MoHPC The Museum of HP Calculators

Search Member List Calendar Help

Welcome back, Valentin Albillo. You last visited: Yesterday, 11:41 PM (**User CP** — Log Out)
View New Posts | View Today's Posts | Private Messages (Unread 0, Total 145)

Current time: 04-22-2019, 01:00 AM
Open Buddy List

HP Forums / HP Calculators (and very old HP Computers) / General Forum / Short & Sweet Math Challenge #21: Powers that be

Pages (2): « Previous 1 **2**

NEW REPLY

Short & Sweet Math Challenge #21: Powers that be

Threaded Mode | Linear Mode

11-09-2016, 12:48 AM **Post: #21**

Paul Dale
Senior Member

Posts: 1,452
Joined: Dec 2013

RE: Short & Sweet Math Challenge #21: Powers that be

> **J-F Garnier Wrote:** (11-08-2016 04:03 PM)
>
> > **Paul Dale Wrote:** (11-08-2016 03:38 PM)
> >
> > If there is one real root > 1 and all of the remaining (possibly complex) roots have |root| < 1, then the polynomial is of the required form.
>
> It may be the element I was missing, but can you explain or justify this statement?
> J-F

I'll try to justify the statement.

Newton's identities allow one to calculate the sum of a power of the roots of a polynomial from its coefficients. The constraints on the polynomials given here (monic with integral coefficients) mean that the sum of the roots to any power will be an integer.

For one root to approximate an integer as the power is raised, it must be the single dominant root. i.e. all other roots must approach zero as they are raise to higher and higher powers. Thus, their absolute value must be strictly less than unity. The dominant root must be one or greater.

Pauli

PM FIND QUOTE REPORT

11-09-2016, 07:45 PM **Post: #22**

J-F Garnier
Senior Member

Posts: 302
Joined: Dec 2013

RE: Short & Sweet Math Challenge #21: Powers that be

Following Paul's approach, I modified my Emu71 program to explore these solutions.
I kept the possibility for the least parameter a0 to be +/-1.

Results:

```
Order 2
1.61803398875 ok x^2-x-1
Order 3
1.83928675521 ok x^3-x^2-x-1
1.46557123188 ok x^3-x^2-1
1.32471795724 ok x^3-x-1
Order 4
1.92756197548 ok x^4-x^3-x^2-x-1
1.38027756910 ok x^4-x^3-1
Order 5
1.96594823665 ok x^5-x^4-x^3-x^2-x-1
1.88851884548 ok x^5-x^4-x^3-x^2-1
1.77847961614 ok x^5-x^4-x^3-x^2+1
1.81240361927 ok x^5-x^4-x^3-x-1
1.70490277604 ok x^5-x^4-x^3-1
1.44326879127 ok x^5-x^4-x^3+x^2-1
1.57014731220 ok x^5-x^4-x^2-1
1.53415774491 ok x^5-x^3-x^2-x-1
Order 6
1.98358284342 ok x^6-x^5-x^4-x^3-x^2-x-1
1.91118343669 ok x^6-x^5-x^4-x^3-x-1
1.80750202302 ok x^6-x^5-x^4-x^3+1
1.71428532914 ok x^6-x^5-x^4-x^2+1
1.74370016590 ok x^6-x^5-x^4-x-1
```

```
1.50159480354 ok x^6-x^5-x^4+x^2-1
1.66040772247 ok x^6-x^5-x^3-x^2-1
Order 7
1.99196419661 ok x^7-x^6-x^5-x^4-x^3-x^2-x-1
1.97504243425 ok x^7-x^6-x^5-x^4-x^3-x^2-1
1.92212800436 ok x^7-x^6-x^5-x^4-x^2-x-1
1.88004410997 ok x^7-x^6-x^5-x^4-x-1
1.85454747658 ok x^7-x^6-x^5-x^4-1
1.74745742449 ok x^7-x^6-x^5-x^4+x^3-1
1.77452005864 ok x^7-x^6-x^5-x^3-1
1.67752174784 ok x^7-x^6-x^5-x^2+1
1.65363458677 ok x^7-x^6-x^5-1
1.54521564973 ok x^7-x^6-x^5+x^2-1
1.60134733379 ok x^7-x^6-x^4-x^2-1
1.59000537390 ok x^7-x^5-x^4-x^3-x^2-x-1
Order 8
1.99603117974 ok x^8-x^7-x^6-x^5-x^4-x^3-x^2-x-1
1.97061782036 ok x^8-x^7-x^6-x^5-x^4-x^3-1
1.96113576083 ok x^8-x^7-x^6-x^5-x^4-x^3+1
1.92172206590 ok x^8-x^7-x^6-x^5-x^4+1
1.90988759678 ok x^8-x^7-x^6-x^5-x^4+x+1
1.88166942009 ok x^8-x^7-x^6-x^5-x^3+1
1.88670294847 ok x^8-x^7-x^6-x^5-x^2-x-1
1.86221396917 ok x^8-x^7-x^6-x^5-x-1
1.84771602509 ok x^8-x^7-x^6-x^5-1
1.83032136835 ok x^8-x^7-x^6-x^5+1
1.83524591514 ok x^8-x^7-x^6-x^4-x^3-x-1
1.74243322086 ok x^8-x^7-x^6-x^4+1
1.65524582449 ok x^8-x^7-x^6-x^2+1
1.57367896839 ok x^8-x^7-x^6+x^2-1
1.72778030821 ok x^8-x^7-x^5-x^4-x^3-x^2-1
1971 candidates (1<root<2)
48 candidates (dominant root)
```

