Bulletin of the *Transilvania* University of Braşov • Vol 13(62), No. 2 - 2020
Series III: Mathematics, Informatics, Physics, 715-722
https://doi.org/10.31926/but.mif.2020.13.62.2.27

LAPACK LIBRARY FOR MATHEMATICAL MODELING USED IN 3D OBJECT RECONSTRUCTION

Bianca Andreea LIXANDRU[1] **and Vlad MONESCU**[*,2]

Abstract

Over time, multiple solutions have been sought for building 3D models in the fastest and most accurate way possible. For this reason, this article proposes the use of LAPACK library for a quick solution of linear systems generated by least square method calculation. A mathematical and physical component used in the reconstruction of models consists in the use of Spherical Harmonics which represent some special functions often used in solving partial differential equations in different scientific areas. This problem finds its place in multiple domains such as medicine, engineering, programming. The target of the work described by this paper is to achieve optimization in the reconstruction of 3D models. We aim to perform real-time model reconstruction for large data sets by using Spherical Harmonics functions.

2000 *Mathematics Subject Classification:* 68-04, 43-04
Key words: spherical harmonics, Legendre polynomials, 3D models, 3D reconstruction, parallel computation, optimization, linear systems

1 Introduction

Nowadays, the reconstruction of 3D objects has become a typical issue that we wanted to address in an efficient manner[5],[8]. We gave special importance to the time in which the reconstruction of an object was carried out, as we tried to bring it as close as possible to real time. To make all this possible we decided to use parallel computation to solve linear systems[7],[1],[6] for problems such as enriching the image scanned by CT[2].

Moreover, we emphasized the use of a well-defined mathematical model based on Spherical Harmonics. The challenge is to calculate Spherical Harmonics values for a large number of given points and a large number of Spherical Harmonics functions[3].

[1]Faculty of Mathematics and Informatics, *Transilvania* University of Braşov, Romania
[2*] *Corresponding author*, Faculty of Mathematics and Informatics, *Transilvania* University of Braşov, Romania

The study proposed by this paper realizes the reconstruction of 3D objects starting from a set of points in space for which we use a series of Spherical Harmonics functions for modeling. The data points are collected from CT images. In this conditions we assume the 3D points are situated on different layers and all the points describe 3D object contour. Usually, the range number of these points is of order of tens on each layer. Thus the whole 3D object is described by not more than 2 hundreds of points. The objective is to find a way to populate the object surface with more points. Thus we extended the number of points and reconstructed the 3D objects with higher accuracy. The given points can be obtained from CT sections, and the problem is not necessarily specific to the field of medicine. Thus the reconstructed model increased the accuracy. Our goal for this research is to achieve the reconstruction of objects in real time.

This paper includes a section for entering the information needed to solve the problem, the algorithm used, the benefits of the LAPACK library, the results obtained on three 3D models used and conclusions.

2 Necessary knowledge

A function which satisfies Laplace equation is called harmonic function. Spherical Harmonics are functions that describe angular parts for some particular case of Laplace's equation solutions.

Spherical Harmonics functions form a basis and this means that any function f expressed in polar coordinates can be represented as a series of Spherical Harmonics:

$$f(\theta,\varphi)=\sum_{l=0}^{\infty}\sum_{m=-l}^{l}c_l^m Y_l^m(\theta,\varphi) \tag{1}$$

The function $Y_l^m(\theta,\varphi)$ is the spherical function given by:

$$Y_l^m(\theta,\varphi)=\sqrt{\frac{2l+1}{4\pi}}\sqrt{\frac{(l-m)!}{(l+m)!}}P_l^m(cos\theta)e^{im\varphi} \tag{2}$$

where

$$P_l^m(x)=(-1)^m(1-x^2)^{\frac{m}{2}}\frac{d^m}{dx^m}P_l(x), l\geqslant 0, m=\overline{0,l} \tag{3}$$

and

$$P_l(x)=\frac{1}{2^l l!}\frac{d^l}{dx^l}(x^2-1)^l \tag{4}$$

The polar coordinates (r, θ, φ) are related to the cartesian coordinates by the next relations:

$$\begin{cases} x = rcos(\theta)sin(\varphi) \\ y = rsin(\theta)sin(\varphi) \\ z = rcos(\varphi) \end{cases} \tag{5}$$

Based on (1) the coefficients c_l^m are calculated using:

$$c_l^m = \int f(\theta, \varphi)\overline{Y_l^m}(\theta, \varphi)d\Omega \tag{6}$$

The more spherical coefficients we have we can recreate a more accurate object using a finite sum.

