**Jim's half range Hermite paper** PhL 10.18.04

**1. Introduction.** The normal Hermite world has interval -∞ to ∞, and here we are going to do 0 to ∞ since this is the form many interesting integrals have.

**2. The method**. Equation (2.4) involves a little trick. If you close 2.2 against $\phi_r$ you get this

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

If you set r = n, you get

$$0 = (x\phi_n, \phi_n) - \alpha_n T_n = S_n - \alpha_n T_n \qquad \text{// hence 2.5}$$

If you set t = n-1, you get

$$0 = (x\phi_{n-1}, \phi_n) - \beta_n T_{n-1}$$

The trick is this. Since monic, you know that $\phi_{n-1} = x^{n-1} +$ rest. Thus, $x\phi_{n-1} = x^n$ + rest'. Thus, we know we can write $x\phi_{n-1} = \phi_n$ + lincom of other $\phi$. Then we have $(x\phi_{n-1}, \phi_n) = T_n$ and we get 2.4.

Backup for a moment and take a higher-level picture of this paper. We know that we can do GQ by the Jacobi matrix method if we have the recurrence relation coefficients which, in this paper, are called $\alpha$ and $\beta$. The problem Jim is going after is this: just how do you calculate these coefficients to a high degree of accuracy? What do I know about this subject? First of all, from Erdelyi, given the weight function, I can in theory compute all the power integrals to get the $c_i$ and then I can get the exact polynomials if I want. Are these power integrals known in closed form? The integral of the weight function alone is $1/2\Gamma(\gamma/2+1/2)$, according to Maple. So yes, I already know all the power integrals of the weight function, so I know the $c_i$ and thus I know the polys! I can then deduce the leading coefficients and that is all I need for the recursion coefficients.

Now back to Jim's method. By "fiddling around" with orthonormality of the $\phi_n$ he comes up with various relationships involving the $\alpha$ and the $\beta$, the things he wants to know. I think he things of (2.10) as his "first recurrence formula", and then 2.12 is a second formula. If you just start with some known $\alpha_0$ and $\beta_0$, you ought to be able to compute the higher ones iteratively. He is able to simplify 2.12 to the form 2.15. He seems to know what $\alpha_0$ is and $\beta_0 = 0$ which is handy.

So here is a scheme to do the computation. however, the claim is that error builds up for some reason as you do the scheme. Jim shows how you can replace your two recurrence relations by a single one for the coefficient $g_n$ and the result is the huge mess 3.5. He shows that for large n (meaning large Y, where $Y = 2n+\gamma$) you have 3.6, and if you try to iterate it in either direction, you multiply your error by 14 in each step -- an interesting problem! A previous work of somebody just did brute force with 75 digits to overcome this problem.

Jim proposes first some kind of "fixed point method" where he uses a rough starting set of $g_n$ and then he somehow iterates. I guess the idea is to use the n+1 and n-1 value to compute the n value in each little step, and this knocks your error DOWN by 14. Maybe you make a bunch of passes over your data until it quits moving, and then you finally have a set of $g_n$ that really satisfy the recursion formula.

Probably the upper point $g_N$ will never get adjusted, so its error will remain. So as you move down from this point, you will have 14x less error in each step in n. This means you are going to have to fine-tune $g_N$ by some method to reduce its error.

So where do the starting values of the $g_n$ come from? Jim proposes direct calculation by the old method up to about n = 8 (before the big error multiplication sets in), and then switch over to an asymptotic formula 3.9 above that. Again, just for good starting values.

He finds that he needs to do about 10 iterations on the mesh with g1 and g51 fixed, and this gives 16 digit accuracy. Do 30 iterations to get 33 digit accuracy!

Having done all this, Jim not proposes what he thinks is an even better method. Here, instead of thinking locally on that linear mesh, you just do the whole thing globally at once using a Jacobi like method using 3.15 as your matrix. This makes good sense to me. Now I think your endpoints can move as well as the interior points. He says really only one iteration is needed here which is sort of first order perturbation theory.

**Summary:** This paper presents two iterative methods to get the recursion coefficients, and then you can use those in the usual Jacobi matrix method to do your half-Hermite Gaussian quadratures. The half-Hermites do not appear in any of my books, but probably Shizgal has information on them.