UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL ATOMIC ENERGY AGENCY
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE: CENTRATOM TRIESTE

H4.SMR/994-5

SPRING COLLEGES IN COMPUTATIONAL PHYSICS

19 May - 27 June 1997

LINEAR ALGEBRA, MINIMIZATION, AND EIGENVALUE EQUATIONS

C. REBBI
Boston University
Department of Physics
590 Commonwealth Ave.
Boston, Massachusetts 01125
U.S.A.

Preliminary notes from a course on Computational Physics - copyright by Claudio Rebbi, 1995 - for distribution to participants only.

Related problems:

i) finding the zeros of several equations in several unknowns;

ii) minimizing a function of several variables;

iii) solving sets of linear equations.

i) $f_i(x_j) = 0 \qquad 1 \le i, j \le N$.

Near the zeros we can expand

$$f_i(x_j) = f_i(x_j^{(0)}) + \left.\frac{\partial f_i}{\partial x_j}\right|_{x_j = x_j^{(0)}} (x_j - x_j^{(0)}) + \dots$$

↗ implicit sum over j

$$\equiv \quad A_{ij} x_j - b_i + \dots \quad \equiv \quad Ax - b + \dots$$

↗ definition

↗ compact notation: x, b are vectors, A is a matrix

As in the algorithm of Newton + Raphson we take $x_i^{(0)}$ to be the first iterate and find a new zero by solving

$$Ax - b = 0,$$

i.e.

$$x^{(1)} = A^{-1} b.$$

Thus i) leads to iii).

ii) Minimization of $F(x_i)$.

At the minimum $f_i(x_j) \equiv \frac{\partial F}{\partial x_i}(x_j) = 0$.

(definition)

This is a set of N equations in N unknowns, so that ii) leads to i).

Vice versa, i) can be reduced to ii) by defining $\tilde{F}(x_j) = \sum_i f_i^2(x_j)$ (or, with suitable weights $w_i > 0$ $\tilde{\tilde{F}}(x_j) = \sum_i w_i f_i^2(x_j)$) and demanding that $\tilde{F}(x_j)$ is minimized (in this case $\min \tilde{F}(x_j) = 0$).

ii) → iii) In the neighborhood of the minimum the first derivatives $f_i(x_j) = \frac{\partial F}{\partial x_i}(x_j)$ can be expanded as

$$f_i(x_j) = f_i(x_j^{(0)}) + \left.\frac{\partial^2 F}{\partial x_i \partial x_j}\right|_{x=x^{(0)}} (x_j - x_j^{(0)}) + \cdots$$

(Hessian matrix)

$$\equiv A_{ij} x_j - b_i$$

and we can take as next iterate $x^{(1)} = A^{-1} b$.

Equivalently, we can think of expanding $F(x_i)$

directly :

$$F(x_i) = F(x_i^{(0)}) + f_i(x_j^{(0)})(x_i - x_i^{(0)}) +$$

$$+ \frac{1}{2} \left.\frac{\partial^2 F}{\partial x_i \partial x_j}\right|_{x=x^{(0)}} (x_i - x_i^{(0)})(x_j - x_j^{(0)}) +$$

$$+ \cdots$$

In the neighborhood of the minimum the terms of higher order will be small and $F(x_i)$ will be well approximated by the quadratic form given by the first three terms in the expansion above.

Using the same notation as in the previous page, we can express this quadratic form as

$$Q(x) = \frac{1}{2} A_{ij} x_i x_j - b_i x_i + c .$$

The minimum of Q is at x solving the equations

$$Ax - b = 0$$

i.e. $$x = A^{-1} b .$$

Then we take as new iterate, etc..

Note: for a well posed problem of minimization the Hessian matrix is positive definite, i.e.

$$x_i A_{ij} x_j > 0 \quad \text{for all } x_i \neq 0 .$$

In 1 dimension at a local minimum

$$f''(x^{(\infty)}) > 0$$

and, by continuity,
$f''(x) > 0$ also in a neighborhood of

$\Rightarrow$

The equivalent of the secant method can also be devised,

e.g., in 2 dimensions, i.e. we wish to solve

and
$$f_1(x_1, x_2) = 0$$
$$f_2(x_1, x_2) = 0 .$$

Let us calculate f_1 in 3 points P_1 P_2 P_3. We can then approximate $f_1(x_1, x_2)$ by a linear function

$$f_1(x_1, x_2) \approx a_{11} x_1 + a_{12} x_2 - b_1$$

which takes the same values as f_1 over $P_1 P_2 P_3$ (plane through 3 points).

Similarly we calculate also f_2 at $P_1 P_2 P_3$ and approximate

$$f_2(x_1, x_2) \approx a_{21} x_1 + a_{22} x_2 - b_1 .$$

As next iterate we use then the common zero of the

two planes, i.e. the solution of

$$Ax - b = 0 .$$

Let us denote this solution by P_4. We repeat then the procedure on the basis of P_2 P_3 P_4 etc...

ii) → ii) For a linear system of eqs.

