CHL Project 324

Tests of the Translation Matrix for Data Corrections of Head Motion

version 1.2

Maher Quraan

January 12, 2006

The translation matrix for data corrections of head motion is tested to verify its properties and to quantify its performance. In particular, the performance is evaluated as a function of the magnitude of the shift, the maximum number of spherical harmonics used, and the accuracy of the correction for various spherical harmonic degrees.

The revisions for the newly developed code used in these tests are as follows.

DC_Trans_T.h	27.2
DC_Trans_T.cc	27.2
DC_Trans_B.h	27.2
DC_Trans_B.cc	27.2
VSMfunction.h	27.1
VSMfunction.cc	27.9
DC_Manager.h	27.4
DC_Manager.cc	27.5

1 Properties of the translation matrix

The translation matrix $T(\vec{u})$ allows shifting the brain sources (expressed in terms of spherical harmonics) by some vector $\vec{u}$. As is the case for any translation, the product of the translation matrix, $T(\vec{u})$, and its inverse, $T(-\vec{u})$, must give the identity matrix

$$T(\vec{u})T(-\vec{u}) = I. \tag{1}$$

The deviation of this product from the identity matrix is characterized in Figure 1 as a function of the magnitude of a translational shift, s, for 3 different rotation matrices computed with a maximum spherical harmonic order, l_{max} = 12, 18, and 22 as labeled on the plots. The translational shift plotted on the x-axis is applied to all three axes, resulting in a shift magnitude

$$|\vec{u}| = \sqrt{3} \times s. \tag{2}$$

The top right plot in Figure 1 shows the deviation of the average diagonal element from 1.0. The remaining plots in this figure characterize the non-diagonal elements, which should

