

Applications of Cholesky Decomposition to
Ab Initio Methods

Thomas Bondo Pedersen and Alfredo Sánchez

Institute of Molecular Science

University of Valencia, Spain

Henrik Koch

Department of Physical Chemistry

University of Trondheim, Norway

December 9, 2002

Cholesky Decomposition

- **DISCLAIMER**

Everything you are about to hear is still at the experimental stage, and in all probability these are not the final words on the subject.

Cholesky Decomposition

- The fundamental doctrine of *ab initio* quantum chemistry:

DON'T MESS WITH THE INTEGRALS

- \implies integral–direct techniques
- Integral–direct methods are (or may be made) open–ended...
- ...but tedious!

Cholesky Decomposition

- Integral-direct CC calculation of one-photon spectra (frequencies and intensities) may require on the order of **100 integral calculations**...
- ...assuming 1 integral calculation per iteration
- Not really reasonable estimate:
 1. In DALTON: Integrals calculated 4 times per iteration (reduced permutation symmetry)
 2. Out-of-core algorithm \implies several integral calculations in each iteration

The calculations presented here have all been carried out with a local version of the DALTON electronic structure program [3]. This implies that the speed-ups to be reported later are (obviously) not globally representative; they are speed-ups of the DALTON program which, in turn, may be slower than some other available code.

Cholesky Decomposition

- Example:

TCCO, C_2 point group symmetry

Integral-direct calculation of CC2 energy, aug-cc-pVTZ basis set

$N = 690$ (345, 345), $O = 31$ (17, 14)

Required memory: ~ 2.6 Gb

Integral time (IBM RS/6000, Power3, 1.6 GFlop): ~ 300 hours

(Appetizer: Same calculation and conditions using Cholesky: ~ 20 hours)

Cholesky Decomposition

- In general, integral-direct is simply too expensive!
- Only place to look is the two-electron integrals....

Cholesky Decomposition

- Properties of the two-electron integral matrix (AO basis)

$$M_{pq} \leftarrow (\alpha\beta|\gamma\delta)$$

- Positive (semi-) definite:

$$\begin{aligned} |M_{pq}| &\leq \sqrt{M_{pp}M_{qq}} \\ \max_{p,q} M_{pq} &= \max_p M_{pp} \end{aligned}$$

- Suggests sparsity \implies screening
- Screening is routinely used to reduce the number of integral evaluations
- Method-specific screening is used (much less routinely) to obtain linear scaling with respect to the size of the molecule

The simplest screening of the integral evaluation can be done in the following way. Suppose that M_{pp} is the largest diagonal. Then calculate the (numerical) upper bound of the off-diagonal elements of each column (or row) q as $\sqrt{M_{pp}M_{qq}}$. If this upper bound is less than some predefined threshold, the entire q^{th} row and column may be disregarded.

In the case of “method-specific” screening, essentially the same procedure can be used except for a scaling with a “typical” quantity such as the SCF density matrix.

Cholesky Decomposition

- The product of two gaussians is a gaussian \implies Resolution of Identity (RI) methods:

$$(\alpha\beta|\gamma\delta) \sim \sum_{AB} (\alpha\beta|A)V_{AB}^{-1}(B|\gamma\delta)$$

- Problems with RI-methods:
 1. Need of an auxiliary basis: pre-optimized? Fitted along the way?
 2. Precision?

Cholesky Decomposition

- There's another way: Cholesky Decomposition
- Mathematically, it exists:

$$M_{pq} = \sum_{J=1}^{N_J} L_{pJ} L_{qJ}$$
$$N_J \leq \text{rank}(\mathbf{M})$$

- Cholesky vectors, $p = 1, 2, \dots, N_J$ and $q = p + 1, \dots, N_J$:

$$L_{pp} = \sqrt{M_{pp} - \sum_{r=1}^{p-1} L_{pr}^2}$$
$$L_{qp} = \frac{M_{qp} - \sum_{r=1}^{p-1} L_{qr} L_{pr}}{L_{pp}}$$

Cholesky decomposition of the two-electron integral matrix was originally suggested by Beebe and Lindenberg [1] in 1977 as a procedure to obtain “simplifications in the generation and transformation of two-electron integrals in molecular calculations”.

Note that if we consider the Cholesky vectors a matrix \mathbf{L} with entries L_{pJ} , we may think of it as an effective integral matrix containing only the non-redundant numerical information necessary to construct the integrals. In this sense, the Cholesky decomposition is similar to “zipping” the integral file.

