Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 00 (2016) 000–000

Procedia Engineering

www.elsevier.com/locate/procedia

25th International Meshing Roundtable

Relaxed Lepp-Delaunay algorithms for the refinement / improvement of triangulations

Pedro A. Rodriguez[a], Maria-Cecilia Rivara[b,*]

[a]*Departamento de Sistemas de Información, Universidad del Bio-Bio, Av. Collao 1202, Concepción, 4051381, Chile*
[b]*Departamento de Ciencias de la Computación, Universidad de Chile, Avenida Beauchef 851, Santiago, 8370456, Chile*

Abstract

We discuss serial and multicore Relaxed Lepp-Delaunay algorithms for triangulation refinement, based on inserting the centroid of associated terminal triangles (that share local longest edge in the mesh), and where a neighborhood parameter K is used to constrain the edge flipping propagation around the terminal edge. Empirical results on a multicore Relaxed Lepp-Delaunay centroid algorithm, show that an efficient and scalable multicore algorithm was obtained.

Peer-review under responsibility of the organizing committee of IMR 25.

Keywords: centroid; Delaunay; improvement; Lepp; Lepp-Delaunay; multicore; relaxed Lepp-Delaunay; refinement; thread;

1. Introduction

Longest-edge refinement algorithms for triangulations, based on bisecting the triangles by the longest-edge, were designed to support the development of adaptive finite element software and to guarantee the construction of refined triangulations that maintain the quality of the input mesh [1]. Later the longest-edge propagating path (Lepp) concept was introduced by Rivara [2] to design both the Lepp-bisection algorithm (an efficient and simple longest-edge algorithm) and Lepp Delaunay algorithms for the automatic construction of quality triangulations.

Lepp-Delaunay algorithms combine the Lepp concept and Delaunay insertion of the selected points. Lepp-centroid algorithm has been studied by Rivara and Calderon [5] and Lepp midpoint algorithms has been studied by Bedregal and Rivara [3]. A study on multicore Lepp-bisection algorithm was presented in [6].

In this paper we propose a Relaxed Lepp-Delaunay method to refine and improve triangulations, where the delaunization step is relaxed by using a parameter K that constrain the edge flipping propagation around the terminal edge. We present empirical results on a multicore relaxed Lepp-Delaunay algorithm for solving the quality triangulation problem. This method generalizes the Lepp-Centroid Delaunay method discussed in [5].

* Corresponding author. Tel.: +56-2-2978-4365 ; fax: +56-2-2689-5531.
E-mail address: mcrivara@dcc.uchile.cl (Maria-Cecilia Rivara)

1877-7058
Peer-review under responsibility of the organizing committee of IMR 25.

2. Lepp-Delaunay centroid method

An edge E is called a terminal edge [2] in triangulation τ if E is the longest edge of every triangle that shares E, while the triangles that share E are called terminal triangles [2]. Note that in 2-dimensions either E is shared by two terminal triangles t_1, t_2 if E is an interior edge, or E is shared by a single terminal triangle t_1 if E is a boundary (constrained) edge. See Figure 1 where edge AB is an interior terminal edge shared by two terminal triangles t_3, t_4.

For any triangle t_0 of a conforming triangulation τ, the longest-edge propagating path of t_0, denoted by $Lepp(t_0)$, is the finite list of increasing triangles $t_0, t_1, t_2, ..., t_{n-1}, t_n$, such that t_i is the neighbor triangle of t_{i-1} on a longest edge of t_{i-1}, for $i = 1, 2, ..., n$ [2]. Note that in general t_{n-1}, t_n are terminal triangles sharing an interior terminal edge.

For improving a triangle t, the first Lepp-Delaunay algorithm [2] repeatedly selects the midpoint of the terminal edge which is Delaunay inserted in the mesh until the triangle t is refined. Later Rivara and Calderon introduced the Lepp-Delaunay centroid algorithm [5] where the centroid of the terminal quadrilateral formed by a couple of terminal triangles is selected for Delaunay point insertion. For an illustration of the centroid algorithm see Figure 1, where for improving t_0, the centroid P of the terminal triangles t_3, t_4 is Delaunay inserted which produces the triangulation of Figure 1 (b). Then, for improving t_0 (that remains in the mesh), the centroid of the terminal triangles t_0, t'_1 is inserted, which destroys t_0.

