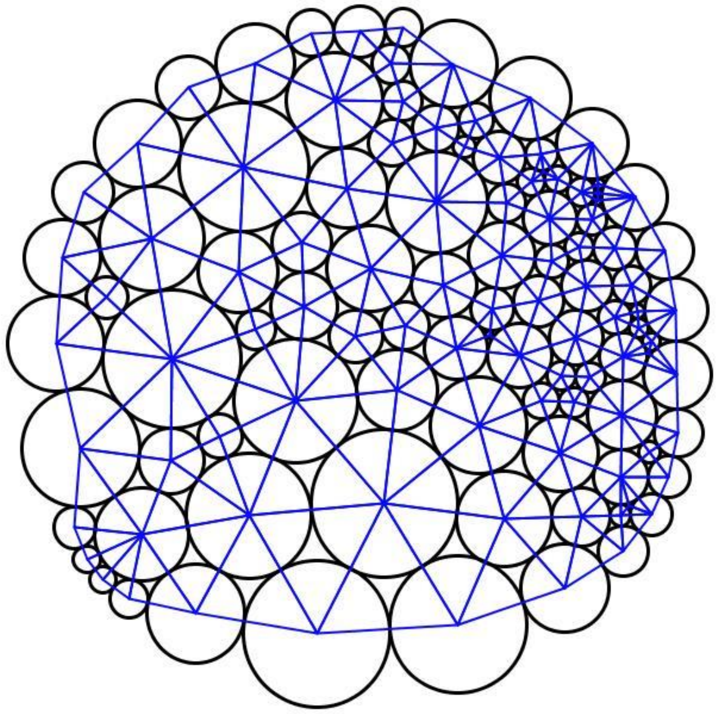
GOPack -- a Matlab Package

by Chuck Collins, Gerald Orick, Ken Stephenson

A circle packing is a configuration of circles satisfying a prescribed pattern of tangencies. The topic was introduced in 1985 by William Thurston [1] and the foundational paper connecting circle packing to conformal mapping is Rodin/Sullivan's proof of Thurston's Conjecture [2]. Since then a comprehensive theory of discrete analytic functions has emerged, one which both mimics and approximates the classical theory. The principal reference is Ken Stephenson's book [3] and the principal computational software is his Java package, ***CirclePack*** [4]; visit www.math.utk.edu/~kens for details.



The ***GOPack*** package in *Matlab* is concerned with the computation of certain types of circle packings. For traditional methods, see [5]. The new GOPack method originated with Gerald Orick; for details see [6].

INPUT: the user provides a triangulation *T* of a disc or a sphere with vertex set *V*.

OUTPUT: ***GOPack*** produces a circle packing *P* for *T*; that is, a collection of circles having a circle *c(v)* for every vertex *v* of *T* such that if *<u,v,w>* is a face of *T*, then *<c(u),c(v),c(w)>* is a triple of mutually tangent circles in *P*.

GOPack package includes:

(Package available on *GitHub* and at www.matworks.com/matlabcentral/fileexchange)

data/

Sample data files are provided (all in *ascii*). The preferred format is the **.p* format of ***CirclePack***. Details on input/output formats are in '*doc/GO_formats.txt*'.

doc/

Main resources are the mathematical literature and in the commented code itself. Minimal documentation is provided.

scripts/

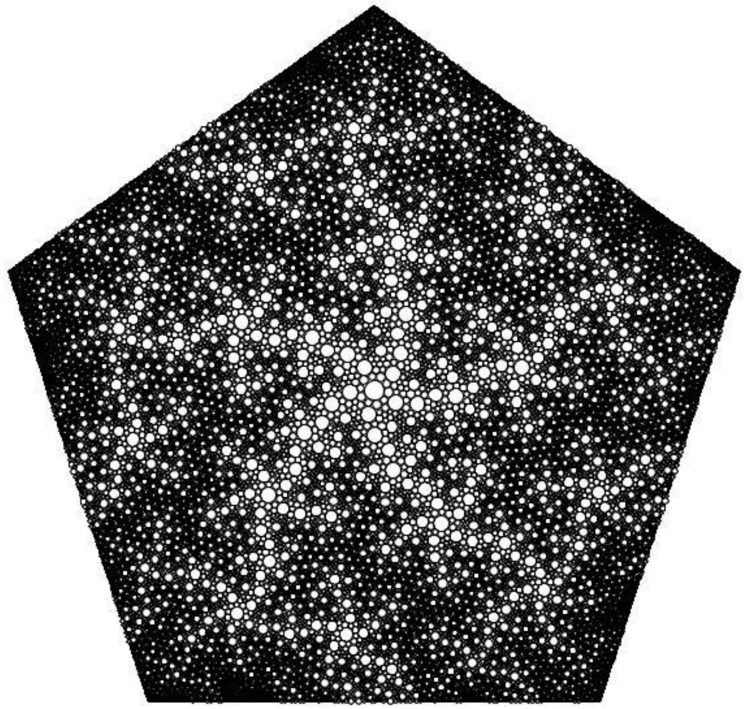
Sample scripts illustrate all the main types of computations. Functions are provided to generate various types of random triangulations in many instance, while in other cases packing files in ***data/*** provide the data.

