Analysis of the Q Sum Rule PhL 12.11.15

In the original bowl doc of Nov 2011 and in my 2015 work I kept getting tails blowing up due to the particular Q function form I was using (see graphs below). When I changed from Bateman (36) to (45) everything started looking clean, as I first learned far below in this doc. // 12.31.15

1. Basics

Motivation: Want to learn more about what can and cannot be done with Q functions in Maple, when adding series of terms.

Past documentation: In the bowl in torroidals area, there are two docs that talk about this
the toroidal cap function.doc
the charged donut.doc
The original mws file that did this work was accidentally deleted.

Origin of the Sum Rule

The sum rule in question comes from here

$$1/\sqrt{a - b\cos(x)} = (1/\pi)\sqrt{2/b}\ \Sigma_{n=0}^{\infty}\ \varepsilon_n\ Q_{n-1/2}(a/b)\cos(nx) \qquad \text{// expansion}$$

$$\int_0^{\pi} dx\ \cos(nx)/\sqrt{a - b\cos(x)} = \sqrt{2/b}\ Q_{n-1/2}(a/b) \qquad \text{// projection} \qquad (10.1.8)$$

You set b = 1 and x = 0 a = z and the first line says

$$1/\sqrt{z - 1} = (1/\pi)\sqrt{2}\ \Sigma_{n=0}^{\infty}\ \varepsilon_n\ Q_{n-1/2}(z) \qquad \text{// expansion}$$

Then the sum rule is this

$$\sqrt{z - 1}\ \Sigma_{n=0}^{\infty}\ \varepsilon_n\ Q_{n-1/2}(z) = (\pi/\sqrt{2}) = 2.221441469 \qquad \text{// sum rule}$$

and this should be true for any z! Can rewrite as

$$\sqrt{ch\xi - 1}\ \Sigma_{n=0}^{\infty}\ \varepsilon_n\ Q_{n-1/2}(ch\xi) = (\pi/\sqrt{2}) = 2.221441469 \qquad \text{// sum rule}$$

Code to compute the sum rule:

Case 1: $\xi = 1$

So I started a new mws file called "the Q sum rule.mws". I enter the usual Bateman P and Q function stuff and the wps(n) routine, and then the following code where we do a solid simple case with $\xi = 1$:

The sum rule

$$\sqrt{ch\xi - 1}\ \Sigma_{n=0}^{\infty}\ \varepsilon_n\ Q_{n-1/2}(ch\xi) \;=\; (\pi/\sqrt{2}) \;=2.221441469$$

```
xi := 1; z := evalf(cosh(xi));
```

$$\xi := 1$$

$$z := 1.543080635$$

```
term := (n) -> eps(n)*Q(n-1/2,xi);    # term in the sum rule
```

$$term := n \to \mathrm{eps}(n)\, Q\left(n - \frac{1}{2}, \xi\right)$$

```
M := 100;
```

$$M := 100$$

```
termstore := array(0..M);
```

$$termstore := \mathrm{array}(0 \,..\, 100, [\ ])$$

```
partialsum := array(0..M);
```

$$partialsum := \mathrm{array}(0 \,..\, 100, [\ ])$$

For each n compute each term one time only and put into the termstore

Build partial series sums in the same loop, so one pass creates M partial sums

I am taking this code from page 12 of "the charged donut.doc"

```
acc := 0;
for n from 0 to M do
  termstore[n] := term(n);   # compute nth term in sum
  acc := acc + term(n);      # accumulate sum of terms to M
    partialsum[n] := acc;
od:
```

$$acc := 0$$

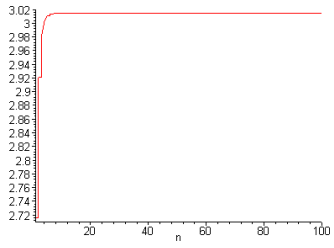
```
unassign('n'):
```

2. Case Studies

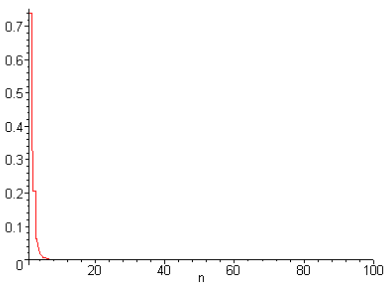
Case 1: $\xi = 1$ ($z = 1.543080635$)

So in this example, I have z = 1.54 with $\xi = 1$. I then do these two plots

```
plot(partialsum[n],n = 1..M);
```




```
plot(termstore[n],n = 1..M);
```



