



# Robert J. Lang Origami

[About](#)[Crease Patterns](#)[Press](#)[Compositions](#)[Books](#)[FAQ](#)

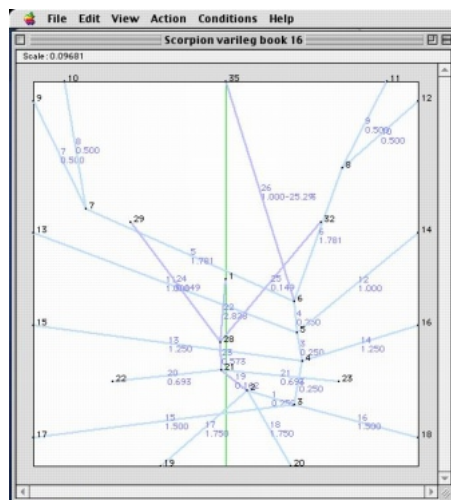
## TreeMaker

Tags: [software](#)

### Background

In 1989, I wrote an article for the magazine *Engineering & Science* about the state of technical folding, which, even then, seemed to be progressing by leaps and bounds due to an infusion of scientific and mathematical principles. In recounting some of the connections between origami, math, and technology, I wrote:

Computing succumbed to the appeal of folded paper when, in 1971, Arthur Appel programmed an IBM System 360 computer to print out simple geometric configurations at the rate of more than one hundred a minute. Ninety percent were considered unsuccessful, but it raises an interesting question: could a computer someday design a model deemed superior to that designed by man? Since so much of the process of design is geometric, the prospect is not as outrageous as it may seem.



A screen shot of the design screen for a scorpion using *TreeMaker 4*. The light blue lines define the tree; the desired lengths are entered as numerical values.

The ensuing decade saw this speculation turned into reality (and no one was more surprised than me!) Within the next few years, the powerful design techniques of circle-river packing had been discovered and systematized by multiple folders, including myself, Toshiyuki Meguro in Japan, and others. Building on these concepts, I set out in the early 1990s to do exactly what I'd speculated on in my *E&S* article: write a computer program that could "design" a non-trivial origami figure based on a description of the number, lengths, and connectedness of the flaps. After several months of work I had succeeded; and the result, was a computer program, which I dubbed *TreeMaker*, since it started with a particular type of stick figure (called a *tree* in graph theory).

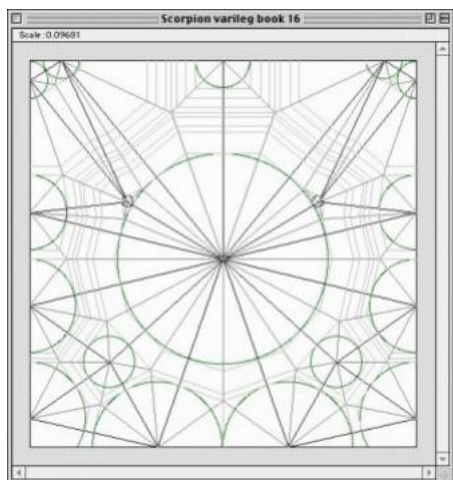
Initially, *TreeMaker* was little more than a mathematical curiosity and a tool for exploring the mathematical theory of how to design a base. The first version did little more than solve the associated circle-packing problem, but gave no clue as to how to fill in the bulk of the creases. Over the years, as my understanding of crease patterns grew, I added what knowledge I gained to *TreeMaker*, adding algorithms for computing molecules, imposing symmetries, and

### Articles by topic

- ◆ admin (0)
- ◆ amusement (12)
- ◆ blog (25)
- ◆ commercial (3)
- ◆ copyright (4)
- ◆ crease patterns (3)
- ◆ design (14)
- ◆ diagramming (2)
- ◆ dialog (1)
- ◆ fine art (9)
- ◆ history (16)
- ◆ links (2)
- ◆ mathematics (10)
- ◆ monumental (2)
- ◆ outdoors (1)
- ◆ paper (3)
- ◆ software (11)
- ◆ tattoos (2)
- ◆ technology (10)
- ◆ videos (1)
- ◆ wildlife (1)

introducing strain into the stick figure, releasing new versions every few years.

