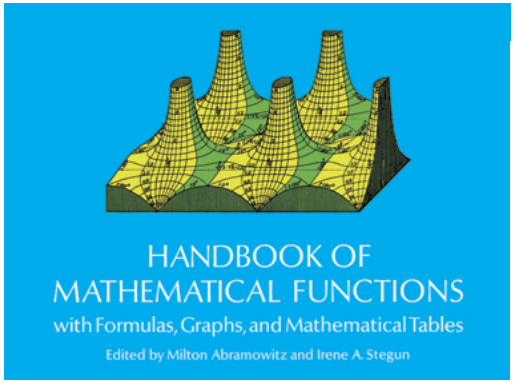
Fungrim: a semantic library of mathematical functions

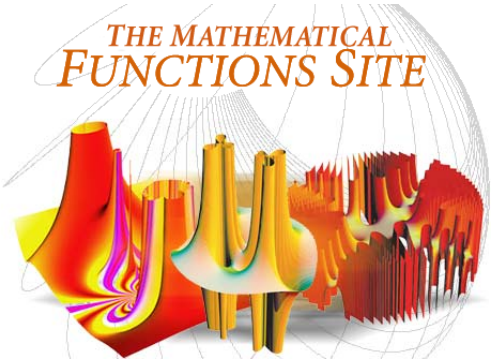
Fredrik Johansson

Big Data in Pure Mathematics, May 21, 2022, online

Special functions references



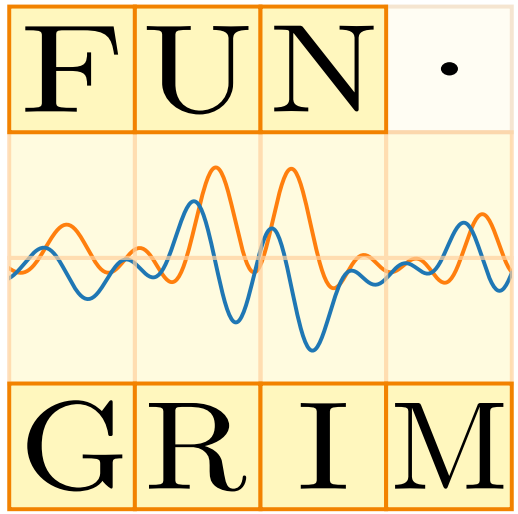
Digital Library of Mathematical Functions



Dynamic Dictionary of Mathematical Functions

- Formal knowledge (theorems)
- Informal knowledge (subject connections; know-how)
- Data (tables)

https://fungrim.org



Why yet another reference for special functions?

- Open source
- Fully computer-readable, symbolic content
- Not restricted to a particular class of functions
- No paper edition size restrictions
- Explicit assumptions in formulas

Backend software

In [3]:
```
from pygrim import *

formula = Where(Sum(1/f(n), For(n, -N, N), NotEqual(n, 0)), Def(f(n),
     Cases(Tuple(n**2, CongruentMod(n, 0, 3)), Tuple(1, Otherwise))))

formula
```

Out[3]:
$$\sum_{\substack{n=-N \\ n\neq 0}}^{N} \frac{1}{f(n)} \quad \text{where } f(n) = \begin{cases} n^2, & n \equiv 0 \pmod 3 \\ 1, & \text{otherwise} \end{cases}$$

In [4]:
```
formula.replace({N:10}).eval()
```

Out[4]:
$$\frac{2317}{162}$$

- Expressions → TeX → (KaTeX) → HTML
- Symbolic and numerical evaluation (uses Flint, Arb)
- Simple syntax (embeds in Python, ...)
- Inert (no evaluation) by default

Backend software: exact and numerical evaluation

In [24]:
```
formula = ((DedekindEta(1 + Sqrt(-1)) / Gamma(Div(5, 4))) ** 12)
formula
```

Out[24]:
$$\left(\frac{\eta\left(1+\sqrt{-1}\right)}{\Gamma\left(\frac{5}{4}\right)}\right)^{12}$$