To check on the sum rule validity, I think pick a stable partial sum such as n = 40 :

```
sum_rule_sum := sqrt(z - 1)*partialsum[40];
```

$$sum_rule_sum := 2.221441471$$

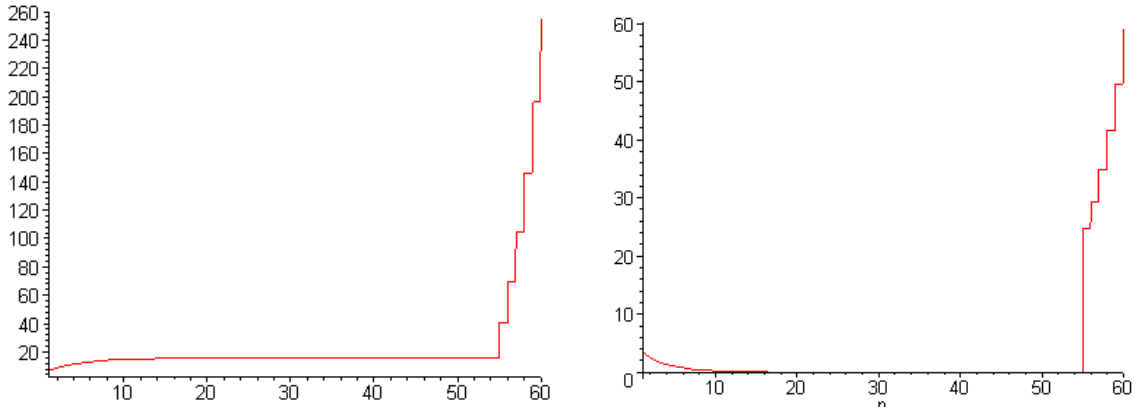
```
desired_sum := evalf(Pi/sqrt(2));
```

$$desired_sum := 2.221441469$$

So for $\xi = 1$, the sum rule is accurate to about 10 decimal places.

Case 2: $\xi = 0.2$ ($z = 1.020066756$)

I run into the same Legendre Q problem I had in Section 10!! Here is what happens with M = 70



Something very bad happens to the term[56] or nearby, terms are on the right. However, if I take the point n = 54 which seems stable, I get

```
sum_rule_sum := sqrt(z - 1)*partialsum[54];
```

$$sum_rule_sum := 2.221432084$$

```
desired_sum := evalf(Pi/sqrt(2));
```

$$desired_sum := 2.221441469$$

and we are accurate to about 5 decimal places. I think the sum is slowly rising though it looks flat, so it might rise to the correct value if only the terms did not blow up!

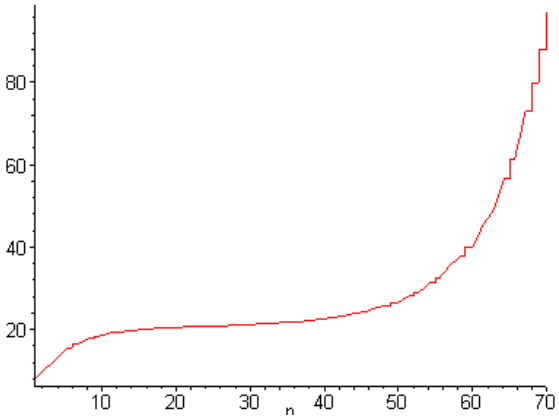
Case 3: $\xi = 0.15$ ($z = 1.011271110$)

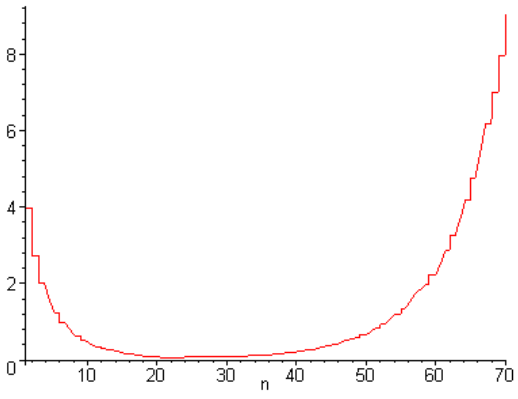
Let's go smaller still and see what happens.