By 1998, I had finished version 4.0 of *TreeMaker*, which, in addition to including many algorithms for the origami design, incorporated a powerful numerical optimization code, *CFSQP* developed by Professor Andre Tits and his students at the University of Maryland (the code is now licensed and maintained by [AEM Design](#); this made it run very fast. And suddenly, *TreeMaker* was no longer an academic curiosity; it had become a powerful tool, capable of constructing the full crease pattern for a wide variety of origami bases.

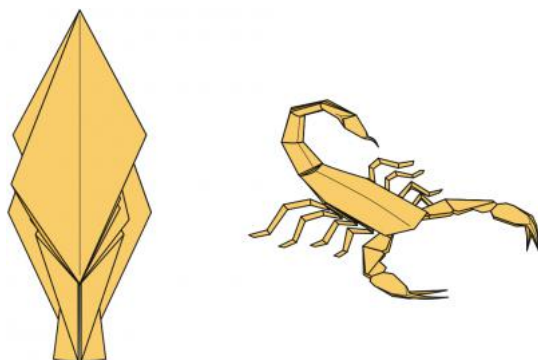


A screen shot of the computed crease pattern for a scorpion using *TreeMaker* 4. Circles corresponding to leaf nodes (terminal flaps) are shown to aid intuition.

In fact, version 4 of *TreeMaker* could solve for crease patterns that I couldn't construct by any other way — by which I mean, using pencil and paper. I (and at this writing, most other composers of technical origami) have usually designed my/our compositions on paper using the geometric concepts collectively described as "circle/river packing." (Those concepts are described in my book, *Origami Design Secrets*; see [its page](#) for details.) No computer is needed for this type of design; one simply sketches circles and the crease patterns known as molecules, constructs the crease pattern, and folds away. Occasionally a bit of algebra is needed to work out an initial reference point or two (see [ReferenceFinder](#) for more on this).

*TreeMaker* allows one to set up quite elaborate relationships between flaps, their lengths, and their angles: far more complex relationships than are possible using pencil-and-paper origami design. Which meant that it was now possible, with *TreeMaker*, to solve for origami bases that truly were more complicated than anything a person could design by hand.

The flip side of this power is: these complex crease patterns are extremely difficult to fold. Since all you're given is the crease pattern, it's up to you to devise a step-by-step folding sequence for all the creases, and the subsequent assembly into a base can also be fiendishly difficult. But the value of *TreeMaker* is that it combines novelty with efficiency: the patterns constructed are commonly the most efficient solutions possible for a given stick figure, and they are just as often totally new structures in the world of origami.



The folded base, and a finished model folded from this base. [See here](#) for a photograph of the origami figure.

And that opens up another question: if you are an origami composer (or wish to be), do you

need to use *TreeMaker*? The answer is: absolutely not. The vast majority of the world's composers of technical origami don't use it; in fact, I don't use it for the majority of my own designs. What I do use it for is for what you might call "rapid prototyping"; quickly examining 3 or 4 (or 5 or 10) different general arrangements of flaps in a base before settling on one particular configuration as the focus of my design. For that purpose, it's an invaluable tool in my arsenal. And for several of my designs, notably the [Scorpion varileg, opus 379](#), [White-Tailed Deer, opus 550](#), [Mule Deer](#), and [Maine Lobster, opus](#) , couldn't have been designed them without it.

## Description

*TreeMaker* is a program for the design of origami bases. You draw a stick figure of the base on the screen; each stick in the stick figure (the "tree") will be represented by a flap on the base. You can also place various constraints on the flaps, forcing them to be corner, edge, or middle flaps, and/or setting up various symmetry relationships (forcing pairs of flaps to be symmetric about a line of symmetry of the paper, for example). Once you have defined the tree, *TreeMaker* computes the full crease pattern for a base which, when folded, will have a projection (roughly speaking, its "shadow") equivalent to that specified by the defining tree. The crease pattern can be printed out, or copied and pasted into another graphics program for further processing. Crease assignment (mountain or valley) are not computed, but with a few simple rules and some exploration by hand, the proper crease assignment can usually easily be found.

