Cancellation in the evaluation of Ai(x) and how to deal with it

Marc Mezzarobba

AriC "Tuesday work session", January 29, 2013

Introduction

- Numerical instability issues that occur when you try to compute the sums of the power series expansions of some functions using finite-precision arithmetic.
- Methods to deal with them—some well-known, other less well-known or new.
- Ref (1st part):
  - Gawronski, W.; Müller, J. & Reinhard, M., SIAM J. Num. An., 2007
  - Reinhard, M., Phd thesis, Universität Trier, 2008
  - Chevillard & Mezzarobba, ARITH 2013.

1 Cancellation

- $e^{-x} = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} x^n$, "fairly large" $x > 0$, say $x = 20$
- $\left| \sum_{n=N}^{\infty} \frac{(-1)^n}{n!} x^n \right| \leqslant \frac{x^N}{N!} \sum_{n=0}^{\infty} \frac{x^n}{n!} \leqslant \frac{x^N}{N!} e^x$
- aim at relative accuracy $\varepsilon = 10^{-10}$, i.e. $|e^{-x} - a| \leqslant 10^{-10} e^{-x}$
- sufficient condition: $\frac{x^N}{N!} e^x \leqslant 10^{-10} e^{-x}$, i.e., $\frac{N!}{x^N} \geqslant e^{2x} 10^{10}$, so $N = 100$ should be okay

```
> x := 20; N := 100;
```

$x := 20$
$N := 100$

```
> evalf(N!/x^N), evalf(exp(2*x)*10^10);
```

$.7362140280e28, .2353852668e28$

- but the result we get does not have a single correct significant digit (and this is really a working precision issue)

```
> Digits:=10:
> add((-20.)^n/n!, n=0..99);
```

$-.12115250e-1$

```
> exp(-20.);
```

$.2061153622e-8$

```
> Digits := 50;
```

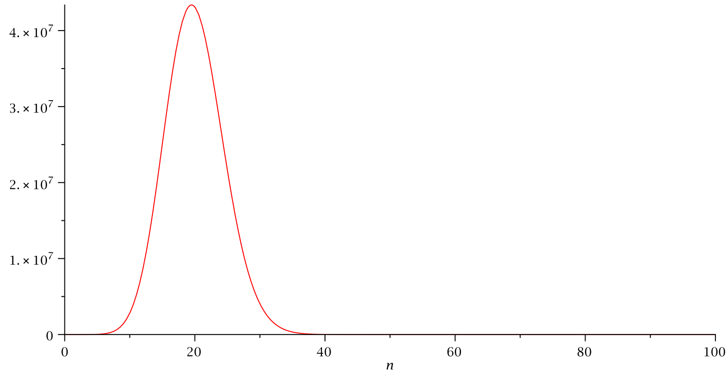
$\text{Digits} := 50$

```
> add((-20.)^n/n!, n=0..99);
```

$.20611536224385578278525921054805014704619e - 8$

- What happened? let's plot the *absolute values* of the coefficients.

```
> plot(20^n/n!, n=0..100);
```




```
> Digits := 19:
> add((-20.)^n/n!, n=0..99);
```

$.2063834819e - 8$

We are subtracting numbers $\gtrsim 5 \cdot 10^7$ to get a result $\approx 10^{-8}$, with only 10 digits of precision. The meaningful contribution of the terms for $0 \leqslant n \lesssim 40$ gets lost in roundoff errors.

- Digits "lost by cancellation" $\approx \log_{10}\left(\max_n |y_n x^n|\right) - \log_{10}|y(z)|$
- Better way (of course!)

```
> Digits := 10;
```

Digits: =10

```
> 1/add((20.)^n/n!, n=0..99);
```

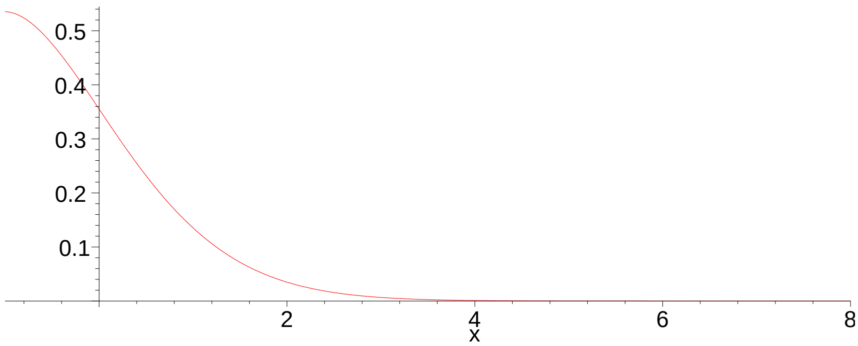
$.2061153623e - 8$

- Rest of the talk: methods analogous to this division (to some extent) for more complicated functions. Specifically $\mathrm{Ai}(x)$.

