Math 467/667: Programming Project 1

1. This question considers approximation of the area

$$A = \int_0^{\pi/2} f(z)dz \qquad \text{where} \qquad f(z) = \frac{1}{\sqrt{1+\tan z}}$$

by various numerical and algebraic techniques.

(i) Use a computer algebra system (or pencil and paper if you prefer) to verify the exact value of A is given by

$$A = \frac{\alpha}{4}\left\{\pi + \frac{3-\alpha^2}{\sqrt{2}}\ln\left(\frac{\alpha-\sqrt{2}}{\alpha+\sqrt{2}}\right) + 4\arctan(\alpha^2 - \alpha\sqrt{2})\right\}$$

where $\alpha = \sqrt{1+\sqrt{2}}$ and then show that $A \approx 1.060233292270744$.

In Mathematica the worksheet

In[29]:= `A1 = Integrate[1 / Sqrt[1 + Tan[z]], {z, 0, Pi / 2}]`

Out[29]= $\frac{1}{2}\sqrt{\frac{1}{2}+\frac{i}{2}}\left(\pi - (-1)^{3/4}\pi + 2\,i\left(\text{ArcCoth}\left[\sqrt{1-i}\right] + (-1)^{3/4}\,\text{ArcCoth}\left[\sqrt{1+i}\right]\right)\right)$

In[30]:= `N[A1, 15]`

Out[30]= $1.060233292270744 + 0. \times 10^{-16}\,i$

In[31]:= `A2 = a / 4 * (Pi + (3 - a^2) / Sqrt[2] * Log[(a - Sqrt[2]) / (a + Sqrt[2])] + 4 * ArcTan[a^2 - a * Sqrt[2]])`

Out[31]= $\frac{1}{4}\,a\left(\pi - 4\,\text{ArcTan}\left[\sqrt{2}\,a - a^2\right] + \frac{(3-a^2)\,\text{Log}\left[\frac{-\sqrt{2}+a}{\sqrt{2}+a}\right]}{\sqrt{2}}\right)$

In[32]:= `A3 = A2 /. a → Sqrt[1 + Sqrt[2]]`

Out[32]= $\frac{1}{4}\sqrt{1+\sqrt{2}}\left(\pi + 4\,\text{ArcTan}\left[1+\sqrt{2} - \sqrt{2\left(1+\sqrt{2}\right)}\right] + \frac{\left(2-\sqrt{2}\right)\text{Log}\left[\frac{-\sqrt{2}+\sqrt{1+\sqrt{2}}}{\sqrt{2}+\sqrt{1+\sqrt{2}}}\right]}{\sqrt{2}}\right)$

In[43]:= `N[A3, 15]`

Out[43]= 1.06023329227074

In[38]:= `DA = N[A1 - A3]`

Out[38]= $-2.22045 \times 10^{-16} + 1.66533 \times 10^{-16}\,i$

indicates the integral found by the computer algebra system and the expression given in the assignment agree and are equal 1.06023329227074.

(ii) Modify the program written in class or write your own to approximate A using the composite Gauss quadrature rule of order $\mathcal{O}(h^{10})$ where $h = (b-a)/m$. Compute

$$A_m = \texttt{comp}(f, 0, \pi/2, m) \qquad \text{and} \qquad E_m = A_m - E$$

for $m = 2^\ell$ and $\ell = 1, 2, \ldots, 12$. The output should look like

l	2^l	Am	Em
1	2	1.060677017116208e+00	4.437248454676190e-04
2	4	1.060389233493366e+00	1.559412226259660e-04
3	8	1.060288268508819e+00	5.497623807904084e-05

Note values for $l > 3$ have been omitted in the above table.

