The Monomial Method

by Scott Allen Burns

Published February 24, 2015

Note: this is a PDF version of the web page: http://scottburns.us/monomial-method/

Summary

The monomial method is an iterative technique for solving equations. It is very similar to Newton's method, but has some significant advantages over Newton's method at the expense of very little additional computational effort.

In a nutshell, Newton's method solves nonlinear equations by repeatedly linearizing them about a series of operating points. This series of points usually converges to a solution of the original nonlinear system of equations. Linearized systems are very easy to solve by computer, making the overall process quite efficient.

The monomial method, in contrast, generates a series of monomial approximations (single term power functions) to the nonlinear system. In applications in science and engineering, the monomial approximations are better than the linear approximations, and give rise to faster convergence and more stable behavior. Yet, when transformed through a simple change of variables into log space, the monomial approximation becomes linear, and is as easy to solve as the Newton approximation!

This presentation is written for someone who has learned Newton's method in the past and is interested in learning about an alternative numerical method for solving equations. If this doesn't match your background, you'll probably not find this writeup very interesting! On the other hand, there may be some interest among those who fancy my artistic efforts because this topic is what first led me into the realm of basins of attraction imagery and ultimately to the creation of my Design by Algorithm artwork.

Newton's Method

Newton's method finds values of x that satisfy $f(x) = 0$. In cases where x cannot be solved for directly, Newton's method provides an iterative procedure that often converges to a solution. It requires that the first derivative of the function, f', be computable (or at least be able to be approximated).

It also requires an initial guess of the solution. That value, or "operating point," gets updated with each iteration. In this presentation, I'll be denoting the operating point with an overbar, $\bar{x}$. Any expression that is evaluated at the operating point is similarly given an overbar, such as $\bar{f}$ or $\bar{f}'$.

The expression for computing a new operating point from the current one by Newton's method is $\bar{x}_{new} = \bar{x} - \bar{f} \,/\, \bar{f}'$. Note that if we're sitting at a solution, $\bar{f} = 0$, the new operating point will be the same as the old one, meaning the process has converged to a solution.

Newton's method isn't guaranteed to converge. Note that if $\bar{f}' = 0$, the new operating point will be undefined and the process will fail. It's also possible for the sequence of operating points to diverge or jump around chaotically. In fact, this undesirable characteristic of Newton's method is responsible for creating some desirable fractal imagery! When Newton's method misbehaves, it is often possible to reëstablish stable progress toward a solution by simply restarting from different operating point.

It’s important to recognize that Newton’s method will solve a linear equations in exactly one iteration. Since Newton’s method is essentially a repeated linearization of the original problem, a linearization of a linear equation is just the original equation, so the first linear solution provided by Newton’s method IS the solution of the original problem. It will be seen later that the monomial method can be interpreted as Newton’s method applied to a reformulation (or transformation) of the original equation in such a way that the reformulated equation is often very nearly a linear expression, which Newton’s method solves easily.

How is Newton’s Method derived?

A function can be approximated about an operating point using the Taylor Series Expansion

$$f(\bar{x} + \Delta x) = \bar{f} + \bar{f'}\,\Delta x \; + \; \bar{f''}\,(\Delta x)^2/2 \; + \; \bar{f'''}\,(\Delta x)^3/6 + \ldots$$

A linear approximation of the function is simply the first two terms: $f(\bar{x} + \Delta x) \approx \bar{f} + \bar{f'}\,\Delta x$. We want the function to equal zero, so we set the linearized function equal to zero and solve for Δx, yielding $\Delta x = -\bar{f}/\bar{f'}$. Since Δx, is really the change in the operating point, $(\bar{x}_{new} - \bar{x})$, we get $\bar{x}_{new} = \bar{x} - \bar{f} \,/\, \bar{f'}$, which is Newton’s method.

Multiple Variables

So far, the expressions for Newton’s method have been presented for the case of a single variable and a single equation. It is also applicable to a system of simultaneous equations containing functions of multiple variables. For most problems, the number of equations in the system must equal the number of variables in order for a solution to be found. If we call X a vector of variables, $x_1, x_2, \ldots, x_n$, and F a vector of functions, $f_1(X), f_2(X), \ldots, f_n(X)$, then Newton’s method can be expressed as

$$X_{new} = \bar{X} - \bar{J}^{-1}\;\bar{F},$$

where J is the Jacobian matrix of first partial derivatives and the bar above any quantity means that it is evaluated at the operating point. Each iteration requires the solution of a system of linear equations, which again is relatively easy for a computer to do.

Additional Notes

The two vectors X and F are column vectors, so the matrix multiplication works correctly. Also, this multivariate version of Newton’s method is sometimes called the “Newton-Raphson” method, but that terminology is becoming less common.

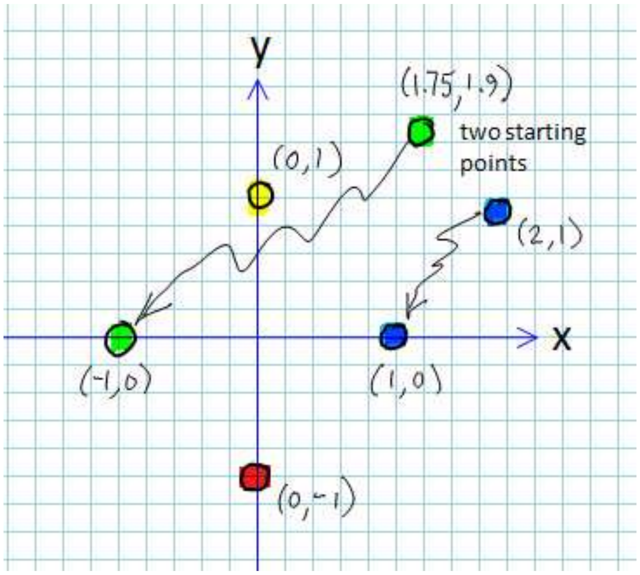
An Example

Here is a system of equations in two variables: $x^4 - 6x^2y^2 + y^4 - 1 = 0$ and $4x^3y - 4xy^3 = 0$. It has four solutions, $(x, y) = (1, 0), (-1, 0), (0, 1), (0, -1)$. Newton’s method for this system is

$$\left\{ \begin{array}{c} x_{new} \\ y_{new} \end{array} \right\} = \left\{ \begin{array}{c} \bar{x} \\ \bar{y} \end{array} \right\} - \left[\begin{array}{cc} 4\,\bar{x}^3 - 12\,\bar{x}\,\bar{y}^2 & -12\,\bar{x}^2\,\bar{y} + 4\,\bar{y}^3 \\ 12\,\bar{x}^2\,\bar{y} - 4\,\bar{y}^3 & 4\,\bar{x}^3 - 12\,\bar{x}\,\bar{y}^2 \end{array} \right]^{-1} \left\{ \begin{array}{c} \bar{x}^4 - 6\,\bar{x}^2\,\bar{y}^2 + \bar{y}^4 - 1 \\ 4\,\bar{x}^3\,\bar{y} - 4\,\bar{x}\,\bar{y}^3 \end{array} \right\}$$