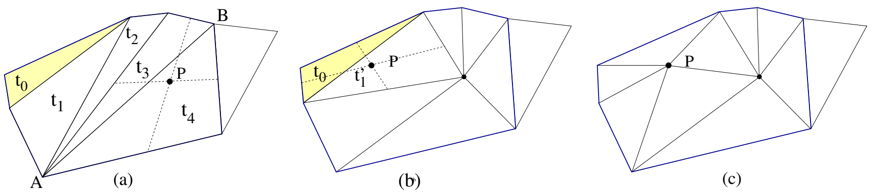

Fig. 1. Lepp Delaunay (centroid) method.

3. A serial relaxed Lepp-Delaunay algorithm

In this algorithm we use a parameter K that allows to define a neighbor set of triangles NS_k that constrain the edge flipping propagation. In this way, we use a quasi-Delaunay point insertion operation.

Algorithm 1 Relaxed-Lepp-Delaunay Algorithm(τ_0, θ_{tol}, K)

Input: τ_0 initial conforming mesh, threshold angle tolerance θ_{tol} and parameter K.
Output: An improved conforming triangulation τ_f.
Find $S \subset \tau$ the set of triangles with smallest angle $< \theta_{tol}$.
while $S \neq \phi$ **do**
  Select a triangle t from S.
  while t remains in τ **do**
    Find Lepp of t and compute the centroid M of the terminal quadrilateral.
    Find set $NS_K(E)$.
    Insert the centroid M by using the relaxed Lepp-Delaunay point insertion (constrained to $NS_k(E)$).
    Update S.
  end while
end while

Definition. Given a terminal edge E, for K=0, the neighbor set of triangles $NS_0(E)$ includes the terminal triangles associated to E. For $K > 0$, the neighbor set of triangles $NS_K(E)$ includes the triangles of $NS_{K-1}(E)$ and its exterior edge-adjacent triangles (see Figure 2).

The serial relaxed Lepp-Delaunay algorithm proceeds as follows: for each bad quality triangle t_0 to be refined, the algorithm finds Lepp(t_0), the terminal edge E, the centroid M of the terminal triangles and a set $NS_K(E)$ over which the quasi-Delaunay point insertion operation is performed. Triangulations (b), (b), (c) of Figure 2 show the NS_K sets for K = 0, 1, 2. Figure 2 (d) shows the quasi-Delaunay mesh obtained after quasi-Delaunay insertion of M for K=2.

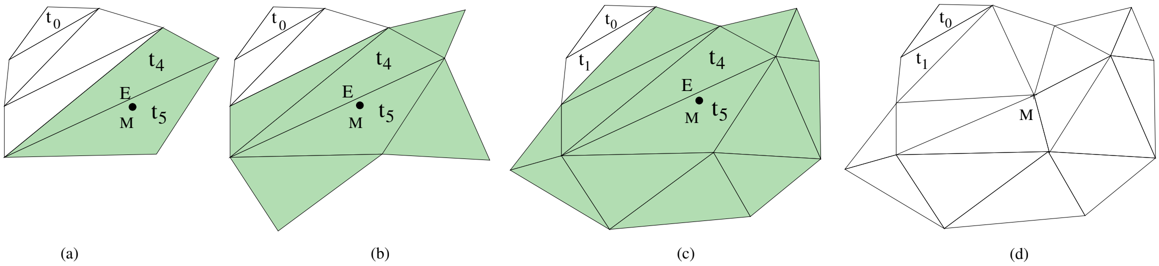

Fig. 2. Shadow triangles identify $NS_K(E)$. (a) $NS_0(E)$ includes the terminal triangles; (b) $NS_1(E)$ includes terminal triangles ($NS_0(E)$) and their immediate neighbors; (c) $NS_2(E)$ includes $NS_1(E)$ and their immediate neighbors; (d) After the quasi-Delaunay insertion of centroid M for K=2.