An interesting point is that solutions with the last parameter a0=+1 are found.
If solutions are restricted to a0=-1, there are only 37 candidates.
Another open point is that some solutions found by my pure numerical method are not there, for instance:

```
1.75487766625 ok x^4-x^3-x^2-1
```

My HP71 program (actually close to Gerson's program, but I explored all roots):

```
10 ! --- SSMC21 ---
20 OPTION BASE 0 @ DIM A(10)
30 C=0 @ C2=0
40 FOR D=2 TO 8
50 DISP "Order";D
60 DIM A(D) @ COMPLEX R(D-1)
70 A(0)=1
80 ! A(D)=-1 ! a0=-1 assumed...
90 K=3^(D-1) ! numbers of coefficient combinaisons
100 K=K*2 ! for a0=+/-1
110 FOR J=0 TO K-1
120 L=J
130 ! build the coefficients
140 IF MOD(L,2) THEN A(D)=1 ELSE A(D)=-1 ! for a0=+/-1
150 L=L DIV 2 ! for a0=+/-1
160 FOR I=D-1 TO 1 STEP -1
170 A(I)=MOD(L,3)-1 @ L=L DIV 3
180 NEXT I
190 ! find roots of polynomial A
200 MAT R=PROOT(A)
210 FOR K=0 TO D-1
220 X=REPT(R(K))
230 IF IMPT(R(K))=0 AND X>1.000001 AND X<2 THEN GOSUB 300
240 NEXT K
250 NEXT J ! next polynomial of order D
260 NEXT D ! next order polynomials
270 DISP C;"candidates (1<root<2)"
280 DISP C2;"candidates (dominant root)"
290 END
300 'T': ! evaluate candidate
310 C=C+1
320 F=1
330 FOR I=0 TO D-1
340 IF I<>K AND ABS(R(I))>=1 THEN F=0
350 NEXT I
360 IF F=0 THEN 480
370 C2=C2+1
380 FIX 11 @ DISP X;"ok"; @ STD
```

```
390 DISP " x^";STR$(D);
400 FOR I=1 TO D-1
410 IF A(I)=1 THEN DISP "+";
420 IF A(I)=-1 THEN DISP "-";
430 IF A(I)<>0 THEN DISP "x";
440 IF A(I)<>0 AND D-I<>1 THEN DISP "^";STR$(D-I);
450 NEXT I
460 IF A(D)>0 THEN DISP "+";
470 DISP STR$(A(D))
480 RETURN
```