In [25]:
```
formula.eval()
```

Out[25]:
$$-\frac{4096}{\pi^9}$$

In [26]:
```
formula.n()
```

Out[26]:
$$\left[-0.13740770743127527951 \pm 3.19 \cdot 10^{-21}\right] + \left[0 \pm 3.32 \cdot 10^{-28}\right] i$$

In [27]:
```
formula.eval().n()
```

Out[27]:
$$\left[-0.13740770743127527951 \pm 3.19 \cdot 10^{-21}\right]$$

Challenge: rigorous semantics

$$\text{Formula} \neq \text{Theorem}$$

Traditionally, reference works AND computer algebra systems neglect issues with types, exceptional points, branch cuts, infinities . . .

Challenge: rigorous semantics

$$\text{Formula} \neq \text{Theorem}$$

Traditionally, reference works AND computer algebra systems neglect issues with types, exceptional points, branch cuts, infinities . . .

> ***The answer might not be valid for certain exceptional values of the parameters.***

- Wolfram Mathematica documentation

Example: what is ${}_1F_1(-1,-1,1)$?

Mathematica:

```
In[•]:= Hypergeometric1F1[n, m, x] /. {m → -1, n → -1, x → 1}

Out[•]= 2

In[•]:= (Hypergeometric1F1[n, m, x] /. {m → n}) /. {n → -1, x → 1}

Out[•]= e
```

PyGrim:

```
>>> f = Hypergeometric1F1(n, m, x)
>>> f.replace({m:-1, n:-1, x:1}).eval()
2
>>> f.replace({m: n}).eval().replace({n:-1, x:1}).eval()
2
```

Example: what is ${}_1F_1(-1,-1,1)$?

`https://fungrim.org/entry/dec042/`

$$ {}_1F_1(-n,b,z) = \sum_{k=0}^{n} \frac{(-n)_k}{(b)_k} \frac{z^k}{k!} $$

Assumptions:
$n \in \mathbb{Z}_{\geq 0}$ and $b \in \mathbb{C}$ and not $(b \in \{0,-1,\ldots\}$ and $b > -n)$ and $z \in \mathbb{C}$

`https://fungrim.org/entry/be533c/`

$$ {}_1F_1(a,b,z) = e^z \, {}_1F_1(b-a,b,-z) $$

Assumptions: $a \in \mathbb{C}$ and $b \in \mathbb{C} \setminus \{0,-1,\ldots\}$ and $z \in \mathbb{C}$

Testing formulas

Formula: $\sqrt{x^2} = x$, Assumptions: $x \in \mathbb{R}$

Testing formulas

Formula: $\sqrt{x^2} = x$, Assumptions: $x \in \mathbb{R}$

```
>>> formula = Equal(Sqrt(x**2), x)
>>> formula.test(variables=[x], assumptions=Element(x, RR))
{x: 0}    ...  True
{x: Div(1, 2)}    ...  True
{x: Sqrt(2)}    ...  True
{x: Pi}    ...  True
{x: 1}    ...  True
{x: Neg(Div(1, 2))}    ...  False
```

Testing formulas

Formula: $\sqrt{x^2} = x$, Assumptions: $x \in \mathbb{R}$

```
>>> formula = Equal(Sqrt(x**2), x)
>>> formula.test(variables=[x], assumptions=Element(x, RR))
{x: 0}    ...  True
{x: Div(1, 2)}    ...  True
{x: Sqrt(2)}    ...  True
{x: Pi}    ...  True
{x: 1}    ...  True
{x: Neg(Div(1, 2))}    ...  False
```

Assumptions: $x \in \mathbb{C} \wedge (\mathrm{Re}(x) > 0 \vee (\mathrm{Re}(x) = 0 \wedge \mathrm{Im}(x) \geq 0))$

```
>>> formula.test(variables=[x], assumptions=And(Element(x, CC),
...    Or(Greater(Re(x), 0), And(Equal(Re(x), 0),
...               GreaterEqual(Im(x), 0)))))
...
Passed 100 instances (75 True, 25 Unknown, 0 False)
```