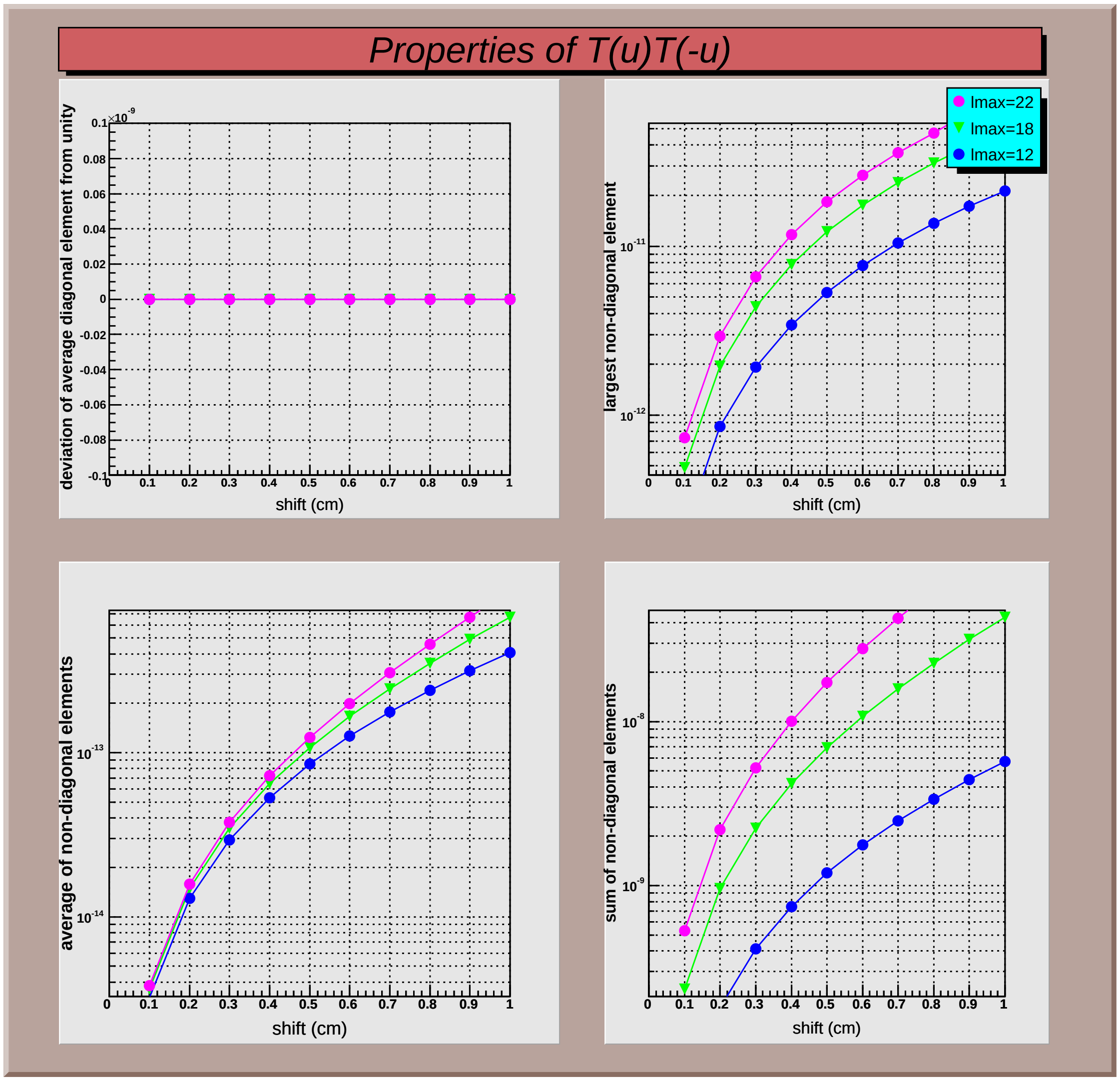

Figure 1: Properties of the matrix $T(\vec{u})T(-\vec{u})$ as a function of shift ($shift = \sqrt(3)|\vec{u}|$).

be zero for the product to be identical to the identity matrix. As the shift magnitude is increased, the accuracy of the translation matrix deteriorates as is evident from the increase in magnitude of the non-diagonal elements. The accuracy of the translation matrix also deteriorates when more spherical harmonic terms are included, as maybe seen from the various curves in Figure 1. This systematic effect will be further quantified in terms of its effect on the signal accuracy below.

2 Performance of the translation matrix

Ignoring all other factors in the problem such as outside sources etc, the effect of an internal source movement by a vector $\vec{u}$ is equivalent to a sensor movement by $-\vec{u}$. The original forward matrix with the sensors in their nominal locations, $L(0)$, changes by ΔL as a result of the sensor movement where

$$\Delta L = L(0) - L(-\vec{u}) \tag{3}$$

Figure 2 shows this difference for a source movement of $\vec{u} = (0.5, -0.3, 1.0)cm$ ($|\vec{u}| = 1.16cm$) as a function of a spherical harmonic degree (i.e. the elements of the forward matrix are summed for all spherical harmonics of the same degree). This difference represents the error in the forward matrix if no translational correction is performed. The original forward matrix can be corrected using the translation matrix by computing the matrix product $L(0)T(\vec{u})$, and the difference

$$\Delta L' = L(0)T(\vec{u}) - L(-\vec{u}) \tag{4}$$

can be computed. This difference is also plotted in Figure 2 and should be null if the translation matrix had infinite accuracy. However, the limited accuracy of the translation matrix results in a systemtic effect that grows with higher spherical harmonic degree. Since the translation matrix is truncated at $l_{max} = 18$, and since translations project a spherical harmonic of a given degree to higher-degree spherical harmonics, at $l = 18$ the translation matrix results in no correction at all. This is particularly significant for sources away from the spherical harmonic expansion centre where higher degree spherical harmonics carry significant weights.

3 Automated unit testing

The automated unit testing implements the tests presented in sections 1 and 2 above. The program
MEG/code/testTrans/testTrans.cc
runs the test described in section 1. The elements of the matrix $T(\vec{u})T(-\vec{u})$ are evaluated with random vectors in the range $(-1, -1, -1)cm < \vec{u} < (1, 1, 1)cm$. An error will result if the elements of the matrix are not similar to the results plotted in Figure 1. The shell script
MEG/code/testTrans/testTrans.bsh
can be used to run the test program. The program
MEG/code/testLTR/testLTR.cc
runs the test described in section 2. When the program is executed, an error will appear if the difference in equation 4 is inconsistent with the results represented by the blue curve in figure 2. The shell script
MEG/code/testLTR/testLTR.bsh
can be used to run the program.

