# Сходимость последовательности производных на множестве натуральных чисел

Георгий Гуляев

7 декабря 2022 г.

## 1. Введение

Математики любят аналогии. В далеком 1961 году была опубликована статья [1], в которой автор попытался ввести операцию подобную взятию прозводной для целых чисел $n \geq 0$.

Другие последователи распространили это определение на рациональные и даже на иррациональные числа, изучили интегралы и доказали несколько интересных теорем [2-3].

В данной статье мы изложим часть их результатов и попытаемся применить компьютер для анализа сходимости последовательности производных натуральных чисел.

## 2. Определение

Согласно литературе [1-3]:

$1' = 0$.

$p' = 1$ для любого простого $p$.

$(a \cdot b)' = a' \cdot b + a \cdot b'$ для любых натуральных a и b (правило Лейбница).

## 3. Примеры

$4' = (2 \cdot 2)' = 2' \cdot 2 + 2 \cdot 2' = 2 + 2 = 4$

$8' = (2 \cdot 4)' = 2' \cdot 4 + 2 \cdot 4' = 4 + 8 = 12$

$10' = (2 \cdot 5)' = 2' \cdot 5 + 2 \cdot 5' = 5 + 2 = 7$

$12' = (3 \cdot 4)' = 3' \cdot 4 + 3 \cdot 4' = 4 + 12 = 16$

$20' = (2 \cdot 10)' = 2' \cdot 10 + 2 \cdot 10' = 10 + 14 = 24$

Здесь, в общем случае, не выполняется важное свойство обычной производной: $(a+b)' = a' + b'$, например

$(10+2)' = 16$, но $10' + 2' = 7 + 1 = 8$, хотя бывают случаи, когда это верно: $(4+8)' = 4' + 8' = 16$

Формула же для производной степени сохраняется, так как является следствием правила Лейбница.

Для простых p:

$(p^2)' = (p \cdot p)' = p' \cdot p + p \cdot p' = 2p$

$(p^3)' = (p \cdot p^2)' = p' \cdot p^2 + p \cdot (p^2)' = p^2 + 2p^2 = 3p^2$

...

Общий случай натурального a:

$(a^2)' = (a \cdot a)' = a' \cdot a + a \cdot a' = 2aa'$

$(a^3)' = (a \cdot a^2)' = a' \cdot a^2 + a \cdot (a^2)' = a^2a' + 2a^2a' = 3a^2a'$

...

## 4. Теоремы

**Теорема 1**. Для любого натурального a справедлива формула:

$(a^n)' = na^{n-1}a'$

**Теорема 2**. Для любого натурального $a$ равенство $a = a'$ выполняется тогда и только тогда, когда $a = p^p$, где p простое.

Таким образом, роль $e^x$ здесь играют числа $p^p$.

**Теорема 3**. Если известно каноническое разложение числа n $>$ 1 на простые множители: $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_k^{\alpha_k}$, то $n' = \alpha_1 \cdot \frac{n}{p_1} + \alpha_2 \cdot \frac{n}{p_2} + \ldots + \alpha_k \cdot \frac{n}{p_k}$

Например, $60' = (2^2 \cdot 3 \cdot 5)' = 2 \cdot (2 \cdot 3 \cdot 5) + (2^2 \cdot 5) + (2^2 \cdot 3) = 60 + 20 + 12 = 92$

**Следствие 1**. Наибольший общий делитель $(n, n') = 1$ тогда и только тогда, когда $n$ не имеет в каноническом разложении степеней $\alpha_i > 1$.

**Теорема 4**. Если $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_k^{\alpha_k}$ и $\alpha_i$ не делится на $p_i$ для всех $i \in [1, k]$, то $(n, n') = \frac{n}{rad(n)}$, где $rad(n) = p_1 \cdot p_2 \cdot \ldots \cdot p_k$ - радикал числа $n$.

## 5. Примеры на компьютере

Вычисление производной n по теореме 3 на языке Julia:

```
function drv(n)
  n >= 0       || return -drv(-n)
  n > 1        || return 0
  s = 0
  for (p,k) in factor(n)
    s+=k*div(n,p)
  end
  s
end
```

Здесь допускается n целое для того чтобы не делать исключений. Предполагается $0' = 0$, а для отрицательных чисел используется формула $(-n)' = -n'$.