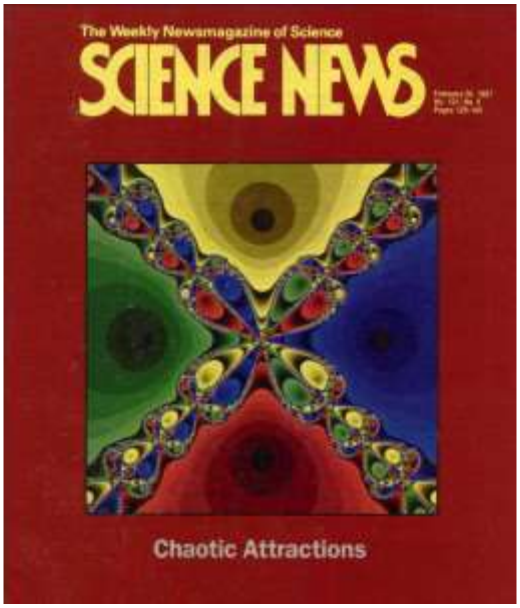
There are many starting points that will cause this recurrence to fail, for example $(0, 0)$. But the vast majority of starting points will lead to convergence to one of the four solutions. Generally speaking, you will converge to the one closest to the starting point. But about 20% of so of the time, Newton’s method will converge to one of the other three more distant solutions. For example, starting with $(x, y) = (2, 1)$, the sequence of Newton’s method iterations are: $(1.50, 0.73)$,$(1.14, 0.49)$,$(0.90, 0.25)$,$(0.89, -0.04)$,$(1.02, 0.02)$, and $(1.00, 0.00)$. This is an example of it converging to the nearest of the four solutions. However, starting with $(x, y) = (1.75, 1.9)$, Newton's method gives: $(1.30, 1.42)$,$(0.95, 1.04)$,$(0.64, 0.73)$,$(0.25, 0.39)$,$(-2.29, -0.07)$,$(-1.74, -0.05)$, $(-1.35, -0.03)$,$(-1.11, -0.02)$,$(-1.02, 0.00)$, and $(-1.00, 0.00)$. In this case, it goes to the third most distant solution. Is there any regularity to this behavior?

One way to visualize the overall behavior of an iterative process in two variables is to construct what is known as a "basins of attraction" plot. We start by assigning colors to each of the four solutions, as shown in the following figure:



Making a color-coded summary of which starting points go to which solutions.

Then we run Newton's method from a large number of different starting points, and keep track of which starting points go to which solutions. The starting point location on the plot is colored according to the solution to which it converged. The final result is a color-coded summary of convergence. The result of this process is the image shown on the cover of Science News from 1987. (This was probably my first "15 minutes of fame," back when fractals were brand new.)



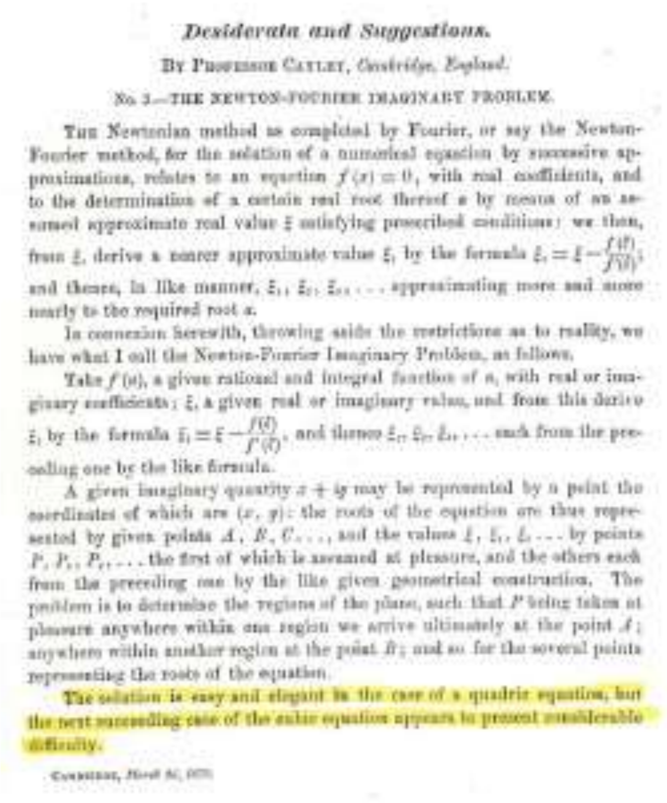
The cover story on the Feb 1987 issue of Science News.

Notice that there are large red, green, yellow, and blue regions that are regions of starting points that all go to the same solution. On the diagonal, however, the colored regions are tightly intermingled, indicating that very tiny changes in starting point can lead to drastic changes in the outcome of Newton's method. This "sensitive dependence to initial conditions" is a hallmark of chaotic behavior, and it captured graphically in the basins of attraction images.

You might be wondering about the different shades of each color. I selected the coloring scheme to reflect the number of iterations it took to get to a solution (to within some pre-defined tolerance). The colors cycle through six shades of each hue, from dark to light and over again, with increasing iteration count. Thus, the dark circles around each of the four solutions contain starting points that all converge to the solution in one iteration of Newton's method. The slightly less dark region around each of them converge in two iterations, and so forth.

The "boundary" of each color is an extremely complex, fractal shape. In fact, it can be demonstrated that *all four* colors meet at *every point* on the boundary!

Although fractal images are a relatively recent concept, mathematicians have had glimpses of this behavior for a long time. In the 1800s, Professor Arthur Cayley of Cambridge was studying the behavior of Newton's method with some very simple equations and found that it was quite difficult to state general rules for associating which starting points end up at which solutions. He wrote a one page paper on the subject in 1879:



Desiderata and Suggestions.

BY PROFESSOR CAYLEY, *Cambridge, England.*

No. 3.—THE NEWTON-FOURIER IMAGINARY PROBLEM.

THE Newtonian method as completed by Fourier, or say the Newton-Fourier method, for the solution of a numerical equation by successive approximations, relates to an equation $f(x)=0$, with real coefficients, and to the determination of a certain real root thereof a by means of an assumed approximate real value ξ satisfying prescribed conditions: we then, from ξ, derive a nearer approximate value ξ_1 by the formula $\xi_1 = \xi - \frac{f(\xi)}{f'(\xi)}$; and thence, in like manner, ξ_2, ξ_3, ξ_4, ... approximating more and more nearly to the required root a.