J-F

EMAIL PM WWW FIND QUOTE REPORT

11-10-2016, 01:53 AM **Post: #23**

Valentin Albillo
Senior Member

Posts: 344
Joined: Feb 2015
Warning Level: 0%

RE: Short & Sweet Math Challenge #21: Powers that be

Hi all:

About a week has elapsed since I posted **S&SMC#21** and in that time some of you have contributed a number of interesting ideas and even actual code implementing them, as well as relevant discussion.

Thanks so much for your interest and contributions, now is the time to give my own *original solution* to the challenge, but first I'll present some theory which will help understand the algorithm I used.

The challenge asked for you to write a program which should find and output all the constants in the range >1 and <2 with the property that their increasing integer powers would be nearer and nearer to *integer* values, outputting both each constant and its minimal polynomial, limited to degree **8** or less and coefficients **-1**,**0**,**+1**, with the leading coefficient in particular being always **+1**. The *Golden Ratio* (Phi) was given as an example of such a constant.

J-F Garnier tried the natural approach of constructing all -1,0,+1 polinomials up to degree 8, finding a candidate root, then computing its increasing integer powers and examining the fractional parts to see if they were converging to either 0.0000... or 0.9999.....

While natural, this approach is doomed to fail because the exponentially growing powers quickly exceed the accuracy of any calculator and the fractional parts get inaccurate or even lost, thus the test eventually fails.

The proper way comes by the way of a little theory, which I'll succintly explain with just the minimum amount of detail to make it short. Given the integer coefficients a(n) of a monic polynomial (leading term 1), say:

```
x^n + a(n-1)*x^(n-1) + ... + a(2)*x^2 + a(1)*x + a(0)
```

any *symmetric* function of the roots can be expressed in terms of the coefficients, where symmetric means that the value of the function is the same (invariant) for all permutations of the roots. For example, the function:

```
Sum of the roots = x1 + x2 + x3+ ... + xN
```

is a symmetric function because its value is the same no matter how the roots are permuted, so it can be computed as an integer function of the coefficients, in this case:

```
Sum of the roots = x1 + x2 + ... + xN = -a(n-1)
```

*(Example: Sum of the roots of [x^3-6*x^2+11*x-6 = 0] = -(-6) = +6. The roots are 1,2,3 and 1+2+3 = 6)*

The same goes for the sum of the roots raised to any integer power, such as for example:

```
Sum of the 10-th powers of the roots = x1^10 + x2^10 + ... + xN^10
```

which, being a symmetric funtion, can also be computed as an integer expression of the coefficients, the expression being more complicated but still delivering an integer result, thus we have:

```
x1^10 + x2^10+ ... + xN^10 = integer value
```

no matter what the roots are and no matter if they are real, complex, or a mix. Now consider what happens when one of the roots is real with an absolute value > 1 while all other roots (real or complex) have an absolute value < 1, such as is the case for the *Golden Ratio*, namely:

```
Min. Polynomial : x^2-x-1
root x1 : +1.61803398875
root x2 : -0.61803398875

Sum of their Nth powers S(N) = x1^N + x2^N = 1.61803398875^N + (-0.61803398875)^N
```

If we the compute S(N) for N=10, 20, 30, .. on a 12-digit HP calc we get:

```
        1.61803398875^10 + (-0.61803398875)^10 = 122.991869381 + 0.0081306187558 = 123 exactly
        1.61803398875^20 + (-0.61803398875)^20 = 15126.9999339 + 0.0000661069613 = 15127 exactly
        1.61803398875^30 + (-0.61803398875)^30 = 1860498.00000 + 0.0000005374904 = 1860498 exactly
```

and you see what's happening: the powers of the root with absolute value > 1 grow *larger and larger* while the powers of the root with absolute value < 1 grow *smaller and smaller*. As their sum has to be an *integer* value and the smaller root's powers are contributing less and less to the sum, the powers of the larger root are forced to approach the integer value of the whole sum by themselves and thus be almost-integer.