2 The Airy Function $\mathrm{Ai}(x)$

- Running example:

```
> plot(AiryAi(x), x=-1..8);
```



```
> AiryAi(0.), D(AiryAi)(0.);
```

$.3550280539, -.2588194038$

- $\mathrm{Ai}''(x) = x\,\mathrm{Ai}(x)$, $\mathrm{Ai}(0) = \dfrac{1}{3^{2/3}\,\Gamma(2/3)} \approx$, $\mathrm{Ai}'(0) = -\dfrac{1}{3^{1/3}\,\Gamma(1/3)}$
- $\mathrm{Ai}(x) = \sum_{n=0}^{\infty} u_n\, x^n$
- $\mathrm{Ai}(x) = \dfrac{1}{3^{2/3}\,\Gamma(2/3)} \underbrace{\sum_{n=0}^{\infty} \dfrac{1\cdot 4 \cdots (3\,n-2)}{(3\,n)!}\, x^{3n}}_{f(x^3)} - \dfrac{1}{3^{1/3}\,\Gamma(1/3)} \underbrace{\sum_{n=0}^{\infty} \dfrac{2\cdot 5 \cdots (3\,n-1)}{(3\,n+1)!}\, x^{3n+1}}_{x g(x^3)}$
- the terms of f and g get large & cancel out

```
> plots:-pointplot([seq([n,abs(coeftayl(AiryAi('x'),'x'=0,n)*10^(n))], n=0..100)]);
> evalf(AiryAi(10));
```

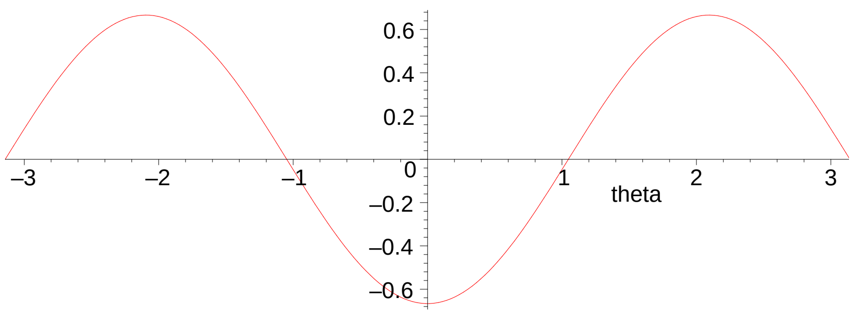
$.1104753255e-9$

- As $x \to +\infty$, we have $\mathrm{Ai}(x) \approx \dfrac{e^{-\frac{2}{3}x^{3/2}}}{2\,\sqrt{\pi}\,x^{1/4}}$.
  Actually $\mathrm{Ai}(z) \sim \dfrac{e^{-\frac{2}{3}z^{3/2}}}{2\,\sqrt{\pi}\,z^{1/4}}$ as $z \to \infty$ in any sector $\{z \in \mathbb{C} | -\theta < \arg z < \theta\}$ with $\theta > 0$.

3 The GMR Method

- $M(r) = \sup_{|z|=r} |y(z)|$
  Ai: $j = e^{\frac{2}{3}i\pi}$, $\mathrm{Ai}(j\,r) \sim \dfrac{e^{\frac{2}{3}r^{3/2}}}{2\,\sqrt{\pi}\,r^{1/4}\,j^{1/4}}$, so $M(r) \sim \dfrac{e^{\frac{2}{3}r^{3/2}}}{2\,\sqrt{\pi}\,r^{1/4}}$ as $r \to \infty$
- order: $\rho = \limsup_{r \to +\infty} \dfrac{\ln \ln M(r)}{\ln r}$
  Ai: $\ln \ln M(r) = \dfrac{3}{2} \ln{(r)} + O(\ln \ln r) \quad \Longrightarrow \quad \rho = \dfrac{3}{2}$
- indicator: $h(\theta) = \limsup_{r \to +\infty} \dfrac{\ln |y(r\,e^{i\theta})|}{r^{\rho}}$
  $|\mathrm{Ai}(r\,e^{i\theta})| \approx \left| e^{-\frac{2}{3}r^{3/2}\exp\left(\frac{3}{2}i\theta\right)} \right| = e^{-\frac{2}{3}r^{3/2}\mathrm{Re}\left(\exp\left(\frac{3}{2}i\theta\right)\right)} = \exp\left(-\dfrac{2}{3}\,r^{\rho}\cos\dfrac{3\,\theta}{2}\right)$
  $h_{\mathrm{Ai}}(\theta) = -\dfrac{2}{3}\cos\left(\dfrac{3}{2}\,\theta\right)$

```
> plot(-2/3*cos(3/2*theta), theta=-Pi..Pi);
```