In connexion herewith, throwing aside the restrictions as to reality, we have what I call the Newton-Fourier Imaginary Problem, as follows.

Take $f(u)$, a given rational and integral function of u, with real or imaginary coefficients; ξ, a given real or imaginary value, and from this derive ξ_1 by the formula $\xi_1 = \xi - \frac{f(\xi)}{f'(\xi)}$, and thence ξ_2, ξ_3, ξ_4, ... each from the preceding one by the like formula.

A given imaginary quantity $x+iy$ may be represented by a point the coordinates of which are (x, y): the roots of the equation are thus represented by given points A, B, C..., and the values ξ, ξ_1, ξ_2... by points P, P_1, P_2,... the first of which is assumed at pleasure, and the others each from the preceding one by the like given geometrical construction. The problem is to determine the regions of the plane, such that P being taken at pleasure anywhere within one region we arrive ultimately at the point A; anywhere within another region at the point B; and so for the several points representing the roots of the equation.

The solution is easy and elegant in the case of a quadric equation, but the next succeeding case of the cubic equation appears to present considerable difficulty.

CAMBRIDGE, *March 3d, 1879.*

Prof. Arthur Cayley's paper on the difficulty of describing the behavior of Newton's method in some cases. (Click to enlarge)

He compared the stable, easy to describe behavior of Newton's method applied to a quadric (second order equation) to the case of a cubic equation, and observed the latter "appears to present considerable difficulty." The Science News image above is the case of a fourth-order equation, one degree above the cubic that he considered. The difficulty he was talking about is the behavior in the regions of tightly intermingled colors, where the shapes of those regions defy description.

You might be wondering about the phrase “Newton-Fourier Imaginary Problem.” First the Fourier part: I’m not sure why Cayley is discussing Fourier in this paper. The Newton-Fourier method is known today as a process different from Newton’s method because it adds some additional computation to each step that can improve convergence. But the method he demonstrates in his paper is clearly the basic Newton’s method as I’ve described above.

Regarding the “imaginary” part, he was actually looking at Newton’s method operating on functions of a complex variable, that is, one that contains a real component and an imaginary component containing the imaginary quantity $i = \sqrt{-1}$. The letter z is usually used to denote a complex variable, *i.e.*, $z = x + iy$. Consider the function $z^4 - 1 = 0$. It has four solutions, just like the example I showed earlier. They are $+1$, -1,$+i$, and $-i$. Newton’s method can be applied just as effectively to functions of a complex variable, and the iteration in this case would be

$$z_{new} = \bar{z} - \left(\frac{\bar{z}^4 - 1}{4\bar{z}^3} \right).$$

Interestingly, if a basins of attraction plot were to be made of Newton’s method applied to $z^4 - 1 = 0$ on the complex plane (real and imaginary axes) instead of the x, y plane, it would be identical to the image on the Science News cover!

The Monomial Method

The monomial method is an equation solving technique that I developed back in the ’80s as part of my research activities at the Univ of IL. It grew out of my work in optimization theory, specifically a method known as geometric programming that uses monomial approximations extensively. I found that the monomial method was just as easy to apply as Newton’s method, but was able to solve certain problems in my field of study (civil engineering structures) far more quickly and stably than Newton’s method. I’ll be demonstrating this in an example later.

Why identical images?

Start with the expression $z^4 - 1 = 0$. Next, substitute $x + iy$ for z and expand the fourth power to give $x^4 + 4x^3yi - 6x^2y^2 - 4xy^3i + y^4 - 1 = 0$. (Remember to substitute $-1$for every instance of i^2. Finally, collect all terms that are real into one equation and all terms that are imaginary into another equation. Divide the $i$out of the second equation, and you’ll be left with the exact same system of equations that I presented earlier; they are equivalent to one another. In fact, the image on the Science News cover is the one on the complex plane, not on the x, y plane!

There are some limitations of the types of problems the monomial method will solve. First, all variables must be positive. There is a logarithmic transformation in the method that will fail if an operating point is not greater than zero. Second, the equations being solved must be algebraic in form, such as $12x^4 - 6.2x^2 + x^{-1.5} - 2.7 = 0$. All algebraic systems of equations can be cast in the general form

$$\sum_{i=1}^{t_k} c_{ik} \prod_{j=1}^{n} x_j^{a_{ijk}} = 0 \quad k = 1, 2, \ldots, n.$$

In each of the k equations, there are t_k terms, each of which has a coefficient, c, followed by a product of variables, x, each raised to a power, a. The coefficients and exponents can be any real number (positive, negative, fractional, etc.).

Let's define u_{ik} to be the ith term of the kth equation, or

$$u_{ik} = c_{ik} \prod_{j=1}^{n} x_j^{a_{ijk}}.$$

Next, we group together all the terms with positive coefficients and those with negative coefficients and recast the system of equations as the *difference* of two sums of terms, $P(X)$ and $Q(X)$, respectively:

$$F_k(X) = P_k(X) - Q_k(X) = \sum_{P\ terms} u_{ik} - \sum_{Q\ terms} u_{ik} = 0 \qquad k = 1, 2, \ldots, n.$$

Note that the coefficients in Q are positive due to the negative sign in the expression. One final manipulation is to bring $Q(X)$ to the right side of the equation and divide it through, giving us the recast form of the system of equations

$$\frac{P_k(X)}{Q_k(X)} = \frac{\sum\limits_{P\ terms} u_{ik}}{\sum\limits_{Q\ terms} u_{ik}} = 1 \qquad k = 1, 2, \ldots, n.$$

I'm hoping to keep this presentation accessible to a wide audience, and I'm sure the math-speak above might be difficult to decipher. So I'll translate it in the setting of an example with just one equation and one variable:

$$12x^4 - 6.2x^2 + x^{-1.5} - 2.7 = 0$$

Using the terminology above, $P(X) = 12x^4 + x^{-1.5}$ and $Q(X) = 6.2x^2 + 2.7.$ The recast form is

$$\frac{12x^4 + x^{-1.5}}{6.2x^2 + 2.7} = 1.$$

So in other words, all those math symbols above simply mean "divide the positive terms by the negative terms and set it equal to 1."