$$\xi := .15$$

$$z := 1.011271110$$

M = 80






```
sum_rule_sum := sqrt(z - 1)*partialsum[40];
```

$$sum_rule_sum := 2.412371839$$

```
desired_sum := evalf(Pi/sqrt(2));
```

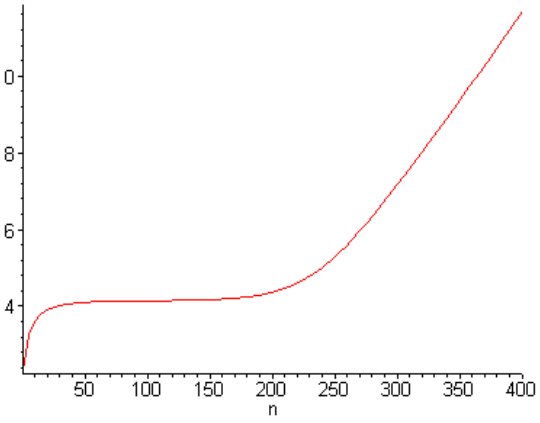
$$desired_sum := 2.221441469$$

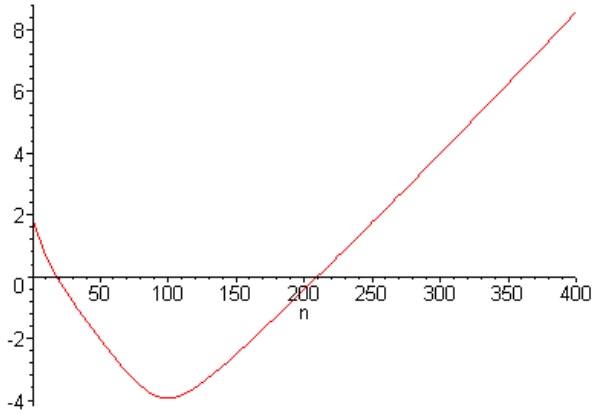
This case is miserable because there is no stable region! I picked off the n = 40 partial sum and it is too high, so probably I am already on the rising tail. Forget the details.

Question: Why does the term size rise again as shown on the far right?

Case 4: $\xi = 0.05$ (z = 1.001250260)

Now I plot log on the curves so I can see things better





I pick off a value at n = 100:

```
sum_rule_sum := sqrt(z - 1)*partialsum[100];
```

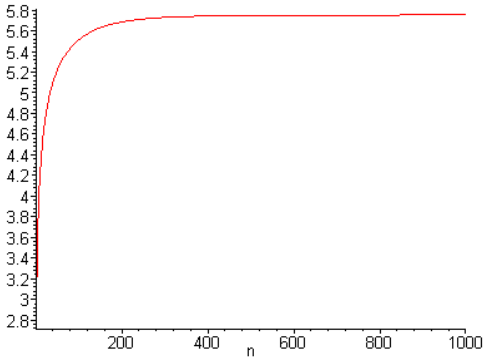
$$sum_rule_sum := 2.229938106$$

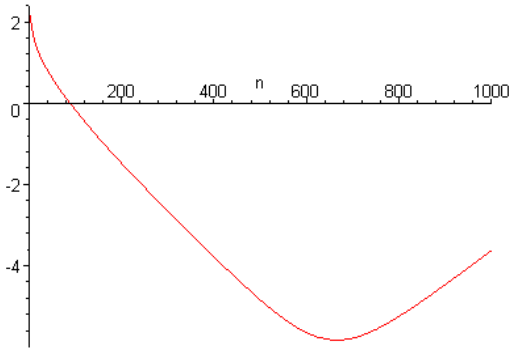
```
desired_sum := evalf(Pi/sqrt(2));
```

$$desired_sum := 2.221441469$$

and am still accurate to 3 decimal places.

Case 5: $\xi = 0.01$ (z = 1.000050000)





I pick off about 620 to get

```
sum_rule_sum := sqrt(z - 1)*partialsum[620];
```