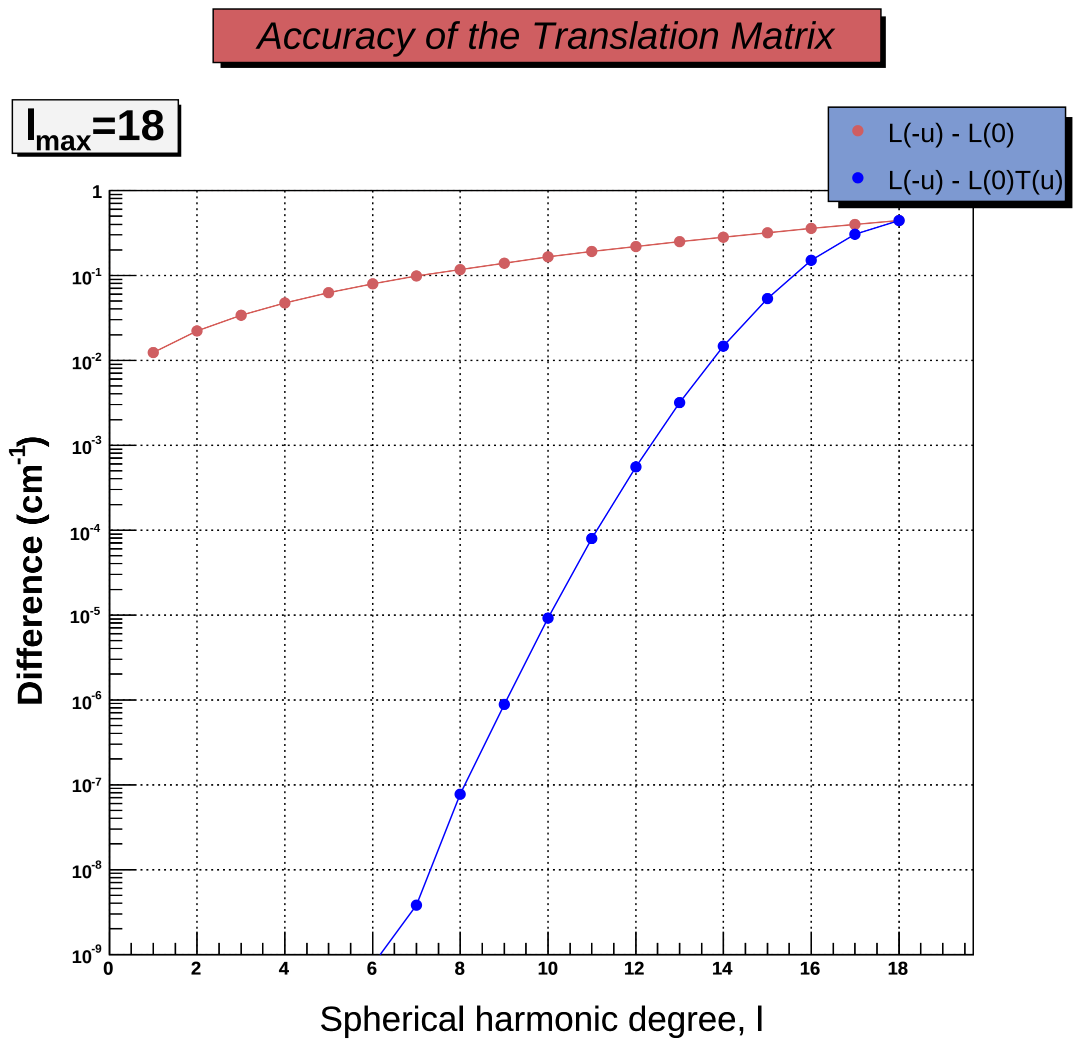

Figure 2: Red: Difference between the forward matrix, $L(-\vec{u})$, with the sensor positions shifted by a vector $-\vec{u} = (0.5, -0.3, 1.0)cm$, and the forward matrix, L(0), with nominal sensor positions. Blue: Difference between $L(0)$ and the product $L(0)T(\vec{u})$ where $T(\vec{u})$ is the translation matrix.

4 Comparison with Matlab

Figure 3 shows the comparison of the translation matrix computed with the C++ code with that computed with the Matlab code for all elements of the translation matrix with $l_{max} = 18$ (a total of 129600 elements). As maybe seen from the figure, the two algorithms yield identical results.

5 CPU performance

The translation matrix is computed in two stages; the first stage is independent of the translation vector and is computed once at the beginning of the data corrections. This is the time consuming part of the calculation. Figure 3 shows the CPU time consumed in this part as a function of the maximum spherical harmonic degree included in the calculation. For a maximum spherical harmonic order $l_{max} = 18$ this calculation takes $\sim 20s$.

