Lecture 11

Linear systems:

- Cholesky method
- QR decomposition

Eigensystems:

- Terminology
- Jacobi transformations
- QR transformation

Cholesky method:

For a symmetric positive definite matrix, one can do an LU decomposition in which $U=L^T$, or $U_{ij}=L_{ji}$.

One therefore has,

$$a_{ii} = \sum_{k=1}^{N} L_{ik}U_{ki} = \sum_{k=1}^{N} L_{ik}L_{ik} = \sum_{k=1}^{i} L^2{}_{ik} = \sum_{k=1}^{i-1} L^2{}_{ik} + L_{ii}$$

$$a_{ij} = \sum_{k=1}^{N} L_{ik}U_{kj} = \sum_{k=1}^{N} L_{ik}L_{jk} = \sum_{k=1}^{i} L_{ik}L_{jk} = \sum_{k=1}^{i-1} L_{ik}L_{jk} + L_{ii}L_{ji}$$

Or,

$$L_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} L_{ik}^2 \right)^{1/2}$$

$$L_{ji} = \frac{1}{L_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} L_{ik}L_{jk} \right) \qquad j = i+1, i+2, \ldots, N$$

If you look at these carefully, you'll see that the L's on the RHS that are needed are already computed. $O(N^3)$

The Cholesky decomposition is quite stable without pivoting. If it fails, it means that your matrix was not (within roundoff accuracy) positive definite. In fact, it can be used as a quick method to determine if a matrix is positive definite!

```
SUBROUTINE choldc(a,n,np,p)
INTEGER n,np
REAL a(np,np),p(n)
    Given a positive-definite symmetric matrix a(1:n,1:n), with physical dimension np, this
    routine constructs its Cholesky decomposition, A = L·L^T. On input, only the upper triangle
    of a need be given; it is not modified. The Cholesky factor L is returned in the lower triangle
    of a, except for its diagonal elements which are returned in p(1:n).
INTEGER i,j,k
REAL sum
do 13 i=1,n
    do 12 j=i,n
        sum=a(i,j)
        do 11 k=i-1,1,-1
            sum=sum-a(i,k)*a(j,k)
        enddo 11
        if(i.eq.j)then
            if(sum.le.0.)pause 'choldc failed'        a, with rounding errors, is not
            p(i)=sqrt(sum)                              positive definite.
        else
            a(j,i)=sum/p(i)
        endif
    enddo 12
enddo 13
return
END
```

QR decomposition:

$$A = QR$$

Where Q is orthogonal $Q^T.Q=1$ and R is upper triangular.

The solution of the system

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

is obtained by rewriting it like this, and then backward substitute.

$$\mathbf{R} \cdot \mathbf{x} = \mathbf{Q}^T \cdot \mathbf{b}$$

This decomposition actually exists for rectangular matrices as well.

The way to perform a QR decomposition is through a method we will discuss soon in the context of eigensystems called the Householder transformation. This transformation consists in multiplying a matrix times an orthogonal factor that can be chosen in such a way as to zero out all elements in a column of a matrix below a given one.

So we a perform Householder transformation that zeroes out all elements below the top leftmost one. Then we perform another zeroing out all elements below a_{22} and so on. Since the product of the Householder matrices is orthogonal, we end up with a formula,

$$H_1 H_2 ... H_N A = H \;\; A = R \Rightarrow A = QR$$

With $Q=H^T$.