This generalizes to all monic polynomials of any degree so that the criterium is that the monic polynomial of degree N must have one real root of absolute value > 1 and all its other N-1 roots, whether real or complex, must have absolute values <1. This way the sums of the powers of the smaller roots will tend to 0 and the powers of the one root with absolute value > 1 will tend to the integer sum and so will tend to being integer themselves.

The algorithm is now quite clear:

- iterate through all monic polynomials of coefficients 1,0,+1 and degrees up to 8 (per the challenge)
- for each polynomial find all its roots and check:
    - that there's only one root of absolute value >1
    - that all other roots have absolute values < 1
- if such a root is found, check if its absolute value is < 2 (per the challenge requirement)
    - if it is, record both the root and its minimal polynomial
- after all polynomials have been examined, sort and ouput the recorded roots/polynomials.

and this is my **11-line** generic implementation for the **HP-71B** (39 statements, 587 bytes), readily adaptable to any **RPL** and **RPN** HP models with a polynomial root-finding capability:

```
>CAT

SSMC21 BASIC 587 bytes

>LIST

        1  DESTROY ALL @ OPTION BASE 1 @ INPUT L @ DIM P(L+1),F(L),T$[60] @ COMPLEX R(L)
        2  INPUT "Sort by C,P ? ","C";X$ @ K=16*(X$="P") @ N=0 @ P(1)=1 @ H=0 @ SETTIME 0
        3  H=H+1 @ F(H)=-1 @ REPEAT @ P(H+1)=F(H) @ IF H<L THEN 3 ELSE MAT R=PROOT(P)
        4  X=0 @ FOR I=1 TO L @ IF ABS(R(I))<1 THEN 5 ELSE IF X THEN 6 ELSE X=REPT(R(I))
```

```
 5  NEXT I @ IF X>1 AND X<2 THEN N=N+1 @ DIM S$(N)[60] @ Z=L+2 @ GOSUB 9
 6  F(H)=F(H)+1 @ UNTIL F(H)>(H#1 AND H#L) @ H=H-1 @ IF H THEN 6
 7  FOR I=1 TO N @ FOR J=I+1 TO N @ IF S$(I)[K]>S$(J)[K] THEN VARSWAP S$(I),S$(J)
 8  NEXT J @ NEXT I @ FOR I=1 TO N @ DISP S$(I) @ NEXT I @ DISP N;TIME @ END
 9  Z=Z-1 @ IF NOT P(Z) THEN 9 ELSE FIX 11 @ X$=" "&STR$(X)&" " @ STD @ T$=""
10  FOR I=1 TO L+1 @ IF P(I) THEN J=2-(P(I)>0) @ T$=T$&"+-"[J,J]&"x^"&STR$(Z-I)
11  NEXT I @ S$(N)=REPLACE$(REPLACE$(X$&T$[2],"^1",""),"x^0","1") @ RETURN
```

(Lines 1-2 do the *initialization*, lines 3-6 do the *search*, lines 7-8 do the *sorting* and *output*, lines 9-11 *format* the output.)

It uses the **Math ROM** (*PROOT*) and the **JPCROM** for structured statements (*REPEAT*, *UNTIL*) and assorted utility functions (*VARSWAP*, *REPLACE$*). The JPCROM functions are mere conveniences and can be dispensed with but PROOT is mandatory or else the program would be much longer and slower.

My program, as written, is *generic* for monic -1,0,+1 polynomials of *any degree*, not just up to 8, and can display its output *sorted* by either constant or (for degree < 10) by minimal polynomial, as well as providing a count of the constants found and timing.