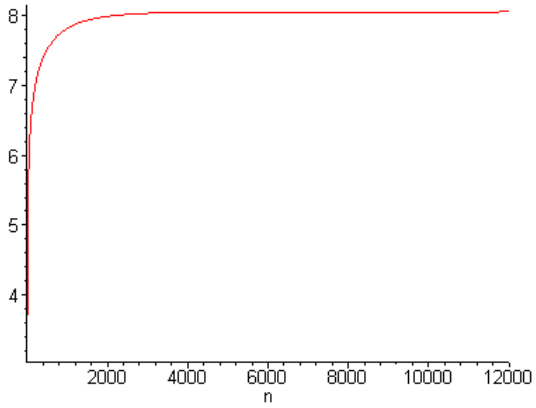
$$sum_rule_sum := 2.221316848$$

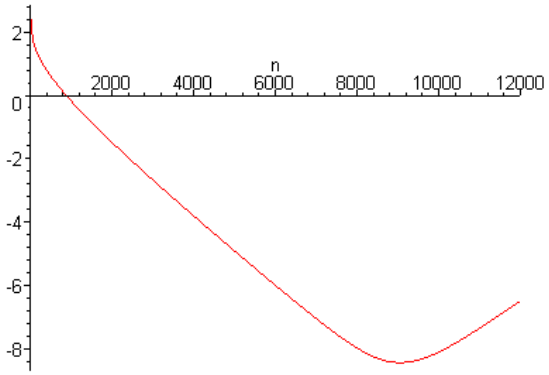
```
desired_sum := evalf(Pi/sqrt(2));
```

$$desired_sum := 2.221441469$$

So despite the tail problems, I am able here to get about 5 places of accuracy, and I am adding a whopping 620 terms to get this result!

Case 5: $\xi = 0.001$ (z := 1.000000500)





Here we look good all the way out to 9000 terms or so before the error sets in!

```
sum_rule_sum := sqrt(z - 1)*partialsum[9000];
```

$$sum_rule_sum := 2.221473455$$

```
desired_sum := evalf(Pi/sqrt(2));
```

$$desired_sum := 2.221441469$$

We are still accurate to 7 places!!

3. Main Point of my doing this sum rule

In my original Appendix B, I was arguing that, despite the asymptotic problems, the sum rule is very well validated for small z values if you pick a stable location. I used this fact to *claim* that probably the capacitance formula is also trustworthy for very small z values. Even today I think this is a good argument.

4. Why does the Q function term tail rise ?

I do know that the problem is with the hypergeometric function, not the associated gamma functions. That is, the problem is in here:

$$Q(\nu,\xi) = Q_\nu(ch\xi) = 2^\nu\,[\Gamma(1+\nu)]^2\,(ch\xi+1)^{-\nu-1}/\Gamma(2+2\nu) * F(1+\nu,1+\nu,2+2\nu,\, 2/(1+ch\xi)) \qquad (7.4.2)$$

$$F(1+\nu,1+\nu,2+2\nu,\, 2/(1+ch\xi))$$

Let's look at the z argument values for the cases above:

ξ	z
1	.7864477330
.2	.9900662906
.15	.9943960266
.05	.9993752606
.01	.9999750006
.001	.9999997500

I know there is a log(z-1) singularity involved here.

5. Why did I pick the Q form I picked from Bateman?

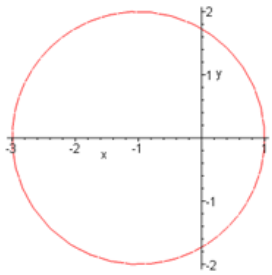
Where did I decide on the forms I have long been using? This was done in "custom Maple Legendre functions.doc" in my Legendre area.

When I picked Bateman (36), it maps z = 1 to z' = 2/(1+z) = 1, so we have a problem.
it maps z = ∞ to z' = 0 where things are simple.

<u>Consider now some other Q forms:</u>

Would it not be better for work involving z near 1 to use some other Q form?

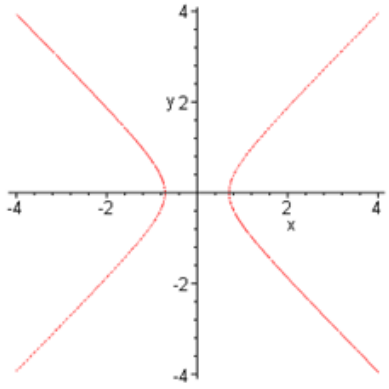
In my doc "convergence for Bateman functions" I show the convergence region for each of his forms.
The one I picked was (36)



(1+z)/2	(15) (33)	inside
2/(1+z)	(18) (36)	outside

and I picked it because it converges on the real axis z > 1. Yes, it has a problem at z = 1.

What about this one:

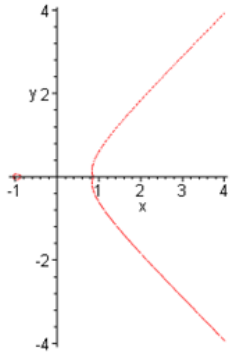


$1-1/z^2$	(24) (42)	left/right	try z = ±2
$z^2/(z^2-1)$	(25) (43)	middle	try z = 0

The real intersection is $1/\sqrt{2}$ = .707.

