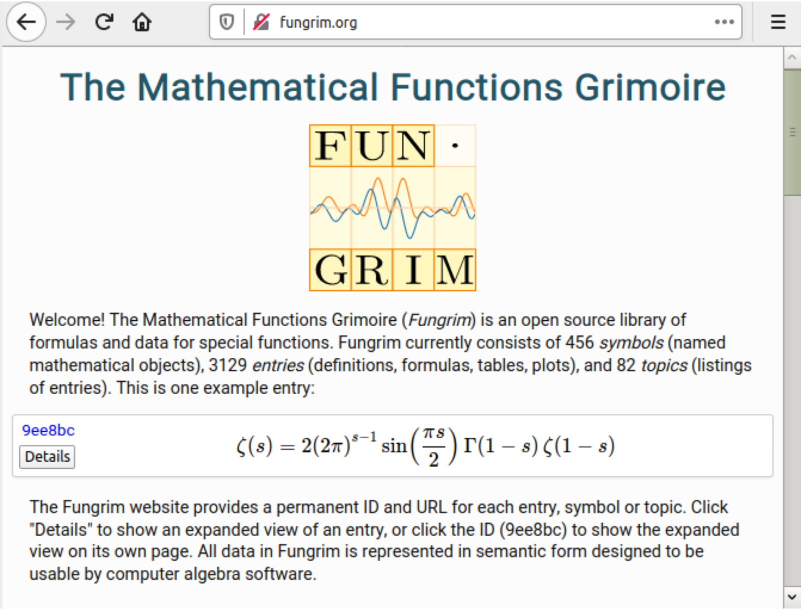
FunGrim: a symbolic library for special functions

Fredrik Johansson (http://fredrikj.net – Inria Bordeaux)

http://fungrim.org

International Congress on Mathematical Software (ICMS) 2020



fungrim.org

The Mathematical Functions Grimoire

FUN · GRIM

Welcome! The Mathematical Functions Grimoire (*Fungrim*) is an open source library of formulas and data for special functions. Fungrim currently consists of 456 *symbols* (named mathematical objects), 3129 *entries* (definitions, formulas, tables, plots), and 82 *topics* (listings of entries). This is one example entry:

9ee8bc

Details

$$\zeta(s) = 2(2\pi)^{s-1} \sin\left(\frac{\pi s}{2}\right) \Gamma(1-s)\,\zeta(1-s)$$

The Fungrim website provides a permanent ID and URL for each entry, symbol or topic. Click "Details" to show an expanded view of an entry, or click the ID (9ee8bc) to show the expanded view on its own page. All data in Fungrim is represented in semantic form designed to be usable by computer algebra software.

grimoire = book of magic formulas

fungrim.org

Browse by topic

- All topics in alphabetical order
- Fundamentals
  - Symbolic expressions
  - Elementary logic and set theory
  - Numbers and infinities
  - Operators
  - Complex plane
- Constants
  - Pi ★
  - Imaginary unit
  - Euler's constant
  - Golden ratio
  - Catalan's constant
- Elementary functions
  - Complex parts
  - Exponential function
  - Natural logarithm
  - Square roots
  - Powers
  - Digamma function ★
  - Beta function
  - Barnes G-function ★
- Hypergeometric functions
  - Gauss hypergeometric function
  - Confluent hypergeometric functions
  - Error functions
  - Airy functions
  - Bessel functions
  - Coulomb wave functions
- Orthogonal polynomials
  - Legendre polynomials
  - Chebyshev polynomials ★
- Zeta and L-functions
  - Riemann zeta function
  - Riemann hypothesis
  - Hurwitz zeta function
  - Stieltjes constants

fungrim.org/topic/Riemann_zeta_function/

Riemann zeta function

Definitions

e0a6a2

Details

Symbol: RiemannZeta — $\zeta(s)$ — Riemann zeta function

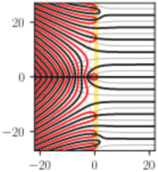
Illustrations

3131df

Details

Image: X-ray of $\zeta(s)$ on $s \in [-22, 22] + [-27, 27]\, i$ with the critical strip highlighted

Big 🔍



fungrim.org/topic/Riemann_zeta_function/

Functional equation

9ee8bc
Details

$$\zeta(s) = 2(2\pi)^{s-1} \sin\left(\frac{\pi s}{2}\right) \Gamma(1-s)\, \zeta(1-s)$$

1a63af
Details

$$\zeta(1-s) = \frac{2\cos\left(\frac{1}{2}\pi s\right)}{(2\pi)^s} \Gamma(s) \zeta(s)$$

Bounds and inequalities

809bc0
Details

$$|\zeta(s)| \le \zeta(\operatorname{Re}(s))$$

3a5eb6
Details

$$|\zeta(s)| < 3 \left|\frac{1+s}{1-s}\right| \left|\frac{1+s}{2\pi}\right|^{(1+\eta-\operatorname{Re}(s))/2} \zeta(1+\eta)$$