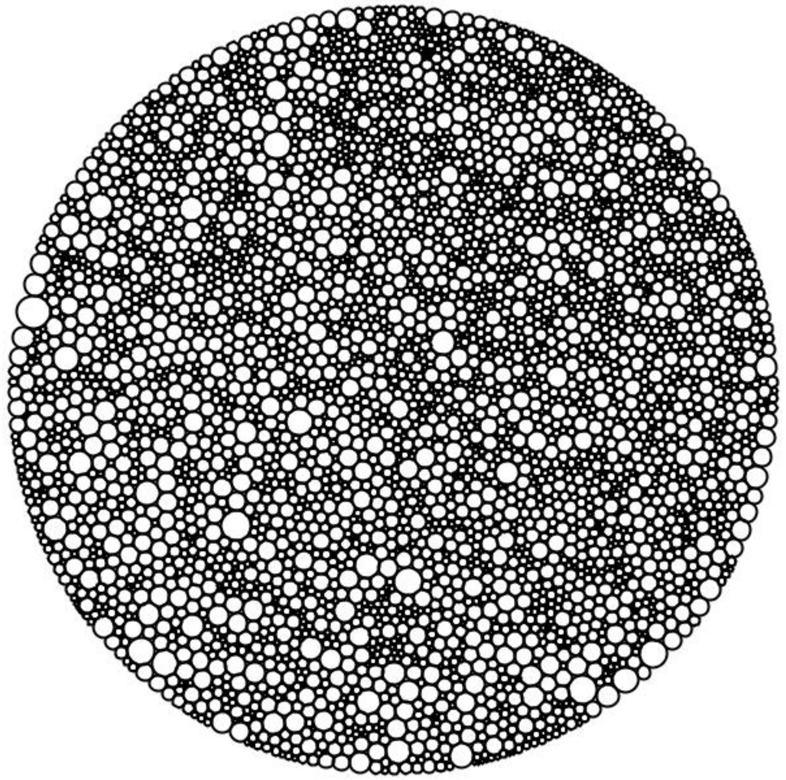
code/

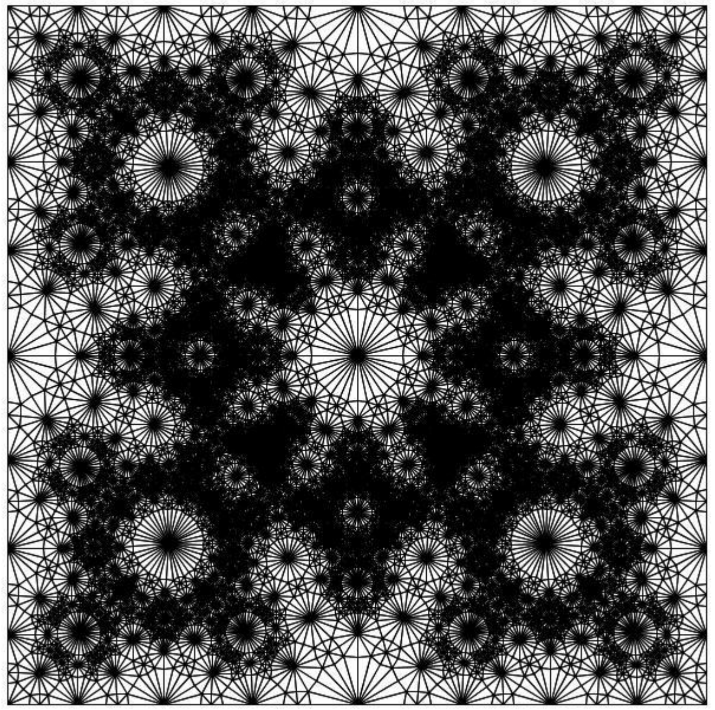
The ***GOPacker*** object is the central device. Additional utility routines are provided.