- in short: $|y(r\,e^{i\theta})| \approx e^{h(\theta)\,r^{\rho}}$ for large r

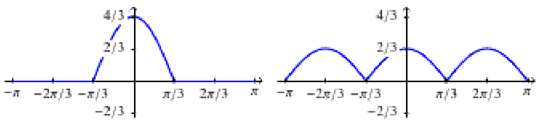
- $\begin{cases} F(z) \approx e^{h_F(\theta) r^\rho} \\ G(z) \approx e^{h_G(\theta) r^\rho} \end{cases}$ (same ρ!) $\Rightarrow \dfrac{G(z)}{F(z)} \approx \exp\big(\overbrace{(h_G(\theta) - h_F(\theta))}^{h_{F/G}} r^\rho\big)$
- $\max_n |y_n z^n| = M(|z|)^{1+o(1)}$
- Thus:

$$\begin{aligned} \text{“lost digits”} &\approx \log_{10}\Big(\max_n |y_n z^n|\Big) - \log_{10}|y(z)| \\ &\approx \log_{10}\frac{M(|z|)}{|y(z)|} = \frac{1}{\ln 10}\ln\frac{M(|z|)}{|y(z)|} \approx \ln\frac{M(|z|)}{|y(z)|} \\ &\approx (r^\rho \max_\varphi h(\rho)) - r^\rho h(\theta) \qquad (z = r\,e^{i\theta}) \\ &= r^\rho (\max h - h(\theta)) \end{aligned}$$

- Idea: find F and $G = y\,F$ such that both h_F and h_G take values close to their maximum in the direction of interest (say, $\theta = 0$).

4 Auxiliary Series for Ai(x)

- First idea: “pull up” bottom of the valley using $F(z) = e^{\sigma z^\rho}$
  What GMR do. Works pretty well for integer ρ. But $e^{z^{3/2}}$ is not an entire function!
- Look what happens when we “shift” the curve by $\dfrac{2\pi}{3}$ to the left/right
- Sum & sum with h_{Ai}



- $\Rightarrow F(z) = \text{Ai}(j\,z)\,\text{Ai}(j^{-1} z)$, $G(z) = \text{Ai}(z)\,\text{Ai}(j\,z)\,\text{Ai}(j^{-1} z)$
- Series expansions? D-finiteness.

```
> restart;
> alias(j = RootOf(_Z^3-1, index=2));
```

```
> with(gfun):
> deqF := holexprtodiffeq(AiryAi(j*x)*AiryAi(j^(-1)*x), y(x));
```

```
> recF := diffeqtorec(deqF, y(x), F(n)):
> collect(op(1, recF), F, factor);
```

$$(-2-4\,n)\,F(n) + (n+1)\,(n+2)\,(n+3)\,F(n+3)$$

```
> evalf(select(type, recF, '='));
```

$$\{F(0) = .1260449191, F(1) = .9188814925e-1, F(2) = .6698748370e-1\}$$

$$F_{n+3} = \frac{2\,(2\,n+1)}{(n+1)\,(n+2)\,(n+3)}\,F_n, \quad F_0 \approx 0.12, \quad F_1 \approx 0.09, \quad F_2 \approx 0.07$$

Obviously $F_n > 0$ for all n.

- Evaluating F is easy.

```
> coefF := rectoproc(recF, F(n), evalfun=evalf):
```

```
> add(coefF(i)*10^i, i=0..150);
```