Another nice feature of the QR decomposition is that there are many procedures for zeroing out components of matrices that operate through an orthogonal matrix. One we will discuss soon is the Jacobi rotation. This one can be used effectively when one requires repeated solution of systems of equations where the matrix A changes "a little" in the sense of,

$$\mathbf{A} \rightarrow \mathbf{A} + \mathbf{s} \otimes \mathbf{t}$$

$$\mathbf{A} = \mathbf{Q} \cdot \mathbf{R} \quad \rightarrow \quad \mathbf{A}' = \mathbf{Q}' \cdot \mathbf{R}' = \mathbf{Q} \cdot (\mathbf{R} + \mathbf{u} \otimes \mathbf{v})$$

$$\mathbf{t} = \mathbf{v} \quad \text{and either} \quad \mathbf{s} = \mathbf{Q} \cdot \mathbf{u} \quad \text{or} \quad \mathbf{u} = \mathbf{Q}^T \cdot \mathbf{s}$$

One then converts R' to upper triangular via Jacobi rotations, which get bundled up in the orthogonal matrix Q.

Eigensystems Terminology:

$$\mathbf{A} \cdot \mathbf{x} = \lambda \mathbf{x}$$

A is an NxN matrix, x is its eigenvector and lambda the eigenvalue associated with the eigenvector. Such system of equations only has solution if,

$$\det |\mathbf{A} - \lambda \mathbf{1}| = 0$$

This is an Nth order equation, so in principle there can be up to N different eigenvalues. Not used numerically in practice.

One can add x times a constant to both members of the equation, therefore "shifting" the value of the eigenvalue without changing the eigenvector. This is useful numerically. It also highlights that there is nothing special about a zero eigenvalue, any eigenvalue can be "shifted" to zero.

Symmetric: $A=A^T$

Hermitian: $A=A^\dagger=(A^T)^*$

Orthogonal: $A.A^T=A^T.A=1$

Unitary: $A^{-1}=A^\dagger$

Normal: $A.\ A^\dagger=A^\dagger.A$

Hermitian matrices have real eigenvalues. Real symmetric matrices therefore also have real eigenvalues.

Any matrix whose columns (or rows) are made of an orthonormal basis of vectors is orthogonal.

Normal matrices are important because if they have N distinct eigenvalues, their eigenvectors form a complete orthogonal basis. Even if they are not distinct they can be made orthogonal (Gram-Schmidt). The matrix of eigenvectors can therefore be made unitary.

Right eigenvectors: $\mathbf{A} \cdot \mathbf{x} = \lambda \mathbf{x}$

Left eigenvectors: $\mathbf{x} \cdot \mathbf{A} = \lambda \mathbf{x}$

The transpose of a right eigenvector of a given matrix is a left eigenvector of the transpose of that matrix . Eigenvalues are the same.

Let X_R be the matrix formed by the right eigenvectors and X_L the matrix of left eigenvectors. Then we have,

$$\mathbf{A} \cdot \mathbf{X}_R = \mathbf{X}_R \cdot \mathrm{diag}(\lambda_1 \ldots \lambda_N) \qquad \mathbf{X}_L \cdot \mathbf{A} = \mathrm{diag}(\lambda_1 \ldots \lambda_N) \cdot \mathbf{X}_L$$

If we left-multiply the left equation by X_L and right-multiply the right one by X_R we have,

$$(\mathbf{X}_L \cdot \mathbf{X}_R) \cdot \mathrm{diag}(\lambda_1 \ldots \lambda_N) = \mathrm{diag}(\lambda_1 \ldots \lambda_N) \cdot (\mathbf{X}_L \cdot \mathbf{X}_R)$$

That means that the matrix of product of eigenvectors commutes with the diagonal matrix of eigenvalues. This means it is diagonal. Therefore left eigenvectors are orthogonal to right ones with different eigenvalues. This is true even for non-normal matrices.

