

07 漸近展開の有名な例

黒木玄

2018-05-20~2019-04-03

- Copyright 2018 Gen Kuroki
- License: MIT <https://opensource.org/licenses/MIT> (<https://opensource.org/licenses/MIT>)
- Repository: <https://github.com/genkuroki/Calculus> (<https://github.com/genkuroki/Calculus>)

このファイルは次の場所できれいに閲覧できる:

- <http://nbviewer.jupyter.org/github/genkuroki/Calculus/blob/master/07%20example%20of%20asymptotic%20expansion.ipynb>
(<http://nbviewer.jupyter.org/github/genkuroki/Calculus/blob/master/07%20example%20of%20asymptotic%20expansion.ipynb>)
- <https://genkuroki.github.io/documents/Calculus/07%20example%20of%20asymptotic%20expansion.pdf>
(<https://genkuroki.github.io/documents/Calculus/07%20example%20of%20asymptotic%20expansion.pdf>)

このファイルは [Julia Box](https://juliabox.com) (<https://juliabox.com>) で利用できる.

自分のパソコンに [Julia言語](https://julialang.org/) (<https://julialang.org/>) をインストールしたい場合には

- [WindowsへのJulia言語のインストール](http://nbviewer.jupyter.org/gist/genkuroki/81de23edcae631a995e19a2ecf946a4f) (<http://nbviewer.jupyter.org/gist/genkuroki/81de23edcae631a995e19a2ecf946a4f>)
- [Julia v1.1.0 の Windows 8.1 へのインストール](https://nbviewer.jupyter.org/github/genkuroki/msfd28/blob/master/install.ipynb) (<https://nbviewer.jupyter.org/github/genkuroki/msfd28/blob/master/install.ipynb>)

を参照せよ. 前者は古く, 後者の方が新しい.

論理的に完璧な説明をするつもりはない. 細部のいい加減な部分は自分で訂正・修正せよ.

目次

- ▼ 1 漸近展開の有名な例
 - 1.1 $F_n(x)$ の定義
 - 1.2 $F_n(x) = O(x^{n+1})$
 - 1.3 $F_0(x)$ の漸近展開
 - 1.4 $F_0(x)$ の漸近展開の別の導出の仕方
 - 1.5 以上のまとめ
 - 1.6 $F_0(1/10)$ の漸近展開を用いた数値計算
 - 1.7 漸近展開の入門書
 - 1.8 指数積分函数
 - 1.9 連分数展開による数値計算

In [1]:

```

1  using Base.MathConstants
2  using Base64
3  using Printf
4  using Statistics
5  const e = e
6  endof(a) = lastindex(a)
7  linspace(start, stop, length) = range(start, stop, length=length)
8
9  using Plots
10 gr(); ENV["PLOTS_TEST"] = "true"
11 #clibrary(:colorcet)
12 clibrary(:misc)
13
14 function pngplot(P...; kwargs...)
15     sleep(0.1)
16     pngfile = tempname() * ".png"
17     savefig(plot(P...; kwargs...), pngfile)
18     showing("image/png", pngfile)
19 end
20 pngplot(; kwargs...) = pngplot(plot!(); kwargs...)
21
22 showing(mime, fn) = open(fn) do f
23     base64 = base64encode(f)
24     display("text/html", """""")
25 end
26
27 using SymPy
28 #sympy.init_printing(order="lex") # default
29 #sympy.init_printing(order="rev-lex")
30
31 using SpecialFunctions
32 using QuadGK

```

1 漸近展開の有名な例

1.1 $F_n(x)$ の定義

n は非負の整数であるとする. $x > 0$ の関数 $F_n(x)$ を

$$F_n(x) = n! e^{1/x} \int_{1/x}^{\infty} e^{-t} t^{-n-1} dt$$

と定める. $F_n(x)$ の $x > 0$ が 0 に近いときの様子を調べたい.