$$Ax - b = 0$$

the solution minimizes the quadratic form

$$Q(x) = (x^T A^T - b^T)(Ax - b) =$$

↑ ↑
let us drop the T from x, b

$$= x A^T A x - b A x - x A^T b - b b$$

The fact that $Q(x)$ is minimized by the solution of $Ax - b = 0$ is obvious, and, in any case, we can also demand

$$\frac{\partial}{\partial x_i} Q(x) = 0 ,$$

which gives $$2A^T A x - 2A^T b = 0 .$$

For non-singular A this is equivalent to

$$Ax - b = 0 .$$

Notice that the minimization of $Q(x)$ is

equivalent to the minimization of

$$\tilde{Q}(x) = \frac{1}{2}\, x A^T A x - x A^T b.$$

If A is already symmetric and positive definit we can consider directly the quadratic form

$$Q = \frac{1}{2}\, x A x - x b$$

and minimize. Finding the minimum is equivalent to solving

$$Ax - b = 0.$$

$$=$$

Methods for solving a system of linear equations:

$$Ax = b.$$

The method of Gaussian elimination:

Let A be given by

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots \\ a_{21} & a_{22} & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix}$$

and let us denote by Γ_1 Γ_2 .. its rows, i.e.

$$\Gamma_1 \equiv a_{11}\ a_{12}\ a_{13}\ \cdots$$

$$\Gamma_2 \equiv a_{21}\ a_{22}\ a_{23}\ \cdots$$

Replacing r_2 with $r_2 - \frac{a_{21} r_1}{a_{11}}$ the matrix

becomes

$$A' = \begin{pmatrix} a_{11} & a_{12} & \cdots \\ 0 & a_{22}' & \cdots \\ & & \text{etc...} \end{pmatrix}$$

But this corresponds to

$$A \rightarrow A' = S_{12} A$$

with

$$S_{12} = \begin{pmatrix} 1 & 0 & 0 & \cdots \\ -\frac{a_{21}}{a_{11}} & 1 & 0 & \\ 0 & 0 & 1 & \\ \vdots & & & \end{pmatrix}$$

Obviously $Ax = b$ is equivalent

to $S_{12} A x = S_{12} b$.

With many similar replacements (Gaussian elimination) the system becomes

$$\hat{A} x = \hat{b}$$

with $\hat{A}$ upper triangular.

(Notice, $\hat{A}$ and A also have the same determinant, (Let $\det \hat{A} = \hat{a}_{11}\hat{a}_{22}\cdots\hat{a}_{NN}$, so that the method (of Gaussian elimination also provides a calculation of the determinant).

With $\hat{A}$ upper triangular

$\hat{A}x = \hat{b}$ is solved immediately by backward substitution

$\Uparrow$

(2) gives x_{N-1} $\quad \hat{a}_{N-1\,N-1}\,x_{N-1} + \hat{a}_{N-1\,N}\,x_N = b_{N-1}$

(1) gives x_N $\quad \hat{a}_{NN}\,x_N = b_N$

Pivoting is crucial: one should not use a_{11} directly, but rather should transpose rows and columns so as to bring into the a_{11} position the matrix element with the largest absolute value, etc..

Operations' count: $O(N)$ operations for every $\emptyset$ introduced in A' A'' — etc, hence $O(N^3)$ operations to bring A to upper triangular form; $O(N^2)$ operations for the backward substitution.

Solving $Ax = b$, finding A^{-1} and det A, (as well as the eigenvalues and eigenvectors of A) are typical operations for which there exist good library subroutines.

A typical library subroutine might be of the form

SOLVE (NV (A, B, M, N, M1, N1, DET, IERR)

and might produce the solution to the equation

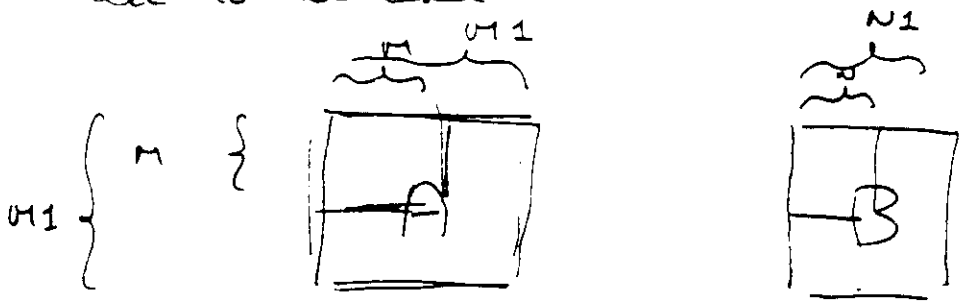
$$AX = B,$$

where A is an $M \times M$ matrix and X, B are $M \times N$ matrices (this is equivalent to solving $Ax = b$ for N different vectors b).

Upon return, the original elements of B might be overwritten with the solution X (note, if on entry $B = I$, then on return it will contain A^{-1}), DET will contain the determinant of A and IERR some error code, in case anything went wrong.

M1 and N1 might specify some physical dimensions of which A and B are imbedded. I.e., it might be that, for convenience, A is dimensioned as a

M1 × M1 matrix, B as a M1 × N1 matrix, of which only the upper M × M, M × N elements are to be used



Because of the way the matrices are stored, both M and M1, N and N1 must be supplied.

Special routines can be used for more efficient solutions in special cases, e.g. A could be symmetric etc. —

L U decomposition:

notice, the gaussian elimination procedure implicitly does implement an L U decomposition.

A is transformed by

$$A \to S_{12} A \to S_{23} S_{13} S_{12} A \to \cdots \text{ until}$$

$$S_{N-1\,N} \cdots S_{12} A = U \quad \text{upper triangular matrix}$$

Thus $A = S_{12}^{-1} \ldots S_{N-1N}^{-1} U$.

