Methods to Compute Approximate Inverses of Matrices

1. The problem

We are interested in computing approximations of the inverse (A^{-1}) of a given matrix A, such that these approximate inverses can be easily used in various iterative methods. An additional aspect of the above problem will also be considered, namely, the applicability of parallel algorithms when computing the approximations and when using them afterwards.
Let denote by G an approximation of A^{-1}.
There are two major applications of the approximate inverses in an iterative method. The first is to use them as preconditioners to the original matrix. Typically we have to perform actions of the form:

$$\mathbf{x}^{(l+1)} = \mathbf{x}^{(l)} - G\mathbf{r}^{(l)},\ l = 1, 2, \cdots,$$

i.e. G is involved in *matrix* $\times$ *vector* multiplications.
The second appears when we compute an incomplete factorization of a matrix partitioned in block form. Consider for simplicity a block tridiagonal matrix

$$A = block\_tridiag(A_{i,i-1}A_{i,i}A_{i,i+1}).$$

Then we compute a preconditioner $C = (D^{-1} + L)(I + DU)$, where L and U are the strictly lower and upper triangular parts of A.
In many applications the matrix A is sparse. The exact inverse will be just a full matrix. A natural condition on G then arises, we can impose that G has some a priori chosen sparsity pattern (the same as A or different) which will make the calculations with G easy and cheap, and also will provide a sufficient accuracy.
Let A be of order n and $\mathcal{S} = \{(i,j),\ 1 \le i \le n;\ 1 \le j \le n\}$. Any proper subset S of $\mathcal{S}$ will be referred to as a sparsity pattern ($S \subset \mathcal{S}$). S_L denotes the corresponding sparsity pattern for the lower triangular matrix and $S_{\tilde{L}}$ denotes the corresponding sparsity pattern for the strictly lower triangular matrix.
Thus, $A \in S$ if $a_{ij} \neq 0 \Longleftrightarrow (i,j) \in S$.

2. Explicit Methods

In these methods an approximation of the inverse A^{-1} of a given nonsingular matrix A is computed explicitly.

Let S be a sparsity pattern. We want to compute $G \in S$, such that

$$(GA)_{ij} = \delta_{ij},\ (i,j) \in S,$$

i.e.

$$\sum_{k:(i,k)\in S} g_{ik} a_{kj} = \delta_{ij},\ (i,j) \in S. \tag{1}$$

Some observations can be made from (1):

- the elements in the ith row of G can be computed independently;
- even if A is symmetric, G in not necessarily symmetric, because g_{ii-1} and g_{i-1i} are, in general, not equal.

Example: Consider

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 3 & -2 & 0 \\ 0 & -2 & 4 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

We want to find G with the same sparsity pattern as A, i.e.

$$G = \begin{bmatrix} g_{11} & g_{12} & 0 & 0 \\ g_{21} & g_{22} & g_{23} & 0 \\ 0 & g_{32} & g_{33} & g_{34} \\ 0 & 0 & g_{43} & g_{44} \end{bmatrix}$$

Then we compute the entries of G using (1):

$$G = \begin{bmatrix} \frac{3}{5} & \frac{1}{5} & 0 & \\ \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & \frac{4}{13} & \frac{6}{13} & \frac{3}{13} \\ 0 & 0 & \frac{1}{7} & \frac{4}{7} \end{bmatrix}$$

and we find

$$GA = \begin{bmatrix} 1 & 0 & -0.40 & 0 \\ 0 & 1 & 0 & -0.33 \\ -0.31 & 0 & 1 & 0 \\ 0 & -0.28 & 0 & 1 \end{bmatrix}$$

For instance, the second row of G is the second row of the matrix

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 3 & -2 & 0 \\ 0 & -2 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1}.$$

This is explained below.
It can be seen that when computing the ith row of G, only certain entries of A are used, namely, for which $(i,k) \in S$ and $(i,j) \in S$. Hence, if

$$S = \{(1,1),(1,2),\{(i,i-1),(i,i),(i,i+1)\}_{i=1}^{n},(n,n-1),(n,n)\}$$

then only the entries of the matrix $A^i = \begin{bmatrix} a_{i-1,i-1} & a_{i-1,i} & a_{i-1,i+1} \\ a_{i,i-1} & a_{i,i} & a_{i,i+1} \\ a_{i+1,i-1} & a_{i+1,i} & a_{i+1,i+1} \end{bmatrix}$ will be used when computing the ith row of G. Furthermore, the ith row of G is the ith row of $\mathcal{G}^{-1}$, where

$$\mathcal{G} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & A^i & & & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}.$$

But, pay attention! If we compute AG then

```
       0.8667   -0.2667   -0.3333         0
AG =   0.4000    1.1846    0.0769   -0.4615
      -0.6667   -0.1026    1.0366    0.3516
            0   -0.3077   -0.1758    0.9121
```

i.e. the matrix G is here a left-hand approximate inverse of A and as such is somewhat less accurate than as an right-hand approximate inverse.
The drawback of the above method is that G is not symmetric in general even if A is symmetric.

