ZEROS OF PERTURBED FUNCTIONS

A toy perturbation theory & its real-world relevance

Nicholas Wheeler, Reed College Physics Department
September 2003

Introduction. In recent work[1] I had occasion to consider this question:

How do the zeros of $f(x)$ respond to the perturbation $f(x) \mapsto f(x)+\epsilon g(x)$?

The question is easily answered if one is content to entrust the computational labor to *Mathematica*. What most surprised me was the realization that, in a long mathematical career, I had not had previous occasion to consider the issue, that I did not know where (if at all) it might be discussed in the literature, and that the answer was (or on its face seemed to be) unfamiliar to me. I was struck by the fact that the question manages to capture—in a simplest-possible setting—the essence of the perturbation theories that students of physics most typically encounter in contexts where distractingly many other things are going on simultaneously. What I appeared to have stumbled upon is, in my view, a "toy perturbation theory" of some potential pedagogical value.

My objective here is to describe that theory and some of its ramifications.

1. The basic idea. Assume $f(x)$ and $g(x)$ to be nicely behaved functions, and that

$$f(x_0) = 0$$

Construct the ϵ-parameterized family of perturbed functions

$$F(x;\epsilon) \equiv f(x) + \epsilon\, g(x)$$

and require that

$$F(x_0 + x_1\epsilon + x_2\epsilon^2 + x_3\epsilon^3 + \cdots;\epsilon) = 0$$

[1] "Measurement of trapped atom temperature: elementary theory of the TOF signal profile" (May 2003), §5. See also §8 in "Measurement of the temperature of trapped atom populations" (notes from the Reed College Physics Seminar of 10 September 2003).

Mathematica's `Series` command supplies

$$\begin{aligned} F(x_0 + x_1\epsilon + x_2\epsilon^2 + x_3\epsilon^3 + \cdots; \epsilon) = f(x_0) &+ A_1(x_1)\,\epsilon \\ &+ A_2(x_1,x_2)\,\epsilon^2 \\ &+ A_3(x_1,x_2,x_3)\,\epsilon^3 \\ &+ A_4(x_1,x_2,x_3,x_4)\,\epsilon^4 + \cdots \end{aligned}$$

where—if we adopt the abbreviations

$$\begin{aligned} & & g &\equiv g(x_0) \\ f' &\equiv f'(x_0) & g' &\equiv g'(x_0) \\ f'' &\equiv f''(x_0) & g'' &\equiv g''(x_0) \\ f''' &\equiv f'''(x_0) & g''' &\equiv g'''(x_0) \\ &\vdots & &\vdots \end{aligned}$$

—the coefficients A_k can be described

$$\begin{aligned} A_1(x_1) &= g + x_1 f' \\ A_2(x_1,x_2) &= x_2 f' + x_1 g' + \tfrac{1}{2}x_1^2 f'' \\ A_3(x_1,x_2,x_3) &= x_3 f' + x_2 g' + x_1 x_2 f'' + \tfrac{1}{2}x_1^2 g'' + \tfrac{1}{6}x_1^3 f''' \\ A_4(x_1,x_2,x_3,x_4) &= x_4 f' + x_3 g' + \tfrac{1}{2}(x_2^2 + 2x_1x_3)f'' + x_1x_2 g'' \\ &\quad + \tfrac{1}{2}x_1^2 x_2 f''' + \tfrac{1}{6}x_1^3 g''' + \tfrac{1}{24}x_1^3 f'''' \\ &\vdots \end{aligned}$$

These expressions are of such a design that the equations

$$\begin{aligned} A_1(x_1) &= 0 \\ A_2(x_1,x_2) &= 0 \\ A_3(x_1,x_2,x_3) &= 0 \\ A_4(x_1,x_2,x_3,x_4) &= 0 \\ &\vdots \end{aligned}$$

can be solved serially for $\{x_1, x_2, x_3, x_4, \dots\}$. *Mathematica*'s `Solve` command supplies

$$\left.\begin{aligned}
x_1 &= -\frac{g}{[f']} \\
x_2 &= \quad \frac{g}{2[f']^3}\Big\{2f'g' - f''g\Big\} \\
x_3 &= -\frac{g}{6[f']^5}\Big\{6[f'g']^2 - 9f'f''g\,g' + 3[f''g]^2 + 3[f']^2 g\,g'' - f'f'''[g]^2\Big\} \\
x_4 &= \quad \frac{g}{24[f']^7}\Big\{24[f'g']^3 + \text{nine more terms}\Big\} \\
&\;\;\vdots
\end{aligned}\right\} \tag{1}$$

The results reported above may seem patternless and off-puttingly complicated (and were certainly tedious to typeset), but if one had practical interest in (say) the structure of x_5 one would allow all preceding detail to remain sequestered within *Mathematica*'s memory and obtain the desired result in a matter of seconds. Very little computation time is involved: all depends upon how fast one can COPY output of the initial `Series` command and PASTE it into the final `Solve` command.

Evidently

$$x_n = (-)^n \frac{g}{n![f']^{2n-1}}\Big\{\text{complicated expression}\Big\} \;:\; n = 1, 2, 3, \ldots$$

and if x_0 is a zero not only of $f(x)$ but also of $g(x)$—whence of $f(x) + \epsilon\, g(x)$—we have $x_1 = x_2 = x_3 = \cdots = 0$: functional perturbation, in such a case, has no effect upon the placement of the zero. Not quite so obvious is the reason why in cases $f' \equiv f'(x_0) = 0$—in cases, that is to say, *when the zero lies at an extremal point—the method fails*: $x_1 = x_2 = x_3 = \cdots = \infty$.