Cholesky Decomposition

- Exploitation of sparsity:
 1. Operation count
 2. Storage
- Not necessarily compatible!
- Sparsity of Cholesky vectors:
 1. $N_J \leq \text{rank}(\mathbf{M})$:
systematic, easy to fully exploit
 2. $p \sim \alpha\beta$:
non-systematic; difficult to exploit for reducing operation counts; gives reduced I/O efficiency (due to scattering) for minimum disk storage

Cholesky Decomposition

- Algorithm:

1. Calculate diagonal, $D_q = M_{qq}$; set $N_J = 0$; find largest diagonal D_p
2. Calculate integral column, M_{qp}
3. Subtraction of previous vectors and scaling:

$$M_{qp} \leftarrow D_p^{-1/2} \left(M_{qp} - \sum_{J=1}^{N_J} L_{qJ} L_{pJ} \right)$$

4. Update diagonal:

$$D_q \leftarrow D_q - M_{qp}^2$$

5. Update counter, $N_J \leftarrow N_J + 1$ and save vector, $L_{qN_J} = M_{qp}$.
6. Find largest diagonal, D_p . If larger than threshold, go to 2

Cholesky Decomposition

- A simple example. Consider the symmetric positive semi-definite matrix

$$\mathbf{M} = \begin{pmatrix} 25 & 10 & 15 & 50 \\ 10 & 5 & 4 & 20 \\ 15 & 4 & 13 & 30 \\ 50 & 20 & 30 & 100 \end{pmatrix}$$

- Cholesky decomposition: initial diagonal $D = (25, 5, 13, 100)$.
 1. $L^1 = (50, 20, 30, 100)/\sqrt{100} = (5, 2, 3, 10)$
 $D \leftarrow D - (25, 4, 9, 100) = (0, 1, 4, 0)$
 2. $L^2 = [(15, 4, 13, 30) - 3(5, 2, 3, 10)]/\sqrt{4} = (0, -1, 2, 0)$
 $D \leftarrow D - (0, 1, 4, 0) = (0, 0, 0, 0)$
- The decomposition detects both the obvious linear dependence (columns 1 and 4) and a “hidden” one (columns 2 and 3).

Cholesky Decomposition

- Pivoting needed for numerical stability.
- Problem: integrals are calculated in shell-distributions.
 - \implies Partial pivoting and span.
 - \implies Possibility of repeated calculations of distributions.

Cholesky Decomposition

- Shell-based algorithm requires a partition of the diagonal into shell-blocks:

$$D = \{D_P\}$$

- Two extreme strategies for decomposing such a sub-block of the integral matrix:

1. Full pivoting: always choose the largest diagonal.

\implies Recalculations, but numerically most stable.

2. No pivoting: finish the integrals in this sub-block.

\implies No recalculations, but numerically less stable.

- Intermediate strategy: If, during the decomposition of the shell-block P, the largest diagonal belongs to a different shell-block, then continue with shell-block P until the diagonal to be treated becomes less than a “span-factor” (≤ 1) times the globally largest diagonal.

Cholesky Decomposition

- The decomposition time is typically on the order of a few integral–direct Fock matrix builds.
- Note that efficient use of sparsity would significantly reduce the decomposition time.

Cholesky Decomposition

- Current limitation: “large” diagonals.
- Suppose the largest diagonal is $D_p = 10^9$. Calculation of the first vector according to

$$L_{q,1} = \frac{M_{qp}}{\sqrt{D_p}}$$

will then introduce an error on the order of 10^{-7} (double precision) into this vector and into the updated diagonal. This error will “propagate” into the vectors calculated later, eventually causing negative diagonals or, at best, reducing precision of the decomposition.

- In practice, “large” diagonals occur for large angular momenta in the Gaussian basis functions, *i.e.*, for large basis sets.

Cholesky Decomposition

- Possible solution: scaling.
- Suppose we have constructed a non-singular, symmetric “scale” matrix \mathbf{S} such that the scaled matrix $\tilde{\mathbf{M}} = \mathbf{S}\mathbf{M}\mathbf{S}$ has a “tractable” diagonal. Cholesky decomposing this matrix results in the Cholesky matrix $\tilde{\mathbf{L}}$: $\tilde{\mathbf{M}} = \tilde{\mathbf{L}}\tilde{\mathbf{L}}^T + \tilde{\epsilon}$, where the last part is the error matrix.
- It is now easy to calculate the Cholesky vectors representing the original matrix \mathbf{M} through $\mathbf{L} = \mathbf{S}^{-1}\tilde{\mathbf{L}}$.
- The resulting error matrix will be $\epsilon = \mathbf{S}^{-1}\tilde{\epsilon}\mathbf{S}^{-1}$.

Cholesky Decomposition

- We have seen that Cholesky decomposition greatly reduces the disk requirement for storing the integrals, thereby avoiding the extensive (and therefore expensive) recalculations inherent to the integral–direct approach.
- It now remains to show that the Cholesky vectors can be used to generate quantum chemical results in a reasonable time and without essential loss of precision.
- The most straightforward way is to “unzip” the integral file, *i.e.*, regenerate the integrals when needed. Such a Cholesky–based integral–generator would then replace the integral program itself in a pseudo–integral–direct approach.