$.5190221519e17$

```
> evalf(AiryAi(10*j)*AiryAi(10*j^(-1)));
```

$.5190221512e17 + 0.\,\mathrm{i}$

- Similar for G:

```
> restart; alias(j = RootOf(_Z^3-1, index=2)): with(gfun):
> deq := holexprtodiffeq(AiryAi(x)*AiryAi(j*x)*AiryAi(j^(-1)*x), y(x));
```

```
> rec := diffeqtorec(deq, y(x), u(n));
```

$$\mathrm{rec} := \left\{ 9\,u(n) + (-60\,n - 90 - 10\,n^2)\,u(n+3) + (n^4 + 18\,n^3 + 119\,n^2 + 342\,n + 360)\,u(n+6), u(0) = \frac{1}{9\,\Gamma\left(\frac{2}{3}\right)^3}, u(1) = 0, u(2) = 0, u(3) = -\frac{1}{72}\,\frac{-4\,\pi^3 + 9\,\sqrt{3}\,\Gamma\left(\frac{2}{3}\right)^6}{\Gamma\left(\frac{2}{3}\right)^3\,\pi^3}, u(4) = 0, u(5) = 0 \right\}$$

```
> recG := eval(subs(n=3*n, op(1,rec)), u=proc(x) G(x/3) end);
```

$$\mathrm{recG} := 9\,G(n) + (-180\,n - 90 - 90\,n^2)\,G(n+1) + (81\,n^4 + 486\,n^3 + 1071\,n^2 + 1026\,n + 360)\,G(n+2)$$

```
> recG := collect(primpart(recG), G, factor);
```

$$\mathrm{recG} := G(n) - 10\,(n+1)^2\,G(n+1) + (3\,n+5)\,(3\,n+4)\,(n+2)\,(n+1)\,G(n+2)$$

$$G(x) = \sum_{n=0}^{\infty} G_n\,x^{3n} \quad \text{where} \quad G_{n+2} = \frac{10\,(n+1)^2\,G_{n+1} - G_n}{(n+1)\,(n+2)\,(3\,n+4)\,(3\,n+5)}$$

- *Not* obvious that $G_n \geqslant 0$. Also:

```
> ini := select(type, rec, `=`);
```

$$\mathrm{ini} := \left\{ u(0) = \frac{1}{9\,\Gamma\left(\frac{2}{3}\right)^3}, u(1) = 0, u(2) = 0, u(3) = -\frac{1}{72}\,\frac{-4\,\pi^3 + 9\,\sqrt{3}\,\Gamma\left(\frac{2}{3}\right)^6}{\Gamma\left(\frac{2}{3}\right)^3\,\pi^3}, u(4) = 0, u(5) = 0 \right\}$$

```
> coefG := rectoproc(subs(ini, {recG, G(0)=u(0), G(1)=u(3)}), G(n), evalfun=evalf):
> add(coefG(i) * 10^(3*i), i=0..50);
```

$.5548985660e16$

```
> evalf(AiryAi(10)*AiryAi(j*10)*AiryAi(j^(-1)*10));
```

$5733914.110 - .204387418e - 3\,\mathrm{i}$

```
> evalf(AiryAi(10)*(AiryAi(10)^2 + AiryBi(10)^2)/4);
```

5733914.120

- Why?

```
> evalf(evalf[100](series(AiryAi(x)*(AiryAi(x)^2+AiryBi(x)^2)/4,x=0, 40)));
```

$.4474948231e - 1 + .5037080542e - 2\,x^3 + .1405330778e - 3\,x^6 + .1738817174e - 5\,x^9 + .1209126352e - 7\,x^{12} + .5378708507e - 10\,x^{15} + .1661161451e - 12\,x^{18} + .3768626043e - 15\,x^{21} + .6545218449e - 18\,x^{24} + .8981063338e - 21\,x^{27} + .9981429333e - 24\,x^{30} + .9167576927e - 27\,x^{33} + .7074985672e - 30\,x^{36} + .4652234199e - 33\,x^{39} + O(x^{40})$

```
> seq(coefG(i)*x^(3*i), i=0..13);
```

.4474948235e − 1, .5037080589e − 2 x^3, .1405330885e − 3 x^6, .1738818307e − 5 x^9, .1209133273e − 7 x^{12}, .5378981594e − 10 x^{15}, .1661913296e − 12 x^{18}, .3783873879e − 15 x^{21}, .6782358308e − 18 x^{24}, .118981\ 9312e − 20 x^{27}, .3906921345e − 23 x^{30}, .2490012860e − 25 x^{33}, .1669355377e − 27 x^{36}, .9824517900e − 30 x^{39}

Summation *is* stable, but the recurrence to compute the coefficients isn't!