The L_{max} variable is an essential parameter of our application and represents the bandwidth that we set, which further dictates the volume of calculations.

As shown in [6], using the least square method we obtained a linear system of equations with $(L_{max}+1)^2$ equations and $(L_{max}+1)^2$ unknowns given by:

$$\sum_{l=0}^{L_{max}} \sum_{m=-l}^{l} c_l^m \sum_{i=1}^{n} Y_l^m(\Omega_i)Y_p^q(\Omega_i) = \sum_{i=1}^{n} R_i Y_p^q(\Omega_i), p = \overline{0, L_{max}}, q = \overline{-p, p} \tag{7}$$

The complex linear system expands to a real linear system $Ax = b$ with $2(L_{max}+1)^2$ equations and $2(L_{max}+1)^2$ unknowns with

$$A = \sum_{i=1}^{n} Y_l^m(\theta_i, \varphi_i)Y_p^q(\theta_i, \varphi_i) \tag{8}$$

and

$$b = \sum_{i=1}^{n} R_i Y_p^q(\theta_i, \varphi_i) \tag{9}$$

Te quantity R_i represents the distance between origin and each point from our data set, and Ω_i is the solid angle defined by the pair (θ_i, φ_i), for each $i = \overline{1, n}$

The linear system is solved with LAPACK specific methods.

Solving the linear system we obtain the coefficients used to obtain a more accurate image of the reconstructed 3D object.

LAPACK offers a strong support for solving systems of linear equations using parallel computation.

In Figure 1 we can see the dimensions of the matrices for which we are going to calculate the solution of the linear system. Matrix A will be a square real matrix of size $2(L_{max}+1)^2$, matrix b will be a real column vector of size $2(L_{max}+1)^2$ and matrix x, the solution of the linear system, will be a column vector of the same size as b.

$$\begin{array}{cc} & \begin{array}{cc}(lmax+1)^2 & (lmax+1)^2\end{array} \\ \begin{array}{c}(lmax+1)^2\\ (lmax+1)^2\end{array} & \begin{bmatrix}\mathbf{Re}A & -\mathbf{Im}A\\ \mathbf{Im}A & \mathbf{Re}A\end{bmatrix}\begin{bmatrix}\mathbf{Re}x\\ \mathbf{Im}x\end{bmatrix} = \begin{bmatrix}\mathbf{Re}b\\ \mathbf{Im}b\end{bmatrix}\end{array}$$

Figure 1: Linear system

3 Algorithm and LAPACK library

In this section of the article we will present the algorithm for the reconstruction of the 3D model. The solution of the linear system is done using the methods from LAPACK. This library offers solutions for both complex and real systems, but we have solved this problem with real systems.

1. Input points in polar coordinates, L_{max}
2. Compute the complex matrix
3. Compute the complex vector
4. A = ComputeRealMatrix(points, L_{max})
5. b = ComputeRealVector(points)
6. clm = SolveSystem(A, b), l = $\overline{0, L_{max}}$, m = $\overline{-l, l}$

Steps (2) and (3) of the algorithm can be solved using formulas (8) and (9). SolveSystem method is used to solve linear system given by A and b, where A is a square matrix and b is a column vector. This function uses a specific method from LAPACK to solve linear system.

Computational time can be improved by using a symmetrization formula to optimize some calculations for negative values of m:

$$P_l^{-m}(x) = (-1)^m \frac{(l-m)!}{l+m!} P_l^m(x), l \geqslant 0, m = \overline{0, l} \tag{10}$$

LAPACK is a standard software library used for numerical linear algebra to solve problems as linear systems, linear least squares, eigenvalue problems, singular value decomposition(SVD).

This library uses associated matrix factorizations(LU, QR, Schur, generalized Schur, SVD, Cholesky) and provides functionality for real and complex matrices, in both single and double precision. LAPACK routines computation is performed by class Basic Linear Algebra Subprograms(BLAS).

The method from LAPACK library used to compute the solution for the linear system with a square matrix A and multiple right-hand sides is **dgesv**.

The signature of this method is: ***dgesv(n, nrhs, A, lda, ipiv, b, ldb, info)***, where n is an integer which represents the number of linear equations, $nrhs$ is an integer which represents the number of column from b matrix, A represents the elements from matrix A, lda is the leading dimension of A, $ipiv$ is an integer array which represents the pivot indices that define the permutation matrix, b represents the elements from matrix b, ldb is an integer for leading dimension of b, $info$ is an integer.