Cholesky Decomposition

- While appealing in terms of ease of implementation, regeneration of the AO two-electron integrals is unfortunately a costly affair for all but the smallest systems and basis sets. Formally, the regeneration scales as $\sim N^5$, when sparsity is not used.
- Sparsity, however, is not readily exploited. The main obstacle is a poor ratio of floating-point operations to memory references caused by the irregular sparsity pattern of the Cholesky vectors. Thus, it turns out that it is often more efficient to simply include the zeros and employ standard BLAS libraries.
- Thus, we have *not* used sparsity in the applications presented here.

Cholesky Decomposition

- SCF calculations — Fock build:

$$\begin{aligned} F_{\alpha\beta} &= \sum_{\gamma\delta} [2(\alpha\beta|\gamma\delta) - (\alpha\delta|\gamma\beta)] D_{\gamma\delta} \\ &= 2 \sum_J L_{\alpha\beta}^J \sum_{\gamma\delta} L_{\gamma\delta}^J D_{\gamma\delta} \\ &\quad - \sum_{kJ} L_{\alpha k}^J L_{\beta k}^J \end{aligned}$$

- $\sim \text{ON}^2 N_J = p_J \text{ON}^3$

Note that the scaling of the Coulomb term is $p_J N^3$, i.e. independent of the number of electrons of the system. For large systems (large O), therefore, the exchange term is clearly dominating. At the same time, it is presumably easier to write an efficient parallel version (minimum communication between nodes) of the Coulomb term which uses also the sparsity of the vectors. In this way, the Cholesky formulation may turn out to be particularly successful for non-hybrid DFT calculations on very large systems.

Cholesky Decomposition

- Canonical MP2 calculations:

$$E_{\text{MP2}} = - \sum_{aibj} \frac{(ia|jb) [2(ia|jb) - (ib|ja)]}{\epsilon_a - \epsilon_i + \epsilon_b - \epsilon_j}$$
$$(ia|jb) = \sum_J L_{ia}^J L_{jb}^J$$

- $\sim O^2 V^2 N_J = p_J O^2 V^2 N$

Cholesky Decomposition

- Algorithm ($(ia|jb)$ integrals not stored):

1. Transform:

$$L_{ia}^J \leftarrow L_{\alpha\beta}^J$$

Store on disk

2. Calculate for as many virtuals ($\#b$) that fit in memory:

$$(ia|j\#b) = \sum_j L_{ia}^J L_{j\#b}^J$$

and calculate energy contribution from this batch

- Memory requirement $\sim O^2V$
- Disk requirement $\sim OVN_J \approx p_J ON^2$
- Notice: may be used integral-direct as well, but requires the number of b -batches integral calculations

Cholesky Decomposition

- Reduction of pre-factor p_J ?
- Second decomposition:

$$(ia|jb) = \sum_K M_{ia}^K M_{jb}^K$$

- Reduced operation count?
- At least, $N_K \leq \min [OV, N_J]$
- But the decomposition requires (expensive) calculation of columns of $(ia|jb)$

Cholesky Decomposition

- CC2 energy calculations:

$$E_{\text{CC2}} = E_{\text{HF}} + \sum_{aibj} (\hat{t}_{ij}^{ab} + 2t_i^a t_j^b - t_j^a t_i^b) (ia|jb)$$

$$\begin{aligned} \Omega_i^a &= \tilde{F}_{ai} + \sum_{bj} \hat{t}_{ij}^{ab} \tilde{F}_{jb} \\ &+ \sum_{ckb} \hat{t}_{ik}^{bc} (kc|ab) \\ &- \sum_{ckj} \hat{t}_{jk}^{ac} (kc|ji) \end{aligned}$$

$$t_{ij}^{ab} = \frac{(ai|bj)}{\epsilon_a - \epsilon_i + \epsilon_b - \epsilon_j}$$

Cholesky Decomposition

- Algorithm similar to MP2 for time-consuming steps
- No doubles arrays stored!
- Time-consuming step(s):

$$t_{ij}^{\#ab} = (\epsilon_i - \epsilon_{\#a} + \epsilon_j - \epsilon_b)^{-1} \sum_J \tilde{L}_{\#ai}^J \tilde{L}_{bj}^J$$

$$Y_{\#ai}^J = \sum_{bj} \hat{t}_{ij}^{\#ab} L_{jb}^J$$

- Operation count $\sim 2O^2V^2N_J = 2p_J O^2V^2N$
- Memory requirement $\sim O^2V$
- Disk requirement $\sim (3OV + O^2)N_J \approx 3p_J ON^2$

Note that a second decomposition (of $(a_i|b_j)$ or of the doubles amplitudes) is far less efficient for CC2 than MP2, as the Y -intermediates must be set up for the full set $J = 1, 2, 3, \dots, N_J$. Furthermore, such a decomposition would have to be done in each iteration since the integrals depend upon the singles amplitudes through the so-called particle and hole transformation matrices [2].