But $$S_{12}^{-1} = \begin{pmatrix} 1 & 0 & 0 & \\ \frac{a_{21}}{a_{11}} & 1 & 0 & \\ 0 & 0 & 1 & \\ & & & \ddots \end{pmatrix}$$ etc.

so that $A = LU$ with L lower triangular; moreover all diag. elements of L are 1.

The decomposition can be done directly:

we demand

$$\begin{pmatrix} 1 & 0 & 0 & \cdots \\ l_{21} & 1 & 0 & \cdots \\ l_{31} & l_{32} & 1 & \cdots \\ \cdots & \cdots & \cdots & \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} & \\ 0 & u_{22} & u_{23} & \\ 0 & 0 & u_{33} & \\ & & & \ddots \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots \\ a_{21} & a_{22} & a_{23} & \cdots \\ a_{31} & a_{32} & a_{33} & \cdots \\ \cdots & \cdots & \cdots & \end{pmatrix}$$

The multiplication gives

$$\begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots \\ l_{21}u_{11} & l_{21}u_{12}+u_{22} & l_{21}u_{13}+u_{23} & \cdots \\ l_{31}u_{11} & l_{31}u_{12}+l_{32}u_{22} & l_{31}u_{13}+l_{32}u_{23}+u_{33} & \text{etc.} \\ & & & \end{pmatrix}$$

Equating the first row to $a_{11}\ a_{12}\ a_{13}$ --

immediately fixes $u_{11}\ u_{12}\ u_{13}$ --

The second row fixes l_{21} first, then $u_{22}\ u_{23}$ --

The third row fixes l_{31} and l_{32} first, then $u_{33}\ a_{34}$ --

etc.

With $A = LU$ the linear system of equations $Ax = b$ can be recast into

$$Ux = y$$
$$Ly = b \qquad (Ly = LUx = b)$$

$Ly = b$ is now solved immediately by forward substitution and $Ux = y$ is solved by backward substitution.

One should rearrange rows and columns so that u_{11} is the largest element, in abs. value, u_{22} the largest in the remaining $(N-1)\times(N-1)$ square etc..

This does not require actual rearrangements, but simply a suitable indexing. The arrays IF(N) and IC(N) will contain the indices of the rows and columns in the order in which they would be rearranged if we had actually performed the substitutions.

For example, write

$$\begin{pmatrix} 4 & 9 & 2 \\ 2 & 8 & 10 \\ 1 & 7 & 3 \end{pmatrix}$$

ir and ic would contain ir = (2 1 3) ic = (3 2 1)

Because 10 (r = 2 c = 3) is the largest element,

9 (r = 1 c = 2) is the largest in the remaining

square $\left(\begin{array}{cc|c} 4 & 9 & \\ \hline 1 & 7 & \end{array}\right)$ etc..

We then use an indirect addressing of rows and columns

```
DO ia = 1, N
DO ja = 1, N
   ib = ir(ia)
   jb = ic(ja)
   use ib jb as indices
```

- - -

An inverting algorithm for symmetric matrices.

Given a symmetric M find M^{-1}.

Write M as

$$M = \begin{pmatrix} A & b \\ b^T & c \end{pmatrix}$$

M = $N \times N$ matrix $A = (N-1)\times(N-1)$ $b = (N-1)\times 1$ c = number

$$M = \begin{pmatrix} \boxed{A} & \boxed{b} \\ \boxed{b^T} & \boxed{c} \end{pmatrix}$$

Write M^{-1} in the form $\begin{pmatrix} X & y \\ y^T & z \end{pmatrix}$,

and solve

$$\begin{pmatrix} X & y \\ y^T & z \end{pmatrix} \begin{pmatrix} A & b \\ b^T & c \end{pmatrix} = \begin{pmatrix} \mathbb{I} & 0 \\ 0 & 1 \end{pmatrix}.$$

This gives

1) $X A + y b^T = \mathbb{I}$,

2) $X b + c y = 0$,

3) $y^T A + z b^T = 0$, or $A y + z b = 0$

4) $y^T b + z c = 1$ $\quad (\text{or } \vec{y} \cdot \vec{b} + z c = 0)$.

3) gives $y = - z A^{-1} b$.

Write this 1) becomes $X A - z A^{-1} b b^T = \mathbb{I}$,

which gives $X = \mathbb{A}^{-1} + z (A^{-1} b)(b^T A^{-1}) =$

$= \mathbb{A}^{-1} + \frac{y y^T}{z}$.

4) fixes z , since it gives $- z b^T A^{-1} b + z c = 1$

i.e. $z = \frac{1}{c - b^T A^{-1} b}$.

2) is also solved : $X b + c y = \left(\mathbb{A}^{-1} + \frac{y y^T}{z} \right) b + c y =$

$= A^{-1} b - y (b^T A^{-1} b) + c y =$

$= - \frac{y}{z} - y (b^T A^{-1} b) + c y =$

$= - (c - b^T A^{-1} b) y + (c - b^T A^{-1} b) y = 0$.

Thus, given $M = \begin{pmatrix} A & b \\ b^T & c \end{pmatrix}$ if we know A^{-1} we can calculate

$$M^{-1} = \begin{pmatrix} A^{-1} + \frac{yy^T}{z} & y \\ y^T & z \end{pmatrix},$$

with

$$v = A^{-1}b$$

$$z = \frac{1}{c - b^T v}$$

$$y = -zv .$$

Notice that we do not need to know A.