1.2 $F_n(x) = O(x^{n+1})$

$x > 0$ のとき $t \geq 1/x$ ならば $t^{-n-1} \leq x^{n+1}$ なので

$$0 < F_n(x) \leq n! e^{1/x} \int_{1/x}^{\infty} e^{-t} x^{n+1} dt = n! x^{n+1}.$$

ゆえに,

$$F_n(x) = O(x^{n+1}) \quad (x \searrow 0).$$

1.3 $F_0(x)$ の漸近展開

$(-e^{-t})' = e^{-t}$ を用いた部分積分によって

$$\begin{aligned} F_n(x) &= n! e^{1/x} \left([-e^{-t} t^{-n-1}]_{1/x}^{\infty} + \int_{1/x}^{\infty} e^{-t} (-(n+1)t^{-n-2}) dt \right) \\ &= n! x^{n+1} - F_{n+1}(x). \end{aligned}$$

ゆえに $x > 0$ において,

$$\begin{aligned}
F_0(x) &= 0!x - F_1(x) = 0!x - 1!x^2 + F_2(x) = \dots \\
&= 0!x - 1!x^2 + \dots + (-1)^n n!x^{n+1} + (-1)^{n+1} F_{n+1}(x) \\
&= \sum_{k=0}^n (-1)^k k! x^{k+1} + (-1)^{n+1} F_{n+1}(x).
\end{aligned}$$

1.4 $F_0(x)$ の漸近展開の別の導出の仕方

$x > 0$ であるとする. $F_0(x)$ は $t = 1/x + u$ という置換によって次のように書き直される:

$$F_0(x) = e^{1/x} \int_{1/x}^{\infty} \frac{e^{-t}}{t} dt = x \int_0^{\infty} \frac{e^{-u}}{1+xu} du.$$

ゆえに

$$\frac{1}{1+z} = \sum_{k=0}^{n-1} (-1)^k z^k + (-1)^n \frac{z^n}{1+z}$$

を $z = xu$ に適用した結果を使うと,

$$\begin{aligned}
F_0(x) &= \sum_{k=0}^{n-1} (-1)^k x^{k+1} \int_0^{\infty} e^{-u} u^k du + (-1)^n x^{n+1} \int_0^{\infty} \frac{e^{-u} u^n}{1+xu} du \\
&= \sum_{k=0}^{n-1} (-1)^k k! x^{k+1} + (-1)^n x^{n+1} \int_0^{\infty} \frac{e^{-u} u^n}{1+xu} du.
\end{aligned}$$

さらに

$$0 < x^{n+1} \int_0^{\infty} \frac{e^{-u} u^n}{1+xu} du \leq x^{n+1} \int_0^{\infty} e^{-u} u^n du = n! x^{n+1}$$

なので

$$x^{n+1} \int_0^{\infty} \frac{e^{-u} u^n}{1+xu} du = O(x^{n+1}) \quad (x \searrow 0).$$

1.5 以上のまとめ

以上をまとめると

$$F_0(x) = \sum_{k=0}^{n-1} (-1)^k k! x^{k+1} + (-1)^n F_n(x) = \sum_{k=0}^{n-1} (-1)^k k! x^{k+1} + O(x^{n+1}) \quad (x \searrow 0).$$

ここで

$$F_n(x) = n! e^{1/x} \int_{1/x}^{\infty} e^{-t} t^{-n-1} dt = x^{n+1} \int_0^{\infty} \frac{e^{-u} u^n}{1+xu} du = O(x^{n+1}) \quad (x \searrow 0).$$

しかし, $x > 0$ で

$$\sum_{k=0}^{\infty} (-1)^k k! x^{k+1} = 0!x - 1!x^2 + 2!x^3 - 3!x^4 + \dots$$

は決して収束しない.

このように全てを足し上げると発散する場合であっても, 有限項と剰余項の和の形式で表わせれば, すべてが well-defined な量だけを使って議論を進めることができる場合がある. 上の場合には剰余項は $(-1)^n F_n(x)$ である.

1.6 $F_0(1/10)$ の漸近展開を用いた数値計算

$x > 0$ のとき $0 < F_n(x) \leq n! x^{n+1}$ となるのであった. $x = 1/10$ のとき $n! x^{n+1}$ が最小になる n は $n = 9, 10$ であることを確認できる. そのとき $n! x^{n+1} = 3.6288 \times 10^{-5}$ である. したがって, $n = 9, 10$ のときの $\sum_{k=0}^{n-1} (-1)^k k! x^{k+1}$ による $F_0(x)$ の近似は最良になると予想される.