From the previous formulae, we have that,

$$\mathbf{X}_R^{-1} \cdot \mathbf{A} \cdot \mathbf{X}_R = \mathrm{diag}(\lambda_1 \ldots \lambda_N)$$

This is just a particular case of a similarity transformation

$$\mathbf{A} \quad \rightarrow \quad \mathbf{Z}^{-1} \cdot \mathbf{A} \cdot \mathbf{Z}$$

These transformations are important because they leave eigenvalues unchanged,

$$\begin{aligned} \det\left|\mathbf{Z}^{-1} \cdot \mathbf{A} \cdot \mathbf{Z} - \lambda\mathbf{1}\right| &= \det\left|\mathbf{Z}^{-1} \cdot (\mathbf{A} - \lambda\mathbf{1}) \cdot \mathbf{Z}\right| \\ &= \det|\mathbf{Z}|\ \det|\mathbf{A} - \lambda\mathbf{1}|\ \det\left|\mathbf{Z}^{-1}\right| \\ &= \det|\mathbf{A} - \lambda\mathbf{1}| \end{aligned}$$

The main strategy for solving eigensystems is to use similarity transformations to "nudge" matrices towards diagonal form.

There are two strategies to implement this idea for diagonalization. Most "canned" routines like EISPACK use a combination of both of them.

The first strategy is to use a finite number of similarity transformations designed to achieve a specific goal, for instance zeroing out a certain off-diagonal component. In general a finite sequence of these operations cannot diagonalize the matrix. One then either uses them to take the matrix to a simple form (e.g. tri-diagonal), or applies the operation repeated times until the off diagonal elements are small enough.

The second strategy is called "factorization methods". Suppose the matrix can be factorized into a right and left factor,

$$\mathbf{A} = \mathbf{F}_L \cdot \mathbf{F}_R \qquad \text{or equivalently} \qquad \mathbf{F}_L^{-1} \cdot \mathbf{A} = \mathbf{F}_R$$

Or, $\mathbf{F}_R \cdot \mathbf{F}_L = \mathbf{F}_L^{-1} \cdot \mathbf{A} \cdot \mathbf{F}_L$

We will come back to the QR method which uses this idea.

Jacobi transformations:

Are just plane rotations designed to zero out some element of the matrix of interest.

When one applies a second Jacobi transformation the element one managed to zero out before will not be zero anymore. However, successive applications of the transformation manage to make the off-diagonal elements smaller and smaller.

$$\mathbf{A}' = \mathbf{P}_{pq}^T \cdot \mathbf{A} \cdot \mathbf{P}_{pq}$$

$$\mathbf{P}_{pq} = \begin{bmatrix} 1 & & & & & & \\ & \cdots & & & & & \\ & & c & \cdots & s & & \\ & & \vdots & 1 & \vdots & & \\ & & -s & \cdots & c & & \\ & & & & & \cdots & \\ & & & & & & 1 \end{bmatrix} \quad \mathbf{A}' = \begin{bmatrix} & \cdots & a'_{1p} & \cdots & a'_{1q} & \cdots & \\ \vdots & & \vdots & & \vdots & & \vdots \\ a'_{p1} & \cdots & a'_{pp} & \cdots & a'_{pq} & \cdots & a'_{pn} \\ \vdots & & \vdots & & \vdots & & \vdots \\ a'_{q1} & \cdots & a'_{qp} & \cdots & a'_{qq} & \cdots & a'_{qn} \\ \vdots & & \vdots & & \vdots & & \vdots \\ & \cdots & a'_{np} & \cdots & a'_{nq} & \cdots & \end{bmatrix}$$

If one writes out explicitly the transformed matrix elements,

$$
\begin{aligned}
a'_{rp} &= ca_{rp} - sa_{rq} \\
a'_{rq} &= ca_{rq} + sa_{rp} \\
a'_{pp} &= c^2 a_{pp} + s^2 a_{qq} - 2sca_{pq} \\
a'_{qq} &= s^2 a_{pp} + c^2 a_{qq} + 2sca_{pq} \\
a'_{pq} &= (c^2 - s^2)a_{pq} + sc(a_{pp} - a_{qq})
\end{aligned}
\qquad r \neq p,\ r \neq q
$$

We now choose the angle of the rotation to zero out a'$_{pq}$

$$\theta \equiv \cot 2\phi \equiv \frac{c^2 - s^2}{2sc} = \frac{a_{qq} - a_{pp}}{2a_{pq}}$$