Thus we can invert M by induction, working out from the inverse of its first element m_{11}

$$\begin{pmatrix} m_{11} & m_{12} & \cdots \\ m_{21}=m_{12} & m_{22} & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix} \Rightarrow \begin{pmatrix} m_{11}^{-1} & m_{12} & \cdots \\ m_{12} & m_{22} & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix} \Rightarrow$$

$$= \begin{pmatrix} \begin{vmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{vmatrix}^{-1} & m_{13} & \cdots \\ m_{13} & m_{23} & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix} \Rightarrow \text{etc.}$$

For example, write a 2×2 matrix

$$1 = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \quad \text{we get} \rightarrow \begin{pmatrix} a^{-1} & b \\ b & c \end{pmatrix} \quad \text{and}$$

$$v = a^{-1}b \qquad z = \frac{1}{c - a^{-1}b^2} \qquad y = -\frac{a^{-1}b}{c - a^{-1}b^2}$$

$$z = \frac{a}{ac - b^2} \qquad y = -\frac{b}{ac - b^2}$$

$$M^{-1} = \begin{pmatrix} a^{-1} + \dfrac{b^2 \cancel{(ac - b^2)}}{(ac-b^2)^{\cancel{2}}\, a} & -\dfrac{b}{ac-b^2} \\ -\dfrac{b}{ac-b^2} & \dfrac{a}{ac-b^2} \end{pmatrix} =$$

$$= \begin{pmatrix} \dfrac{1}{a} \quad \dfrac{ac - b^2 + b^2}{ac - b^2} = \dfrac{c}{ac-b^2} & -\dfrac{b}{ac-b^2} \\ -\dfrac{b}{ac-b^2} & \dfrac{a}{ac-b^2} \end{pmatrix}$$

The analytic calculation looks rather complicated, but the iteration procedure is very simple.

With very sparse matrices, iterative techniques for the solution of a system of linear eqs.

$$Ax = b$$

may be more efficient than the methods illustrated above.

If A is symmetric and positive definite, solving the eq. $Ax=b$ is equivalent to finding the minimum of

$$Q(x) \equiv \frac{1}{2} x Ax - bx .$$

If A is not symmetric and positive definite we may replace the original system of eqs. with

$$A^T A x = A^T b ,$$

or with $A^+ A x = A^+ b$ (A^+ = hermitian conjugate of A)

if A is complex.

$M = A^T A$ or $M = A^+ A$ is then

symmetric and positive definite (or hermitian and p.d.).

•Let us study then the problem of minimizing

$$Q(x) = \frac{1}{2} x Ax - bx .$$

Search along the steepest descent.

Let us start from a given x_0. We can calculate

$$g_0 = -\operatorname{grad} Q\Big|_{x=x_0} = -Ax_0 + b.$$

We may now replace x_0 with

$$x_1 = x_0 + \beta g_0,$$

where β is a number which we will adjust so that the minimum of Q along the line starting at x_0 and directed along g_0 occurs at x_1.

Later we will have to move along a direction $h \neq g$, so let us say that we move along the line

$$x_1 = x_0 + \beta h_0,$$

with $h_0 = g_0$ to begin with.

Let us find β.

at x_1 the negative gradient will be

$$g_1 = -Ax_1 + b = -Ax_0 - \beta A h_0 + b = g_0 - \beta A h_0.$$

But, if Q is minimized along the line, then g_1 must be orthogonal to the direction of the line, i.e. h_0:

$$g_1 h_0 = 0.$$

This fixes β:

$$g_1 \cdot h_0 = 0 \quad \Rightarrow \quad g_0 \cdot h_0 - \beta\, h_0 A h_0 = 0$$

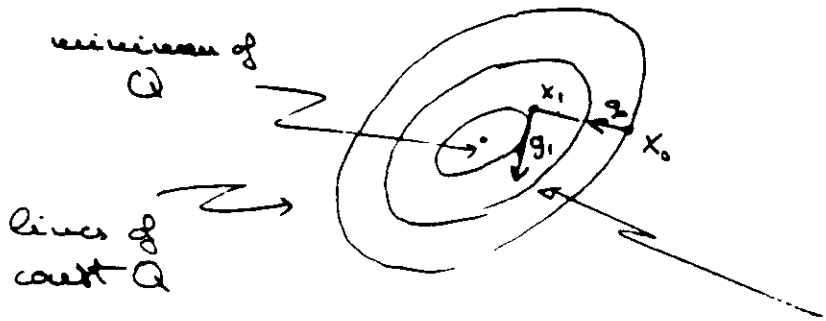
i.e.

$$\beta = \frac{g_0 \cdot h_0}{h_0 A h_0}.$$

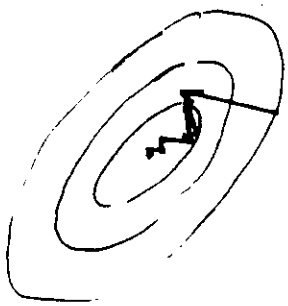
Do we now move in direction g_1?

No, g_1 is orthogonal to the original g_0 and the direction of steepest descent (i.e. g_0) is not the best.

See it in 2 dimensions:



g_1 is orthogonal to g_0 and clearly does not represent the best direction to approach the minimum



If we kept on going along the steepest descent we would approach the solution along a path like the one on the left, with very poor convergence indeed.

Rather, we move along a direction h_1 such that the new gradient is orthogonal to h_0 <u>after</u> the move.

Find the new direction of minimization:

set $h_1 = g_1 + \alpha_0 h_0$ ← (correction term).

