on GAUSS/LAGUERRE QUADRATURE

Almost as a joke last night, after stumbling upon the two old Fortran subroutines of mine from 1992 that I will duplicate on the back, I typed them in anew to reexamine just how rapidly those abscissae x_i and weigths w_i succeed in estimating the known integral

$$\int_0^\infty e^{-x} \sin x \; dx = 1/2 \quad \text{via the sums} \quad S_n = \sum_{i=1}^{N} w_i \sin x_i \; .$$

Here the relevant polynomials begin with $L_0(x) = 1$, $L_1(x) = 1 - x$, $L_2(x) = 1 - 2x + x^2/2$, $L_3(x) = 1 - 3x + 3x^2/2 - x^3/6$ and march onward via the recurrence relation

$$(n+1)\; L_{n+1}(x) = (2n+1-x)\; L_n(x) - n\; L_{n-1}(x) \quad ,$$

whereas the recommended weights w_i obey $x_i/w_i = (n+1)^2\; [L_{n+1}(x_i)]^2$.

To cut this long story short, in this new millenium:

n	S_n	$S_n - 0.5$
1	0.841 470 984 808	0.**341470984808**
2	0.432 459 454 680	-0.0**67540545320**
3	0.496 029 827 481	-0.00**3970172519**
4	0.504 879 279 460	0.00**4879279460**
5	0.498 903 320 956	-0.00**1096679044**
6	0.500 049 474 798	0.0000**49474798**
7	0.500 038 911 995	0.0000**38911995**
8	0.499 987 753 735	-0.0000**12246265**
9	0.500 001 352 423	0.00000**1352423**
10	0.500 000 204 965	0.000000**204965**
11	0.499 999 888 715	-0.000000**111285**
12	0.500 000 018 908	0.0000000**18908**
13	0.500 000 000 114	0.000000000**114**
14	0.499 999 999 155	-0.000000000**845**
15	0.500 000 000 205	0.000000000**205**
16	0.499 999 999 985	-0.0000000000**15**
17	0.499 999 999 995	-0.00000000000**5**
18	0.500 000 000 002	0.00000000000**2**
19	0.500 000 000 000	0.000000000000
20	0.500 000 000 000	0.000000000000

18.335 digression to the roots of Laguerre polynomials: November 3, 1992

Invoked via: call LAGWTS (nlag).

```
      subroutine LAGWTS (n)

      implicit double precision (a-h,o-z)

      common / XWLAG / XLAG(100),WLAG(100)


      x = 1.0d0/n
      xprev = 111.1d0


      do 39 kase=1,n


      do 29 icase=1,50
      call LAGUERRE (n, kase-1, x, pred,dred, poly)
      if (abs(x-xprev).lt.1.0d-14) go to 31
      xprev = x
29    x = x - pred/dred


31    x = x - pred/dred
      XLAG(kase) = x

      call LAGUERRE (n+1, 0, x, pred,dred, poly)

      weight = x / ((n+1)*poly)**2
      WLAG(kase) = weight

39    x = x + 0.01d0


      return
      end
```

```
      subroutine LAGUERRE (n,known,  x, pred,dred, poly)

      implicit double precision (a-h,o-z)

      common / XWLAG / XLAG(100),WLAG(100)


      pn2 = 1.0d0
      pn1 = 1.0d0 - x

      do 29 k=2,n
      poly= ((2*k-1-x)*pn1-(k-1)*pn2)/k
      pn2 = pn1
29    pn1 = poly

      deri = -n
      if (abs(x).gt.1.0d-8) deri=n*(poly-pn2)/x


      prod = 1.0d0
      sum = 0.0d0

      if (known.le.0) go to 91

      do 79 k=1,known
      term = x - XLAG(k)
      prod = prod*term
79    sum = sum + 1.0d0/term

91    pred = poly/prod
      dred = (deri - poly*sum)/prod


      return
      end
```