2nd comments on Jim papers PhL 1.17.05

Time has passed, I have reviewed many things, my perspective on these papers is slightly altered.

I cut and pasted the sections below and put them into the binder with the related paper.

1. "Recurrence Relations for Ortho Polys and Efficient Algorithms". notes 1.17.05

I now think of this as the "first paper" and it should be entitled "Rehash of Wilf", I will soon get a look at Wilf (done, p55) In some recent notes "Hilbert Spaces" (see ortho poly binder) I have formalized the finite Hilbert Space $|x_i>$ that Jim discusses here. The main idea is that this is only a Hilbert space if x_i are the zeros of p_{n+1} because only then do you have an eigenvalue equation $X\ |x_i> = x_i\ |x_i>$ whose eigenvectors are linearly independent and can thus make a basis for the Hilbert space. This equation appears not in the abstract like this, but as
$\Sigma_{m=0}^{N} <n \mid X \mid m><m \mid x_i> = x_i\ \phi_n(x_i)$ where these matrix elements of X are the Jacobi matrix J_{nm}.

New realizations from re-reading this paper are:

- $1/S_i$ is w(x) in the finite position space
- $S_i = \Sigma_{n=0}^{N} \phi_n(x_i)^2$
- $A_i = 1/S_i$ = Gaussian weights, so you can compute them as in previous item
- the $\delta_{k,0}$ formula that I used in my own GQ proof has a generalization to $\delta_{k,j}$
- it is useful to do a double expansion of f(x) in product of $\phi_n\ \phi_m$
- x_i = nodes of p_{n+1} makes the $|x_i>$ really define a finite Hilbert Space

The example of the spherical Laplace solution makes use of the $|x_i>$ Hilbert Space properties. You have (4.3) as your assumed solution and you want to compute the A_L coefficients. The $|x_i>$ space orthogonality lets you extract A_L as shown in (4.5).

Now where is the "approximation" here? Suppose we know the zeros x_i of p_{n+1} exactly. I think the Hilbert space of the $|x_i>$ is exact. The approximation is that in (4.3) the sum over L stops at N. Only then can you use the $H_{N|+1}$ Hilbert space result to get (4.5). So you are approximating the radial function as a polynomial in r up to r^N. Probably for reasonably small N you can see that the higher partial waves are fading rapidly if the boundary condition is reasonably smooth, and then you would know how high you have to go in N to get a good answer. So this *really is* a numerical solution of a differential equation in an unusual manner. We don't write a difference equation and iterate it or do Runge Kutta.

What about Jim's "remarkably" statement? Well, it is true that we diagonalize the matrix problem and this gives us the "nodes" x_i and the "eigenvectors" $\phi_n(x_i)$. It is true that in doing so, we don't make use of the weight function w(x), nor do we make use of the integration range (a,b). Instead of these more conventional inputs to the problem, the inputs required by Jim are the recursion coefficients.

I have to wonder: in what situation would you actually know these recursion coefficients and not know the w(x) and the (a,b) ? Maybe the point is only that w and a,b are not used anywhere. This information is replaced by other information α_n and β_n. Another point of this paper is that your analysis work in the $|x_i>$ space is relatively efficient.

In my Jim email, I had no real questions on this paper, only comments.

Further note: I think you could argue that in (4.5) Jim is simply replacing the normal continuous orthogonality dx integral with a GQ sum -- remember that $1/S_i$ are the weights. So from that point of

view, he has done nothing new here. He is just saying: "I can approximate an integral by GQ". We know the weights for $P_L(x)$ etc.

2. "Automatic Computation of Zeros of Bessel Functions and other Special Functions". 1/17/05

I now know how to scale the ortho polys to get the symmetric matrix, see notes elsewhere (Addendum notes in the ortho poly binder).

So in this paper we have a quick Wilf rehash, a sine example, and then we apply to Bessel functions. Since these are not ortho polys, the perfect Wilf method does not work, and you have to worry about a certain error term R. Recall that the Wilf method for ortho polys is exact for any N and you can use it to compute the zeros to any degree of accuracy you want. For example, with a small N like N = 4, if you grind enough on the matrix, you can compute the 5 x_i of ϕ_5 to 1000 places of accuracy. Of course if you want the zeros of ϕ_{81} you then have to select N = 80.