The program

```
x=[0., 0.5384693101056829,
   -0.5384693101056829, 0.9061798459386638, -0.9061798459386638]
A=[x.^0 x.^1 x.^2 x.^3 x.^4]'
B=[2,0,2/3,0,2/5]
W=A\B

function myf(x)
    1/sqrt(1+tan(x))
end
function quad(f)
    W'*f.(x)
end
function myg(j,x)
    h/2*myf(x*h/2+a+h*j+h/2)
end
function comp(f)
    s=0
    for i=0:(m-1)
        s=s+quad(x->myg(i,x))
    end
    return s
end

using Printf
@printf("#%3s %6s %22s %22s\n","l","2^l","Am","Em");
for l=1:12
    global a,b,m,h
    m=2^l
    a=0
    b=pi/2
    h=(b-a)/m
```

```
    ap=comp(myf)
    ex=1.060233292270744
    er=ap-ex
    @printf("%4d %6d %22.15e %22.15e\n",l,m,ap,er)
end
```

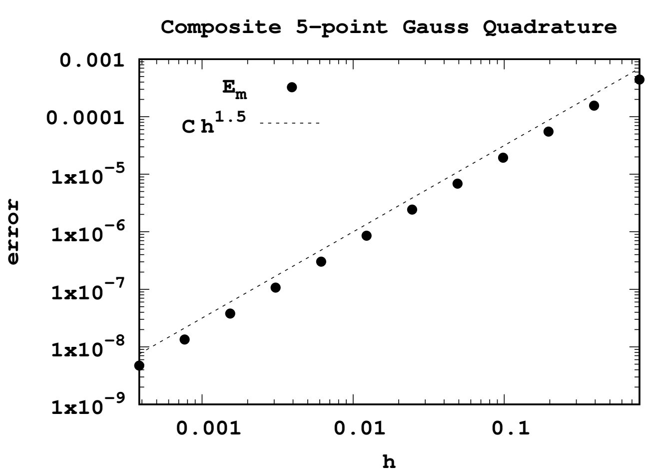
when run produces the output

```
#  l    2^l                     Am                     Em
   1      2  1.060677017116208e+00  4.437248454636222e-04
   2      4  1.060389233493366e+00  1.559412226219692e-04
   3      8  1.060288268508819e+00  5.497623807504404e-05
   4     16  1.060252702176081e+00  1.940990533744191e-05
   5     32  1.060240149969156e+00  6.857698412021662e-06
   6     64  1.060235716000310e+00  2.423729565492749e-06
   7    128  1.060234149041726e+00  8.567709819384817e-07
   8    256  1.060233595159112e+00  3.028883677202998e-07
   9    512  1.060233399353375e+00  1.070826309845785e-07
  10   1024  1.060233330129361e+00  3.785861690808190e-08
  11   2048  1.060233305655643e+00  1.338489852287239e-08
  12   4096  1.060233297002993e+00  4.732249037076031e-09
```

Note that the first three lines of output agrees with the sample output provided in the statement of the question.

(iii) Plot $\log E_m$ versus $\log h$ for the output obtained in the previous question. Do the points lie on a straight line? What is the slope of the line? What does the slope of the line indicate about the numerically observed order of convergence for this calculation? Is the numerically observed order of convergence consistent with what was expected theoretically? Explain.

The graph is



Note that the points lie on a straight line with a slope approximately equal to 1.5. This indicates that the order of convergence is $\mathcal{O}(h^{1.5})$. As the theoretical order of convergence was supposed to be $\mathcal{O}(h^{10})$, this is much less than what was expected. The reason for this is apparently because $\tan z \to \infty$ as $z \to \pi/2$ with the result that the function $1/\sqrt{1+\tan z}$ is not well approximated by polynomials near $z = \pi/2$.

(iv) Make the change of variables $y = \tan z$ to transform the integral appearing in question (i) to the form

$$\int_0^\infty g(y)dy.$$

Write down an explicit formula for $g(y)$.

Noting that

$$z = \arctan y \qquad \text{it follows that} \qquad dz = \frac{dy}{1+y^2}.$$

The limits of the transformed integral may be found by noting $y \to \infty$ as $z \nearrow \pi/2$ and $y = 0$ when $z = 0$. Thus,

$$\int_0^{\pi/2} \frac{dz}{\sqrt{1+\tan z}} = \int_0^\infty \frac{dy}{(1+y^2)\sqrt{1+y}}.$$

Therefore,

$$g(y) = \frac{1}{(1+y^2)\sqrt{1+y}}.$$

(v) Show that the further change of variables $x = 2y/(1+y) - 1$ transforms the integral in question (iv) into

$$\int_{-1}^1 h(x)\sqrt{1-x}\,dx \qquad \text{where} \qquad h(x) = \frac{2^{-1/2}}{1+x^2}.$$