Согласно теореме 2, $n' = n$ бывает только тогда, когда $n = p^p$, $p$ - простое. Из примеров мы видели, что в других случаях может быть как $n' > n$, так и $n' < n$.

Попробуем изучить вопрос: всегда ли будет последовательность производных $n', n'', n''', \ldots$ стремиться к бесконечности или же нет?

Для этого напишем функцию вычисляющую 50 производных.

```
function go(n)
  k = 0
  d = big(drv(n))
  while k<50&&d>0
    println(d)
    d = drv(d)
    k+=1
  end
end
```

Как можно увидеть из нижеследующих примеров, есть последовательности сходящиеся к 1, но есть и, предположительно, расходящиеся. Мы берем только 50 значений и, естественно, пока не можем доказать точно, что та или иная последовательность расходится.

```
go(5*977) = 982 493 46 25 10 7 1
go(13*977) = 990 1443 631 1
go(7*7*13) = 231 131 1
go(7*11*13) = 311 1
go(5*7*11) = 167 1
go(3*5*7) = 71 1
go(2*3*3) = 21 10 7 1
go(5*409) = 414 501 170 129 46 25 10 7 1
go(2*3*5*7*11) = 2927 1
go(2*3*5*7*11*13) = 40361 1

julia> go(2*3*5*7*11*13*17*19*23)
334406399
9835475
4893565
978718
564671
1
```

```
julia> go(3*5*7*11*13*17*19*23*29)
3343015913
107839254
89866051
8341703
534455
111431
1

go(7*11*13*17*19*23*29) = 107850959 1

julia> go(13*17*19*23*29)
745945
154019
9951
3731
911
1

go(19*23*29*31*37) = 2760049 1

go(7*11*13*17*19*23*29) = 107850959 1

julia> go(2*3*5*7*11*13*17*19*23*29*31*37)
11819186711467
149759057759
1

julia> go(3*5*7*11*13*17*19*23*29)
3343015913
107839254
89866051
8341703
534455
111431
1

go(19*23*29*31*37) = 2760049 1
```

```
julia> go(13*17*19*23*29)
745945
154019
9951
3731
911
1
```

```
julia> go(2*3*5*7*11*13*17*19*23*29)
9920878441
36880858
23794635
12690487
654407
60479
504
1164
1564
1724
1728
6912
34560
179712
1002240
5246208
28684800
169585920
906315264
5002566912
25727535360
133783218432
760954279680
4348176926976
21740884641792
...
```

```
julia> go(big(2)*3*5*7*11*13*17*19*23*29*31*37*41*43*47*53*59*61)
201015517717077830328949
1
```

## 6. Анализ результатов

Первое ощущение такое, что все случайно и может быть все что угодно. Как только натыкаемся на простое, а это довольно случайное событие, так получаем сходимость к 1.

Интересно, что многие числа и в том числе некоторые простые не могут быть производными. Из 1229 простых до 10000, 700 не являются производными какого-либо числа, остальные 529 - являются:

$5, 7, 13, 19, 31, 41, 43, 59, 61, 71, 73, 101, 103, 109, 113, 131, 139, ...$ - являются

$2, 3, 11, 17, 23, 29, 37, 47, 53, 67, 79, 83, 89, 97, 107, 127, 137, ...$ - нет

То есть вероятность наткнуться на простое здесь уменьшается еще более чем вдвое. То что выше я привел много примеров сходящихся последовательностей производных не означает, что расходящихся меньше. Например, все числа вида $4k$ приводят к расходящейся последовательности.

Доказать это несложно: $(4k)' = 4(k + k')$, то есть производная $4k$ не меняет вид числа, при этом при $k = 1$ все производные будут равны 4 (зацикливание), а при $k > 1$ последовательность $n', n'', n'''...$ будет неограниченно расти.

