Ben Sauerwine
Condensed Matter Research
Progress Report for February 7 2006

I have found a proof of the Gessel-Viennot algorithm in this paper:

http://www.cc.gatech.edu/grads/c/Deeparnab.Chakrabarty/RESEARCH/REPORTS/LozengeTilings.pdf

Here is the logic behind it in simpler words:

For two paths, suppose I have two non-touching paths going from 1S to 1E and 2S to 2E. Certainly, I get no fewer than P(1S, 1E) P(2S, 2E) different choices where P combinatorially counts the paths. Now suppose that my choices have touched. Consider only the first point where they have. In this case, by selecting the ending portions of each path, I see that each touching path can be rewritten as two paths P(1S, 2E) P(2S, 1E). Subtracting, I have P(1S, 1E) P(2S, 2E) - P(1S, 2E) P(2S, 1E).

Now I must invoke the principle of inclusion and exclusion to extend this to larger numbers of paths: using this two-path situation at the basis case, consider that I add one additional starting and ending point to my array. I'd like to consider situations where it might or might not cross the other paths. Call a routing a permutation (ACBD), for example, the set of paths where the path starting at A ends at A's stated end, B ends at C's stated end, C ends at B's stated end, et cetera. I notice that here lies the beginnings of a Matrix determinant. There is no easy way to put the recursion in words, but by the inclusion-exclusion principle, one ends up subtracting odd permutations and adding even.

The Long-Term Behavior of Two Paths as the Free End Grows:

I have modified the program from GVLongterm2.nb to slowly increase the length of several paths and examine how the entropy per unit changes. It does seem to still be approaching its limit, but the rate of approach becomes rather slow. See GVLongterm3.nb.

Virial Theorem for an Octagon:

Consider the problem of counting the number of states for an octagon. I will take a number N of upward-moving paths and downward-moving paths (this becomes more complex for periodic conditions, where the paths may "wrap around". This way, all of the paths end at the same place and therefore eliminate the problem of wraparound.

The 45-degree rotated squares become a sort of gas with potentially long-range effects, in the sense that each one creates a hard wall where certain rotated squares above, below, to the left or right of this particular square may never pass this one. I notice, however, that on the octagon the region where these rotated squares might appear is not a rectangular grid: due to the streaks of 45-degree rhombi and rather is octagonal itself due to the presence of the cuts on the corners. Looking at figure 2 in "**A formula for the number of tilings of an octagon by rhombi**", I see that this effect is limited only to a few squares: namely, that for a particular rotated square, it may never pass only the four connected to it on "strings" of 45-degree rhombi. Squares attached to other "strings" might well pass, and so even as an approximation using two single binomials to select the number of hard walls along the top and bottom of the rectangle the octagon is inscribed in is inappropriate. I'll have to think a bit about how to find a "rotated square gas" counting function and also an additional term to help weight countings since a degree of repulsion will occur (as studied in GVLongterm notebook series) as the "strings" of 45-degree rhombi that the rotated squares are attached to will maximize their counts towards the center of the region between the four impassable neighbors discussed earlier.

Let me first consider a one-45 degree rhombus gas (as depicted in figure 1 in **A formula for the number of tilings of an octagon by rhombi**). It is clear that this model does indeed have a rectangular lattice of sites that this octagon might occupy, and that if the integer portion of the height of the octagon has a contribution of N squares and the width has a contribution of M squares, that there are (N + 1) (M + 1) sites that the rotated square might occupy. I will first experiment by counting the number of these with the "string" repulsion effect from the hard walls generated by the top and bottom of the octagon.

[Please see squaregas.nb for results].

<u>Implement the known Counting Algorithm on the Octagon:</u>

I examined the algorithm from **A formula for the number of tilings of an octagon by rhombi**, and realized on trying to implement it that the general-case formula actually requires one to pick the locations of each of these rotated squares! This indicates that the algorithm must grow at least as fast as $T_{a,b,c,d} \to O\left[(ac)^{bd}\right]$, without even considering the issue of actually calculating the determinants.

Just finding a way to enumerate the indices $X \times Y$ is a nontrivial feat.

[Please see squaregas.nb for implementation.]