For the limits, note that

$$x\Big|_{y=0} = \frac{2y}{1+y} - 1\Big|_{y=0} = -1 \qquad \text{and} \qquad \lim_{y\to\infty} x = \lim_{y\to\infty} \frac{2y}{1+y} - 1 = 2 - 1 = 1.$$

Now, solving for y yields

$$x(1+y) = 2y - (1+y), \quad x + xy = y - 1, \quad x + 1 = y - xy \quad \text{so} \quad y = \frac{1+x}{1-x}.$$

Consequently, by the chain rule

$$\frac{dy}{dx} = \frac{d}{dx}\frac{1+x}{1-x} = \frac{(1-x)+(1+x)}{(1-x)^2} = \frac{2}{(1-x)^2}.$$

Therefore,

$$\int_0^\infty \frac{1}{(1+y^2)\sqrt{1+y}} = \int_{-1}^{1} \frac{1}{\left(1+\left(\frac{1+x}{1-x}\right)^2\right)\sqrt{1+\left(\frac{1+x}{1-x}\right)}} \cdot \frac{2}{(1-x)^2}\,dx$$
$$= \int_{-1}^{1} \frac{1}{\left((1-x)^2+(1+x)^2\right)\sqrt{2/(1-x)}} \cdot 2\,dx$$
$$= \int_{-1}^{1} \frac{\sqrt{2}\sqrt{1-x}}{2+2x^2}\,dx = \int_{-1}^{1} \frac{\sqrt{1-x}}{\sqrt{2}(1+x^2)}\,dx$$

verifies that $h(x) = 2^{-1/2}/(1+x^2)$.

(vi) Define the weighted inner product and norm as

$$(f,g) = \int_{-1}^{1} f(x)g(x)\sqrt{1-x}\,dx \qquad \text{and} \qquad \|f\| = \sqrt{(f,f)}.$$

Use a computer algebra system (or pencil and paper if you prefer) to find the orthonormal polynomials p_n with respect to this inner product for $n = 0, 1, \ldots, 8$.

In Mathematica the worksheet

```
In[4]:= dp = Function[{f, g}, Integrate[f * g * Sqrt[1 - x], {x, -1, 1}]]

Out[4]= Function[{f, g}, ∫_{-1}^{1} f g √(1 - x) dx]

In[5]:= nm = Function[f, Sqrt[dp[f, f]]]

Out[5]= Function[f, √(dp[f, f])]

In[9]:= N0 = 8
        For[k = 0, k ≤ N0, k++,
         w[k] = x^k;
         For[j = 0, j < k, j++,
          w[k] = w[k] - dp[v[j], x^k] * v[j]];
         v[k] = Simplify[w[k] / nm[w[k]]];
         Print["v[", k, "]=", v[k]]]

Out[9]= 8
```

$$v[0] = \frac{\sqrt{3}}{2\times 2^{1/4}}$$

$$v[1] = \frac{\sqrt{7}\,(1+5x)}{8\times 2^{1/4}}$$

$$v[2] = \frac{\sqrt{11}\,(-17+14x+63x^2)}{64\times 2^{1/4}}$$

$$v[3] = \frac{\sqrt{15}\,(-23-225x+99x^2+429x^3)}{256\times 2^{1/4}}$$

$$v[4] = \frac{\sqrt{19}\,(827-1364x-9438x^2+2860x^3+12155x^4)}{4096\times 2^{1/4}}$$

```
v[5] = Sqrt[23] (1207 + 17615 x - 15210 x^2 - 90610 x^3 + 20995 x^4 + 88179 x^5) / (16384 × 2^(1/4))

v[6] = 1/(131072 × 2^(1/4)) 3 Sqrt[3]
  (-22181 + 54930 x + 512805 x^2 - 303620 x^3 - 1661835 x^4 + 312018 x^5 + 1300075 x^6)

v[7] = 1/(524288 × 2^(1/4)) Sqrt[31] (-33511 - 650573 x + 834309 x^2 +
  6319495 x^3 - 2860165 x^4 - 14820855 x^5 + 2340135 x^6 + 9694845 x^7)

v[8] = 1/(16777216 × 2^(1/4))
  Sqrt[35] (2492243 - 8234600 x - 96763884 x^2 + 84746536 x^3 + 566985650 x^4 -
  208134360 x^5 - 1037918700 x^6 + 141430680 x^7 + 583401555 x^8)
```

