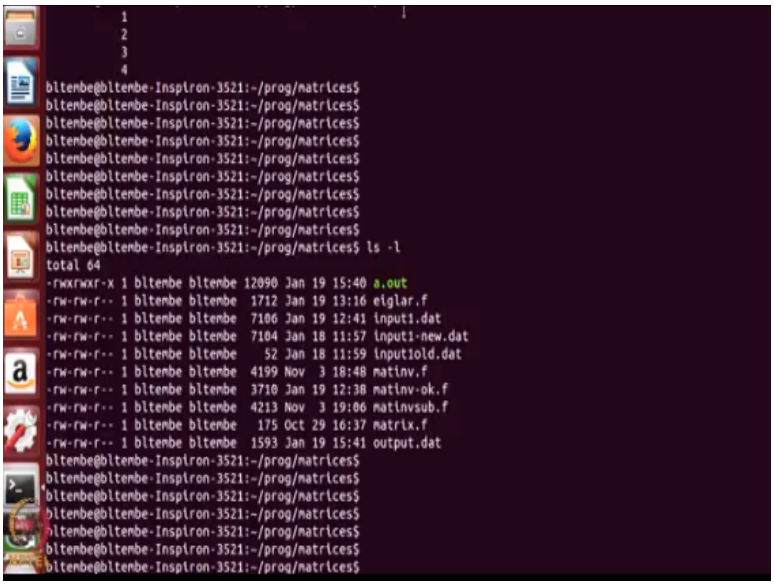
Escape and shift column, okay. So now I will compile this program by typing gfortran matinv-okay.f. when I compiled that, so I have got a result like this. So, you will see that I compiled the program by typing vi matin-ok. f so it compiled and has given me a new a.out, I will execute this program by typing. /a .out

(Refer Slide Time: 17:02)



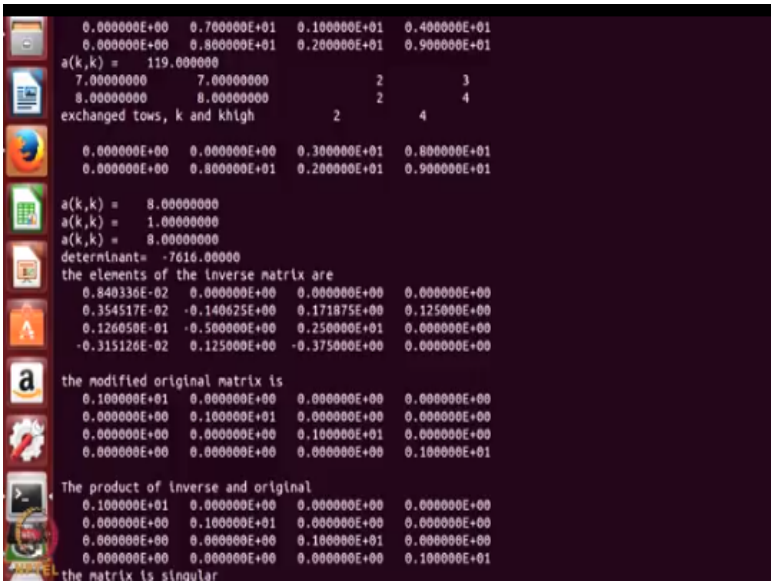
So, when I typed a.out it gave this 1 2 3 4 this output is just to indicate that I did my 1 st column 2 nd column 3 nd column and 4 th column. So my results are now output.dat. You see this line output.dat, it has created a new output.dat. How do I know that is the new output.dat this output.dat was created at 12:41 now let us do ls-l again when I do ls-l again.

(Refer Slide Time: 17:38)



See this ls-l just now I created a new output.dat by executing this a.out, okay. Since I executed this a.out I got a new look at this earlier one was 12:41 so the new one is at 15:41. I have just executed at this time now I want to edit the output.dat. vi.