With non-ortho polys like Bessel functions, it does not work this way. If you pick N = 4, you won't get the first 4 zeros of $J_0(x)$ to lots of accuracy. In his example, by choosing N = 100, Jim can compute the first 58 zeros of $J_0(x)$ to 14 places, but his claim is poorly stated IMHO. At this value of N, no amount of grinding on the tridiagonal matrix is going to improve the results, because there is this intrinsic error R that is finite at any finite N. Nevertheless, at least Jim is showing that you can extend the Wilf method to non ortho poly functions and at least get useful results if you are willing to go with a high N. And of course remember that each $J_\upsilon(x)$ has an infinite number of zeros since not polys. The idea is what is important here. The Coulomb Wave and Associated Legendre are other examples of non-ortho special functions that you can apply this method to.

I still think a reader would say: if I want the zeros of some function, I will just do Newton Raphson and the fact that it is slow does not matter because I will only do it once. Jim's idea might be useful if there were some real-time application requiring computation on the fly of zeros of functions that are always somehow changing. Yes, the FORTRAN code is concise, but so would code to issue Newton calls. So even after all this time, I am still pressed to see why this stuff is interesting or really useful. Unless of course you want the first 100,000 zeros, then Jim's efficiency does matter.

I had no first-round questions for Jim on this paper.

3. Gautschi paper "Evaluating certain slowly converging sums..." (my own title)

First I will just quote my remarks from my earlier comments:

"I was impressed by the result of this paper, although I don't know how these series arise in structural mechanics where (I guess) a plate presses against some object. I can see that, if p = 2, and z near 1, you would have to add up about 5 billion terms of the series to get e-20 accuracy, whereas his "trick" does it no problem. He transforms the sum in question into an integral with a certain interval and weight function w(t), which of course then has some weird ortho polys associated with it. These polys have a recursion formula. If you *knew* the α and β coefficients, you could define an iteration whose limit is the integral you want. Unfortunately, *why* this is so is not explained in this paper, but in another. Similarly, he quotes the method for finding the α and β, but that too is in some other paper. At least he puts the α and β values in an appendix. A good feature of the paper is that there are lots of experimental results showing how well the method actually works with closed-form test sums. I was happy to see similar experimental results appearing in later Jim papers. I presume this guy Gautschi is a world guru on ortho polys, maybe the Szego of his era. " [I think that is a correct assumption, he just did a new book in Aug 2004]

So, the "computation work" needed in this paper is two-fold. First, you have to compute the α and β recursion coefficients by some "well-known" but mysterious method, using the moments shown in (3.2). [I now know how to do this, not so mysterious now.] Second, you have to pick a good large value of υ, and then do backward iteration of the recursion shown in (2.12) until you get $\upsilon = 0$, which he claims is a cheap calculation, and you can see that the α and β appear in this formula.

Recall that Miklos showed me how to accelerate a poorly convergent sum by converting it into an integral and then doing GQ on that integral. That is NOT the method used here.

You can see that Gautschi has been motivated to compute these arcane sums, all good fun. Other than that his method refers to the recursion formula for ortho polys, I don't see much connection to the world of the Jim papers. It is just "something else you can do" with ortho poly recursion formulas. And of course I had no first-round questions for Jim on this third-party paper.

4. " GQ for two classes of log weight functions."

This paper is really just an early buggy draft of the B&B #1 paper, so no need to look at it ever again.

5. Ball & Beebe #1: "Efficient High-Accuracy Quadrature for Two Classes of Log Weight Functions"

I have written a whole new set of notes on this paper now, page by page, and it all makes sense. Also, I have in those notes answered my own questions to Jim.

6. Ball & Beebe #2: " The QUADLOG package"

I reread my first-round notes, they are fine. This is all just nuts and bolts, installing, testing, etc.