We will move along h_1 to

$$x_2 = x_1 + \beta_1 h_1$$

and, at the minimum, the new gradient will be

$$g_2 = g_1 - \beta_1 A h_1.$$

β_1 is fixed by $g_2 h_1 = 0$, i.e.

$$\beta_1 = \frac{h_1 g_1}{h_1 A h_1}.$$

If we also demand

$$g_2 h_0 = 0$$

this gives

$$0 = g_1 h_0 - \beta_1 h_0 A h_1$$

$g_1 h_0 = 0$ from previous step

i.e. $$h_0 A h_1 = 0.$$

This fixes h_1, indeed we get

$$h_0 A g_1 + \alpha_0 h_0 A h_0 = 0$$

or
$$\alpha_0 = - \frac{g_1 A h_0}{h_0 A h_0} .$$

These conditions will be maintained throughout all iterations, namely

From the given h_i g_i we calculate the next minimum along h_i

$$x_{i+1} = x_i + \beta_i h_i$$

by demanding $g_{i+1} h_i = 0$,

i.e., since $g_{i+1} = g_i - \beta_i A h_i$

$$\beta_i = \frac{h_i g_i}{h_i A h_i} .$$

The previous construction of h_i, determined by the condition

$$h_{i-1} A h_i = 0$$

insures that $g_{i+1} h_{i-1} = 0$ as well (since $g_i h_{i-1} = 0$).

We will then define a new direction of minimization

$$h_{i+1} = g_{i+1} + \alpha_i h_i$$

such that $h_{i+1} A h_i = 0$, which gives

$$\alpha_i = - \frac{g_{i+1} A h_i}{h_i A h_i} .$$

This algorithm is referred to as the method of conjugate gradients.

By construction, it gives

A) $g_i h_j = 0$ for $j = i-1$, $j = i-2$;

B) $h_i A h_j = 0$ for $j = i-1$.

It also gives

C) $g_i g_j = 0$ for $j = i-1$,

since from $h_j = g_j + \alpha_{j-1} h_{j-1}$ it follows that g_j is a linear combination of h_j and h_{j-1} and $g_i h_{i-1} = g_i h_{i-2} = 0$.

But one can prove that eqs. A) B) C) are actually true for all $j < i$.

By induction: assume it proved up to i, prove it then for $i+1$.

A) $g_{i+1} h_j = 0$?

this is true by construction for $j = i$, for $j < i$ we use

$$g_{i+1} = g_i - \beta_i A h_i$$

and $g_i h_j = 0$, $h_j A h_i = 0$ which are both true for $j < i$ by the inductive hypothesis.

C) $g_{i+1} g_j = 0$?

From $h_j = g_j + \alpha_{j-1} h_{j-1}$ we get

$g_{i+1} g_j = g_{i+1} (h_j - \alpha_{j-1} h_{j-1}) = 0$ because A) has now been proven true up to $i+1$.

B) $h_{i+1} A h_j = 0$?

This is true by construction for $j = i$.

For $j < i$ we use $h_{i+1} = g_{i+1} + \alpha_i h_i$ and the identity to prove becomes

$$(g_{i+1} + \alpha_i h_i) A h_j = 0 \quad \text{for} \quad j < i .$$

But $h_i A h_j = 0$ for $j < i$ because of the inductive hypothesis; so we must prove

$$g_{i+1} A h_j = 0 \quad \text{for} \quad j < i .$$

From $g_{j+1} = g_j - \beta_j A h_j$ we get

$$g_{i+1}(A h_j) = g_{i+1} \frac{g_j - g_{j+1}}{\beta_j}$$

and, with $j < i$, $j+1 \le i$ the r.h.s. is 0 on account of C.

Note that the orthogonality and conjugate orthogonality relations

$$g_i g_j = 0 \qquad i \neq j$$
$$h_i A h_j = 0 \qquad i \neq j$$

imply that, if the dimensionality of the race is n, the algorithm must terminate in at most n steps (i.e., starting from g_0 and h_0, g_n and h_n must turn out to vanish, so that x_n is the required minimum).

In practice, the algorithm frequently converges to a high degree of accuracy in a number of steps much smaller than n.

From A) B) and C) more equally serviceable relations follow, e.g.

$g_i = h_i - \alpha_{i-1} h_{i-1}$ gives $h_j A g_i = 0$ for $j > i$;

$A h_i = \dfrac{g_i - g_{i+1}}{\beta_i}$ gives $g_j A h_i = 0$ for $j > i+1$;

$$\beta_i = \frac{h_i g_i}{h_i A h_i} = \frac{(g_i + \alpha_{i-1} h_{i-1}) g_i}{(g_i + \alpha_{i-1} h_{i-1}) A h_i} = \frac{g_i^2}{g_i A h_i},$$

and, especially,

$$\alpha_i = -\frac{g_{i+1} A h_i}{h_i A h_i} = \frac{g_{i+1}(g_{i+1}-g_i)}{\beta_i (h_i A h_i)} = \left(\text{with } \beta_i = \frac{h_i g_i}{h_i A h_i}\right)$$

$$= \frac{g_{i+1}(g_{i+1}-g_i)}{h_i g_i} = \frac{g_{i+1}(g_{i+1}-g_i)}{(g_i + \alpha_{i-1} h_{i-1}) g_i} = (\text{because of } A) =$$

$$= \frac{g_{i+1}(g_{i+1}-g_i)}{g_i^2} = (\text{because of } c) = \frac{g_{i+1}^2}{g_i^2}$$

with no reference to the matrix A.

In this form the algorithm generalizes to the non-linear case.