3. Implicit Methods

These methods require that A is factored first. In practice they can be used mainly for band or "envelope" matrices.
Suppose $A = LD^{-1}U$ is a triangular matrix factorization of A. If A is a band matrix then L and U are also band matrices.
Let

$$L = I - \tilde{L};\ U = I - \tilde{U},$$

where $\tilde{L}$ and $\tilde{U}$ are strictly lower and upper triangular matrices correspondingly.
The following **Lemma** can be stated.
Lemma: Using the above notations it can be shown that

(i) $A^{-1} = DL^{-1} + \tilde{U}A^{-1}$,

(ii) $A^{-1} = U^{-1}D + A^{-1}\tilde{L}$.

Proof:

$$A = LD^{-1}U \Longrightarrow A^{-1} = U^{-1}DL^{-1}$$
$$\Longrightarrow (I - \tilde{U})A^{-1} = DL^{-1} \Longrightarrow A^{-1} = DL^{-1} + \tilde{U}A^{-1}.$$

Also

$$A^{-1}(I - \tilde{L}) = U^{-1}D \Longrightarrow A^{-1} = U^{-1}D + A^{-1}\tilde{L}.$$

□

Since DL^{-1} is lower triangular and $\tilde{U}$ is upper triangular , using (i) we can compute entries in the upper triangular part of A^{-1} with no need to use entries of L^{-1}. Similarly, using (ii) we can compute entries of the lower triangular part of A^{-1} without U^{-1}.
Suppose A is a block banded matrix with a semibandwidth p and we want to form A^{-1} also as block banded with a semibandwidth $q:\ q \geq p$. The identities (i) and (ii) can be used then for the computation of the upper and lower parts of A^{-1}.

Algorithm:

for $r = n, n-1, \cdots, 1$

$$(A^{-1})_{r,r} = D_{r,r} + \sum_{s=1}^{min(q,n-r)} \tilde{U}_{r,r+s}(A^{-1})_{r+s,r}$$

for $k = 1, 2, \cdots, q$

$$(A^{-1})_{r-k,r} = \sum_{s=1}^{min(q,n-r+k)} \tilde{U}_{r-k,r-k+s}(A^{-1})_{r-k+s,r} \rightsquigarrow (i)$$

$$(A^{-1})_{r,r-k} = \sum_{t=1}^{min(q,n-r+k)} (A^{-1})_{r,r-k+t}\tilde{L}_{r-k+t,r-k} \rightsquigarrow (ii)$$

endfor

endfor

Remarks:

- The algorithm consists only of $matrix \times matrix$ operations.
- There is no need to compute any entries outside the bands.
- If A is symmetric then executing only (i) or only (ii) will be enough.
- It can be seen that $(A^{-1})_{nn} = D_{nn}^{-1}$.

There are two drawbacks of this algorithm. It requires first the factorization $A = LD^{-1}U$ and even A is s.p.d., the band part of A^{-1}, which is computed, need not to be s.p.d.

Example: Consider an s.p.d. matrix

$$G = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -3 \\ 1 & -3 & 4 \end{bmatrix}.$$

Then

$$G_{band} = \begin{bmatrix} 1 & -2 & 0 \\ -2 & 5 & -3 \\ 0 & -3 & 4 \end{bmatrix}$$

is indefinite.

4. A general framework for computing approximate inverses

It turns out that both the explicit and implicit methods can be characterized as methods to compute best approximations of A^{-1} in some norm. The basic idea is due to Kolotilina and Yeremin (1986-1989).

Let a sparsity pattern S be given. Consider the functional

$$F_W(G) = \|I - GA\|_W^2 = tr(I - GA)W(I - GA)^T,$$

where the weight matrix W is s.p.d. If $W \equiv I$ then $\|I - GA\|_I$ is the Frobenius norm of $I - GA$.
Clearly $F_W(G) \geq 0$. If $G = A^{-1}$ then $F_W(G) = 0$. Hence, we want to compute the entries of G in order to minimize $F_W(G)$, i.e. to find $\hat{G} \in S$, such that

$$\|I - \hat{G}A\|_W \leq \|I - GA\|_W, \ \forall G \in S.$$

The following properties of $tr(\cdot)$ will be used:

$$trA = trA^T, \ tr(A + B) = trA + trB.$$

Then

$$\begin{aligned} F_W(G) &= tr(I - GA)W(I - GA)^T \\ &= tr(W - GAW - W(GA)^T + GAW(GA)^T) \\ &= trW - trGAW - tr(GAW)^T + trGAWA^TG^T. \end{aligned} \tag{2}$$