## The Genesis of *TreeMaker 5*

In recent years, I've had the pleasure of working with Professor Erik Demaine and Martin L. Demaine at MIT on various aspects of the mathematical theory of origami. One problem we have attacked has been the challenge of computing the full crease pattern for a *TreeMaker* base, and the related (and much harder) problem of proving that the algorithms of tree theory always give a correct solution (i.e., a flat-foldable base), or if they don't, under what circumstances they do. In my initial development of tree theory, I had found early versions of *TreeMaker* to be an invaluable tool for testing and exploring mathematical concepts, and so in 2003 I set out to update *TreeMaker* to use in this further exploration.

The crease assignment problem is deceptively challenging, because we can identify *most* of the creases by simple inspection:

- All ridge creases are valley creases;
- All gusset creases are mountain creases;
- *Most* axial creases are mountain creases.

The problem is, we don't know *which* of the axial creases are mountain and which are valley. And the last family of crease — hinge creases — were completely unspecified. Fortunately, one could almost always find the crease assignment very quickly by experimenting by folding the pattern produced by *TreeMaker 4*, but it was an annoying lingering hole in the theory that I desperately wanted to plug.

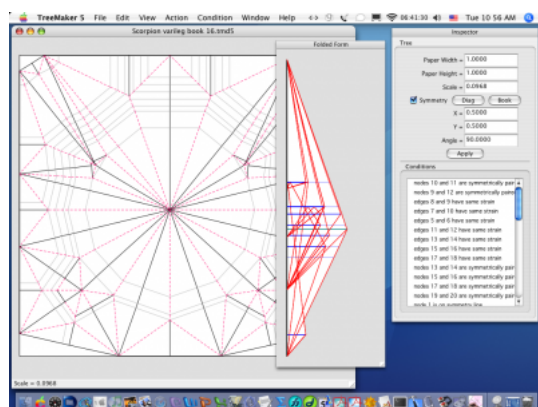
Although it turned out to be far more complicated than I'd originally imagined, by late 2005, I had an algorithm for full crease pattern generation that worked for a wide range of structures and that was implemented in *TreeMaker 5*. Over the same period, I migrated the code base from an Apple Macintosh-specific GUI based on Metrowerks PowerPlant to the cross-platform class library [wxWidgets](#), which offered the promise — now realized — of fully cross-platform versions of *TreeMaker*: this last accomplished with the invaluable assistance of Brazilian programmer Carlos Furuti for GNU/Linux porting and Polish programmer and [wxWidgets](#) team member Włodzimierz 'ABX' Skiba for Windows porting (not to mention all-around help and advice on programming and [wxWidgets](#) from both of them).

One more nice development was that although CFSQP was incredibly faster than my own hand-rolled optimization code, the roughly 1000x improvement in numerical computing speeds over the last decade meant that my own optimization code was now fast enough. This meant that I could release the full *TreeMaker* code. And so I have; you'll find it below.

## What's New in *TreeMaker 5*

Like its predecessors, *TreeMaker 5* lets you draw a stick figure that represents the base you

are after, specifying the lengths and connections between flaps, and lets you set various types of constraints that enforce symmetries in the base (e.g., mirror symmetry) and in the crease pattern (e.g., forcing particular crease angles). What *TreeMaker 5* adds to the mix is:



Screen shot of *TreeMaker 5*, showing the full crease pattern, folded form, and the new Inspector for editing the design.

- Full mountain-valley crease assignments;
- An x-ray image of the folded form of the base;
- Numerous new options for display and simplified editing.