4. Practical behavior of the serial algorithm as a function of *K*

We used the serial relaxed Lepp-Delaunay algorithm for studying both the evolution of the angle distribution and the number and percentage of the non-Delaunay triangles obtained in the final mesh for different values of K. Table 1 summarizes these results for $\theta_{tol} = 30^o$. This includes the size of the meshes, number and percentage of non-Delaunay triangles and the execution time for different values of K. Note that the final meshes have approximately the same number of elements (vertices and triangles), but the number and percentage of non-Delaunay triangles in the final meshes are different. Note that when K=0 the algorithm only inserts the centroid into a couple of terminal triangles without carrying out edge flipping operations.

Table 1. Final meshes and Percentage of Delaunay triangles for input and final meshes obtained from different values of K, threshold angle 30^o.

	Vertices	Triangles	Non-Delaunay Triangles (NDT)	Percentage of NDT	Time (ms)
Init Mesh →	2,999,998	5,999,953	0	0	0
Final mesh, K=0	10,929,370	21,841,912	2,552,911	11.69	330,511
Final mesh, K=1	10,885,375	21,753,875	40,004	0.1839	370,841
Final mesh, K=2	10,867,452	21,718,098	44	0.000203	331,952
Final mesh, K=3	10,864,202	21,711,627	12	0.000055	362,422
Final mesh, K=4	10,863,910	21,711,078	10	0.000046	368,190
Final mesh, K=7	10,863,878	21,711,007	0	0.000000	445158
Final mesh, K=10	10,863,826	21,710,922	0	0.000000	589,493

As expected the percentage of non-Delaunay triangles obtained decreases when the value of K increases. However this remains very low for $K \geq 3$, which suggests that either K=2 or K=3 is a good parameter value.

Table 2 summarizes the distribution of the smallest angles (between 0 and 60 degrees) in triangle percentage for the initial and final meshes, for K=0,1,2,3,4,7,10. Note that good quality meshes formed by triangles with good internal angles (threshold 30^o) are obtained even when the mesh is not fully Delaunay. Note that even for K=0 all the bad quality triangles (needle, cap, etc) are eliminated from the mesh.

Table 2. Distribution (in triangle percentage) of smallest angles for different values of K, $\theta_{tol} = 30^o$.

Angle distribution (triangle %)						
Degrees	$0^o - 10^o$	$10^o - 20^o$	$20^o - 30^o$	$30^o - 40^o$	$40^o - 50^o$	$50^o - 60^o$
Initial Mesh	6.11	16.99	24.37	25.63	9.59	7.32
Final Meshes (triangle %) for different values of K						
K \ Degrees	$0^o - 10^o$	$10^o - 20^o$	$20^o - 30^o$	$30^o - 40^o$	$40^o - 50^o$	$50^o - 60^o$
K=0	0	0	0	42.0955	44.6932	13.2113
K=1	0	0	0	36.9587	48.7855	14.2558
K=2	0	0	0	37.0057	48.7315	14.2627
K=3	0	0	0	37.0166	48.7239	14.2595
K=4	0	0	0	37.0170	48.7235	14.2595
K=7	0	0	0	37.0169	48.7235	14.2595
K=10	0	0	0	37.0169	48.7236	14.2595

5. Practical performance of the parallel Relaxed Lepp-Delaunay algorithm

Given an input triangulation τ, a set $S \subset \tau$ of bad quality triangles and a parameter K. Then for each triangle t in S, the parallel relaxed algorithm proceeds as follows: (1) Lepp(t), and centroid M of the terminal quadrilateral are computed; (2) The $NS_K(E)$ set is found; (3) If $NS_K(E)$ is computed without detecting collisions, then the centroid M is Delaunay inserted into the mesh. Otherwise, the computation is stopped and the core proceeds to pick up a new triangle from S. Algorithm 2 summarizes the parallel relaxed Lepp-Delaunay centroid algorithm:

Algorithm 2 Multicore Lepp-Delaunay algorithm

Input: τ_0 an initial mesh, threshold angle tolerance θ_{tol}, parameter K.
Output: An improved conforming triangulation τ_f.
Find $S \subset \tau$ the set of triangles with smallest angle $< \theta_{tol}$.
while $S \neq \phi$ **do**
  Take a triangle t from S.
  while t remains in τ **do**
    Find Lepp(t).
    Find $NS_K(E)$ and compute centroid M of the terminal quadrilateral.
    if Collision is detected while computing a Lepp(t) or NS_K **then**
      Destroy Lepp(t) and take a new triangle t from S.
    else
      Lock the triangles of NS_K.
      Insert the centroid M into the mesh by using relaxed Lepp-Delaunay point insertion.
      Update S.
    end if
  end while
end while

We have used a computer with two Intel Xeon E5-2660 processors (20 physical cores, 10 core per socket) for testing the algorithm behavior. We used several triangulations of sets of randomly generated points over a rectangle. The input domain was divided in a grid of rectangles in order to distribute the triangles and the workload between the threads. Empirical work shows that the multicore algorithms (for $K \leq 3$) have good scalable behavior until 20 processors are used, In Table 3 and Figure 3 we present results for the meshes of Table 1 (input and final meshes of approximately 6 millions and 21.7 millions of triangles respectively, for $\theta_{tol} = 30^o$). Table 3 shows the efficiency behavior and Figure 3 shows the speedup behavior.

Table 3. Performance measure: efficiency; threshold angle 30^o; Intel Xeon E5550.

Efficiency							
K	1P	2P	4P	8P	10P	16P	20P
0	1.0	0,79	0,73	0,68	0,67	0,77	0,66
2	1.0	0,76	0,68	0,64	0,61	0,75	0,64
3	1.0	0,68	0,65	0,61	0,60	0,72	0,60
4	1.0	0,66	0,58	0,53	0,52	0,53	0,44

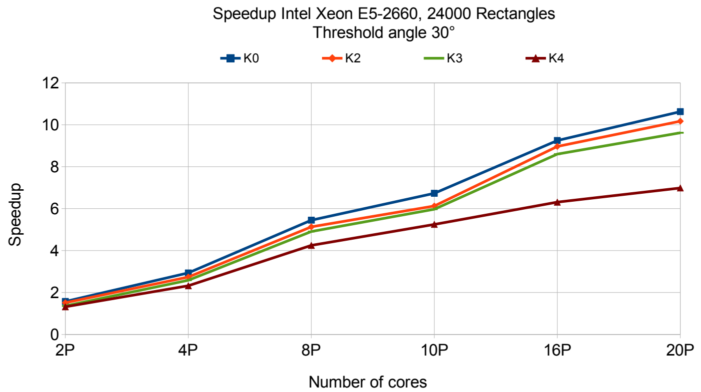

Fig. 3. Speedup for K=0,2,3,4, threshold angle 30^o. Intel Xeon E5-2660, 2, 4, 8, 10,16 and 20 cores.

Acknowledgements

Work partially supported by Departamento de Ciencias de la Computación, Universidad de Chile, Departamento de Sistemas de Información and Research Group GI150115/EF, Universidad del Bio-Bio. Powered@NLHPC: This research was partially supported by the supercomputing infrastructure of the NLHPC (ECM-02)

References

[1] Maria-Cecilia Rivara. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. International Journal for Numerical Methods in Engineering, 20(4):745–756, 1984.
[2] Maria-Cecilia Rivara. New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations. International Journal for Numerical Methods in Engineering, 40(18):3313-3324, 1997.
[3] C. Bedregal C, M-C. Rivara, A study on size-optimal longest edge refinement algorithms. In: proceedings of the 21st International Meshing Roundtable, pp. 121–136, (2013).
[4] M-C. Rivara, Lepp-bisection algorithms, applications and mathematical properties. Appl. Numer. Math., 59(9):2218–2235, (2009).
[5] Maria-Cecilia Rivara and Carlo Calderon. Lepp terminal centroid method for quality triangulation. Computer-Aided Design, 42(1):58–66, 2010.
[6] M-C. Rivara, P. Rodriguez, R. Montenegro, G. Jorquera. Multithread parallelization of lepp-bisection algorithms. Appl. Numer. Math., 62(4):473–488, (2012).