Let $F(x)$ be any function

we start from x_0 and calculate $g_0 = -\text{grad} F\Big|_{x_0}$

We select a direction h_0, which the first time will coincide with g_0

$$h_0 = g_0.$$

We move along the line

$$x = x_0 + \beta h_0$$

and we look for the minimum of F along this line

*)
Let the new minimum be at x_i (now, of course, $i=1$, but by using a generic index i we can formulate the basic iterative cycle). We calculate the (negative) new gradient $g_i = -\text{grad } F\big|_{x_i}$

and the parameter

$$\alpha_{i-1} = \frac{g_i^2}{g_{i-1}^2}$$

With this we define the direction for the new search

$$h_i = g_i + \alpha_{i-1} h_{i-1}$$

We find the minimum along this line and go to step *).

To the extent that $F(x)$ is well approximated by a quadratic form $Q(x)$ this procedure will converge to the minimum of F. Convergence is not guaranteed, but, by construction, the values of the $F(x_i)$ will form a non-increasing (in general decreasing) sequence. If the sequence does not converge to the required degree of approximation to the minimum, one can always restart the algorithm from the final value found for x. This will typically be closer to the minimum and the approximation of F with a quadratic form will be better.
In very many cases the algorithm works in a quite satisfactory manner.

Calculation of eigenvalues and eigenvectors of a symmetric (a hermitian) matrix.

1) bring the matrix to tridiagonal form

$$\hat{M} = \begin{pmatrix} a_1 & b_1 & 0 & 0 \\ b_1 & a_2 & b_2 & 0 \\ 0 & b_2 & a_3 & \ddots \\ 0 & 0 & \ddots & \end{pmatrix},$$

2) calculate the determinant

$$\det(\hat{M} - \lambda \hat{I}),$$

3) find the zeros (eigenvalues),

4) calculate the eigenvectors.

Reduction to tridiagonal form: Householder method

$$M = \begin{pmatrix} a & \vec{v}^T \\ \vec{v} & A \end{pmatrix}$$

dim $M = n$

$a = M_{11}$

$\vec{v} = n-1$ component vector

$A = (n-1) \times (n-1)$ matrix.

Consider the matrix

$$U = \begin{pmatrix} 1 & \vec{0}^T \\ \vec{0} & P \end{pmatrix}$$

with $P = \mathbb{I} - 2\vec{u}\vec{u}^T$ and $\vec{u}^T\vec{u} = 1$.

U is symmetric and orthogonal. The symmetry is obvious. Then

$$P^T P = P^2 = (\mathbb{I} - 2\vec{u}\vec{u}^T)(\mathbb{I} - 2\vec{u}\vec{u}^T).$$

$$= \mathbb{I} - 4\vec{u}\vec{u}^T + 4\vec{u}(\vec{u}^T\vec{u})\vec{u}^T = \mathbb{I},$$
$$\qquad\qquad = 1$$

so that U is indeed orthogonal.

The similarity transformation

$$M \to M' = U^T M U = U M U \text{ gives}$$

$$M' = \begin{pmatrix} a & (P\vec{v})^T \\ (P\vec{v}) & PAP \end{pmatrix}.$$

We fix vector $\vec{u}$ so that $P\vec{v} = \vec{w}$, where $\vec{w}$ is of the form $(w, 0, 0, 0 \cdots 0)$.

Then M' will be of the form

$$\begin{pmatrix} m_{11} & m_{12} & 0 & 0 & \cdots \\ m_{21} & & & & \\ 0 & & & & \\ 0 & & & & \\ \vdots & & & & \end{pmatrix} \qquad \begin{matrix} m_{22} \\ \\ \\ \\ \\ M'' \end{matrix}$$

(with $m_{11} = a$, $m_{12} = m_{21} = \omega$)

The method can then be applied to the $n-1 \times n-1$ matrix M'' etc.. and in $n-2$ steps the whole matrix will be brought to tridiagonal form.

We must still determine $\vec{u}$. We have

$$\vec{w} = P\vec{v} = \vec{v} - 2\vec{u}(\vec{u}^T\vec{v}). \qquad *)$$

The orthogonality of P (or a direct calculation from the above equation*)) gives $\vec{w}^T\vec{w} = \vec{v}^T\vec{v}$, hence

$$w = \pm|\vec{v}|.$$ Moreover, we also have (from *)

$$\vec{v}^T\vec{w} = \vec{v}^T\vec{v} - 2(\vec{v}^T\vec{u})(\vec{u}^T\vec{v}),$$
$$\pm v_1|\vec{v}| = |\vec{v}|^2 - 2(\vec{u}^T\vec{v})^2.$$

This allows us to solve for u^Tv. We choose the sign

that makes the r.h.s. the largest and find $u^T \cdot v$ from

$$(u^T \cdot v)^2 = \frac{|\vec{v}|^2 \mp v_1 |\vec{v}|}{2}$$

We can now finally solve for $\vec{u}$

$$\vec{u} = \frac{\vec{v} - \vec{\omega}}{2(\vec{u}^T \cdot \vec{v})} = \frac{1}{\sqrt{2(|\vec{v}|^2 \mp v_1 |\vec{v}|)}} (v_1 \mp |\vec{v}|, v_2, v_3 \dots) .$$

Assume now that M is tridiagonal.
We must calculate

$$\det (M - \lambda \mathbb{I}) .$$

$M - \lambda \mathbb{I}$ is also tridiagonal.