NOTE: To run examples, the user should add **code/**, **data/**, and **scripts/** to their *Matlab* path.

Examples from our scripts:

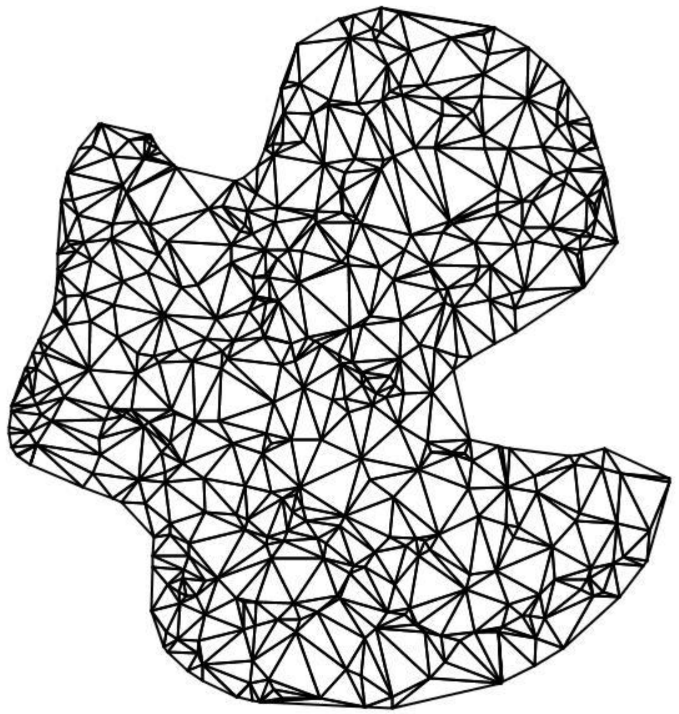

TwistPent_example.m
~ 40000 circles

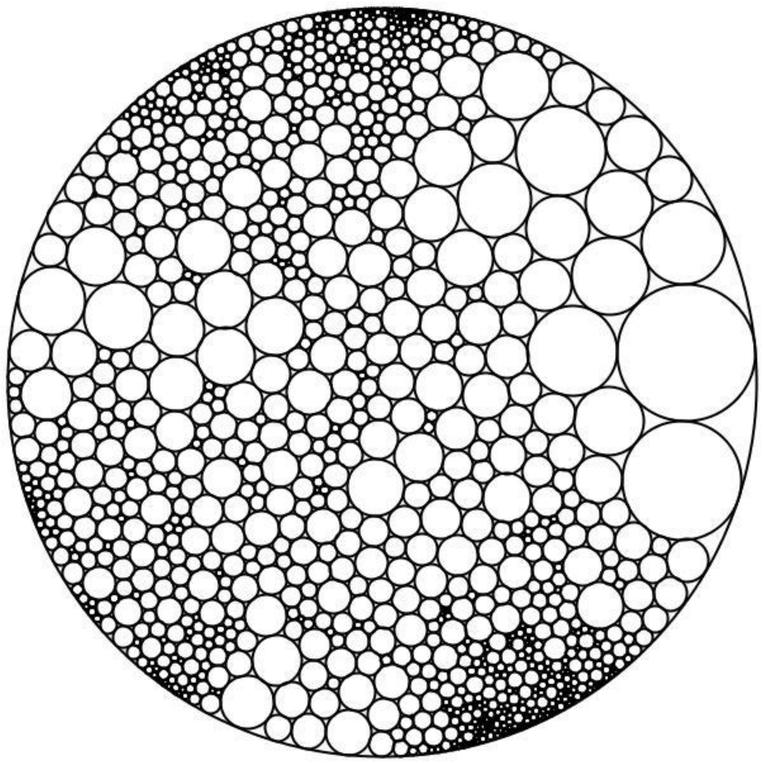

RandDisc_example.m
~ 5000 circles


Lace_l3_example.m (embedding)
~ 42000 circles

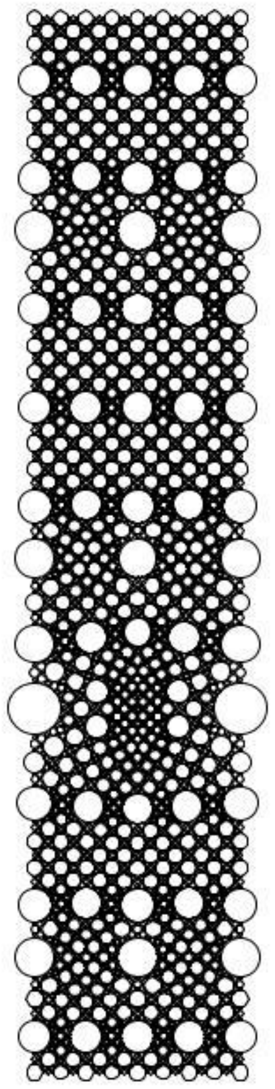
Examples from our scripts (cont):