indicates that

$$p_8(x) = \frac{\sqrt{35}}{16777216 \cdot 2^{1/4}} \Big(2492243 - 8234600x - 96763884x^2 + 84746536x^3 + 566985650x^4 - 208134360x^5 - 1037918700x^6 + 141430680x^7 + 583401555x^8 \Big).$$

(vii) Find the eight roots x_k of $p_8(x)$ and the corresponding weights w_k such that

$$\int_{-1}^{1} x^j \sqrt{1-x}\, dx = \sum_{k=0}^{7} w_k x_k^j \qquad \text{for} \qquad j = 0, 1, \ldots, 15.$$

For reference the roots and weights you find should be consistent with

```
k            x_k                    w_k
0    -0.9624795445887677    0.1340407182534346
1    -0.8075678953806377    0.2835409515409297
2    -0.5496419355080006    0.3727176289987073
```

Note values for $k > 2$ have been omitted from the above table.

Continuing the worksheet from the previous problem compute the roots of $p_8(x)$ as

```
In[14]:= R8 = x /. Solve[v[8] == 0, x]
Out[14]= {-0.962..., -0.808..., -0.550..., -0.222...,
  0.135..., 0.474..., 0.753..., 0.936...}

In[15]:= R8n = N[R8, 16]
Out[15]= {-0.9624795445887677, -0.8075678953806377,
  -0.5496419355080006, -0.2215282765736194, 0.1349372926691484,
  0.4742968263639596, 0.7532724966821605, 0.9362867939115145}
```

To find the weights first compute the integrals

$$B_j = \int_{-1}^{1} x^j \sqrt{1-x}\, dx \qquad \text{for} \qquad j = 0, 1, \ldots, 7$$

using Mathematica as

```
In[8]:= For[j = 0, j < 8, j++,
         B[j] = dp[x^j, 1]]

In[11]:= N[Table[B[j], {j, 0, 7}], 16]

Out[11]= {1.885618083164127, -0.3771236166328253,
         0.5926228261372970, -0.2334574769631776, 0.3444722212533599,
         -0.1701640028429831, 0.2409924143584388, -0.1342917910867198}
```

Then find the weights by solving the system of linear equations

$$\sum_{k=0}^{7} w_k x_k^j = B_j \qquad \text{for} \qquad j = 0, 1, \ldots, 7.$$

Note that similar equations for $j = 8, \ldots, 15$ are automatically satisfied by the orthogonality properties of the the polynomials. The Julia program

```
x=[ -0.9624795445887677, -0.8075678953806377, -0.5496419355080006,
    -0.2215282765736194, 0.1349372926691484, 0.4742968263639596,
    0.7532724966821605, 0.9362867939115145]
A=[x.^0 x.^1 x.^2 x.^3 x.^4 x.^5 x.^6 x.^7]'
B=[1.885618083164127, -0.3771236166328253, 0.5926228261372970,
    -0.2334574769631776, 0.3444722212533599, -0.1701640028429831,
    0.2409924143584388, -0.1342917910867198]

w=A\B
using Printf
@printf("#%3s %22s %22s\n","k","x[k]","w[k]")
for k=1:8
    @printf("%4d %22.16f %22.16f\n",k,x[k],w[k])
end
```

calculates the weights and produces the output

```
#  k                   x[k]                   w[k]
   1    -0.9624795445887677     0.1340407182534332
   2    -0.8075678953806377     0.2835409515409342
   3    -0.5496419355080006     0.3727176289986990
   4    -0.2215282765736194     0.3865591072659580
   5     0.1349372926691484     0.3306471833577256
   6     0.4742968263639596     0.2290526625904178
   7     0.7532724966821605     0.1172417908442615
   8     0.9362867939115145     0.0318180403126977
```

which agrees to within rounding error of the desired output.