5 Unstable Recurrences and Miller's Method

- $c_n = n!^2 G_n$
  $\frac{c_{n+2}}{(n+2)!^2} = \frac{10\,c_{n+1}/n!^2 - c_n/n!^2}{(n+1)(n+2)(3n+4)(3n+5)}$
  $c_{n+2} = \frac{(n+1)(n+2)}{(3n+4)(3n+5)}(10\,c_{n+1} - c_n)$
  $\underbrace{\frac{(3n+4)(3n+5)}{(n+1)(n+2)}}_{\to 9 \text{ as } n\to\infty} c_{n+2} - 10\,c_{n+1} + c_n = 0$
- so the recurrence "looks like" $9\,u_{n+2} - 10\,u_n + 1 = 0$
  $9X^2 - 10X + 1$, roots $1/9$ and 1
  $u_n = 1$, $u_n = 9^{-n}$
- Theorem (Perron-Kreuser): either $\frac{c_{n+1}}{c_n} \to 1$ or $\frac{c_{n+1}}{c_n} \to \frac{1}{9}$
  for G: either $G_n \approx \frac{1}{n!^2}$ ("dominant") or $G_n \approx \frac{1}{9^n\,n!^2}$ ("minimal")
- Wimp 1984
- $(c_n), (d_n)$ such that $d_n \gg c_n$ — in our case $c_{n+1}/c_n \to 1$, $d_{n+1}/d_n \to 1/9$
- define (u_n) by $u_0 \approx c_0$, $u_1 \approx c_1$ (rounding err on ini, assume the rest is exact)
  in fact $\begin{pmatrix} u_0 \\ u_1 \end{pmatrix} = (1-\delta)\begin{pmatrix} c_0 \\ c_1 \end{pmatrix} + \varepsilon \begin{pmatrix} d_0 \\ d_1 \end{pmatrix}$
  so $u_n = \underbrace{(1-\delta)\,c_n}_{\text{small}} + \underbrace{\varepsilon\,d_n}_{\text{large}}$
- Miller's method. Idea: run the rec backwards; minimal $\rightsquigarrow$ dominant
- Algorithm (assume exact real arithmetic):
  - Choose $N \gg 0$.
  - Set $u_N = 1$, $u_{N+1} = 0$.
  - Compute $u_{N-1}, \ldots, u_1, u_0$.
  - Return $\frac{c_0}{u_0} u_n$.
- Output of Miller $\approx (c_n)$.
- Proof:
  $v_n^{(N)} = \text{output} = \frac{c_0}{u_0} u_n$
  $v_n^{(N)} = \alpha^{(N)} c_n + \beta^{(N)} d_n$
  $\begin{cases} v_{N+1}^{(N)} = 0 \\ v_0^{(N)} = c_0 \end{cases} \Rightarrow \begin{cases} \alpha^{(N)} c_{N+1} + \beta^{(N)} d_{N+1} = 0 \\ \alpha^{(N)} c_0 + \beta^{(N)} d_0 = c_0 \end{cases}$
  Solving this linear system, we get

$$\Delta(n) = \begin{vmatrix} c_{N+1} & d_{N+1} \\ c_0 & d_0 \end{vmatrix},$$

$$\alpha^{(N)} = \frac{-c_0\,d_{N+1}}{\Delta(N)} = \frac{-c_0\,d_{N+1}}{d_0\,c_{N+1} - c_0\,d_{N+1}} \to 1, \qquad \beta^{(N)} = \frac{c_0\,c_{N+1}}{\Delta(N)} \to 0,$$

so *for fixed* n, $v_n^{(N)} \to c_n$ *as* $N \to \infty$ (and pretty fast actually).

6 Proofs and Error Bounds

- Idea: $G_n = \frac{1}{2\pi i} \oint \frac{G(z)}{z^{3n+1}} \mathrm{d}z$.
  $\left| \frac{\mathrm{Ai}(z)}{\widetilde{\mathrm{Ai}}(z)} - 1 \right| \leqslant \frac{5}{48} \cos(\theta/2)\, r^{-3/2}$.
  Saddle-point method: determine where the weight of the integral is concentrated; estimate the integral *while bounding all estimation errors* starting from the formula on Ai.
  We get $G_n \sim \gamma_n = \frac{1}{4\sqrt{3}\,\pi\, 9^n\, n!^2}$ with $\left| \frac{G_n}{\gamma_n} - 1 \right| \leqslant 2.4\, n^{-1/4}$ for all $n \geqslant 1$.
- Corollary: $G_n \geqslant 0$. We were not able to find a significantly simpler proof!
- The (much) more precise estimate is useful to bound the errors in Miller's algo.
  Bound the "method error" as outlined above (if we have precise estimates for c_n and d_n, we can bound $|u_n - c_n|$.
- Roundoff errors: see Matthiej & van der Sluis, Numerische Mathematik, 1976.
- Throw in some routine (though somewhat tedious) error analysis, get a rigorous multiple-precision evaluation algorithm à la MPFR.