Mixed_tiling_example.m
~10000 vertices



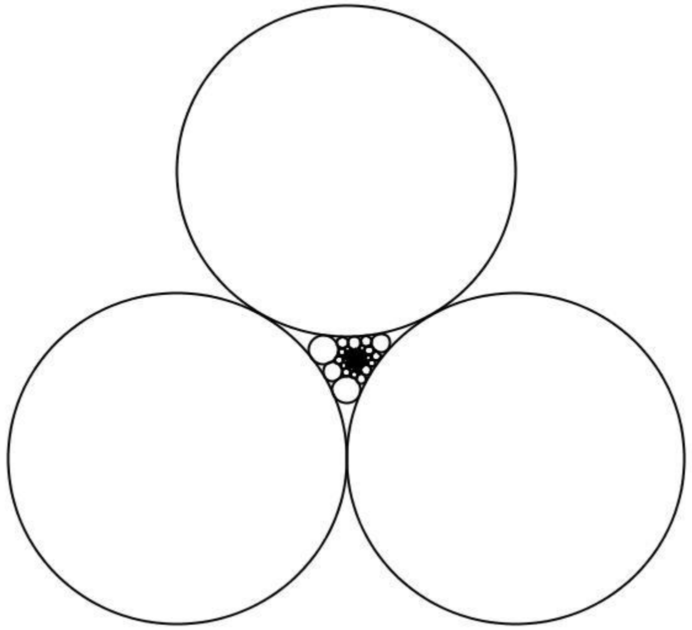


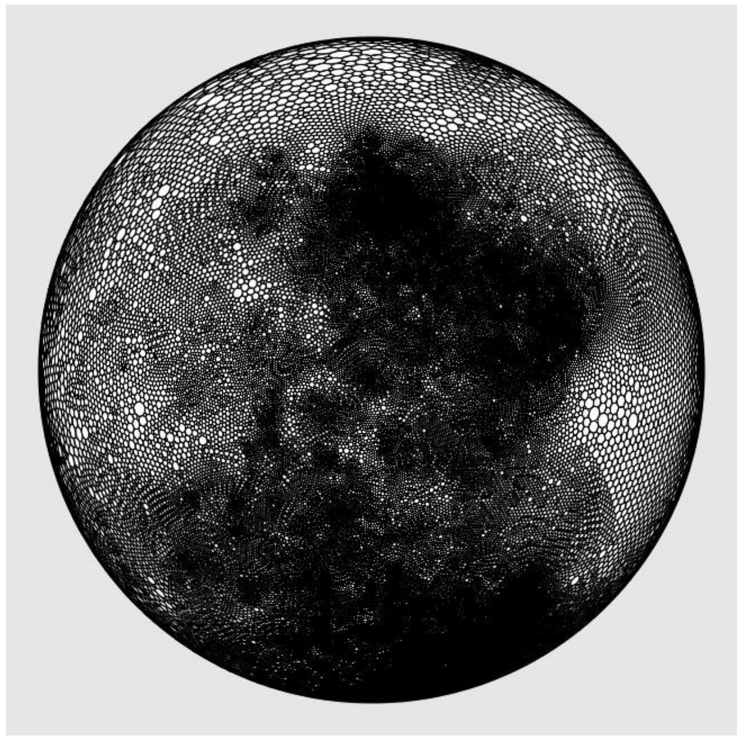
RandJordan_exmaple.m
~500 vertices
(triangulation and its hyperbolic circle packing)



Circle Packings of the Sphere: (run ***BrainSph_example.m***)

Circle packings for triangulations of a sphere are computed in ***GOPack*** using Euclidean data, giving packings of the disc such as that on the left below. By default, however, when saved they are stereographically projected to circles on the Riemann sphere, with spherical radii and centers in spherical coordinates and normalized to put the centroid of the tangency points at the origin in 3-space. ***GOPack*** cannot display on the sphere, but reading the output file into ***CirclePack*** allows pictures like that on the right below.