(viii) Use the weighted eight-point Gauss quadrature method developed above to approximate the area A. What is the error in this approximation? How does the composite formula used in question (ii) compare in terms of computational effort?

Modification of the Julia program in the previous question led to

```
x=[ -0.9624795445887677, -0.8075678953806377, -0.5496419355080006,
    -0.2215282765736194, 0.1349372926691484, 0.4742968263639596,
    0.7532724966821605, 0.9362867939115145]
A=[x.^0 x.^1 x.^2 x.^3 x.^4 x.^5 x.^6 x.^7]'
B=[1.885618083164127, -0.3771236166328253, 0.5926228261372970,
    -0.2334574769631776, 0.3444722212533599, -0.1701640028429831,
    0.2409924143584388, -0.1342917910867198]

w=A\B
f(x)=2^(-1/2)/(1+x^2)
approx=w'*f.(x)
exact=1.060233292270744
error=approx-exact
println("A=",approx)
println("error=",error)
```

with the output

```
A=1.0602323209055566
error=-9.713651873966e-7
```

After all the work was done to find the roots and weights, it took only 8 functional evaluations of $h(x)$ to approximate A to within about 9.7×10^{-7}. This error level is similar to taking $l = 7$ in the answer to question (ii). Thus, the composite formula took

$$2^7 \cdot 5 = 640$$

functional evaluations to arrives at a similar accuracy. Neglecting all the work needed to construct the new formula, which was anyway not much different that making the original Gauss quadrature rule in the first place, we conclude that the new weighted formula is $640/8 = 100$ times more efficient.

(ix) [Extra Credit and Math/CS 667] Let `quad_w(`f`)` be the weighted eight-point method above and let `quad(`f`)` be the standard five-point Gauss rule. Show that

$$\int_{b-h}^{b} \phi(t)\sqrt{b-t}\,dt = \frac{h^{3/2}}{2^{3/2}} \int_{-1}^{1} \phi\Big(\frac{xh}{2} + b - \frac{h}{2}\Big)\sqrt{1-x}\,dx.$$

Setting $h = (b-a)/m$ leads to the hybrid composite quadrature formula

$$\int_a^b f(x)dx \approx \texttt{hybrid(}f, a, b, m\texttt{)} = \sum_{j=0}^{m-2} \texttt{quad(}g_j\texttt{)} + \texttt{quad\_w(}\psi\texttt{)}$$

where $f(x) = \phi(x)\sqrt{b-x}$,

$$g_j(x) = \frac{h}{2} f\Big(\frac{xh}{2} + a + hj + \frac{h}{2}\Big) \quad \text{and} \quad \psi(x) = \frac{h^{3/2}}{2^{3/2}} \phi\Big(\frac{xh}{2} + b - \frac{h}{2}\Big).$$

Numerically determine the order of convergence of this method by approximating

$$\frac{1}{\sqrt{2}} \int_{-1}^{1} \frac{\sqrt{1-x}}{1+x^2} dx$$

following a similar procedure as in questions (ii) and (iii). Is there a way to fix this method so it converges faster?

Consider the change of variables given by

$$t = \frac{xh}{2} + b - \frac{h}{2} \qquad \text{and} \qquad dt = \frac{h}{2}\, dx.$$

Since $t = b$ when $x = 1$ and $t = b - h$ when $x = -1$ the limits are correct and

$$\begin{aligned} \int_{b-h}^{b} \phi(t)\sqrt{b-t}\, dt &= \int_{-1}^{1} \phi\Big(\frac{xh}{2} + b - \frac{h}{2}\Big) \sqrt{b - \Big(\frac{xh}{2} + b - \frac{h}{2}\Big)} \cdot \frac{h}{2}\, dx \\ &= \frac{h^{3/2}}{2^{3/2}} \int_{-1}^{1} \phi\Big(\frac{xh}{2} + b - \frac{h}{2}\Big) \sqrt{1-x}\, dx. \end{aligned}$$