7. Jim " Half Range Hermites and Gaussian Quadratures with them" (SIAM 2003)

Just another application of the general theory I now know about. Integral of interest is (1.1) which has that "half range" for (a,b), namely $(0,\infty)$. Jim does a Gautschi modified Stieltjes' procedure to compute the recursion coefficients, because then you have your Jacobi matrix. You can then grind on this to get your x_i nodal values and your numerical eigenvectors like $f_n(x_i)$. Recall (B&B1 (9) for example) that you can then compute the Gaussian weight $1/S_i$ by just adding up the squares of these $f_n(x_i)$ values, summed over the vector index n. You are then all set to do GQ.

Jim starts with recursion coefficients in the Gautschi form (2.2) with α_n and β_n. Things turn out to be pretty messy, as in (2.10) and (2.15). He then finds a way to convert these two recursions into just one for g_n and then he can get the α_n and β_n from g_n as in (3.2) and (3.3). However, he then runs into numerical error problems with his recursion for g_n which is (3.5). This leads to various gymnastics and some numerical work, but he gets the job done somehow. In the end, he compares his results with those of others.

Notice that this is not a "log" paper, he is just trying to do the *basic* routine GQ.

Notice also that these half-Hermites do not fall under the umbrella of "classical ortho polys", so many pieces of information (such as Rodriguez) are not available. I did not see how the interval (a,b) fell out of Erdelyi's general theory, so maybe these are classical after all. In either case, it seems that there is not very much info on these ortho polys, so that motivated Jim to do this paper. I was at first unable to derive 2.9 and 2.11, but have since done these in pen notes. So my mysteries, just grunt work.

In my first round comments to Jim, I had this to say:

********** do it this way: **********

"The recursion relation implies

$(\phi_r, \phi_{n+1}) = (x\phi_r, \phi_n) - \alpha_n (\phi_r, \phi_n) - \beta_n (\phi_r, \phi_{n-1})$ where $(\phi_r, \phi_r) = T_r$

Setting r = n gives

$$0 = (x\phi_n, \phi_n) - \alpha_n T_n = S_n - \alpha_n T_n \quad \Rightarrow \quad \alpha_n = S_n / T_n \qquad (2.5)$$

where $S_n \equiv (x\phi_n, \phi_n)$, and setting r = n-1 gives

$$0 = (x\phi_{n-1}, \phi_n) - \beta_n T_{n-1} = T_n - \beta_n T_{n-1} \quad \Rightarrow \quad \beta_n = T_n / T_{n-1} \qquad (2.4)$$

In the last step above, since the ϕ_n are monic, we know $\phi_{n-1} = x^{n-1} +$ rest. Thus, $x\phi_{n-1} = x^n$ + rest'. Thus, we can write $x\phi_{n-1} = \phi_n$ + lincom of other ϕ, so $(x\phi_{n-1}, \phi_n) = T_n$ by orthogonality. "

In any event, I understand the 14x/stage error-magnification problem encountered in the g_n iteration, and how you got around it by two methods:

(1) pick good starting values for the g_n . Think of them like beads on a string, and run your little recursion-relation tool over that string maybe 10 times to get the beads positioned correctly. The endpoints have to stay fixed, so the g_N error does not go away.

(2) Adjust the entire string of beads at once, globally, using the matrix iteration (3.16). This converges much faster because ALL information is used at once rather than in piecemeal fashion. Also, I presume this lets g_N be adjusted along with the other beads. [By the way, I think this notion of global use of information is why the Jacobi matrix method for GQ works so well numerically.]

I guess that Steen published a table in which most of the digits are wrong! (your page 9).

I have a comment/question about computing the α and β here, but it is embedded in the next sections. I guess it seems that these things could be gotten in closed form and just evaluated with some gamma functions.

****

My last remark above refers to the fact that we can always write ϕ_n as in Erdelyi using the matrix form page 158 (4) if we know the weights c_n which in this case are all just gamma functions. But of course as n gets large the matrix gets large, and probably this is not too useful.