The second stage uses the table constructed in the first stage to compute a translation matrix for a given translation vector $\vec{u}$. Figure 2 shows the CPU time consumed in this process. This part of the calculation is fast and it takes $\sim 8ms$ for a maximum spherical harmonic order $l_{max} = 18$.

6 Conclusions

The validity of the translational corrections code has been established and its performance has been assessed. The translational corrections accuracy is highly dependent on the spherical harmonic degree. The low-order spherical harmonics are corrected with high accuracy exceeding 1 part in 10^4 for spherical harmonics up to degree $\sim l = 10$ and deteriorating down to almost no correction at $l = 18$. The high-degree spherical harmonics contribute more to sources further away from the spherical harmonics expansion centre (currently chosen to be around the helmet "centre"), and will therefore be only partially corrected for translational head motion. This limited accuracy is inherent the model and not an artifact of implementation.

Automated unit testing has been implemented to ensure that future modifications which may result in errors or significant performance deterioration will be detected. The results of the C++ algorithms were compared to the results from the Matlab algorithms and found to be identical.

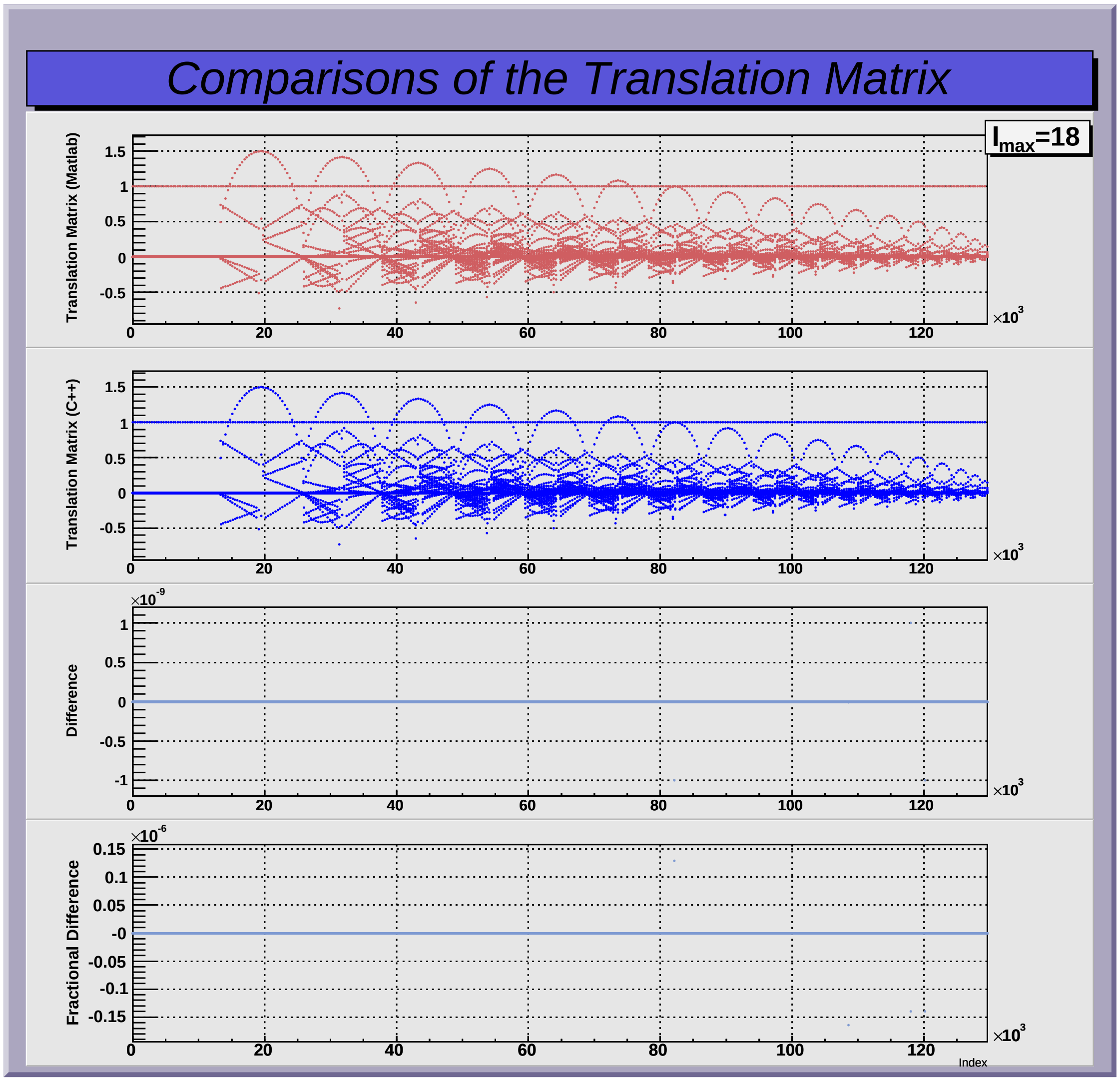

Figure 3: Elements of the translation matrix as computed using the Matlab algorithms (top), and the C++ algorithms (second from top). The third plot is the difference between the first two, and the bottom plot is the fractional difference.

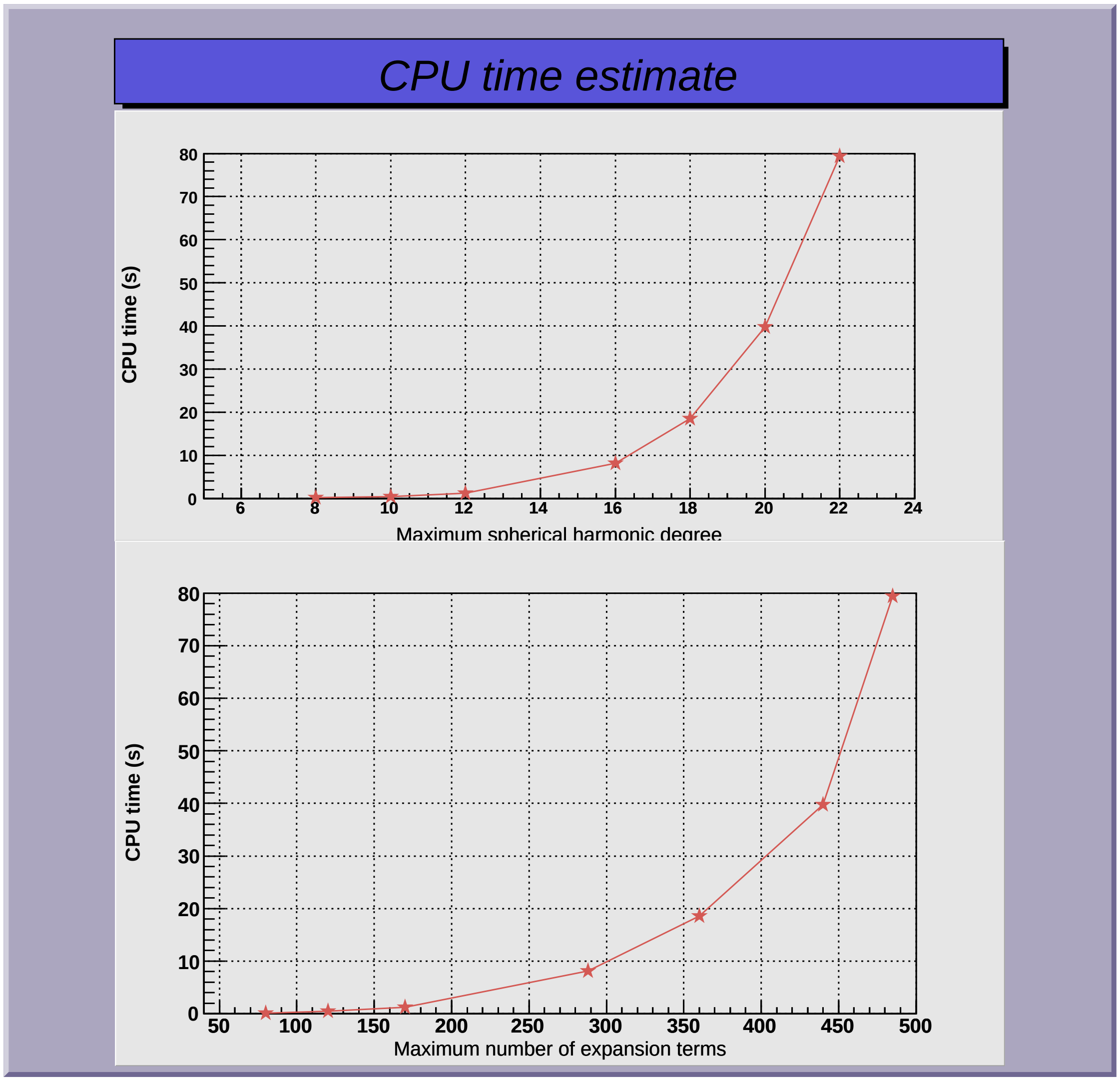

Figure 4: CPU time spent in calculating the translation-vector independent part of the translation matrix.