Нетрудно заметить, что здесь $4 = 2^2$ является частным случаем общего $p^p$, ибо здесь все то же самое $(p^p k)' = p^p(k + k')$. Поэтому имеет смысл рассматривать только числа $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot ... \cdot p_k^{\alpha_k}$ для которых $\alpha_i < p_i, i = 1, 2, ...k$.

В приведенных выше примерах я везде брал $\alpha_i = 1$. Однако и в этом случае попадается много расходящихся. Возьмем простое число 7. Умножая его на разные числа k и проверяя результат на сходимость, увидим, что при при $k = 3, 6, 7, 11, 15, 18, 23, ...$ - последовательность сходится, а при $k = 5, 9, 13, 17, 19, 21, 25, ...$ - расходится.

Заметим, что у нас появился метод для строгого доказательства, что последовательность расходится. А именно, она расходится когда существует $i$ для которого $\alpha_i \geq p_i$. С учетом этого напишем еще одну функцию, которая будет возвращать 1, если последовательность сходится, -1, если последовательность расходится и 0, если определить не удалось.

```
function explore(n)
  drv(n,f) = sum(map(x -> x[2]*div(n,x[1]), f))
  l = [n]; up = 0
  while up<20
    m = l[end]
    if isprime(m) return 1 end
    f = factor(m).pe
    if length(filter(x -> x[1]<=x[2],f)) > 0 return -1 end
    push!(l,drv(m,f))
    if m < l[end] up+=1 end
  end
  0
end
```

Выход из программы по количеству возрастаний последовательности (переменная up). Если их больше 20 и определить сходимость не удалось, то завершаем программу и возвращаем 0.

```
julia> filter(x -> x[2] == 0,map(x -> (x,explore(x)),2:10000000))
Tuple{Int64, Int64}[]

julia> filter(x -> x[2] == 1,map(x -> (x,explore(x)),2:10000000))
2589482-element Vector{Tuple{Int64, Int64}}:

julia> filter(x -> x[2] == -1,map(x -> (x,explore(x)),2:10000000))
7410517-element Vector{Tuple{Int64, Int64}}:
```

Оказалось, что до n = 10000000, существует 2589482 сходящихся последовательностей, начинающихся с n, а все остальные расходящиеся. 0 в качестве возвращаемого значения не встретилось ни разу. То есть случай $\alpha_i \geq p_i$ всегда встречается раньше 20-ти возрастаний.

Это хорошо видно на примере go(2*3*5*7*11*13*17*19*23*29), приведенном в параграфе 5. Как только последовательность дошла до $504 = 2^3 * 3^2 * 7$ стало ясно, что она расходится, ибо для степени $2^3, 3 \geq 2$.

Этот пример еще интересен тем, что сначала производные уменьшаются, а потом начинают расти. Теперь при помощи небольшой модификации функции explore нетрудно найти примеры сходящихся и расходящихся последовательностей с колебаниями возрастания и убывания.

```
julia> go(40250)
51775
23910
24737
882
1281
631
1

julia> go(45150)
63185
12642
14441
2070
2919
1411
100
140
188
192
640
2368
7168
36864
245760
1851392
12976128
120127488
1012858880
...
```

## 7. Выводы

Мы попытались при помощи программирования на языке Julia исследовать вопросы сходимости последовательности производных $n', n'', n''', \ldots$ от натуральных чисел.

Выяснили, что последовательность начинает расходиться как только в ней встречается число $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_k^{\alpha_k}$ для которого существует $i$ такое, что $\alpha_i \geq p_i$.

При исследовании на компьютере в пределах его вычислительных возможностей (проверено до $10^9$) все последовательности либо быстро сходились, либо в них встречалось число $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_k^{\alpha_k}$ с $\alpha_i \geq p_i$. Неопределенных последовательностей не обнаружено.

## Ссылки


[1] E. J. Barbeau, Remarks on arithmetic derivative, Canad. Math. Bull, 4 (1961), 117–122.

[2] V. Ufnarovski and B. Åhlander, How to differentiate a number, J. Integer Seq., 6 (2003)

[3] J. Kovič, The Arithmetic Derivative and Antiderivative, J. Integer Seq, Vol. 15 (2012)