The left/right choice here gives you z = .7 to infinity, so stays away from z = 1. There are 2 terms for (42) it is true. But I see now that both blow up at μ = 0 so no go.

What about (48)?



$[(2\sqrt{z^2-1})/(z+\sqrt{z^2-1})]$ (28) (46) right and z=-1 try z = 0 =>div

$[(z+\sqrt{z^2-1})/(2\sqrt{z^2-1})]$ (30) (48) left

Where do I get the red curves from?

z := x+I*y;
implicitplot(abs((1-z)/2) = 1, x=-4..4,y=-4..4 , scaling = CONSTRAINED, grid = [100,100]);

The F(a,b,c,z) converges inside a unit circle about the origin, so the boundary is |z| = 1. So if you want to ask about a form where z' = (1-z)/2, then your locus will be | (1-z)/2| = 1. So that explains my red curves.

Now let's write out the form (48) for Q to see if it is even viable:

$$Q_\nu(z) = \sqrt{\pi/2}\,\Gamma(\nu+1/2)\,(z^2-1)^{-1/4}\,(z-\sqrt{z^2-1})^{\nu+1/2}[\Gamma(\nu+1)]^{-1}\;F(1/2,1/2,1/2-\nu;(z+\sqrt{z^2-1})/(2\sqrt{z^2-1}))$$

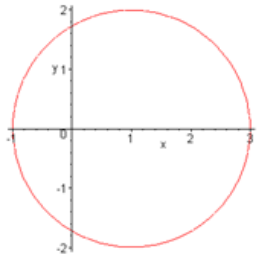
$$+(1/\sqrt{\pi})\,2^{-\nu-1}\,\Gamma(\nu+1)\Gamma(-1/2-\nu)(z^2-1)^{-1/2-\nu/2}\,F(1+\nu,1+\nu,\nu+3/2;(z+\sqrt{z^2-1})/(2\sqrt{z^2-1}))$$

It is long and messy, that is for sure. If ν = n-1/2, we have ν+1/2 = n, and -ν-1/2 = -n and second term has a gamma pole so this is a NO GO. Nice try. Rule out (48).

What about (34) ? Has a μ pole at 0, so Rule out (34).

OK, so I am stuck really with (36) for use on z > 1,

But in my capacitance study I want z near 1, so *for that* why not use the most basic form (32) inside this circle? But (32) for Q has μ=0 poles, so no good. Rule out (32).



(1-z)/2 (14) (32) inside
2/(1-z) (19) (37) outside

How about (38)? Same μ=0 pole, so Rule out (38).

How about (45) which converges z > 0 which is pretty good. Write this one out

$$Q_\nu(z) = \sqrt{\pi}\,\Gamma(1+\nu)\,(z+\sqrt{z^2-1}\,)^{-1-\nu}\;\;(\Gamma(\nu+3/2))^{-1}\;\;F\left(1/2,\,1+\nu;\,3/2+\nu;\,\frac{z-\sqrt{z^2-1}}{z+\sqrt{z^2-1}}\,\right)$$

Now if ν+1/2 = n I see no problems.

Now if I write $z = ch\xi$ $\quad z-\sqrt{z^2-1} = ch\xi - sh\xi = e^{-\xi}$

$z+\sqrt{z^2-1} = ch\xi + sh\xi = e^{+\xi}$

Then the above says

$$Q_\nu(ch\xi) = \sqrt{\pi}\,\Gamma(1+\nu)\,(e^{\xi})^{-1-\nu}\;\;(\Gamma(\nu+3/2))^{-1}\;\;F(1/2,\,1+\nu;\,3/2+\nu;\,e^{-2\xi})$$

But as ξ⟶ 0, z⟶ 1 and we hit the cut. Is my locus picture wrong? Well no, I reconstructed it and it is fine

```
z := x + I*y;
```

$$z := x + Iy$$

```
implicitplot(abs((z-sqrt(z^2-1))/(z+sqrt(z^2-1))) = 1, x=-0.1..0.1,y=-4..4 , scaling =
CONSTRAINED, grid = [100,100], thickness = 2);
```



The locus as I now see requires that $|e^{-2\xi}| = 1$ which requires ξ = iθ and then $z = ch\xi = ch(i\theta) = \cos(\theta)$ so this seems to say that z = cos(θ) which would be a horizontal red line from -1 to 1, so what gives here? OK, the above plot is wrong as you find out . The locus is really (-1,1) so in fact we do hit the branch point after all . So Rule out (45). I guess I could try it and see what it does. Maybe try this in the sum rule test program "the Q sum rule.mws".