~175,000 vertices, euclidean


On the sphere in ***CirclePack***

Commands:

The ***code/*** subdirectory should be added to *Matlab's* path.
The central object is a ***GOPacker***. Here are its main methods:

- **readpack('filename')**: Read in the combinatorics of a triangulation *T*. This must be a triangulation of a topological disc or topological sphere. The file may be in the *.p format common for CirclePack (e.g., ***data/lace42000_K.p***) or a simple listing of triples of vertices, as can be produced in many software packages (e.g. ***data/rawTri***). Set the *Matlab* path to read/store data in desired locations.

- **riffle(N)**: This carries out up to *N* (default, 20) repacking computation cycles. Each cycle includes: **(A)** The layout of circles corresponding to boundary vertices (if *T* is a sphere it has three faux boundary vertices). **(B)** A weighted Tutte-type embedding of the interior vertices; this is were linearity enters and sparse matrices are used. **(C)** Computation of new "effective" radii. The code then returns to step **(A)** for the next cycle.

- **setMode(m,[crns, [angs]]):** mode 1 is for maximal packing, mode 2 for polygonal packing. A polygonal packing requires boundary vertices (default to 4) to serve as corners. These may be specified in a packing file along with the combinatorics or given specifically in the function call, else they will be chosen at random. Default is to specify equal angles at all corners; optional angles may be specified, but can fail to pack without underlying symmetries.

(cont.)

- **packStatus()**: Provide summary information on the combinatorics and the mode of the packing. The key measure of numerical quality is given by maximal '*visualError*'. Repacking is generally considered to be done when the worst visual error is less than 1%, meaning that whenever circles for vertices u and v are intended to be tangent, then the distance between their centers varies by less that .01 from the sum of their radii.

- **show([options]):** Display the circle packing in a figure. By default, this displays the circles, but **show('label','circle')** will show circle indices as well, and **show('object','face')** will show the embedding of *T* determined by circle centers. For more sophisticated images, read packings into ***CirclePack***.

- **writepack('filename',[e_opt])**: Packing data is saved in the **.p* format associated with ***CirclePack***. If *T* triangulates a sphere, the packing is normalized and saved with spherical radii and centers; the sphere packing cannot be viewed, but the packing can be read into ***CirclePack*** and viewed there. If a maximal packing of a topological discs, then by default it is save with hyperbolic radii and centers, appropriate for manipulation in the unit disc (the hyperbolic plane). In any case, if the packing flag **e_opt** is true, the packing will be saved in Euclidean form.

Random Circle Packings:

Random triangulations are important in many disciplines, and circle packing is a robust way to embed them. Hence we include functions for creating geometrically random triangulations using Poisson Point Processes and Delaunay triangulations; see our samples in **scripts/**.

randomDisc(N): geometrically random triangulation of the disc, **N** interior vertices.

randomSphere(N): geometrically random triangulation of a sphere, **N** vertices.

randomRectangle(N,[asp]): geometrically random triangulation of a rectangle, optionally with aspect **asp**.

randomSquare(N): geometrically random square, **N** vertices.

randomTri(N,B,X,Y): geometrically random triangulation of a Jordan domain with boundary curve **(X,Y), N** interior and **B** boundary vertices.

Caveats:

GOPack handles only *maximal packings* of the disc or sphere and *polygonal packings* in certain situations. There are many circle packings not amenable to the ***GOPack*** algorithm (see [5], [6]) but which can generally be computed in ***CirclePack*** using the traditional packing algorithm. ***GOPack*** is best with packings having large numbers of vertices; when the algorithm applies it is generally far superior to traditional methods: not only is it faster, but it avoids layout problems that can confound traditional methods. C++ versions of the ***GOPack*** algorithm will soon be available in ***CirclePack***.

References:

[1] William Thurston, “The finite Riemann mapping theorem”, hour lecture, Intern. Symposium, Purdue University, 1985.

[2] B. Rodin and D. Sullivan, “The convergence of circle packings to the Riemann mapping”, J. Diff. Geom., 26, 1987.

[3] Ken Stephenson, “Introduction to Circle Packing: the Theory of Discrete Analytic Functions”, Cambridge University Press, New York, 2005.

[4] Ken Stephenson, ***CirclePack***, open source java software package for creation, manipulation, display, and analysis of circle packings, www.math.utk.edu/~kens/CirclePack

[5] C. Collins and K. Stephenson, “A circle packing algorithm”, Computational Geometry: Theory and Applications, 25, 2003.

[6] C. Collins, G. Orick, and K. Stephenson, “A linearized circle packing algorithm”, to appear.

Note: the principal contact for questions is Ken Stephenson, Mathematics, Univ. of Tenn., Knoxville