We now apply a process called *condensation* to $P(x)$ and $Q(x)$ separately. Condensation is the process of approximating a multi-term expression with a single-term expression (also called a *monomial*). It does this using a famous relationship between the arithmetic mean and the geometric mean of a set of positive numbers

$$\sum_i n_i \delta_i \geq \prod_i n_i^{\delta_i}.$$

The δ_i are *weights*, or fractional numbers that sum to 1. When all the weights are equal, then the left side is the familiar arithmetic mean (average) of the numbers, and the right side is the less known geometric mean. For example, if $n = \{4, 5, 6\}$ and all $\delta = 1/3,$ the geometric mean is $(4 + 5 + 6)/3 = 5$ and the geometric mean is $\sqrt[3]{4 \times 5 \times 6} = 4.932.$

The inequality becomes more useful to us if we make the substitution $n_i = u_i/\delta_i,$ giving

$$\sum_i u_i \geq \prod_i \left(\frac{u_i}{\delta_i}\right)^{\delta_i}.$$

This is where the magic happens. Instead of thinking of the u_i as simple numbers, we can think of them as *monomials*, just as we've defined earlier:

$$\sum_i c_i \prod_j x_j^{a_{ij}} \geq \prod_i \left(\frac{c_i}{\delta_i} \prod_j x_j^{a_{ij}} \right)^{\delta_i} .$$

OK, lots of math. But let's step back here and see what we have. The left side of this expression is simply our familiar expression for $P(X)$or $Q(X),$ the positive or negative terms in an equation. The right side, after everything gets multiplied out, is simply a single term, a monomial, something like $3.1\ x_1^{-4.1}\ x_2^5\ x_3^{9.2}.$ This is a monomial approximation to the polynomial expression on the left. The monomial will exactly equal the polynomial when evaluated at the operating point, and will approximate its value at other values of $X.$

So now let's complete the monomial method, first in words, and later with math. We start by condensing P and Q each to monomials. The ratio of two monomials is again another monomial, so $P/Q = 1$ is an expression setting a monomial equal to 1. Remember from high school math that the logarithm of a product of numbers is equal to the sum of the logarithms of the numbers (the basis of the slide rule, for those of you as old as I am). The log of this monomial equation becomes a linear equation, which is simple to solve, giving rise to the new operating point (after transforming back out of log space with the exponential function, e^x).

The one missing piece to the puzzle is how to handle the weights, $\delta.$ It's pretty simple: they are calculated as the fraction that each term in P or Q contributes to the total, when evaluated at the current operating point. See how this makes the sum of the δ equal to 1?

It's time to continue our simple example above. We had $P(X) = 12x^4 + x^{-1.5}$ and $Q(X) = 6.2x^2 + 2.7.$ Let's select an operating point of $x = 1.5,$ which gives us these weights:

$$\delta_1^+ = \frac{12\ (1.5)^4}{12\ (1.5)^4 + (1.5)^{-1.5}} = 0.991 \quad \delta_2^+ = \frac{(1.5)^{-1.5}}{12\ (1.5)^4 + (1.5)^{-1.5}} = 0.009$$

$$\delta_1^- = \frac{6.2\ (1.5)^2}{6.2\ (1.5)^2 + 2.7} = 0.838 \quad \delta_2^- = \frac{2.7}{6.2\ (1.5)^2 + 2.7} = 0.162$$

The +/- superscript on δ indicates if the term came from P or $Q.$ The monomial approximation to $P/Q = 1$ is

$$\frac{\left(\left(\frac{12}{0.991}\right)^{0.991} x^{4\ (0.991)}\right)\left(\left(\frac{1}{0.009}\right)^{0.009} x^{-1.5\ (0.009)}\right)}{\left(\left(\frac{6.2}{0.838}\right)^{0.838} x^{2\ (0.838)}\right)\left(\left(\frac{2.7}{0.162}\right)^{0.162}\right)} = 1.463\ x^{2.276} = 1.$$

Taking the log of this gives $\ln 1.463 + 2.276 \ln x = 0.$ Solving for $\ln x$gives $-0.1672,$ which leads to $x = 0.8460.$ This is the new operating point, which completes one iteration of the monomial method. Additional iterations will yield the sequence $\bar{x} = 0.8273,$ $\bar{x} = 0.8268,$ $\bar{x} = 0.8268,$ and a solution has been found.

A Computationally Efficient Way

It is apparent from that last example that the monomial method seems to involve much more computation than Newton's method. The Newton iteration is simply $x_{new} = \bar{x} - (12\bar{x}^4 - 6.2\bar{x}^2 + \bar{x}^{-1.5} - 2.7)\ /\ (48\bar{x}^3 - 12.4\bar{x} - 1.5\bar{x}^{-2.5})$, which generates the series of operating points from $\bar{x} = 1.5$: $\bar{x} = 1.1875$, $\bar{x} = 0.9835$, $\bar{x} = 0.8721$, $\bar{x} = 0.8322$, $\bar{x} = 0.8269$, $\bar{x} = 0.8268$, $\bar{x} = 0.8268$.

Although Newton's method is converging more slowly, the number of computations necessary is far less, especially considering that raising a number to a fractional power takes much more internal computation on a computer than simple multiplication or division.

Fortunately, there is an alternate way to compute the monomial method iteration that is more on par with the computational effort required for Newton's method. Without getting into the details of the derivation, which are available elsewhere, the monomial method iteration can be stated as the following matrix equation:

$$A\ \Delta Z = -\ln\left(\bar{P}\ /\ \bar{Q}\right),$$

where ΔZ is the change in the log of X, $\Delta Z = \ln X_{new} - \ln \bar{X}$, and the matrix A has as its j th column and kth row the value

$$A_{jk} = \sum_{P\ terms} a_{ijk}\ \delta^{+}_{ik} - \sum_{Q\ terms} a_{ijk}\ \delta^{-}_{ik}.$$

For the single variable example earlier, $A = 4\delta^{+}_{1} - 1.5\delta^{+}_{2} - 2\delta^{-}_{1}$, which when evaluated at the operating point $\bar{x}$ gives $A = 2.276$. Since $\bar{P} = 61.29$ and $\bar{Q} = 16.65$, the right side of the equation is $-\ln(\bar{P}\ /\ \bar{Q}) = -1.303$. Solving for ΔZ gives -0.5729 and therefore $x_{new} = \exp(\ln \bar{x} + \Delta Z) = 0.8460$. This version requires far less computation than the monstrous equation used earlier!

Let me restate the math expressions above in words. Recall that each term in P and Q has an associated weight, δ. To create the A matrix, simply multiply each exponent, a, of each variable by the weight associated with the term in which it appears. Gather together these products by common variable and add or subtract the product according to whether the term is in P or Q, respectively.

