UNIVERSITY OF CALIFORNIA, SANTA CRUZ

BERKELEY · DAVIS · IRVINE · LOS ANGELES · MERCED · RIVERSIDE · SAN DIEGO · SAN FRANCISCO | SANTA BARBARA · SANTA CRUZ

BIOMOLECULAR ENGINEERING DEPARTMENT
1156 HIGH STREET
SANTA CRUZ, CALIFORNIA 95064

+1 (831) 459-5745
FAX +1 (831) 459-4829

PHONE: +1-831-459-4250
`karplus@soe.ucsc.edu`
`http://www.soe.ucsc.edu/~karplus`
May 16, 2008

Judges
California State Science Fair

Dear Judges,

Whenever a young student presents an ambitious science fair project, there is immediately a question: how much of this is the student's work, and how much the work of the mentor or parent? The same question comes up on PhD dissertation defenses and in a somewhat different form for co-authored papers in faculty promotion cases. I will attempt to answer that question in the case of Abe's project on Sierpiński polygons.

Obviously, Abe did not invent trigonometry or calculus to solve the problems he was faced with, but his in-school education in math only extends as far as Algebra 1 and Geometry. He learned some calculus for science fair last year, to be able to predict how high a ball would bounce based on the coefficient-of-restitution theory (that the velocity coming out of the bounce is a constant times the velocity going into the bounce). This inspired him to buy *Calculus for Dummies* with his own money, and read it several times for fun, though not to work out any exercises. This is not the text I would have chosen for him to learn calculus and trigonometry, as the presentation is quite shallow, but it suffices to give him a flavor of the subject.

He was inspired to study the Sierpiński polygons by seeing a program that produced Sierpiński triangles using the Chaos Game. He is a fairly competent programmer in the Scratch programming language, so I thought that his project would be mainly a programming exercise, with perhaps some direction from me to prove that the Chaos Game produces Sierpiński triangles. He ended up going in a different direction, though.

He was fascinated by recursive definition of the objects and wanted to write a recursive program to draw them. Since the Scratch programming language does not support recursion, I suggested that he replicate code to "unroll" the recursion a few levels. He didn't do this in exactly the way I would have, but his works well enough. He already knew how to draw regular polygons with "turtle graphics" and so the main task for him was to figure out how much to scale the polygons in the recursion.

He needed some guidance for coming up with the summation formula for scaling factor. I added construction lines for him for the hexagon case, but he applied his own knowledge of trigonometry and turtle geometry to get the formula. I also showed him construction lines for the decagon case, and again he worked out formula. We worked together on the general case—I'm not certain now how much input I provided. He wrote the program for computing scaling factor himself, though he did use procedures I had written for outputting numbers (Scratch has rather limited I/O capabilities).

To come up with the closed-form formula, I suggested the idea of packing circles rather than polygons, and drew a figure for him. He derived the first formula for the scaling factor from that diagram. He wrote a program to compute the closed form and was excited to notice that the numbers came out exactly the same as the summation formula for every fourth entry (something neither of us had expected).

We drew the figure for decagons together, and he worked out that exactly the same reasoning we used for circles would work for polygons with $4k + 2$ sides. I drew him a figure for octagons, and helped him figure out the formula for polygons with $4k$ sides. I then drew him a figure for polygons with an odd number of sides, and left him to work out the formula for that himself. The formula he worked out turned out to work for all numbers of sides.

We looked a little on the web then, and found an undergrad paper that presented the summation formula, but we did not find a paper that presented the closed-form formula. While it seems unlikely that this is a novel result, the possibility exists.

He read about fractal dimension in a book on fractals and was interested in the idea. It took him quite a while to wrap his mind around the idea of non-integer dimensions, since he had a strong geometric intuition about what dimensions were that did not permit non-integer interpretations. I think he now understands how the self-similarity definition of fractal dimension generalizes the geometric definition he internalized years ago, but we did not look at other definitions of fractal dimension, so he does not yet have tools for understanding non-self-similar fractals.

He computed fractal dimension for Sierpiński polygons and came up with his own conjectures about how it behaved. He needed some guidance to compute the limit though, as this was stretching his calculus skills. I suggested replacing $\lfloor (n-1)/4 \rfloor$ with $n/4 + c$, to approximate the non-differentiable function with a differentiable one. I also showed him how to use maple to take limits.

Once he was convinced that his conjecture that the limit was one was correct, I pointed him to L'Hopital's rule in his *Calculus for Dummies* book. He also needed to have the chain rule pointed out, before he could take the derivatives correctly. He got a bit bogged down in the algebra for the derivatives, so I suggested the change of variables $x = \pi/n$, which helped him simplify the problem.

In summary, he worked through all parts of the problem himself, but he did have considerable guidance to help him past rough spots and to teach him the tools he needed. He could not have done the project alone, but he put in over 100 hours of work spread out over 6 months and learned a lot of fairly sophisticated math. We can honestly say that the project is his work.

Sincerely,

Kevin Karplus
Professor of Biomolecular Engineering
Graduate and Undergraduate Director,
Bioinformatics