Interestingly, our experience shows that the decomposition of the amplitudes requires more vectors than that of the $(a_i|b_j)$ integrals alone. Seemingly, therefore, the orbital energy denominators reduce the sparsity of the amplitudes.

Cholesky Decomposition

- The advantage of CC2 over MP2 is, however, not in energy calculations but in the possibility of calculating excitation energies and response properties, notably frequency-dependent.
- In general, a major bottleneck in response (or excitation energy) calculations is the number of equations that need to be solved.
- If the equations may be solved simultaneously (e.g. more than one excitation energy is needed) then the integrals can be calculated once per iteration (provided there is enough memory for all intermediates).
- That is not the case with a Cholesky based algorithm which would in general depend linearly on the number of equations.
- Calls for new methods of solution....

Cholesky Decomposition

- We have seen that the Cholesky procedure is well-suited for perturbation-based lower-order methods like MP2 and CC2.
- The main reason is the ease with which integrals of the type $(ia|jb)$ are constructed from the Cholesky vectors.
- For more accurate methods, however, we need to be able to treat integrals with four virtual indices.

Cholesky Decomposition

- The archetypical example would be the so-called CCSD B-term:

$$\Omega_{aibj}^B = \sum_{cd} t_{ij}^{cd} (ac|bd)$$

- A straightforward Cholesky implementation could be

$$\Omega_{aibj}^B = \sum_J \sum_c \sum_e \tilde{T}_{ac}^J \sum_d t_{ij}^{cd} \tilde{T}_{bd}^J$$

- Scaling: $\sim p_J O^2 N^4$
- Comparing with the scaling of the conventional implementation ($O^2 N^4/4$), remembering the rule-of-thumb $p_J \approx 10$, we see that this simple approach is severely slower than the conventional one.

Cholesky Decomposition

- Only general solution is to use sparsity.
- In AO-basis:

$$\Omega_{\alpha i \beta j}^B = \sum_{\gamma \delta} t_{ij}^{\gamma \delta} (\alpha \gamma | \beta \delta)$$

- May be viewed as “O² exchange contributions to the Fock matrix”.
- In this sense, an efficient implementation of the exchange contribution in SCF would give important hints about how to handle the B-term and its equivalents in related methods like MP4 or CCSD(T).

Cholesky Decomposition

- Main conclusion: it's worthwhile!
 - reduces scaling through prefactor (in the sense that the fundamental scaling is the same: MP2 scales as N^5 etc.)
 - reduces storage demands (disk and memory)
 - messes with the integrals in a non-essential way
- Outlook:
 - need an integral program designed for the decomposition
 - density-based decomposition for SCF/DFT (linear scaling?)
 - CC2 properties
 - use of $\alpha\beta$ -sparsity [non-trivial!]
 - efficient parallelization
 - applications to bio-molecules such as light-harvesting complexes and DNA fragments.

Cholesky Decomposition

- Acknowledgments:
 - MOLPROP
 - Berta Fernández
 - Hans Aagren

References

- [1] N. H. F. Beebe and J. Lindenberg. Simplifications in the Generation and Transformation of Two-Electron Integrals in Molecular Calculations. *Int. J. Quantum Chem.*, 12:683, 1977.
- [2] O. Christiansen, H. Koch, and P. Jørgensen. The second order approximate coupled cluster singles and doubles model CC2. *Chem. Phys. Lett.*, 243:409, 1995.
- [3] T. Helgaker, H. J. Aa. Jensen, P. Jørgensen, J. Olsen, K. Ruud, H. Ågren, A. A. Auer, K. L. Bak, V. Bakken, O. Christiansen, S. Coriani, P. Dahle, E. K. Dalskov, T. Enevoldsen, B. Fernández, C. Hättig, K. Hald, A. Halkier, H. Heiberg, H. Hettema, D. Jonsson, S. Kirpekar, R. Kobayashi, H. Koch, K. V. Mikkelsen, P. Norman, M. J. Packer, T. B. Pedersen, T. A. Ruden, A. Sanchez, T. Saue, S. P. A. Sauer, B. Schimmelpfennig, K. O. Sylvester-Hvid, P. R. Taylor, and O. Vahtras. *DALTON, a molecular electronic structure program*, Release 1.2, 2001. <http://www.kjemi.uio.no/software/dalton/dalton.html>.