4 Results and discussion

The study proposes two options for approaching the creation of linear system matrices: the first option involves recalculating the values for the real and imaginary part at each step, and the second option proposes storing the common part Y_p^q of formula (10) to optimize the calculations.

For our experiment we used three 3D models: a small one with 422 points, a medium one with 1394 points and a big one with 4242 points, iterate over multiple L_{max} values and obtained the following results with optimizations:

422 points model			
L_{max}	System Size	Compute System	Solve System
3	32x32	0.13s	0.006s
5	72x72	0.635s	0.01s
7	128x128	1.8s	0.007s
10	242x242	6.65s	0.008s
20	882x882	1min 31s	0.019s
30	1922x1922	8min 9s	0.098s
40	3362x3362	26min 25s	0.428s
50	5202x5202	1h 4min	1.2s

1394 points model			
L_{max}	System Size	Compute System	Solve System
3	32x32	0.397s	0.008s
5	72x72	1.96s	0.007s
7	128x128	5.95s	0.006s
10	242x242	21s	0.008s
20	882x882	5min 20s	0.024s
30	1922x1922	26min 37s	0.097s
40	3362x3362	1h 24min	0.365s
50	5202x5202	3h 31min	1.181s

4242 points model			
L_{max}	System Size	Compute System	Solve System
3	32x32	1.21s	0.006s
5	72x72	5.66s	0.006s
7	128x128	18.32s	0.007s
10	242x242	1min 4s	0.007s
20	882x882	15min 30s	0.33s
30	1922x1922	1h 18min 29s	0.325s
40	3362x3362	4h 15min 45s	0.355s

According to the above results, we could observe a significant improvement in the time required to find the solution of the linear system. This is possible with the help of parallel computation, the size no longer becomes an impediment and we can solve the linear system almost instantly.

5 Conclusions

Due to the improvements made to our implementation we obtained a much better computation time than the results obtained by [6], which was possible with the help of LAPACK library.

Finding the solution of linear systems has become a solvable problem that we no longer have to worry about, as we managed to find a solution in almost real time. Parallel computation has taken an essential place in our work in order to get the best time possible.

From the point of view of optimizations, the use of values in formula 10 proved to be an inspired choice, as we observed significant improvements in computing time.

Our target is to perform these reconstructions in real time for a large number of points. The goal has been achieved, as we can calculate in almost real time for a set of given 3D points of the order of hundreds. The challenge is to use a larger set of 3D points in order of thousands. As we can easily remark the time to calculate the linear system matrix increases dramatically as we get a larger bandwidth L_{max} while the time to solve the resulted linear system is usually less than 1 second. To increase the calculation speed for system elements calculation we can use the parallel computation.

References

[1] Greenbaum, A., *Iterative methods for solving linear systems*, Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[2] Commandeur, F., *Segmentation of prostate from CT scans using combined voxel random forests classification with spherical harmonics regularization*, Oscar Acosta, Antoine Simon, Juan David Ospina Arango, 2015.

[3] Dillenseger, J.-L. *Spherical harmonics based intrasubject 3-D Kidney modeling/registration technique applied on partial information*, Hélène Guillaume, Jean-Jacques Patard, 2007.

[4] Kämäräinen, T., *Harmonic analysis of surface instability patterns on colloidal particles*, Janne Raula, Mariko Ago, Jani Seitsonen, 2018.

[5] Mousa, M., *Efficient spherical harmonics representation of 3D objects*, Chaine R., Akkouche S., Galin Liris E., 2007.

[6] Richter, D., Abdellaoui, S, Bekkaoui, F., Monescu, V., Strassman, G., *Mathematical modeling of anatomical structures by means of spherical harmonics*, In 5th IEE International Symposium on Biomedical Imaging, 2008.

[7] Saad, Y. *Iterative methods for sparse linear systems*, SIAM, Philadelphia, 2003.

[8] Zou, C., Zhao, G. and Hancock, E.R., *Reconstructing 3D facial shape using spherical harmonics*, In: Foggia P., Sansone C., Vento M. (eds) Image Analysis and Processing – ICIAP 2009. ICIAP 2009. Lecture Notes in Computer Science, vol 5716. Springer, Berlin, Heidelberg (2009), 949-957.