For example, for polynomials up to **degree 4**, sorting by constant:

```
>RUN

? 4
Sort by C,P ? C

1.32471795724 x^3-x-1
1.38027756910 x^4-x^3-1
1.46557123188 x^3-x^2-1
1.61803398875 x^2-x-1
1.83928675521 x^3-x^2-x-1
1.92756197548 x^4-x^3-x^2-x-1

6 .76
```

i.e., it found **6** constant in 0.76 seconds (**Emu71** on an old 2.4 GHz single-core).
The same run sorted by polynomial (same count and timing):

```
>RUN

? 4
```

```
Sort by C,P ? P

1.61803398875 x^2-x-1
1.32471795724 x^3-x-1
1.46557123188 x^3-x^2-1
1.83928675521 x^3-x^2-x-1
1.38027756910 x^4-x^3-1
1.92756197548 x^4-x^3-x^2-x-1
```

For polynomials up to **degree 6**, sorted by constant:

```
>RUN

? 6
Sort by C,P ? C

1.32471795724 x^3-x-1
1.38027756910 x^4-x^3-1
1.44326879127 x^5-x^4-x^3+x^2-1
1.46557123188 x^3-x^2-1
1.50159480354 x^6-x^5-x^4+x^2-1
1.53415774491 x^5-x^3-x^2-x-1
1.57014731220 x^5-x^4-x^2-1
1.61803398875 x^2-x-1
1.66040772247 x^6-x^5-x^3-x^2-1
1.70490277604 x^5-x^4-x^3-1
1.74370016590 x^6-x^5-x^4-x-1
1.77847961614 x^5-x^4-x^3-x^2+1
1.81240361927 x^5-x^4-x^3-x-1
1.83928675521 x^3-x^2-x-1
1.88851884548 x^5-x^4-x^3-x^2-1
1.91118343669 x^6-x^5-x^4-x^3-x-1
1.92756197548 x^4-x^3-x^2-x-1
1.96594823665 x^5-x^4-x^3-x^2-x-1
1.98358284342 x^6-x^5-x^4-x^3-x^2-x-1
```

i.e.: **19** constants found in 15.25 seconds.

Finally, for polynomials up to **degree 8** sorted by constant, as per the challenge requirements:

```
>RUN

? 8
Sort by C,P ? C

1.32471795724 x^3-x-1
1.38027756910 x^4-x^3-1
1.44326879127 x^5-x^4-x^3+x^2-1
1.46557123188 x^3-x^2-1
1.50159480354 x^6-x^5-x^4+x^2-1
1.53415774491 x^5-x^3-x^2-x-1
1.54521564973 x^7-x^6-x^5+x^2-1
1.57014731220 x^5-x^4-x^2-1
1.57367896839 x^8-x^7-x^6+x^2-1
1.59000537390 x^7-x^5-x^4-x^3-x^2-x-1
1.60134733379 x^7-x^6-x^4-x^2-1
1.61803398875 x^2-x-1
1.65363458677 x^7-x^6-x^5-1
1.66040772247 x^6-x^5-x^3-x^2-1
1.67752174784 x^7-x^6-x^5-x^2+1
1.70490277604 x^5-x^4-x^3-1
1.71428532914 x^6-x^5-x^4-x^2+1
1.72778030821 x^8-x^7-x^5-x^4-x^3-x^2-1
1.74370016590 x^6-x^5-x^4-x-1
1.74745742449 x^7-x^6-x^5-x^4+x^3-1
1.77452005864 x^7-x^6-x^5-x^3-1
1.77847961614 x^5-x^4-x^3-x^2+1
1.80750202302 x^6-x^5-x^4-x^3+1
1.81240361927 x^5-x^4-x^3-x-1
1.83524591514 x^8-x^7-x^6-x^4-x^3-x-1
1.83928675521 x^3-x^2-x-1
1.84771602509 x^8-x^7-x^6-x^5-1
1.85454747658 x^7-x^6-x^5-x^4-1
1.86221396917 x^8-x^7-x^6-x^5-x-1
1.88004410997 x^7-x^6-x^5-x^4-x-1
1.88670294847 x^8-x^7-x^6-x^5-x^2-x-1
1.88851884548 x^5-x^4-x^3-x^2-1
```

```
1.91118343669 x^6-x^5-x^4-x^3-x-1
1.92212800436 x^7-x^6-x^5-x^4-x^2-x-1
1.92756197548 x^4-x^3-x^2-x-1
1.96594823665 x^5-x^4-x^3-x^2-x-1
1.97061782036 x^8-x^7-x^6-x^5-x^4-x^3-1
1.97504243425 x^7-x^6-x^5-x^4-x^3-x^2-1
1.98358284342 x^6-x^5-x^4-x^3-x^2-x-1
1.99196419661 x^7-x^6-x^5-x^4-x^3-x^2-x-1
1.99603117974 x^8-x^7-x^6-x^5-x^4-x^3-x^2-x-1
```