Here's a two-variable example to demonstrate:

$$\frac{11664x_1x_2^{-4} + 54x_1^2x_2^{-4} + 21.6x_1^{-2}}{10.8x_1^4x_2^{-4} + 4.32} = 1 \quad \frac{6998.4x_1^4x_2^{-7} + 18x_1^4x_2^{-6} + 8398x_2^{-3}}{12.96x_1^4x_2^{-4} + 5.184} = 1.$$

Let's identify the weights (shown in correspondence to their associated term) as

$$\frac{\delta^{+}_{11},\ \ \delta^{+}_{21},\ \ \delta^{+}_{31}}{\delta^{-}_{11},\ \ \delta^{-}_{21}} \quad \frac{\delta^{+}_{12},\ \ \delta^{+}_{22},\ \ \delta^{+}_{32}}{\delta^{-}_{12},\ \ \delta^{-}_{22}}.$$

The A matrix will be computed as (rows are by equation and columns are by variable)

$$A = \left[\begin{array}{cc} \delta^{+}_{11} + 2\delta^{+}_{21} - 2\delta^{+}_{31} - 4\delta^{-}_{11} & -4\delta^{+}_{11} - 4\delta^{+}_{21} + 4\delta^{-}_{11} \\ 4\delta^{+}_{12} + 4\delta^{+}_{22} - 4\delta^{-}_{12} & -7\delta^{+}_{12} - 6\delta^{+}_{22} - 3\delta^{+}_{32} + 4\delta^{-}_{12} \end{array}\right].$$

If we start with $(x_1, x_2) = (2, 10)$, the weights will evaluate as

$$\frac{0.3008,\quad 0.0028,\quad 0.6964}{0.0040,\quad 0.9960} \qquad \frac{0.0013,\quad 0.00003,\quad 0.9986}{0.0040,\quad 0.9960}$$

and matrix A will evaluate as

$$A = \begin{bmatrix} -1.1023 & -1.1985 \\ -0.0105 & -2.9895 \end{bmatrix}.$$

After evaluating the right-hand side [$-\ln(P/Q)$], we have the linear system

$$A = \begin{bmatrix} -1.1023 & -1.1985 \\ -0.0105 & -2.9895 \end{bmatrix} \begin{Bmatrix} \Delta z_1 \\ \Delta z_2 \end{Bmatrix} = \begin{Bmatrix} -0.5810 \\ -0.4798 \end{Bmatrix},$$

which solves for $\Delta z_1 = 0.3539$ and $\Delta z_2 = 0.1593$. The new operating point becomes $x_1 = 2.849$ and $x_2 = 11.726.$

Relationship Between Newton’s Method and the Monomial Method

Again, without delving into details that are available elsewhere, the monomial method applied to $P(X) - Q(X) = 0$ can be shown to be equivalent to Newton’s method applied to

$$\ln \frac{P(e^{z})}{Q(e^{z})} = 0.$$

Note that Newton’s method is operated in the space of $z = \ln x$ instead of in the space of x.

Newton’s method in matrix form is $J\ \Delta X = -F$, where J is the Jacobian matrix of first partial derivatives. The monomial method can be expressed similarly in terms of partial derivatives as

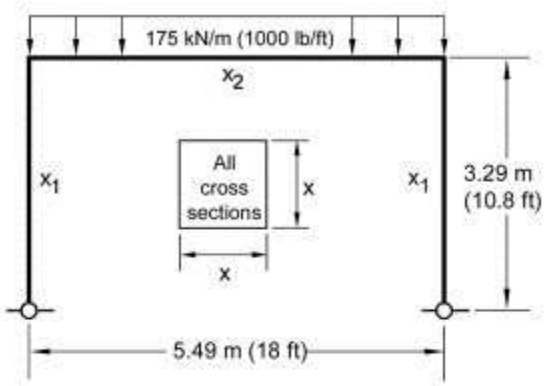
$$\left[D_{\bar{P}}^{-1}\ \bar{J}_P\ D_{\bar{X}} - D_{\bar{Q}}^{-1}\ \bar{J}_Q\ D_{\bar{X}}\right] \Delta Z = -\ln\left(\bar{P}\ /\ \bar{Q}\right).$$

$\bar{J}_P$ and $\bar{J}_Q$ in this expression are the Jacobian matrices of the P and Q functions, evaluated at the operating point. The D matrices are diagonal matrices (all zeros except the values on the main diagonal): $D_{\bar{X}}$ has the values of the operating point, $\bar{X}$, on the diagonal; $D_{\bar{P}}$ and $D_{\bar{Q}}$ have the values of $\bar{P}$ and $\bar{Q}$ on the diagonal (and they are inverted, so the diagonals are just $1/\bar{P}$ and $1/\bar{Q}$).

Why Bother?

So it seems the monomial method requires a little more computation. What do we stand to gain?

The following example is what convinced me that I was on to something with the monomial method. It demonstrates that for some applications, the difference in behavior between the monomial method and Newton’s method can be striking. It is most clearly demonstrated by constructing basins of attraction plots, as we did earlier.



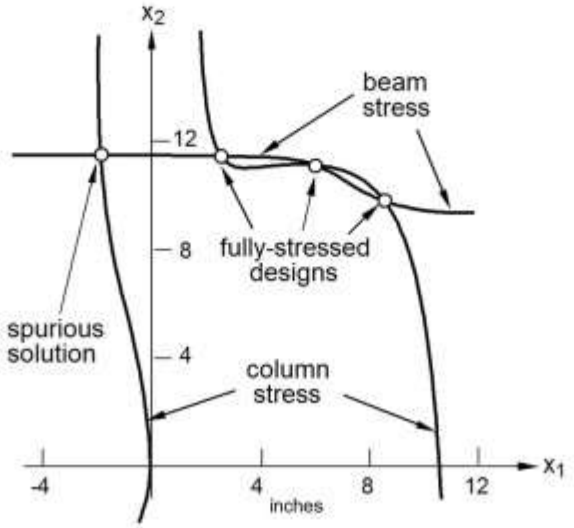
The design of a civil engineering frame structure comprising three members.

This example comes from the field of civil engineering structural design. One of the simplest frame structures, called a portal frame, has only three members, two vertical columns and a horizontal beam. The design goal is to select the dimensions of the members (the size of their square cross sections) so that the frame safely supports a load of 1000 pounds per foot along the beam. Assuming the two columns are of the same size, there are only two design variables, x_1 and x_2, that we seek to determine. Without getting into the engineering details, the example presented earlier

$$11664x_1x_2^{-4} + 54x_1^2x_2^{-4} + 21.6x_1^{-2} - 10.8x_1^4x_2^{-4} - 4.32 = 0$$

$$6998.4x_1^4x_2^{-7} + 18x_1^4x_2^{-6} + 8398x_2^{-3} - 12.96x_1^4x_2^{-4} - 5.184 = 0$$

are the governing equations for this example. In other words, values of x_1 and x_2 that satisfy these equations are dimensions of the structure that we seek. These solutions are known as “fully-stressed designs” and can be thought of as particularly good designs because are safe and yet economical because they don’t unnecessarily waste building materials.