Euler-Maclaurin formula

792f7b
Details

$$\zeta(s) = \sum_{k=1}^{N-1} \frac{1}{k^s} + \frac{N^{1-s}}{s-1} + \frac{1}{N^s} \left(\frac{1}{2} + \sum_{k=1}^{M} \frac{B_{2k}}{(2k)!} \frac{(s)_{2k-1}}{N^{2k-1}}\right) - \int_N^\infty \frac{B_{2M}(t - \lfloor t \rfloor)}{(2M)!} \frac{(s)_{2M}}{t^{s+2M}}\, dt$$

fungrim.org/topic/Riemann_zeta_function/

809bc0

Details

$$|\zeta(s)| \le \zeta(\operatorname{Re}(s))$$

Assumptions: $s \in \mathbb{C} \;\mathbin{\operatorname{and}}\; \operatorname{Re}(s) > 1$

TeX:

```
\left|\zeta\!\left(s\right)\right| \le \zeta\!\left(\operatorname{Re}(s)\right)

s \in \mathbb{C} \;\mathbin{\operatorname{and}}\; \operatorname{Re}(s) > 1
```

Definitions:

Fungrim symbol	Notation	Short description
Abs	$\lvert z \rvert$	Absolute value
RiemannZeta	$\zeta(s)$	Riemann zeta function
Re	$\operatorname{Re}(z)$	Real part
CC	$\mathbb{C}$	Complex numbers

Source code for this entry:

```
Entry(ID("809bc0"),
    Formula(LessEqual(Abs(RiemannZeta(s)), RiemannZeta(Re(s)))),
    Variables(s),
    Assumptions(And(Element(s, CC), Greater(Re(s), 1))))
```

Example: http://fungrim.org/entry/0b5b04/

$$G_{2k}\left(\frac{a\tau + b}{c\tau + d}\right) = (c\tau + d)^{2k} G_{2k}(\tau)$$

Assumptions: $k \in \mathbb{Z}_{\geq 2}$ and $\tau \in \mathbb{H}$ and $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \text{SL}_2(\mathbb{Z})$

```
Entry(ID("0b5b04"),
    Formula(Equal(EisensteinG(2*k, (a*tau+b)/(c*tau+d)),
        (c*tau+d)**(2*k) * EisensteinG(2*k, tau))),
    Variables(k, tau, a, b, c, d),
    Assumptions(And(Element(k, ZZGreaterEqual(2)),
        Element(tau, HH),
        Element(Matrix2x2(a, b, c, d), SL2Z))))
```

Example: http://fungrim.org/entry/375afe/

$$|\pi(x) - \operatorname{li}(x)| < \frac{\sqrt{x}\log(x)}{8\pi}$$

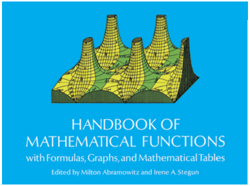
Assumptions: $x \in \mathbb{R} \text{ and } x \geq 2657 \text{ and } \mathrm{RH}$

References:

- L. Schoenfeld (1976). Sharper bounds for the Chebyshev functions $\theta(x)$ and $\psi(x)$. II. Mathematics of Computation. 30 (134): 337-360. DOI: 10.2307/2005976

```
Entry(ID("375afe"),
    Formula(Less(Abs(PrimePi(x) - LogIntegral(x)),
        (Sqrt(x) * Log(x)) / (8 * Pi))),
    Variables(x),
    Assumptions(And(Element(x, RR), GreaterEqual(x, 2657),
        RiemannHypothesis)),
    References("L. Schoenfeld (1976). ..."))
```

Why yet another reference work on special functions?



Digital Library of Mathematical Functions

THE MATHEMATICAL FUNCTIONS SITE

WIKIPEDIA
The Free Encyclopedia

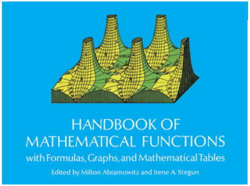
Dynamic Dictionary of Mathematical Functions

DRMF

LMFDB

. . .

Why yet another reference work on special functions?



Digital
Library of
Mathematical
Functions

THE MATHEMATICAL
FUNCTIONS SITE

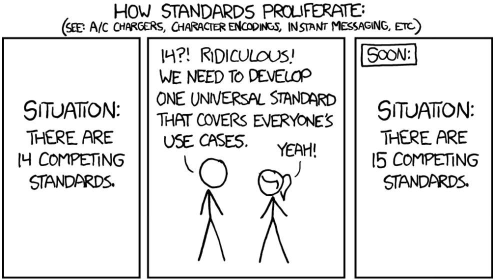
WIKIPEDIA
The Free Encyclopedia

Dynamic
Dictionary of
Mathematical
Functions

DRMF

LMFDB

. . .