i.e.: it found **41 constants** in 239.96 seconds, just shy of 4 min. For fun, a run for polynomials up to **degree 10** finds **85** constants in less than an hour.

That's all. Thanks again for your contributions, I hoped you enjoyed the challenge, see you all in **S&SMC#22**.

Best regards.
V.

PM FIND EDIT ✕ QUOTE REPORT

11-10-2016, 08:51 AM (This post was last modified: 11-10-2016 08:52 AM by Paul Dale.) **Post: #24**

Paul Dale
Senior Member

Posts: 1,452
Joined: Dec 2013

RE: Short & Sweet Math Challenge #21: Powers that be

Valentin, your explanation is much better than mine but at least I was on the right path.

An interesting challenege as always.

- Pauli

PM FIND QUOTE REPORT

11-10-2016, 07:47 PM (This post was last modified: 11-10-2016 07:55 PM by J-F Garnier.) **Post: #25**

J-F Garnier

Posts: 302

Senior Member Joined: Dec 2013

RE: Short & Sweet Math Challenge #21: Powers that be

> **Valentin Albillo Wrote:** ⇨ (11-10-2016 01:53 AM)
>
> ```
> >RUN
>
> ? 8
> Sort by C,P ? C
>
> 1.32471795724 x^3-x-1
> 1.38027756910 x^4-x^3-1
> ...
> 1.99196419661 x^7-x^6-x^5-x^4-x^3-x^2-x-1
> 1.99603117974 x^8-x^7-x^6-x^5-x^4-x^3-x^2-x-1
> ```
>
> i.e.: it found **41 constants** in 239.96 seconds, just shy of 4 min.

It was an interesting and fun challenge, and the opportunity for me to learn something in math.

I have two comments:

1) It seems that your program doesn't explore all the polynomials. It uses basically the same algorithm that I used in my previous thread (based on Paul's idea), and I got not **41** but **48** constants.
I had a look at your program, but had some difficulties to understand the logic of the polynomial scanning. However, I was able to hack your program to display all the tested polynomials, and indeed some are missing:

```
? 4
Sort by C,P ? p
1.00000000000 1
1.00000000000 x-1
1.00000000000 x^2+1
1.00000000000 x^2-1
1.00000000000 x^2-x+1
1.00000000000 x^2-x-1
1.00000000000 x^3+1
```

```
1.00000000000 x^3+x+1
1.00000000000 x^3+x-1
1.00000000000 x^3-1
1.00000000000 x^3-x+1
1.00000000000 x^3-x-1
1.00000000000 x^3-x^2+1
1.00000000000 x^3-x^2+x+1
1.00000000000 x^3-x^2+x-1
1.00000000000 x^3-x^2-1
1.00000000000 x^3-x^2-x+1
1.00000000000 x^3-x^2-x-1
1.00000000000 x^4+x-1
1.00000000000 x^4+x^2+x-1
1.00000000000 x^4+x^2-1
1.00000000000 x^4+x^2-x-1
1.00000000000 x^4-1
1.00000000000 x^4-x-1
1.00000000000 x^4-x^2+x-1
1.00000000000 x^4-x^2-1
1.00000000000 x^4-x^2-x-1
1.00000000000 x^4-x^3+x-1
1.00000000000 x^4-x^3+x^2+x-1
1.00000000000 x^4-x^3+x^2-1
1.00000000000 x^4-x^3+x^2-x-1
1.00000000000 x^4-x^3-1
1.00000000000 x^4-x^3-x-1
1.00000000000 x^4-x^3-x^2+x-1
1.00000000000 x^4-x^3-x^2-1
1.00000000000 x^4-x^3-x^2-x-1
36
```