Consider then a tridiagonal symmetric matrix $T \equiv T_n$. Let $T_1, T_2, \dots T_n$ be the matrices formed by the first $1, 2, \dots n$ rows and columns:

T_3 { T_2 { $T_1 = t_{11}$ t_{12} 0 / t_{12} t_{22} t_{23} / 0 t_{23} t_{33} / ... / ... t_{nn} }.

We have $\mathrm{Det}\, T_i = (\mathrm{Det}\, T_{i-1})\, t_{ii} - (\mathrm{Det}\, T_{i-2})\, t_{i\,i-1}^2$

With $\mathrm{Det}\, T_1 = t_{11}$, $\mathrm{Det}\, T_0 = 1$ this allows us to calculate $\mathrm{Det}\, T_n$ by iteration. The procedure is very fast, so $\mathrm{Det}\,(M - \lambda \mathbb{I})$ can be calculated quickly for any given value of λ, and this can be used in a program that looks for the zeros of the determinant, i.e. the eigenvalues.

Once the eigenvalues are found, the eigenvectors can be determined by solving $n-1$ among the n eqs

$$M\vec{v} = \lambda \vec{v},$$

considering $n-1$ components of $\vec{v}$ as unknowns and one component as given.

The eigenvectors $\vec{U}$ corresponding to a definite eigenvalue λ can also be found by solving the equations

$$(M - \lambda \underline{\underline{\Pi}} + \epsilon \underline{\underline{\Pi}})\, \vec{U} = \vec{U}_0 ,$$

where $\vec{U}_0$ is an arbitrary vector (not orthogonal to $\vec{U}_\lambda$) and ϵ is a small number. The solution of this equation enhances the component of $\vec{U}_0$ along the direction of the eigenvector $\vec{U}_\lambda$.
Indeed, the solution

$$\vec{U} = (M - \lambda \underline{\underline{\Pi}} + \epsilon \underline{\underline{\Pi}})^{-1} \vec{U}_0 ,$$

in the basis formed by the eigenvectors themselves takes the form

$$U_{\lambda'} = \frac{U_{0\lambda'}}{\lambda' - \lambda + \epsilon}$$

so that, for very small ϵ, $U_\lambda = \frac{U_{0\lambda}}{\epsilon}$

dominates over all the others $U_{\lambda'}$.
The above procedure can be repeated a few times to enhance the component along $\vec{U}_\lambda$ even more.

Reduction to tridiagonal form: the Lanczos algorithm.

Let us call the matrix to be brought to tridiagonal form H. It is a real symmetric or complex hermitian matrix.
We construct a basis of vectors $|\psi_1\rangle$ $|\psi_2\rangle$ ... such that H is tridiagonal in that basis.
(It is convenient to use Dirac's notation:
$\vec{\psi}_1 \equiv |\psi_1\rangle$, $\vec{\psi}_1^{\,+} \equiv \langle\psi_1|$, $\vec{\psi}_1^{\,+} \cdot H\vec{\psi}_2 \equiv \langle\psi_1|H|\psi_2\rangle$ etc.)

We choose $|\psi_1\rangle$ arbitrarily (or by some motivated guess) normalized so that $\langle\psi_1|\psi_1\rangle = 1$.

$|\psi_1\rangle$ and $H|\psi_1\rangle$ in general span a 2 dimensional subspace (otherwise, if $H|\psi_1\rangle = c|\psi_1\rangle$, $|\psi_1\rangle$ is an eigenvector, we have solved part of our problem, and can proceed to another guess $|\psi_2\rangle$).

We define $|\psi_2\rangle$ as a linear combination of $H|\psi_1\rangle$ and $|\psi_1\rangle$ such that $|\psi_2\rangle$ is also normalized to 1 and that it is orthogonal to $|\psi_1\rangle$.
We write

→

$$|\psi_2\rangle = c_2 \left(H|\psi_1\rangle - a_{11}|\psi_1\rangle \right).$$

Then $\langle \psi_1 | \psi_2 \rangle = 0$ gives $a_{11} = \langle \psi_1 | H | \psi_1 \rangle$ and c_2 can be easily determined

$$\left(c_2 = \left[\langle \psi_1 | H H | \psi_1 \rangle - a_{11}^2 \right]^{-\frac{1}{2}} \right).$$

We now determine $|\psi_3\rangle$ as a linear combination of $H|\psi_2\rangle$ $|\psi_2\rangle$ and $|\psi_1\rangle$ such that $\langle \psi_3 | \psi_2 \rangle = 0$, $\langle \psi_3 | \psi_1 \rangle = 0$ and $\langle \psi_3 | \psi_3 \rangle = 1$:

$$|\psi_3\rangle = c_3 \left(H|\psi_2\rangle - a_{22}|\psi_2\rangle - a_{12}|\psi_1\rangle \right).$$

Our conditions give $a_{22} = \langle \psi_2 | H | \psi_2 \rangle$

$a_{12} = \langle \psi_1 | H | \psi_2 \rangle$ and c_3 is again immediately found.

In general, we construct $|\psi_{n+1}\rangle$ from $|\psi_n\rangle$ and $|\psi_{n-1}\rangle$ as

$$|\psi_{n+1}\rangle = c_{n+1} \left(H|\psi_n\rangle - a_{nn}|\psi_n\rangle - a_{n-1\,n}|\psi_{n-1}\rangle \right)$$

in such a way that

$$\langle \psi_{n+1} | \psi_n \rangle = \langle \psi_{n+1} | \psi_{n-1} \rangle = 0 \,, \quad \langle \psi_{n+1} | \psi_{n+1} \rangle = 1.$$

By construction, every $|\psi_n\rangle$ is orthogonal to $|\psi_{n-1}\rangle$ and $|\psi_{n-2}\rangle$. We can prove however that $|\psi_n\rangle$ is orthogonal to all $|\psi_i\rangle$ with $i < n$.