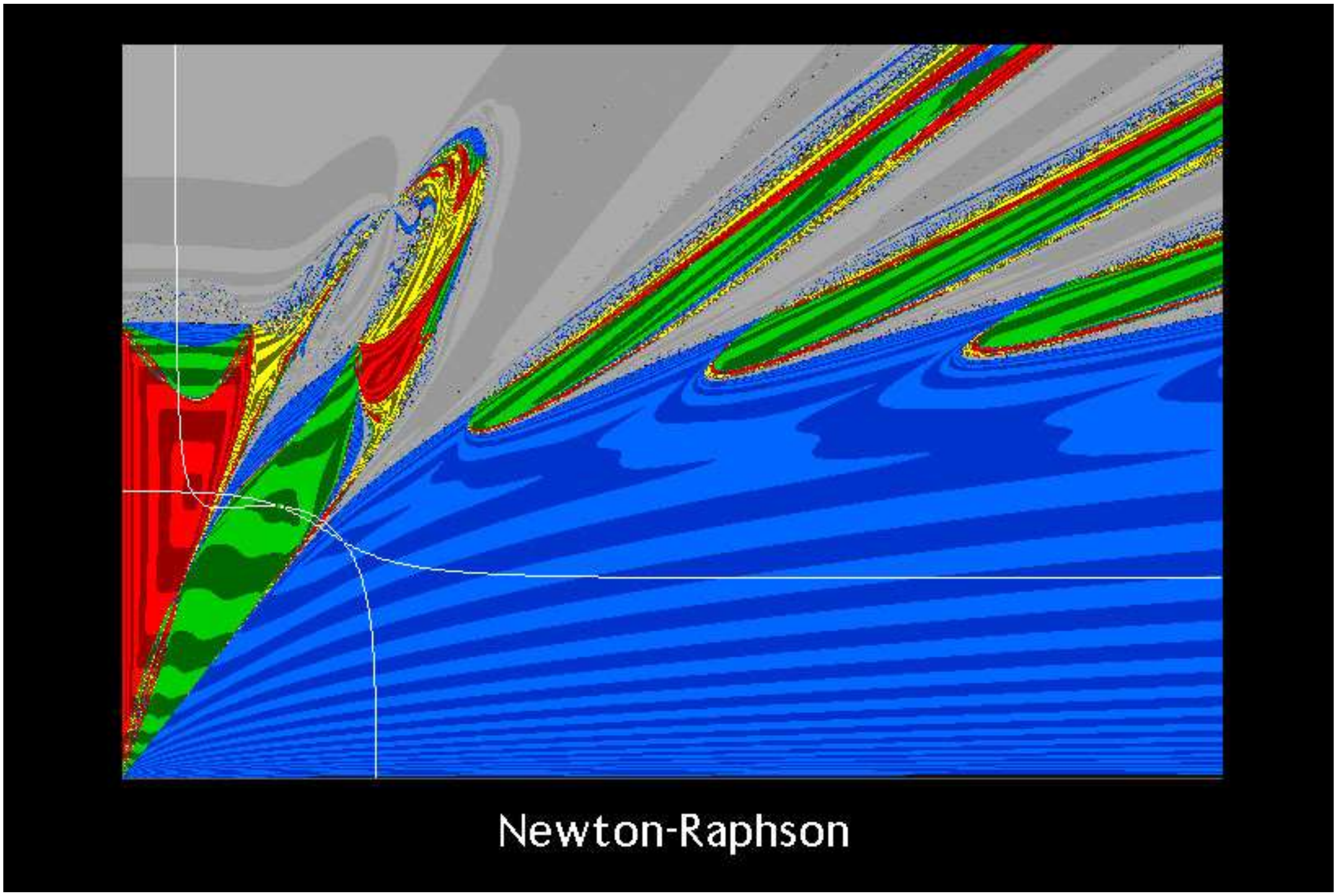


A plot of the two equations being solved.

We can plot the two equations on the $x_1 - x_2$ plane, as shown in the figure above. The curve that runs more or less horizontally comes from one equation, whereas the two curves running vertically are two “branches” of the other equation. Since the equations are nonlinear, there are several solutions that satisfy them, shown as small circles where the curves intersect. Three of the four solutions fall in the positive quadrant of the "design space," and represent physically realizable solutions. The fourth solution has a negative value for x_1. Although that point formally satisfies the equations, it is not a real solution because the square cross section cannot have a negative dimension. We call that a "spurious" solution.

The three positive solutions have an interesting interpretation from an engineering point of view. The leftmost of the positive designs (with coordinates of approximately $x_1 = 3$ and $x_2 = 11$) has a heavy horizontal beam and relatively slender columns. This design resembles the ancient "post and lintel" type construction, where the beam does most of the work and the columns just prop up the ends of the beam without exerting too much effort. In contrast, the rightmost design has beam and column dimensions that are more similar. In this case, there are substantial forces being developed in the corners of the frame, and much of the load is carried by the bending action transmitted between the beam and the columns. It is a very different sort of load-carrying mechanism, one that is more recent in structural engineering practice with the advent of structurally rigid frame connections. I recall my PhD advisor, Narbey Khachaturian, telling me that the various solutions represent distinct load paths, or preferred ways that the frame likes to "flex its muscles."

To create basins of attraction plots, we need to assign colors to the four solutions. Going from left to right, let's select yellow, red, green, and blue. Let's solve the equations using Newton's method first. Here are the basins of attraction:



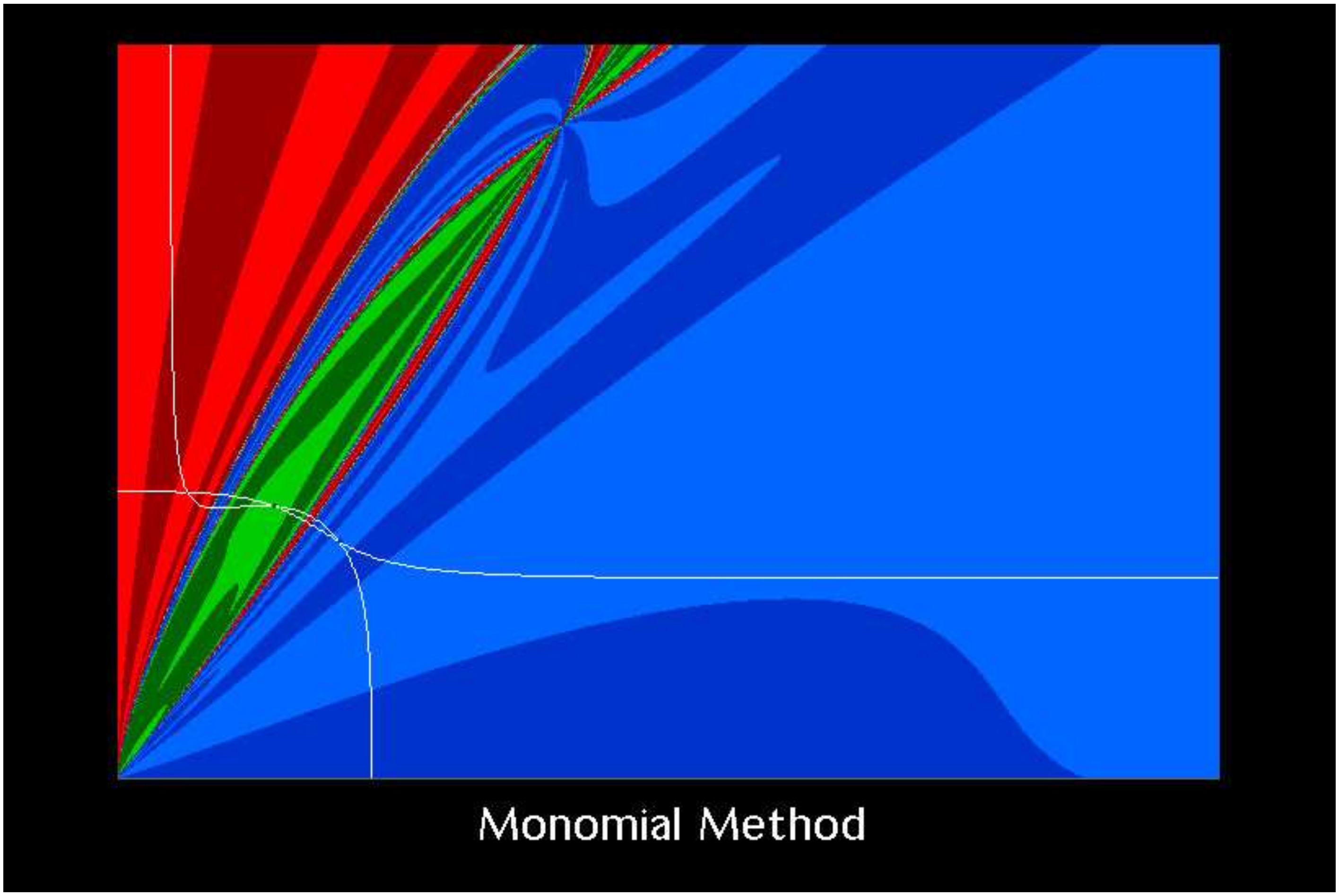
Basins of attraction for the frame example, as solved by Newton's method. (I created this image decades ago, back when I was still calling it "Newton-Raphson.")

I'm showing just the positive quadrant of the design space, and the curves are shown in white. The first thing you might notice is the large region of gray. That is a fourth color I selected to indicate when Newton's method has failed, in this case, due to numerical overflow. In other words, any starting point in a gray region produced a series of operating points that just kept getting bigger and bigger until the computer could no longer represent the numbers and gave up with an error condition.

The two shades of each color represent whether the number of iterations it took to converge (or fail) was odd or even. Thus, the spacing of these stripes gives an indication of how fast the method converges (narrower stripes means slower).

Another interesting observation is all the yellow in the plot. These are starting points that end up at the spurious, meaningless solution, even though the process was initiated from a perfectly reasonable starting point. This is particularly troublesome. There are lots of places where very good initial designs, in the vicinity of the best designs, end up with useless results by Newton's method!

Now let's compare this to the monomial method:



Basins of attraction for the frame example as solved by the monomial method.

What a world of difference! First of all, there are very few starting points that run into numerical difficulty. Second, the spacing between the stripes of color shades are much larger, indicating that convergence is happening very quickly, typically within just a few iterations instead of dozens. Third, and perhaps most importantly, no starting points end up at the meaningless solution (simply because the monomial method can't converge there).

Special Properties

The monomial method exhibits some unusual properties, not shared by Newton’s method, that are responsible for the improved performance in certain types of applications.

- Multiplication of an equation by a positive polynomial has no effect on the monomial method iteration. With Newton’s method, it is often suggested that equations be put in some sort of “standard form” by multiplying through by powers of the variables. This can improve the performance of Newton’s method in some cases. There is no reason to manipulate the equations in this way for the monomial method. It will give the same sequence of operating points in all cases of this type of manipulation.

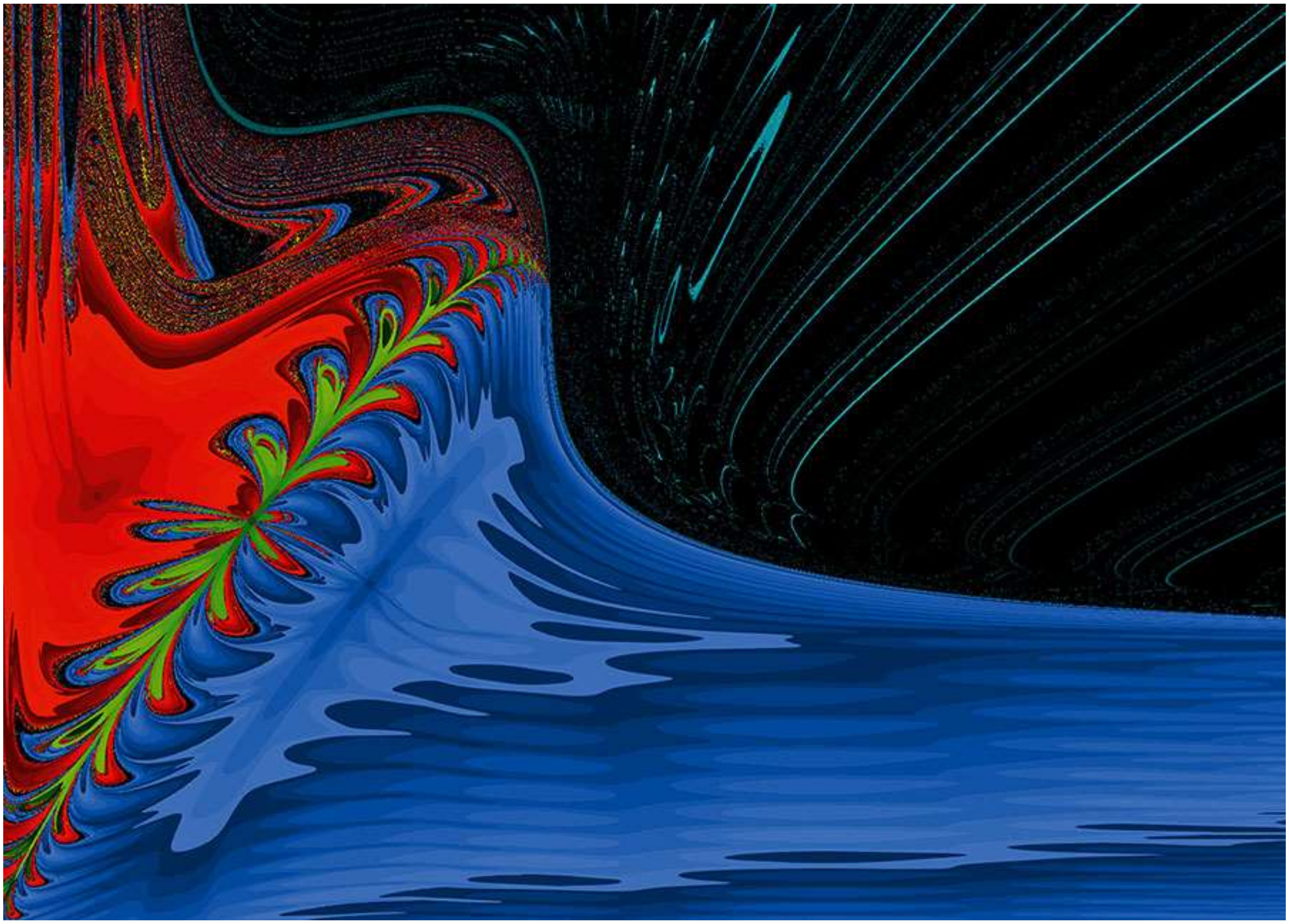
- There is an automatic scaling of variables and equilibrating of equations built into the monomial method. Newton’s method benefits greatly from these actions, but they must be done outside of, and in addition to, the regular Newton iteration. By examining the form of the monomial method involving Jacobian matrices (presented earlier), it can be seen that scaling and equilibrating are taking place automatically via the diagonal matrices that pre- and post-multiply the Jacobian matrices.