One can rearrange equations a bit, $\tau \equiv \dfrac{s}{1+c}$

$$a'_{pp} = a_{pp} - ta_{pq}$$

$$
\begin{aligned}
a'_{qq} &= a_{qq} + ta_{pq} \\
a'_{rp} &= a_{rp} - s(a_{rq} + \tau a_{rp}) \\
a'_{rq} &= a_{rq} + s(a_{rp} - \tau a_{rq})
\end{aligned}
$$

And then one can easily see that if one computes the sum of the squares of the off-diagonal components,

$$S = \sum_{r \neq s} |a_{rs}|^2$$

$$S' = S - 2|a_{pq}|^2$$

To see this notice (through a little of trig algebra) that $|a'_{rq}|^2 + |a'_{rp}|^2 = |a_{rq}|^2 + |a_{rp}|^2$

One eventually obtains a matrix that is therefore diagonal up to machine precision. Better yet, since one obtained it through a set of orthogonal transformations (and the product of orthogonal matrices is orthogonal),

$$\mathbf{D} = \mathbf{V}^T \cdot \mathbf{A} \cdot \mathbf{V} \qquad \mathbf{V} = \mathbf{P}_1 \cdot \mathbf{P}_2 \cdot \mathbf{P}_3 \cdots$$

The only thing left for practical implementation is to decide which element to rotate. Jacobi's original prescription was to sweep the upper triangular portion of the matrix and pick the largest element. For large matrices this is expensive. So another possibility is to go element by element. Numerical implementations also set to zero elements that are smaller than the diagonal by a certain amount and then test and skip such elements in further sweeps.

In practice, one needs about 6-10 sweeps, or 3-5n^2 rotations, each of about 4n operations, so one is looking to 12-20n^3 operations.

The QR and QL algorithms:

As we discussed before, any matrix can be decomposed into a product of an orthogonal and an upper (or lower) triangular matrix,

$$\mathbf{A} = \mathbf{Q} \cdot \mathbf{R}$$

Consider the matrix formed by inverting the factors, $\mathbf{A}' = \mathbf{R} \cdot \mathbf{Q}$

Since Q is orthogonal, we can solve for R in the first equation and we get,

$$\mathbf{A}' = \mathbf{Q}^T \cdot \mathbf{A} \cdot \mathbf{Q}$$

Similar considerations hold for a QL decomposition. The algorithm consists in forming a sequence,

$$\mathbf{A}_s = \mathbf{Q}_s \cdot \mathbf{L}_s$$

$$\mathbf{A}_{s+1} = \mathbf{L}_s \cdot \mathbf{Q}_s$$

A (long) theorem (see Bulirsch and Stoer) shows that A_s tends asymptotically to a triangular matrix. Therefore the eigenvalues are trivial.

The rate at which the off-diagonal elements converges to zero is

$$a_{ij}^{(s)} \sim \left(\frac{\lambda_i}{\lambda_j}\right)^s$$

To accelerate the convergence, one can use the technique of "shifting": the matrix $\mathbf{A} - k\mathbf{1}$ has eigenvalues λ_i-k. Therefore if one decomposes such matrix instead of A, one would have convergence,

$$\frac{\lambda_i - k_s}{\lambda_j - k_s}$$

Ideally, one would like to choose k_s close to the first eigenvalue, then one would quickly zero out the first row of the matrix, and so on. In practice, one does not know such eigenvalue. A good guess is to consider the 2x2 matrix formed by a_{11},a_{12},a_{21},a_{22} and compute its eigenvalues as a guidance. For subsequent eigenvalues one takes the corresponding 2x2 sub-matrix.

Summary

- Cholesky and QR decompositions can be used to solve systems of equations.
- Jacobi transformations can be used to zero out elements of matrices.
- QL and QR algorithms efficiently yield a triangular matrix, and therefore the eigenvalues.