In [2]: 1 [(n, factorial(n)*(1/10)^(n+1)) for n in 7:13]

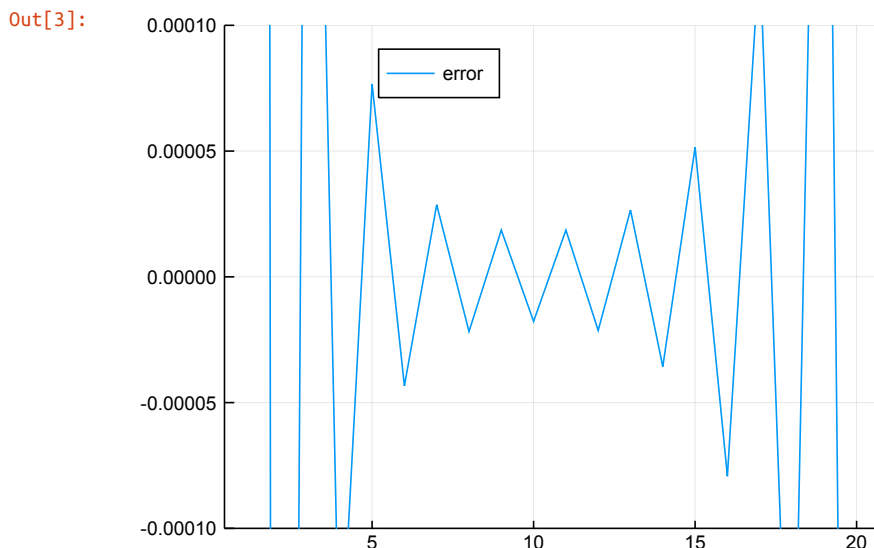
Out[2]: 7-element Array{Tuple{Int64,Float64},1}:
 (7, 5.0400000000000002e-5)
 (8, 4.0320000000000002e-5)
 (9, 3.6288000000000002e-5)
 (10, 3.6288000000000002e-5)
 (11, 3.99168000000000024e-5)
 (12, 4.7900160000000004e-5)
 (13, 6.2270208000000005e-5)

In [3]: 1 F0(x) = exp(1/x)*quadgk(u->exp(-1/(x*u))/u, 0, 1)[1]
 2 F0_ae(x, n=10) = sum(k->(-1)^k*factorial(k)*x^(k+1), 0:n-1)
 3
 4 @show Y = F0(1/10)
 5 @show Y_ae9 = F0_ae(1/10, 9)
 6 @show Y_ae10 = F0_ae(1/10, 10)
 7
 8 n = 1:20
 9 plot(size=(500,350))
 10 plot!(ylims=(-0.0001,0.0001))
 11 plot!(legend=:top)
 12 plot!(n, F0_ae.(1/10, n) .- Y, label="error")

Y = F0(1 / 10) = 0.09156333393978826

Y_ae9 = F0_ae(1 / 10, 9) = 0.09158192

Y_ae10 = F0_ae(1 / 10, 10) = 0.091545632



1.7 漸近展開の入門書

このように $x > 0$ で決して収束しないべき級数であっても、適切な解釈のもとで数学的に十分な意味を持つことがある。このような理由で数学では発散級数に関するたくさんの深い研究がある。

その方面の入門書としては次の文献がある。非常に面白い本なのでおすすめできる。

- 大久保謙二郎・河野實彦共著『[漸近展開](https://www.amazon.co.jp/dp/4316376306)』, 新しい応用の数学12, 教育出版, 1976, 1996 (<https://www.amazon.co.jp/dp/4316376306>).

1.8 指数積分関数

$F_0(x)$ は実用的にも重要な関数である。それと本質的に同じ

$$E_1(z) = e^{-z} F_0(1/z) = \int_z^\infty \frac{e^{-t}}{t} dt = \int_0^1 \frac{e^{-z/u}}{u} du$$

は指数積分関数と呼ばれる特殊関数の1つである($t = z/u$)。この関数の数値計算については

- <http://nbviewer.jupyter.org/github/stevengj/18S096/blob/iap2017/pset3/pset3-solutions.ipynb>
 (<http://nbviewer.jupyter.org/github/stevengj/18S096/blob/iap2017/pset3/pset3-solutions.ipynb>)

が非常に面白い解説になっている. 特殊関数の数値計算の最適化に興味がある人は是非とも閲覧して欲しい.