For instance, the polynomials x^2+x+1 and x^2+x-1 are not tested, as well as the x^3+x^2... polynomials.

2) my second comment is what I already mentioned in my previous message: the value 1.75487766625, root of x^4-x^3-x^2-1, is not found by this algorithm, although its powers clearly tend to an integer (quite quickly actually):

```
x^10 276.992792634
x^11 486.088465506
...
x^21 134643.001528
```

```
x^22 236281.996298
...
x^27 3932464.99924
x^28 6900995.00047
```

The roots of the polynomial x^4-x^3-x^2-1 are:

```
(.122561166877,-.744861766662)
(.122561166877,.744861766662)
(-1,0)
(1.75487766625,0)
```

Clearly, the roots 1 or -1 should not cause to reject the polynomial under test, since powers of -1 or +1 will not impact the decimals of the powers.

Thanks again for this challenge!

J-F

EMAIL PM WWW FIND QUOTE REPORT

11-11-2016, 01:13 AM **Post: #26**

Valentin Albillo
Senior Member

Posts: 344
Joined: Feb 2015
Warning Level: 0%

RE: Short & Sweet Math Challenge #21: Powers that be

Hi, **Mike**:

> **Mike (Stgt) Wrote:** ⇨ (11-10-2016 11:51 AM)
>
> So the reason for the almost integer effect of increasing powers is just the "symetry" of roots r1 and r2 with abs(r1)<1 and abs(r2)>1 ?

The reason for the *almost-integer i*ncreasing powers is, in short:

- all *symmetric functions* of the roots of any polynomial can be expressed in terms of the coefficients of the polynomial, where *symmetric* means that the function remains the same and has the *same* value for *every* permutation of the roots. For instance, the sum of the Nth-powers of the roots of a quintic polynomial is symmetric because:

```
          S = x1^5 + x2^5 + x3^5 + x4^5 + x5^5    (initial permutation)
            = x1^5 + x2^5 + x3^5 + x5^5 + x4^5    (second permutation)
                          ...
            = x5^5 + x4^5 + x3^5 + x2^5 + x1^5    (last permutation)
```

so you see, no matter how you permute the roots the sum stays the same.

- if the polynomial is *monic* (leading coefficient is 1) and has *integer* coefficients, then any symmetric function of the roots is *integer* as well because it can be computed from the integer coefficients using just addition, subtraction and multiplication of their integer values. Thus, in particular, *the sum of the roots raised to any integer power is mandatorily integer* as well, regardless of the absolute values of the roots or whether they're real and/or complex.

-now, if any number is >1 in absolute value, its increasing powers will grow ever *bigger*, while if the number is <1 in absolute value its increasing powers will grow ever *smaller*, tending to 0.

- so, for those monic polynomials which have just the **one** root >1 in absolute value while all other roots are <1 in absolute value, the sum of the powers of the roots will essentially equal the power of the large root alone plus change, and as the sum has to mandatorily be integer, then the power of the larger root tends to be integer, *almost-integer* so to say. The only difference with an integer is the sum of the powers of all other roots, whichs tends to 0 for each root and so for the sum of all of them.

I hope this explanation makes it clear for you, thanks a lot for your interest.