Test Bateman (45) for Q.

For the sum rule work, does it work any better than Bateman (36)??

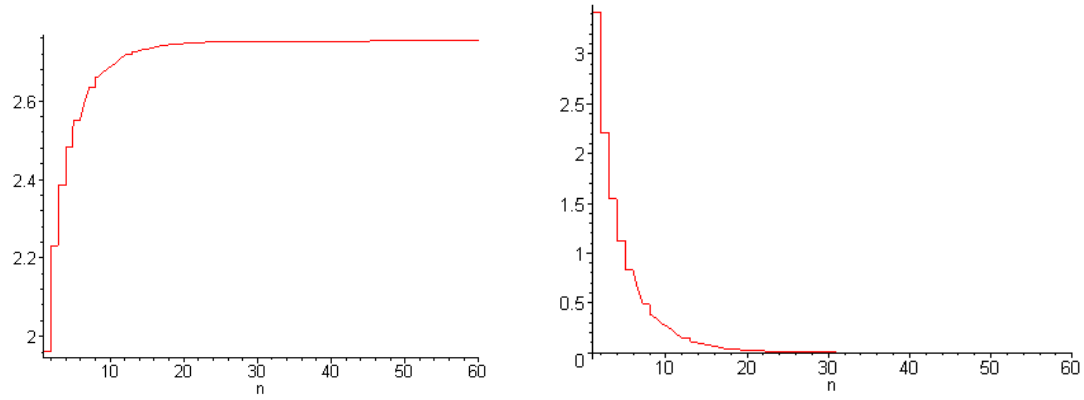
Case 1: ξ = 1, looks the same

```
sum_rule_sum := sqrt(z - 1)*partialsum[70];
                          sum_rule_sum := 2.221441471
desired_sum := evalf(Pi/sqrt(2));
                          desired_sum := 2.221441469
```

and accuracy exactly the same.

Case 2: $\xi = 0.2$ (z = 1.020066756)

Curves are perfectly clean compared to the mess last time! Here they are




```
sum_rule_sum := sqrt(z - 1)*partialsum[55];
                          sum_rule_sum := 2.221433876
desired_sum := evalf(Pi/sqrt(2));
                          desired_sum := 2.221441469
```

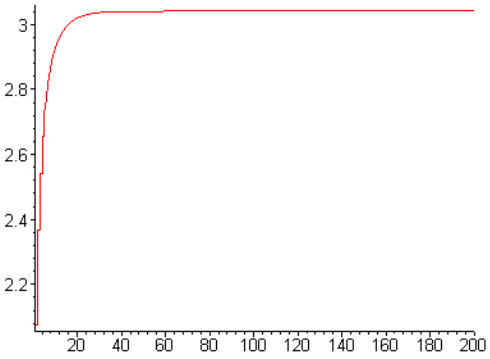
This seems to avoid that ripup problem I was getting with form (36) for Q. Suppose I increase M in this same example: it stays perfectly clean!!!! I can now compare out at 200 terms:

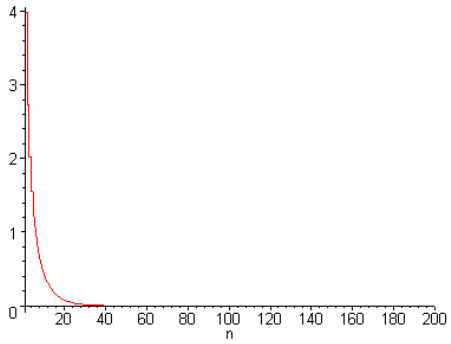
```
sum_rule_sum := sqrt(z - 1)*partialsum[200];
                          sum_rule_sum := 2.221441488
desired_sum := evalf(Pi/sqrt(2));
                          desired_sum := 2.221441469
```

Astounding accuracy!!! Let's keep trying the other cases.

Case 3: $\xi = 0.15$ (z = 1.011271110)

Again totally clean.





And

```
sum_rule_sum := sqrt(z - 1)*partialsum[200];
```

$$sum_rule_sum := 2.221441514$$

```
desired_sum := evalf(Pi/sqrt(2));
```

$$desired_sum := 2.221441469$$

.

Case 4: $\xi = 0.05$ (z = 1.001250260)

Again it is perfectly flat all the way.

Case 5: $\xi = 0.01$ (z = 1.000050000)

Again flat all the way. Try the log plot,

I cannot find any problems with this new Q function! Wow, that was a lucky thing to stumble on today, especially after all the trouble I was having yesterday.