The figure to the right shows *TreeMaker 5*'s take on the scorpion design shown above, with the complete mountain/valley-assigned crease pattern, the folded form of the base, and the new "Inspector" window for editing the tree and its conditions. *TreeMaker 5* is cross-platform: screen shots of supported platforms can be seen at these links:




- [Mac OS X Tiger](#)
- [GNU/Linux/Fedora Core 4](#)
- [GNU/Linux/Ubuntu](#)
- [Microsoft Windows XP](#)
- [Microsoft Windows 2000](#)

## Downloads

The current version of *TreeMaker* is version **5.0.1**. The [version history](#) is here. I will update this page as new versions are released to fix bugs or introduce new features. Email me at [treemaker@langorigami.com](mailto:treemaker@langorigami.com) if you find any bugs (but please check the *Known Issues* list below first). This software is free software with no warranty of merchantability or fitness for any particular purpose; use it at your own risk, and enjoy!

## Binary Downloads

At the moment, *TreeMaker* is available as downloadable binary for three platforms: Mac, GNU/Linux i386, and Windows. The documentation is part of the application, and can be printed from the Help menu.

- Apple Macintosh OS X 10.4+: [TreeMaker.dmg](#) [7.2 MB]   
Download and mount the disk image and drag the TreeMaker application to your Applications folder, then double-click to run. *TreeMaker 5* is a Universal Binary that will run natively on Intel Macs; it requires System 10.4 or later.
- Apple Macintosh OS 9, OS X 10.3.9 or earlier: [TreeMaker 4.0 Manual](#) With Mac OS versions earlier than 10.4, you will have to use *TreeMaker 4.1* running in Classic mode. Because of the licensing terms associated with *TM4*'s use of CFSQP, you'll need to email me the information described in the manual and I will email you the executable. (I strongly urge you to upgrade your Mac; *TreeMaker 5* offers many enhancements over version 4.1.) 
- GNU/Linux i386: [TreeMaker5Linux.sh\\_zip](#) [4.5 MB]   
To install, unzip the archive, then execute the file, as in "sh TreeMaker5Linux.sh" (you'll need write permission to the installation directory, or be the superuser). For details and requisites, [click here](#). Tested on Mandriva 2006, Fedora Core 4, and Ubuntu 5.10.

- Microsoft Windows: [treemaker5-setup.exe.zip](#) [4.9 MB]  
To install, just execute file "treemaker5-setup.exe" and follow the on-screen instructions.



## Standalone Documentation

The *TreeMaker* documentation is included in the application, accessible and printable from the Help menu, but you can also download the documentation and view it separately without the application. Download this archive and expand it; then open the file "index.htm" with any web browser.

- Documentation: [treemaker\\_help.zip](#) [2.8 MB]

## Source Code

The *TreeMaker* source code is released under the GNU Public License, which is included in this download.

- Source Code: [TreeMaker\\_src.zip](#) [3.5 MB]
- Linux users should also download and read this patch: [tm5LinuxPatch081118A.tar.gz](#) [204 kB]

The source package includes full source code plus build systems for Mac, GNU/Linux, and Windows from the common code base. If you would like to try porting *TreeMaker* to a system not listed above (iPhone anyone?), I'd be happy to add you to the official development team.

## Underlying Theory (English+Japanese)

The *TreeMaker 4.0 Manual* [pdf, 784KB] contains a chapter that describes the underlying theory in abbreviated form. (A much fuller description of the theory (and a lot more besides) can now be found in my book, *Origami Design Secrets (2nd Edition)*.) Computer scientist Yohsuke Furuta has translated the theory chapter of the *TreeMaker 4* manual into a bilingual edition (English+Japanese), so if you'd like to read the basic theory (in either language), you can [download it here](#) [pdf, 892KB].

## Known Issues

*TreeMaker 5* has gone through considerable testing and is fairly stable. if you find any bugs, please let us know and we'll try to address them in a future bugfix release.

- There are currently no known issues in version 5.0.1.

copyright © 2004–2016 robert j. lang • contact robert j. lang