Fungrim entry: 799894

$$\left|R_F(x,y,z) - A^{-1/2}\left(1 - \frac{E}{10} + \frac{F}{14} + \frac{E^2}{24} - \frac{3EF}{44} - \frac{5E^3}{208} + \frac{3F^2}{104} + \frac{E^2F}{16}\right)\right| \le \frac{0.2\left|A^{-1/2}\right| M^8}{1-M} \quad \text{where } A = \frac{x+y+z}{3},\ X = 1 - \frac{x}{A},\ Y = 1 - \frac{y}{A},\ Z = 1 - \frac{z}{A},\ E = XY + XZ + YZ,\ F = XYZ,\ M = \max(|X|, |Y|, |Z|)$$

Assumptions: $x \in \mathbb{C}$ and $y \in \mathbb{C}$ and $z \in \mathbb{C}$ and $((x \neq 0 \text{ and } y \neq 0) \text{ or } (x \neq 0 \text{ and } z \neq 0) \text{ or } (y \neq 0 \text{ and } z \neq 0))$ and $\max(|\arg(x) - \arg(y)|, |\arg(x) - \arg(z)|, |\arg(y) - \arg(z)|) < \pi$ and $\left|1 - \frac{3x}{x+y+z}\right| < 1$ and $\left|1 - \frac{3y}{x+y+z}\right| < 1$ and $\left|1 - \frac{3z}{x+y+z}\right| < 1$

Fungrim entry: 799894

$$\left|R_F(x,y,z) - A^{-1/2}\left(1 - \frac{E}{10} + \frac{F}{14} + \frac{E^2}{24} - \frac{3EF}{44} - \frac{5E^3}{208} + \frac{3F^2}{104} + \frac{E^2F}{16}\right)\right| \le \frac{0.2\left|A^{-1/2}\right| M^8}{1-M}$$

where $A = \frac{x+y+z}{3}$, $X = 1 - \frac{x}{A}$, $Y = 1 - \frac{y}{A}$, $Z = 1 - \frac{z}{A}$, $E = XY + XZ + YZ$, $F = XYZ$, $M = \max(|X|, |Y|, |Z|)$

Assumptions: $x \in \mathbb{C}$ and $y \in \mathbb{C}$ and $z \in \mathbb{C}$ and $((x \neq 0 \text{ and } y \neq 0) \text{ or } (x \neq 0 \text{ and } z \neq 0) \text{ or } (y \neq 0 \text{ and } z \neq 0))$ and $\max(|\arg(x) - \arg(y)|, |\arg(x) - \arg(z)|, |\arg(y) - \arg(z)|) < \pi$ and $\left|1 - \frac{3x}{x+y+z}\right| < 1$ and $\left|1 - \frac{3y}{x+y+z}\right| < 1$ and $\left|1 - \frac{3z}{x+y+z}\right| < 1$

```
>>> test_fungrim_entry("799894")
{x: Div(1, 6), y: Add(1, ConstI), z: ConstI}    ...  True
{x: Sqrt(2), y: 3, z: Div(1, 2)}    ...  True
...
Passed 100 instances (99 True, 1 Unknown, 0 False)
```

Testing the whole database

- A few hours in total (100 random inputs per entry)
- About 75% of entries effectively testable (right now)
- First run found errors in 24 out of 2618 entries
  - 4× wrong formula (sign error, etc.)
  - 6× incorrect assumptions
  - 14× wrong metadata / malformatted expressions

Future

- Frontend and backend improvements
  - Use Calcium: `https://fredrikj.net/calcium/`
  - Other infinite objects (functions, limits, sets)
- Formal semantics, ideally backed by a real theorem prover. Joint effort with MathLib?
  - Resolve subtle type issues
  - Want formally proved theorems as well as tentative theorems that are only informally tested/reviewed
- Joint effort with LMFDB?
  - Symbolic knowledge about modular forms and L-functions
- Integration with computer algebra software

Thank you!