(Refer Slide Time: 18:09)



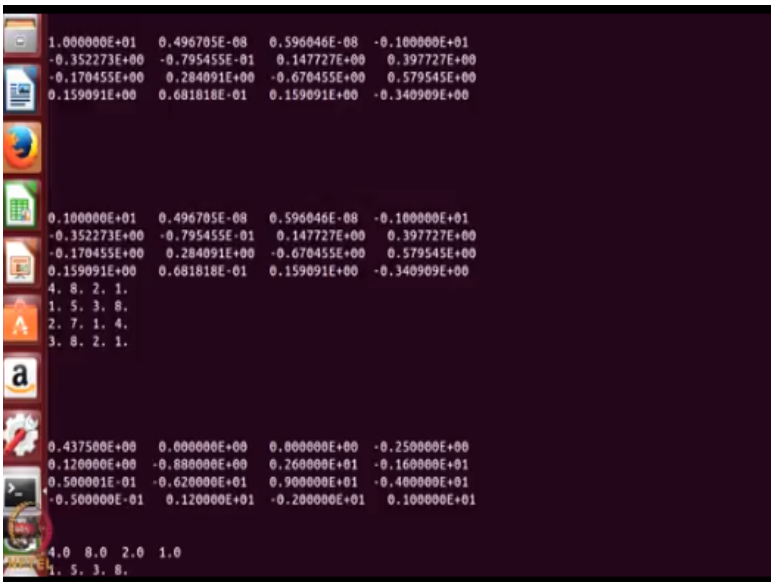
So, when I edit that what it did it read the original matrix on your screen you are not seeing all those things, but the original matrix is the first element is 119 2 0 0 3 0 3 8. And the third line is 0 7 1 4 you are seeing the third line here and the fourth line is 0 8 2 and 9. So this is my original matrix and the first akk was 119 okay and it tells you whether it exchanges rows or not. Whenever it exchanges rows it will tell you it has exchanged rows k and khigh okay.

Second time akk was 8 there was no need to exchange third time akk is 1 and fourth time akk is 8. So, these are my 4 operations my determinant now is the product of all the diagonal elements. What are my diagonal elements, first value is 119, second one is 8, third one is 1 and fourth one is 8, okay? So, determinant is the product of these values okay. Then what are the elements of the inverse?

It is writing now these are all the elements of the inverse which are obtained by taking the matrix b and repeatedly converting it into the inverse matrix. Then to make sure that the inverse is actually the inverse, what I will do is I multiply this inverse matrix by the original matrix, remember the original matrix was aold. So the product of the original and the inverse is written here, see this is the last particular 4/4 matrix, 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1.

So this identity matrix I obtained in two different ways. One by taking the inverse and multiply by the original matrix, second way by converting matrix A into the identity matrix by changing the columns and rows by the elementary operations that we have described in detail. So, before I go for that let us edit that input.dat so that you know what it is.

(Refer Slide Time: 20:48)