Further, as we are interested in minimizing F_W w.r.t. G, we consider the entries $g_{i,j}$ as variables. The necessary condition for a minimizing point are then

$$\frac{\partial F_W(G)}{\partial g_{ij}} = 0, \ (i, j) \in S. \tag{3}$$

From (2) and (3) we get

$$-2(WA^T)_{ij} + 2(GAWA^T)_{ij} = 0,$$

or

$$(GAWA^T)_{ij} = (WA^T)_{ij}, \ (i, j) \in S. \tag{4}$$

The equations (4) may or may not have a solution, depending on the particular matrix A and the choice of S and W.
Example: Let A be s.p.d. Choose $W = A^{-1}$ which is also s.p.d.

$$\Longrightarrow (GA)_{ij} = \delta_{ij}, \ (i, j) \in S,$$

i.e. the formula for the explicit method can be seen as a special case of the more general framework for computing approximate inverses using weighted Frobenius norms.

Example: Let $W = (A^T A)^{-1}$.

$$\Longrightarrow (G)_{ij} = (A^{-1})_{ij},\ (i,j) \in S,$$

which is the formula for the implicit method. In this case the entries of G are the corresponding entries of the exact inverse.

Example: Let $W = I$.Then

$$F_W(G) = n - tr(GA)$$

and

$$(GAA^T)_{ij} = (A^T)_{ij},\ (i,j) \in S.$$

This method is also explicit.
We can expect that such methods will be accurate only if all elements of$A$ which are not used in the computations are zero or are relatively small. In some cases the quality of the computed approximation G to A^{-1} can be significantly improved using diagonal compensation of the entries of A which are outside S.

Example: Let A be symmetric and five-diagonal. Suppose we know that the two of the off-diagonals contain small entries. Such matrix appears if we solve the anisotropic problem, for instance:

$$-\frac{\partial^2 u}{\partial x^2} - \varepsilon \frac{\partial^2 u}{\partial y^2} = f,$$

where $\varepsilon > 0$ is small.
We choose a tridiagonal sparsity pattern S_3 for G, where the the two nonzero off-diagonals will correspond to the off-diagonals of A, containing bigger elements, i.e. they are not necessarily next to the main diagonal. Then we construct an approximate inverse in the following way:

Step 1: Let $\tilde{A}$ be A with deleted small entries, i.e. $\tilde{A} \in S_3$.

Step 2: Compute $\tilde{G}$: $(\tilde{G}A)_{ij} = \delta_{ij},\ (i,j) \in S_3$.

Step 3: Find $G = \bar{G} + D$, where $\bar{G} = \frac{1}{2}(\tilde{G} + \tilde{G}^T)$ and D is diagonal, computed from the following imposed condition on G, i.e.

$$GA\mathbf{e} = \mathbf{e},$$

and $\mathbf{e} = (1, 1, \cdots, 1)^T$.

The diagonal compensation technique prescribes the s.p.d. property of A.

5. Constructing a symmetric and positive definite approximate inverse

For some methods (as the preconditioned Chebyshev iteration method) it is of importance to use s.p.d. preconditioners. The methods described till now do not guarantee that G will be such a matrix.

In order to compute s.p.d. approximate inverse of an s.p.d. matrix, we proceed as follows.

Let S be a symmetric sparsity pattern. We seek G of the form

$$G = L_G^T L_G, L_G \in S_L.$$

Clearly G will be s.p.d.

Theorem. A matrix G of the form $G = L_G^T L_G$ which is an s.p.d. approximation of A^{-1} can be computed from the following relation:

$$min_{X \in S_L} \frac{\frac{1}{n} tr XAX^T}{(det(XAX^T))^{\frac{1}{n}}} = \frac{\frac{1}{n} tr L_G A L_G^T}{(det(L_G A L_G^T))^{\frac{1}{n}}}. \tag{5}$$

Proof:

$X \in S_L$ is lower triangular. Let $X = D(I - \tilde{X})$, where $\tilde{X} \in S_{\tilde{L}}$ is strictly lower triangular. Then $\tilde{X} = I - D^{-1}X$. Let denote also $D = diag(d_1, d_2, \cdots, d_n)$. Then

$$\frac{\frac{1}{n} tr XAX^T}{(det(XAX^T))^{\frac{1}{n}}} = \frac{\frac{1}{n}\sum_i (XAX^T)_{ii}}{(det(X)^2 det(A))^{\frac{1}{n}}}$$

$$= \frac{\frac{1}{n}\sum_i \left(D(I - \tilde{X})A(I - \tilde{X})^T D\right)_{ii}}{(det(X)^2 det(A))^{\frac{1}{n}}} = \frac{\frac{1}{n}\sum_i d_i^2 \left((I - \tilde{X})A(I - \tilde{X})^T\right)_{ii}}{(\prod_i d_i^2)^{\frac{1}{n}} (det(A))^{\frac{1}{n}}}$$

$$(6) \qquad = \frac{\frac{1}{n}\sum_i \alpha^2}{(\prod_i \alpha^2)^{\frac{1}{n}}} \cdot \frac{\left(\prod_i((I-\tilde{X})A(I-\tilde{X})^T)_{ii}\right)^{\frac{1}{n}}}{(det(A))^{\frac{1}{n}}}$$

$$= Expression\_A \cdot Expression\_B.$$

In the above notations $\alpha_i^2 = d_i^2 \left((I-\tilde{X})A(I-\tilde{X})^T\right)_{ii}$.
It can be seen from (6) that $Expression\_B$ does not depend on d_i. The problem of minimizing $Expression\_B$ is a particular case of the already considered problem of minimizing the functional $F_W(G)$ with a special choice of the corresponding matrices - $W = A, \ A = I, \ G = \tilde{X}$. In other words, the solution of the problem

$$(7) \qquad min_{\tilde{X}\in S_{\tilde{L}}} \prod_i \left((I-\tilde{X})A(I-\tilde{X})^T\right)_{ii} = min_{\tilde{X}\in S_{\tilde{L}}} tr(I-\tilde{X})A(I-\tilde{X})^T$$

will be also the solution of minimizing $Expression\_B$.
Further, $Expression\_A \geq 1, \ \forall\alpha$, being the ratio of the arithmetic and geometric mean, and takes the value 1 when $\alpha_i^2 = 1$.
Thus, we minimize $Expression\_A$ computing

$$(8) \qquad d_i = \frac{1}{\left((I-\tilde{X})A(I-\tilde{X})^T\right)_{ii}^{\frac{1}{2}}}.$$

Let $\tilde{L_G}$ be the solution of (7). Note that it is strictly lower triangular. Let the entries d_i of D are computed from the relations (8) where instead of $\tilde{X}$ $\tilde{L_G}$ is used. Then the matrix $L_G^T L_G$, where $L_G = D(I - \tilde{L_G})$, will be the searched approximation of A^{-1}:

- $(L_G A L_G^T)_{ii} = 1$ by construction;
- The equality (5) gives a measure of the quality of the approximate inverse constructed.

□

Example: Let $A = tridiag(-1, 4, -1)$. Find $L_G^T L_G$ - an approximate inverse of A, where L_G is bidiagonal. Thus, $S_{\tilde{L}} = \{\{(i-1,i)\}_{i=2}^n\}$.
First we compute a strictly lower bidiagonal matrix $\tilde{L}$ from the condition

$$(\tilde{L}A)_{i,j} = (A)_{i,j}, \ i,j \in S_{\tilde{L}},$$

which gives us

$$\tilde{L} = \begin{bmatrix} 0 & & & & & \\ \frac{1}{4} & 0 & & & & \\ & \frac{1}{4} & 0 & & & \\ & & & \ddots & & \\ & & & \frac{1}{4} & 0 & \\ & & & & \frac{1}{4} & 0 \end{bmatrix}.$$

Then d_i are found to be

$$d_1 = \frac{1}{2}, d_i = \frac{2}{\sqrt{15}},\ i = 1, 2, \cdots, n.$$

Thus,

$$L_G = \begin{bmatrix} \frac{1}{2} & 0 & & 0 & 0 \\ \frac{1}{2\sqrt{15}} & \frac{2}{\sqrt{15}} & & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & & \frac{1}{2\sqrt{15}} & \frac{2}{\sqrt{15}} \end{bmatrix},$$

$$L_G^T L_G = \begin{bmatrix} \frac{4}{15} & \frac{1}{15} & 0 & & 0 & 0 \\ \frac{1}{15} & \frac{17}{60} & \frac{1}{15} & & 0 & 0 \\ & & & \ddots & & \\ 0 & 0 & & & \frac{1}{15} & \frac{17}{60} \end{bmatrix},$$

and

$$L_G^T L_G A = \begin{bmatrix} 1 & 0 & -\frac{1}{15} & & 0 & 0 \\ \frac{7}{15} & 1 & \frac{7}{60} & & 0 & 0 \\ & & & \ddots & & \\ 0 & 0 & & & \frac{7}{60} & 1 \end{bmatrix}.$$