We proceed by induction. We assume that the property holds true up to $|\psi_n\rangle$ and prove it true for $|\psi_{n+1}\rangle$. We only need to prove that

$$\langle \psi_i | \psi_{n+1} \rangle = 0 \quad \text{for} \quad i < n-1,$$

since for $i = n-1$ and $i = n$ the equality is satisfied by construction.
With $i < n-1$ we have $\langle \psi_i | \psi_n \rangle = \langle \psi_i | \psi_{n-1} \rangle = 0$ by the induction hypothesis, so that

$$\langle \psi_i | \psi_{n+1} \rangle = \langle \psi_i | \{ c_{n+1} (H|\psi_n\rangle - a_{nn}|\psi_n\rangle - a_{n-1,n}|\psi_{n-1}\rangle) \}$$
$$= c_{n+1} \langle \psi_i | H | \psi_n \rangle .$$

But from $|\psi_{i+1}\rangle = c_{i+1}(H|\psi_i\rangle - a_{ii}|\psi_i\rangle - a_{i-1,i}|\psi_{i-1}\rangle)$ it follows that $H|\psi_i\rangle$ is a linear combination of $|\psi_{i+1}\rangle$, $|\psi_i\rangle$, $|\psi_{i-1}\rangle$. Since $i < n-1$, all these vectors are orthogonal to $|\psi_n\rangle$ by the induction hypothesis. It follows that

$$\langle \psi_i | H | \psi_n \rangle = 0 \quad \text{and} \quad \langle \psi_i | \psi_{n+1} \rangle = 0 .$$

Q.E.D.

Then, by the Lanczos algorithm, we can construct a basis $|\psi_1\rangle$, $|\psi_2\rangle$ --- of orthonormal vectors. But from the relation

$$|\psi_{n+1}\rangle = C_{n+1}\left(H|\psi_n\rangle - a_{nn}|\psi_n\rangle - a_{n-1,n}|\psi_{n-1}\rangle\right),$$

i.e.

$$H|\psi_n\rangle = a_{nn}|\psi_n\rangle + a_{n-1,n}|\psi_{n-1}\rangle - \frac{|\psi_{n+1}\rangle}{C_{n+1}},$$

it follows that H has matrix elements only between $|\psi_n\rangle$ and $|\psi_n\rangle$, $|\psi_n\rangle$ and $|\psi_{n-1}\rangle$ and $|\psi_n\rangle$ and $|\psi_{n+1}\rangle$. Indeed,

$$\langle\psi_n|H|\psi_n\rangle = a_{nn}$$

and

$$\langle\psi_n|H|\psi_{n-1}\rangle \left(= \langle\psi_{n-1}|H|\psi_n\rangle^*\right) = a_{n-1,n}.$$

Then the Lanczos algorithm brings H into tridiagonal form.

H' can be diagonalized following the procedure described before.

Notice:

- if only the eigenvalues are sought, it is not necessary to keep the vectors $|\psi_{u-2}\rangle \ldots |\psi_1\rangle$ in memory. Only the current $|\psi_u\rangle$ and $|\psi_{u-1}\rangle$ are needed to calculate $|\psi_{u+1}\rangle$ and only the coefficients a_{uu} and $a_{u-1\,u}$ must be stored. This can be particularly useful when dealing with very large, but very sparse matrices.

- the construction of the $|\psi_u\rangle$ can be truncated, the lowest eigenvectors (in the space spanned by $|\psi_1\rangle \ldots |\psi_{n\,max}\rangle$) can be found, and the procedure can be repeated starting from the lowest eigenvecta. Generally this converges to the true lowest eigenvectors of H. This allows one to find the lowest eigenvalues and eigenvectors of a very large, sparse $N \times N$ matrix H without having to deal with actual $N \times N$ matrices and a characteristic equation of degree N.

Alternative possibilities:

Let the eigenvectors of M be such that

$$\lambda_1 \leq \lambda_2 \quad \cdots \quad \leq \lambda_N$$

(a, if complex $\mathrm{Re}\,\lambda_1 \leq \mathrm{Re}\,\lambda_2$ ...).

If only the eigenvector corresponding to the lowest eigenvalue λ_1, or the eigenvector corresponding to the λ of largest modulus (let this be λ_N) are sought, one can use relaxation techniques.

For instance, the sequence

$$\vec{U}_i \to \vec{U}_{i+1} = M\vec{U}_i \to \frac{\vec{U}_{i+1}}{|\vec{U}_{i+1}|}$$

(this last step to keep the vector normalized to 1)

will converge to $\vec{U}_N$.

Vice versa, if we approximate the evolution

$$\frac{d\vec{U}}{dt} = -M\vec{U}$$

write

$$\vec{U}_i \to \vec{U}_{i+1} = \vec{U}_i - \epsilon M\vec{U}_i \to \frac{\vec{U}_{i+1}}{|\vec{U}_{i+1}|}$$

and take care that the step ϵ is small enough that

$$|1 - \epsilon\lambda_1| > |1 - \epsilon\lambda_N|,$$

then the sequence will converge to the eigenvector corresponding to λ_1.

Using the orthogonality properties of eigenvectors, these methods can be generalized so as to produce a few of the lowest (or largest) eigenvectors, but they are not well suited to find a large number of eigenvectors.