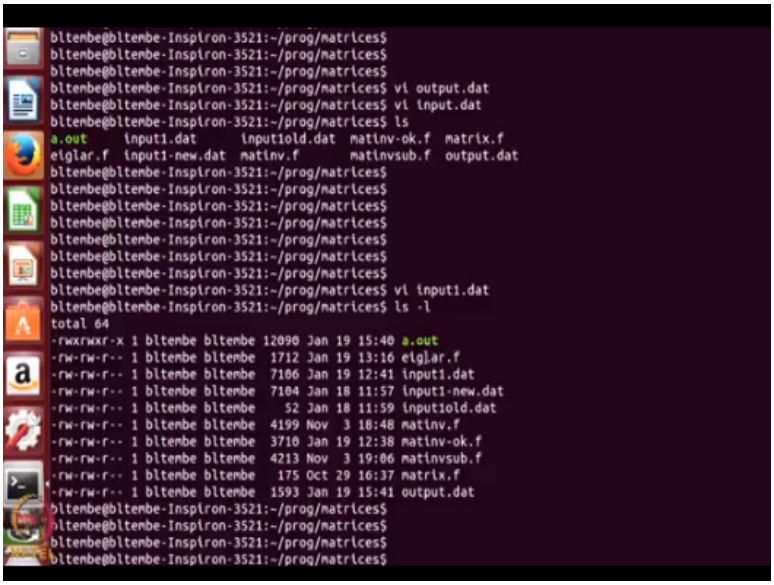


So vi, okay I am not seeing anything. So it is called input 1.dat so I will do let me show here, you see here these are all my files in my matrices subdirectory. The file is called input 1.dat. so when I say vi input.dat it gave me nothing because there was no file input.dat it gave me a blank file,

so I came out and now I will edit vi. So this is my matrix, since my screen and your screen are different this is my matrix.

What was my matrix 119 2 0 0 3 0 3 8 0 this was my original matrix, okay. And I used it to do all the calculations okay so since we have now found how to find the inverse of a matrix let me also execute the program for diagonalization. What was the program for diagonalization?

(Refer Slide Time: 22:06)

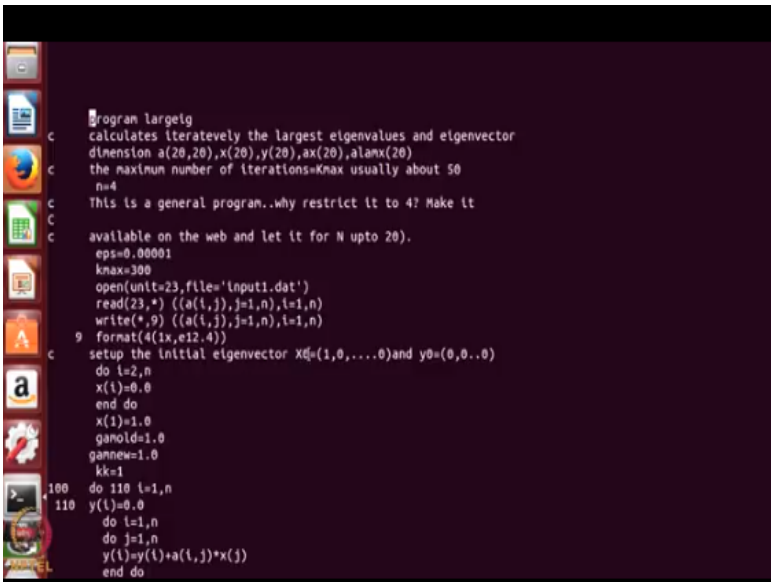


So, these are all my files in the matrices directory. The program for diagonalization was eiglar. f that was the program for diagonalizing matrices. Use that program to illustrate how you can get a diagonal matrix, okay. So what I shall do now is I shall compile that, how do I compile? G, so before I compile let us just see what that file is eiglar. f I am editing that file.

(Refer Slide Time: 22:06)

Okay so again there is nothing. So it is eiglarge. f remember what I did I tried to edit it as eiglar. g so there is no file called eiglar. g so it gave me an error, so I shall edit now vi eiglar. f,

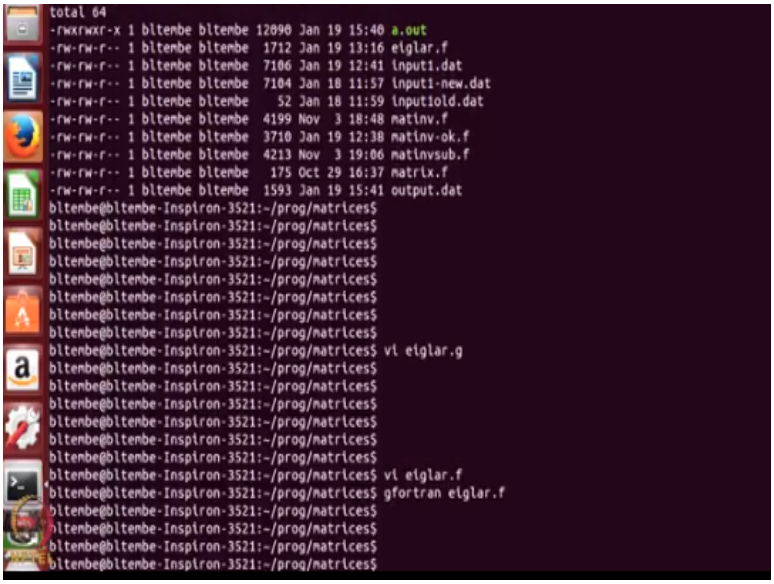
(Refer Slide Time: 23:11)