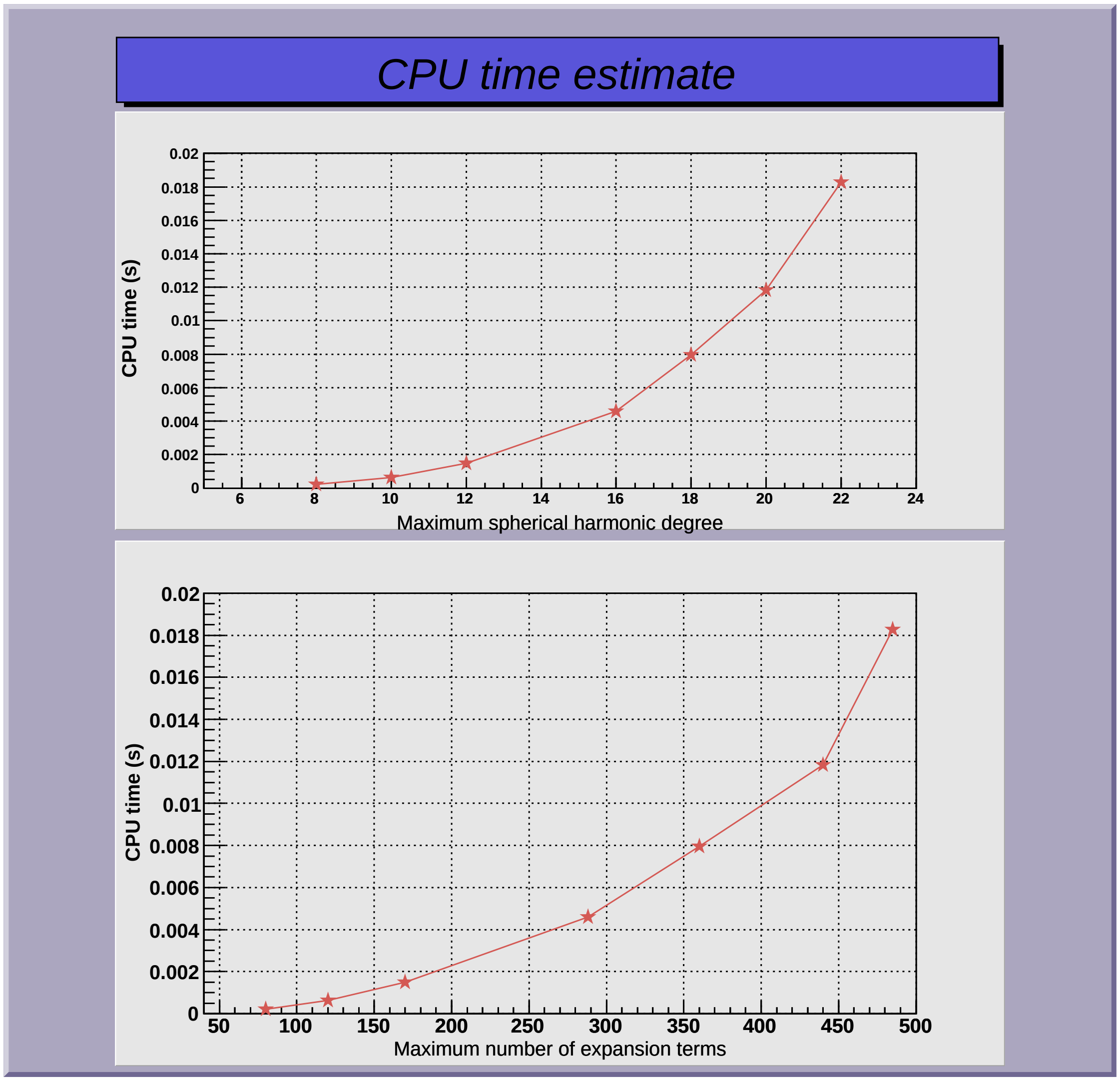

Figure 5: CPU time spent in calculating the translation matrix for a given translation vector.

References

[1] Harold Wilson. Internal report CI-803-1103 “Continuous Head Localization and Correction of MEG Data for Head Motion”, Nov 2003.