# Letter to the Editor
# Comment on "Fast Determination of the Optimal Rotational Matrix for Macromolecular Superpositions" [J. Comp. Chem. 31, 1561 (2010)]

**GERALD R. KNELLER**[1,2,3]

[1]*Centre de Biophysique Moléculaire, CNRS; Rue Charles Sadron, 45071 Orléans, France*
[2]*Université d'Orléans, Chateau de la Source-Av. du Parc Floral, 45067 Orléans, France*
[3]*Synchrotron Soleil; L'Orme de Merisiers, 91192 Gif-sur-Yvette, France*

*Received 27 April 2010; Accepted 23 May 2010*
*DOI 10.1002/jcc.21607*
*Published online 26 July 2010 in Wiley Online Library (wileyonlinelibrary.com).*

**Abstract:** Recently Liu et al. published a fast algorithm to solve the eigenvector problem arising in the quaternion-based method for the rotational superposition of molecular structures (J Comput Chem 2010, 31, 1561.). In this Comment, it is shown that the construction of the $4 \times 4$ matrix to be diagonalized—and not the diagonalization itself—represents the dominating part of the computational effort for the quaternion-based solution of the rotational superposition problem if molecules with more than about 100 atoms are considered.

 J Comput Chem 32: 183–184, 2011

**Key words:** rotational superposition; structural alignment; quaternions

The optimal superposition of rigid molecular structures using quaternions for the representation of the rotation has been treated independently by many researchers in the past.[1–7] A review can be found in ref. 8. The use of quaternions as angular variables leads to a an elegant, numerically robust and efficient formulation of the superposition problem, in which the latter is reduced to diagonalizing a symmetric $4 \times 4$ matrix depending on the coordinates of the molecular structures to be superposed. Recently Liu et al. published a particularly rapid method to solve this eigenvector problem. The gain in execution speed must be put in relation to the CPU time needed for the construction of the matrix to be diagonalized. The computational effort for the calculation if this matrix grows, in fact, linearly with the number of atoms in the coordinate sets to be superposed. If we consider two coordinate sets $\{\mathbf{x}_\alpha\}$ and $\{\mathbf{y}_\alpha\}$, which are both related to a common rotation center, the superposition problem reads

$$m(q) = \sum_{\alpha=1}^{N} w_\alpha (\mathbf{D}(q) \cdot \mathbf{x}_\alpha - \mathbf{y}_\alpha)^2 = \min(q). \tag{1}$$

Here $\{w_\alpha\}$ is a normalized set of weights for each atom pair $\{\mathbf{x}_\alpha, \mathbf{y}_\alpha\}$, and $\mathbf{D}(q)$ is the rotation matrix expressed in the quaternion parameters $q_0, q_1, q_2, q_3$,

$$\mathbf{D} = \begin{pmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(-q_0q_3+q_1q_2) & 2(q_0q_2+q_1q_3) \\ 2(q_0q_3+q_1q_2) & q_0^2+q_2^2-q_1^2-q_3^2 & 2(-q_0q_1+q_2q_3) \\ 2(-q_0q_2+q_1q_3) & 2(q_0q_1+q_2q_3) & q_0^2+q_3^2-q_1^2-q_2^2 \end{pmatrix}. \tag{2}$$

Using the normalization $q_0^2+q_1^2+q_2^2+q_3^2 = 1$ as a side constraint, the minimization of the target function leads to the eigenvector problem (here the notation of ref. 7 is used)

$$\mathbf{M} \cdot \mathbf{q} = \lambda \mathbf{q}, \tag{3}$$

where $\mathbf{M}$ has the form

$$\mathbf{M} = \sum_\alpha w_\alpha \begin{pmatrix} (\mathbf{x}_\alpha - \mathbf{y}_\alpha)^2 & -2\mathbf{u}_\alpha^T \\ -2\mathbf{u}_\alpha & (\mathbf{x}_\alpha + \mathbf{y}_\alpha)^2 \mathbf{1} - 2\mathbf{P}_\alpha \end{pmatrix}, \tag{4}$$

***Correspondence to:*** G. R. Kneller; e-mail: gerald.kneller@cnrs-orleans.fr

**Table 1.** Total CPU Time for the Superposition of Some Arbitrarily Chosen Structures from the PDB.

| PDB code | Name | No. of res. | No. of atoms | CPU time (ms) |
|---|---|---|---|---|
| 1JXT | Crambin | 46 | 787 | 0.0884 |
| 193L | Lysozyme | 129 | 1012 | 0.1100 |
| 1ABE | L-Arabinose-binding protein | 306 | 2316 | 0.2370 |
| 3BA0 | Macrophage metalloelastase | 365 | 2990 | 0.3030 |
| 1CTS | Citrate synthase | 437 | 3444 | 0.3480 |
| 3JUX | Protein translocase subunit secA | 813 | 6589 | 0.6550 |