Reegards.
V.

PM FIND EDIT QUOTE REPORT

11-11-2016, 01:15 AM **Post: #27**

Valentin Albillo
Senior Member

Posts: 344
Joined: Feb 2015
Warning Level: 0%

RE: Short & Sweet Math Challenge #21: Powers that be

Hi, **Pauli**:

Paul Dale Wrote: (11-10-2016 08:51 AM)

> Valentin, your explanation is much better than mine but at least I was on the right path.

Absolutely, you nailed it.

> **Quote:**
>
> An interesting challenege as always.

Thank you very much for your kind comment and interest, hope to see you back in S&SMC#22 due next month or so.

Regards.
V.

PM FIND EDIT ✕ QUOTE + REPORT

11-11-2016, 02:10 AM **Post: #28**

Valentin Albillo
Senior Member

Posts: 344
Joined: Feb 2015
Warning Level: 0%

RE: Short & Sweet Math Challenge #21: Powers that be

Hi, **J-F**:

> **J-F Garnier Wrote:** ⇒ (11-10-2016 07:47 PM)
>
> I have two comments:
>
> 1) It seems that your program doesn't explore all the polynomials. [...] However, I was able to hack your program to display all the tested polynomials, and indeed some are missing: [...] For instance, the polynomials x^2+x+1 and x^2+x-1 are not tested, as well as the x^3+x^2... polynomials.

I made use of mathematical equivalences and polynomial theory to prune the number of polynomials to be tested to the barest minimum necessary. For instance, your polynomial:

```
    x^2+x+1
```

doesn't have any real roots, just two conjugate complex roots. For polynomials with real coefficients, all complex roots come out in conjugate pairs which of course have the same absolute value and thus can't be *unique*, i.e., even if their absolute value were >1 there are *two* of them, not just *one* as required.

Your other polynomial, namely:

```
x^2+x-1
```

is equivalent to **x^2-x-1** if you simply change the variable from **x** to **-x**, which of course doesn't alter its absolute value at all, so you only need to generate and check **x^2-x-1**, which I did. In short, I choose the coefficients in such a way that those redundant polynomials aren't generated while searching.

> **Quote:**
>
> 2) my second comment is what I already mentioned in my previous message: the value 1.75487766625, root of x^4-x^3-x^2-1, is not found by this algorithm, although its powers clearly tend to an integer (quite quickly actually)

Your number **1.75487766625** is indeed a bona-fide **Pisot number** (which is the proper mathematical name for the numbers which have this *almost-integer* powers property) as can be readily checked like you did.

However, although it's indeed a root of the **4th-degree** polynomial you mention, namely:

```
x^4-x^3-x^2-1
```

this is *not* its *minimal polynomia*l. Your number is a root of infinite polynomials with integer coefficients but *only one of them has the minimum degree possible*, and in this case it is the **3rd-degree** polynomial:

x^3-2*x^2+x-1

which neither your second program nor my original solution generates or checks because it is *not* a **-1,0,+1** polynomial as per the challenge requirements because it does have a **-2** coefficient. That's why neither your program nor mine should have found it and *that's correct*, the -2 coefficient places it squarely out of the specified requirements.

> **Quote:**
>
> Thanks again for this challenge!

You're welcome, I'm glad you enjoyed it and thank you very much for your interest, your kind comments and, above all, the time you took to deal with it, much appreciated. Hope to see you back in **S&SMC#22** due next month or so.

Best regards.
V.

PM FIND EDIT QUOTE REPORT

« Next Oldest | Next Newest » Enter Keywords Search Thread

Pages (2): « Previous 1 **2** NEW REPLY

View a Printable Version

Send this Thread to a Friend

Subscribe to this thread

User(s) browsing this thread: Valentin Albillo*

Contact Us | The Museum of HP Calculators | Return to Top | Return to Content | Lite (Archive) Mode | RSS Syndication English (American) ▼ Go