Okay so this is my file again since my view and your view is slightly different. So this is my program large eigenvalue. So, it calculates iteratively the largest eigenvalue, so I define exactly the same dimension as in the old case. I read from input 1.dat so read all the values of aij and now I will set up an initial eigenvector. My initial eigenvector is 1,0,0,0 and y,0,0 so this is setting my initial eigenvector.

Okay 1,0,0,0 and y is another 0. So what I do in this program is I keep on multiplying with matrix A that eigenvector X many times. So, that when I multiply by the largest value only the 1st eigenvalue will remain.

(Refer Slide Time: 24:04)



So now I will execute that, so I am going to compile this program again gfortran eiglar. f, okay. So when I did that gfortran eiglar. f it gave me no error message that means the program is executed properly now I shall execute that; how do I execute? ./a.out.

(Refer Slide Time: 24:53)

So, when I execute these are all the results of my execution. So let us see how it is executed. So when I typed ./a.out it first wrote my original matrix. And this first vector was 1 so the first

column tells me how many iterations I am doing in this program. So first time the eigenvalue was 1 the x column had 1 and all the rest was 0, second time I got my eigenvalue as 119.000 and this was my eigenvector.

When I iterated again that 119 became 119.0504231 okay 10-2 1.48 times 10-3 1.69 10-3 when I iterate again the next one is 119.05 and this is my eigenvalue eigenvector is 1 2.53*-2 1.56*-3 and I iterate again you see 6 th iteration it has not done the 7th iteration. Because the parameter I gave 10-5 was the error I mentioned. So, if my new eigenvector it does not change from the old one by more than 1 part in 10-5 it will stop the calculation.

1 st value was 1 second value was 2.53625 the next line is 2.53643, okay. So it is changing much less than the fourth digit. So, it converges at this point, so this is my eigenvector. So to now make sure that this is really the eigenvector what was my equation? AX=lambda X. What I am typing here is AX and lambda X values. What is AX I take matrix A multiply on the vector X that will be one way.

The other way is I just take the lambda and multiply the X. so this is my X in one case I multiplies lambda into that eigenvalue and in the other case I have taken A into that eigenvalue.so the top line is AX bottom line is lambda X this is a 4/4 case. You see that both AX and lambda X are identical to the 5th digit you see that 109.05072, 109.05072. These are exactly the same in this case it only differs 01968, 019678 only differs in the 5 th digit, okay.

So, this means I have a converge value. Now what we have seen, we have seen that you have a matrix inversion program as well as a matrix diagonalization program which just gets the largest eigenvalue. When we operated both these, we got the correct inverse as well as the largest eigenvalue. I would urge you to practice this in great detail and any questions you have you can send it to us we will reply to you.

And you have to practice all these programming because when you actually program there will be many errors in your program. So it is only when you master the art of correcting the errors you progress in programming. So, this concludes our session on matrices. Next time we will start

with fitting a curve to a given particular data points that is called curve fitting and that curve fitting requires matrix inversion.

Therefore, we did the matrix inversion first and then curve fitting. So, next lecture we will do curve fitting then we will do little bit of integration, differential equations, random numbers, okay and then we will go to molecular dynamics after that. So I conclude my lecture here. Thank you.