The CPU time for the diagonalization of the matrix $\mathbf{M}$ is $t_q = 1.08 \times 10^{-2}$ ms. More explanations are given in the text.

with

$$\mathbf{u}_\alpha = \mathbf{x}_\alpha \wedge \mathbf{y}_\alpha, \tag{5}$$

$$\mathbf{P}_\alpha = \mathbf{x}_\alpha \cdot \mathbf{y}_\alpha^T + \mathbf{y}_\alpha \cdot \mathbf{x}_\alpha^T. \tag{6}$$

Table 1 gives the CPU time for the superposition of some molecular structures of different size taken from the Protein Data Bank (PDB)[9,10] and Figure 1 displays these data together with a linear fit,

$$t(N) = \alpha + \beta N, \tag{7}$$

with $\alpha = 1.10 \times 10^{-2}$ ms and $\beta = 9.78 \times 10^{-5}$ ms. The intercept $\alpha$ corresponds precisely to the average CPU time for the diagonalization of $\mathbf{M}$. The calculations have been performed with a Fortran program on an Apple Macintosh laptop computer with a 2.53 GHz Intel dual core processor, using the GNU Fortran 95 compiler[11] and subroutine DSYEV from the LAPACK library[12] for the diagonalization of $\mathbf{M}$.

The results show clearly that there is no point in optimizing the diagonalization of $\mathbf{M}$ if molecular structures with more than about 100 atoms are considered. For Crambin, one of the smallest proteins in the PDB, the ratio of the CPU times for preparing and diagonalizing $\mathbf{M}$ is 8 and grows to 60 for the largest protein considered in this study (Protein translocase subunit secA, PDB code 3JUX).

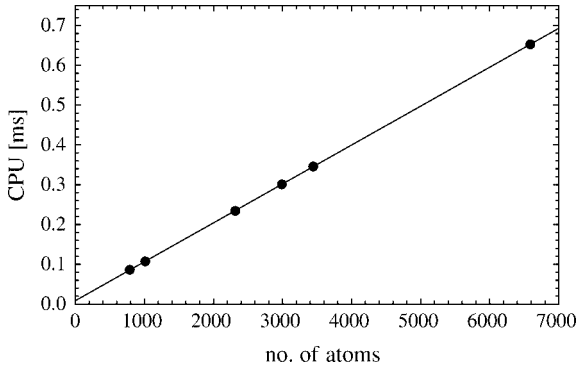


**Figure 1.** Total CPU time for the superposition of the protein structures listed in Table 1 (PDB entries 1JXT, 193L, 1ABE, 3BA0, 1CTS, 3JUX in increasing order with respect to the number of atoms).

## References

1. Horn, B. K. P. J Opt Soc Am A 1987, 4, 629.
2. Diamond, R. Acta Cryst A 1988, 44, 211.
3. Kearsley, S. Acta Cryst A 1989, 45, 208.
4. Kneller, G. Mol Sim 1991, 7, 113.
5. Coutsias, E.; Seok, C.; Dill, K. J Comput Chem 2004, 25, 1849.
6. Theobald, D. L. Acta Cryst A 2005, 61, 478.
7. Kneller, G. J Comput Chem 2005, 26, 1660.
8. Flower, D. J Mol Graph 1999, 17, 238.
9. Berman, H. M. Acta Crystallograph Sect A 2008, 64, 88.
10. Kirchmair, J.; Markt, P.; Distinto, S.; Schuster, D.; Spitzer, G. M.; Liedl, K. R.; Langer, T.; Wolber, G. J Med Chem 2008, 51, 7021.
11. The GNU fortran project. Accessed on 29 June 2010, Available at: http://gcc.gnu.org/fortran/.
12. Anderson, E.; Bai, Z.; Bischof, C.; Blackford, S.; Demmel, J.; Dongarra, J.; Du Croz, J.; Greenbaum, A.; Hammarling, S.; McKenney, A.; Sorensen, D. LAPACK Users' Guide. Society for Industrial and Applied Mathematics, 3rd ed.; Philadelphia, USA, 1999.