- The monomial method was shown to be equivalent to Newton’s method applied to expressions of the form $\ln \sum_i \exp(\ln c_i + \sum_j a_{ij} z_j)$. When one term dominates the others, then the summation over $i$effectively disappears and only the one term remains. The log and the exp functions cancel each other out, leaving just a linear expression, $\ln c_i + \sum_j a_{ij} z_j$. Since Newton’s method applied to a linear expression gives a solution in just one iteration, the monomial method inherits this behavior.

- When initiating the solution process from a very distant starting point, one term of each equation tends to dominate (assuming the terms in an equation have different powers of x.) This makes the system appear to the monomial method as a nearly linear system, and initial iterations move very rapidly toward a solution, in contrast to Newton’s method, which can take dozens of iterations to recover from a distant operating point. This “asymptotic behavior” is evident in the frame example presented earlier. See this paper for details.

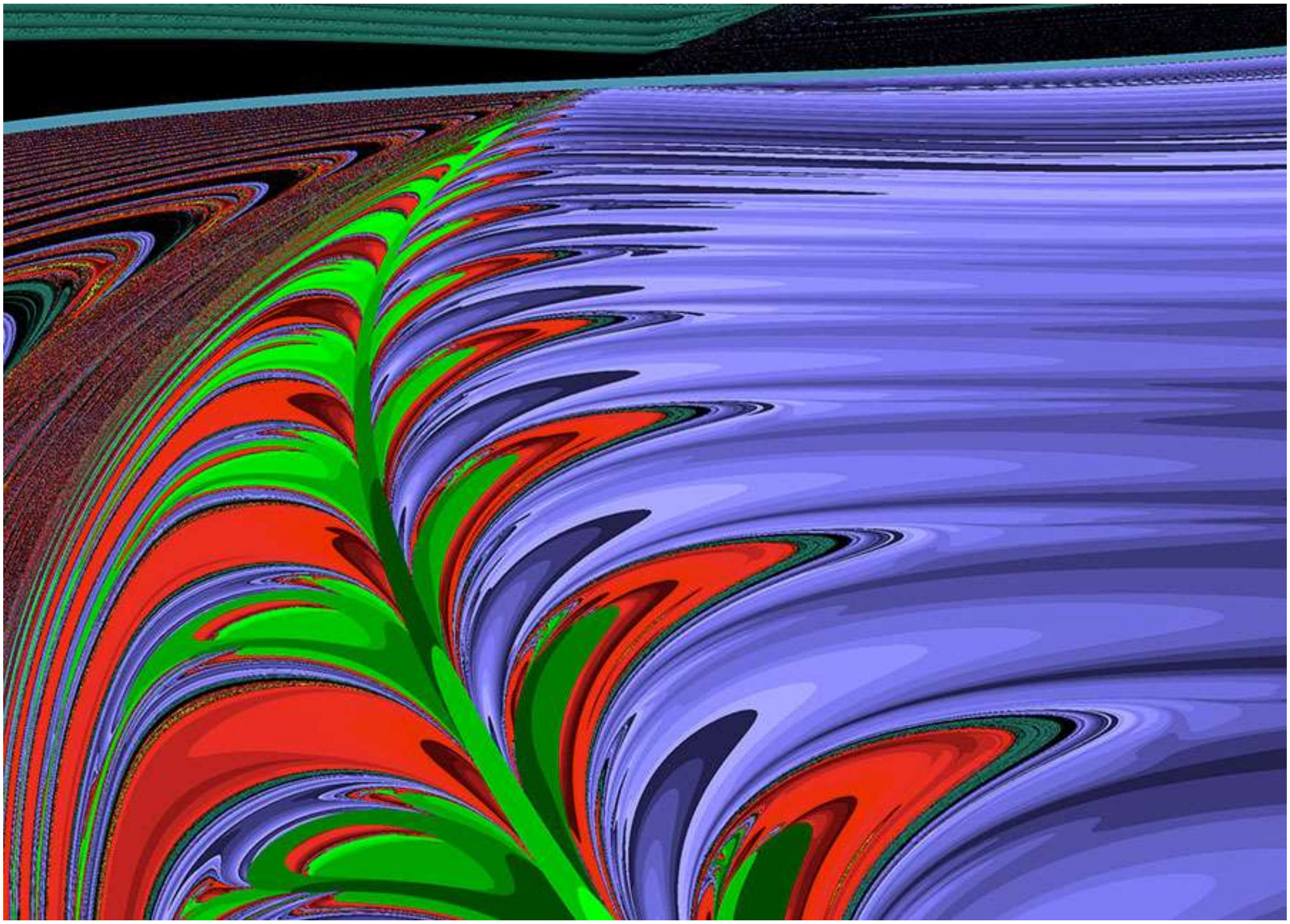
Connection to Artwork

The basins of attraction study I conducted at the University of Illinois in the late ’80s sparked my interest in creating abstract art using these methods, leading to my Design by Algorithm business. In particular, one of the early pieces I offered (image #34) was based on the frame example:



This is the frame example solved by another method called “steepest descent.” Can you see the three positive frame solutions? This image also appeared on the cover of a math textbook in the ’90s.

Although I’ve retired this image from the set of currently offered pieces, I have since cropped and rotated this image to form what is now known as Image 78 on the next page



The basins of attraction construction has been a fascinating “paintbrush” to work with over the past several decades. I have grown to where I can deliberately design math expressions for desired outcomes and express my artistic intent more fully.