FunGrim goals and principles

- Open source
- Fully computer-readable, symbolic content
- Complex variables, arbitrary mathematical functions
- No paper edition size restrictions
- Rigorous semantics + explicit conditions/assumptions

Backend software: PyGrim

`https://github.com/fredrik-johansson/fungrim/`

Generating the website

- Symbolic expressions $\to$ TeX $\to$ (KaTeX) $\to$ HTML
- Cross-references, index pages...

Symbolic and numerical evaluation, testing

- Symbolics engine in Python + Computational libraries (Flint, Arb, ...)

Grim formula language

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]:

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

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

Out[4]:

$$\frac{2317}{162}$$

- Simple syntax (embeds in Python, ...)
- Simple functional, mathematical semantics
- Inert (no evaluation) by default
- NOT a general-purpose programming language
- Documentation: `http://fungrim.org/grim/`

Grim formula language

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

Symbolic engine

Implemented:

- Direct evaluation of most mathematical functions (symbolic and/or numerical with Arb)
- Predicates involving numbers, boolean logic
- Simple inferences (e.g. $x \in \mathbb{Q} \implies x \in \mathbb{R}$)
- Finite set operations
- Exact calculation in $\overline{\mathbb{Q}}$ + some symbolic arithmetic
  - Spin-off project: `http://fredrikj.net/calcium/` - C library for exact real and complex arithmetic

Not implemented:

- Calculus operators (limits, integrals, derivatives, etc.)
- Advanced inferences (requiring SAT solving, LP, CAD, etc.)
- Infinite set comprehensions
- Most operations on power series, matrices...

Consistent semantics

Traditional point of view (reference works AND computer algebra systems): formulas are only correct “modulo special cases” (exceptional points, branch cuts, infinities . . .)

Consistent semantics

Traditional point of view (reference works AND computer algebra systems): formulas are only correct "modulo special cases" (exceptional points, branch cuts, infinities ...)

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

[Wolfram Language (*Mathematica*) Documentation, 2019]

Consistent semantics

Traditional point of view (reference works AND computer algebra systems): formulas are only correct "modulo special cases" (exceptional points, branch cuts, infinities . . .)

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

[Wolfram Language (*Mathematica*) Documentation, 2019]

- Burden is on user to check details / fill in gaps
- Cannot be used in mechanical theorem proving
- Automated testing is futile

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 ${}_1F_1(-1,-1,1)$?

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

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

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

`http://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 $z \in \mathbb{C}$

Simplification with free variables

```
>>> (x / x).eval()
Div(x, x)

>>> (x / x).eval(assumptions=Element(x, CC))
Div(x, x)

>>> (x / x).eval(assumptions=And(Element(x, CC),
...                                NotEqual(x, 0)))
1
```

```
>>> Sin(Pi * n).eval()
Sin(Mul(Pi, n))

>>> Sin(Pi * n).eval(assumptions=Element(n, ZZ))
0
```

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

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

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

Fungrim entry: 799894

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

Assumptions: $x \in \mathbb{C}$ and $y \in \mathbb{C}$ and $z \in \mathbb{C}$ and $((x \ne 0 \text{ and } y \ne 0) \text{ or } (x \ne 0 \text{ and } z \ne 0) \text{ or } (y \ne 0 \text{ and } z \ne 0))$ 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
  - $4\times$ wrong formula (sign error, etc.)
  - $6\times$ incorrect assumptions
  - $14\times$ wrong metadata / malformatted expressions

Future development?

- Database format (not needed for < 10000 entries)
- Automatically generated content? (Like DDMF, Wolfram Functions Site.)
- Website interface (page layout, search, test reports)
- Improved test code, and test reports on the website
- User-friendly backend library (documentation, easy installation)
- JavaScript and Julia implementations of symbolic expressions
- Easy submissions (with automatic testing?)
- Integration with other projects

Formulas as rewrite rules

In [16]:
```
fungrim_entry("ad6c1c")
```

Out[16]:

$$\mathrm{Entry}\left(\mathrm{ID}(\text{``ad6c1c''}), \mathrm{Formula}\left(\sin(a)\sin(b) = \frac{\cos(a-b)-\cos(a+b)}{2}\right), \mathrm{Variables}(a,b), \mathrm{Assumptions}(a \in \mathbb{C} \;\text{ and }\; b \in \mathbb{C})\right)$$

In [17]:
```
(Sin(2) * Sin(Sqrt(2)))
```

Out[17]: $\sin(2)\sin\left(\sqrt{2}\right)$

In [18]:
```
(Sin(2) * Sin(Sqrt(2))).rewrite_fungrim("ad6c1c")
```

Out[18]:

$$\frac{\cos\left(2-\sqrt{2}\right)-\cos\left(2+\sqrt{2}\right)}{2}$$

- Assumptions are checked automatically
- Need better pattern matching, good search tools to be truly useful
- Hard problem: automatic formula simplification

Thank you!