1.9 連分数展開による数値計算

以下は $x = 1/z$ のときの

$$G(z) = F_0(1/z) = e^z E_1(z) = e^z \int_0^1 \frac{e^{-zu}}{u} du$$

の連分数展開による数値計算である. この関数の連分数展開については

- 一松信著 『特殊関数入門』数学選書, 森北出版, 1999 (<https://www.amazon.co.jp/dp/4627038216>)

の第3章の最後のp.74にある例3.10および第6章のpp.132-133にある例6.4を参照せよ. この本も非常に面白い本なのでおすすめできる.

```
In [4]: 1 ▼ function G_cf(z; n::Int=2)
2         cf = 1 + (n+1)/z
3 ▼       for i = n:-1:1
4           cf = z + (1+i)/cf
5           cf = 1 + i/cf
6       end
7       return 1 / (z + 1/cf)
8     end
9
10  G(z) = exp(z)*quadgk(u->exp(-z/u)/u, 0, 1)[1]
```

Out[4]: G (generic function with 1 method)

```
In [5]: 1  sympy.init_printing(order="lex")
2  z = symbols("z")
3 ▼ for n in 1:3
4     cf = G_cf(z, n=n)
5     display(cf)
6  end
```

$$z + \frac{1}{1 + \frac{1}{z + \frac{2}{1 + \frac{2}{z}}}}$$

$$z + \frac{1}{1 + \frac{1}{z + \frac{2}{1 + \frac{2}{z + \frac{3}{1 + \frac{3}{z}}}}}}$$

$$z + \frac{1}{1 + \frac{1}{z + \frac{2}{1 + \frac{2}{z + \frac{3}{1 + \frac{3}{z + \frac{4}{1 + \frac{4}{z}}}}}}}}}$$

```
In [6]: 1  sympy.init_printing(order="rev-lex")
2  display(series(sum((-1)^n*factorial(n)*z^(n+1) for n in 0:9), n=11))
3  for n in 1:3
4  display(series(G_cf(1/z, n=n), n=11))
5  end
```

$$-362880z^{10} + 40320z^9 - 5040z^8 + 720z^7 - 120z^6 + 24z^5 - 6z^4 + 2z^3 - z^2 + z$$

$$z - z^2 + 2z^3 - 6z^4 + 24z^5 - 108z^6 + 504z^7 - 2376z^8 + 11232z^9 - 53136z^{10} + O(z^{11})$$

$$z - z^2 + 2z^3 - 6z^4 + 24z^5 - 120z^6 + 720z^7 - 4896z^8 + 35712z^9 - 269568z^{10} + O(z^{11})$$

$$z - z^2 + 2z^3 - 6z^4 + 24z^5 - 120z^6 + 720z^7 - 5040z^8 + 40320z^9 - 360000z^{10} + O(z^{11})$$

n = 1, 2, 3 の連分数近似であっても、大きめの z における近似が相当にうまく行っていることを以下で確認する。

```
In [7]: 1  G_cf(4), G(4)
```

Out[7]: (0.20636792452830188, 0.20634564990105575)

```
In [8]: 1  x = 0.05:0.05:4
2  @time y = G.(x)
3  @time y_cf1 = G_cf.(x, n=1)
4  @time y_cf2 = G_cf.(x, n=2)
5  @time y_cf3 = G_cf.(x, n=3)
6  plot(size=(500,350))
7  plot!(title="y = G(x) = e^x E_1(x)", titlefontsize=11)
8  plot!(xlims=(0,maximum(x)), ylims=(0,2.3))
9  plot!(x, y, label="numetical integration")
10 plot!(x, y_cf1, label="continued fractional approximation n=1", ls=:dash)
11 plot!(x, y_cf2, label="continued fractional approximation n=2", ls=:dash)
12 plot!(x, y_cf3, label="continued fractional approximation n=3", ls=:dash)
```

0.544783 seconds (1.15 M allocations: 65.165 MiB, 7.58% gc time)
 0.084327 seconds (259.28 k allocations: 12.983 MiB)
 0.000033 seconds (179 allocations: 4.000 KiB)
 0.000029 seconds (179 allocations: 4.000 KiB)