This verifies the change of variables. Now, the Julia program

```
Gx=[0., 0.5384693101056829,
   -0.5384693101056829, 0.9061798459386638, -0.9061798459386638]
GA=[Gx.^0 Gx.^1 Gx.^2 Gx.^3 Gx.^4]'
GB=[2,0,2/3,0,2/5]
Gw=GA\GB

Wx=[ -0.9624795445887677, -0.8075678953806377, -0.5496419355080006,
    -0.2215282765736194, 0.1349372926691484, 0.4742968263639596,
    0.7532724966821605, 0.9362867939115145]
WA=[Wx.^0 Wx.^1 Wx.^2 Wx.^3 Wx.^4 Wx.^5 Wx.^6 Wx.^7]'
WB=[1.885618083164127, -0.3771236166328253, 0.5926228261372970,
    -0.2334574769631776, 0.3444722212533599, -0.1701640028429831,
    0.2409924143584388, -0.1342917910867198]
Ww=WA\WB

exact=1.060233292270744

function myh(x)
    2^(-1/2)/(1+x^2)
```

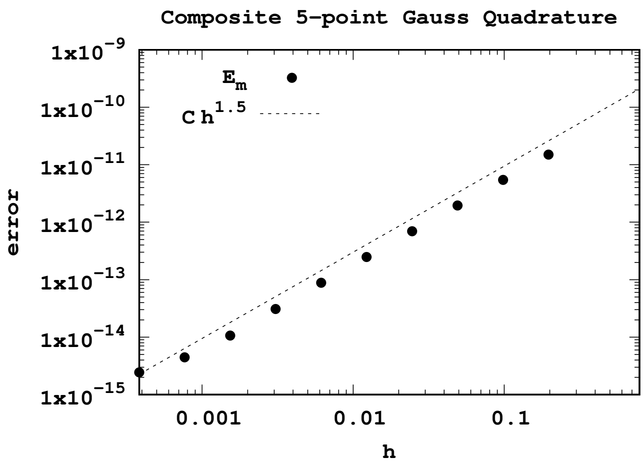
```
end
function myf(x)
    myh(x)*sqrt(1-x)
end

function Gquad(f)
    Gw'*f.(Gx)
end
function Wquad(f)
    Ww'*f.(Wx)
end

function Gmyg(j,x)
    h/2*myf(x*h/2+a+h*j+h/2)
end
function Wmyg(x)
    (h/2)^(3/2)*myh(x*h/2+b-h/2)
end
function comp(f)
    s=0
    for i=0:(m-2)
        s=s+Gquad(x->Gmyg(i,x))
    end
    s=s+Wquad(x->Wmyg(x))
    return s
end

using Printf
@printf("#%3s %6s %22s %22s\n","l","2^l","Am","Em");
for l=1:12
    global a,b,m,h
    m=2^l
    a=-1
    b=1
    h=(b-a)/m
    ap=comp(myf)
    er=ap-exact
    @printf("%4d %6d %22.15e %22.15e\n",l,m,ap,er)
end
```

with output

```
#  l    2^l                     Am                     Em
   1      2  1.060233193109279e+00 -9.916146548327731e-08
   2      4  1.060233291769965e+00 -5.007791958888674e-10
```

3	8	1.060233292285723e+00	1.497912904824261e-11
4	16	1.060233292276199e+00	5.455191853798169e-12
5	32	1.060233292272703e+00	1.959099549253551e-12
6	64	1.060233292271441e+00	6.974421040695233e-13
7	128	1.060233292270992e+00	2.478017790963349e-13
8	256	1.060233292270833e+00	8.859579736508749e-14
9	512	1.060233292270775e+00	3.086420008457935e-14
10	1024	1.060233292270755e+00	1.065814103640150e-14
11	2048	1.060233292270748e+00	4.440892098500626e-15
12	4096	1.060233292270746e+00	2.442490654175344e-15

led to the graph



which shows the numerical rate of convergence is disappointingly still $\mathcal{O}(h^{1.5})$, though the actual error is significantly less than it was before. It seems the rate of convergence is being spoiled by taking the last interval over which `quad(`g_j`)` is applied too close to the singularity at $x = 1$. To fix the method, I would try to vary the size of h so it was larger for the last piece of the integral represented by `quad_w(`ψ`)`.