Calcium: computing in exact real and complex fields

Fredrik Johansson

ISSAC 2021
July 21, 2021 Online

Motivation

Current computer algebra systems are quite good when it comes to working in (structures over) fields like these:

$$\mathbb{F}_7 \qquad \mathbb{Q} \qquad \mathbb{Q}(\sqrt{2})$$

These fields are more problematic:

$$\mathbb{R} \qquad \mathbb{C}$$

Example: real numbers in SageMath

```
sage: sqrt(RDF(2)) ** 2          # floating-point "fields"
2.0000000000000004
```

Example: real numbers in SageMath

```
sage: sqrt(RDF(2)) ** 2          # floating-point "fields"
2.0000000000000004
```

```
sage: sqrt(RBF(2)) ** 2                 # balls, intervals
[2.00000000000000 +/- 1.30e-15]
```

Example: real numbers in SageMath

```
sage: sqrt(RDF(2)) ** 2          # floating-point "fields"
2.0000000000000004
```

```
sage: sqrt(RBF(2)) ** 2                      # balls, intervals
[2.00000000000000 +/- 1.30e-15]
```

```
sage: sqrt(QQbar(2)) ^ 2                   # algebraic numbers
2.000000000000000?
sage: sqrt(QQbar(2)) ^ 2 == 2
True
```

Example: real numbers in SageMath

```
sage: sqrt(RDF(2)) ** 2          # floating-point "fields"
2.0000000000000004
```

```
sage: sqrt(RBF(2)) ** 2                # balls, intervals
[2.000000000000000 +/- 1.30e-15]
```

```
sage: sqrt(QQbar(2)) ^ 2              # algebraic numbers
2.000000000000000?
sage: sqrt(QQbar(2)) ^ 2 == 2
True
```

```
sage: sqrt(2) ^ 2                    # symbolic expressions
2
```

Calcium

$e^{\pi} - \pi$

$\mathbb{C}$a

Calcium

- C library for exact real and complex numbers, polynomials, matrices
- Includes a Python interface
  In progress: Julia interface (in Nemo.jl)
- `http://fredrikj.net/calcium/`
- **Demo notebook**:
  `https://mybinder.org/v2/gh/fredrik-johansson/calcium/HEAD?filepath=doc%2Fintroduction.ipynb`

Problem 1: correctness

$$X = 2\log(\sqrt{2} + \sqrt{3}) - \log(5 + 2\sqrt{6}) \qquad (X = 0)$$

$$A = \begin{pmatrix} 0 & X \\ 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & X + e^{-1000} \\ 0 & 0 \end{pmatrix}$$

Problem 1: correctness

$$X = 2\log(\sqrt{2}+\sqrt{3}) - \log(5+2\sqrt{6}) \qquad (X = 0)$$

$$A = \begin{pmatrix} 0 & X \\ 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & X + e^{-1000} \\ 0 & 0 \end{pmatrix}$$

Maple 2020, SageMath 9.2 `SymbolicRing`: $\text{rank}(A) = 1$

Mathematica 12.2: $\text{rank}(B) = 0$

Problem 1: correctness

$$X = 2\log(\sqrt{2}+\sqrt{3}) - \log(5+2\sqrt{6}) \qquad (X = 0)$$

$$A = \begin{pmatrix} 0 & X \\ 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & X + e^{-1000} \\ 0 & 0 \end{pmatrix}$$

Maple 2020, SageMath 9.2 `SymbolicRing`: $\text{rank}(A) = 1$

Mathematica 12.2: $\text{rank}(B) = 0$

Calcium 0.3:

```
>>> X = 2*log(sqrt(2)+sqrt(3)) - log(5+2*sqrt(6))
>>> A = ca_mat([[0,X],[0,0]]); A.rank()
0
>>> B = ca_mat([[0,X+exp(-1000)],[0,0]]); B.rank()
1
```

Problem 2: efficiency

```
N = 1/16*(44*(7*sqrt(2)-10)*sqrt(sqrt(2)+2)*sqrt(-17*sqrt(2)+26)
+ 2*(11*(7*sqrt(2)-10)*sqrt(sqrt(2)+2)*sqrt(-17*sqrt(2)+26)-...
(...this goes on for several screens...)
```

I have to check if this value is equal to (...). Sadly it keeps loading for hours (at 6 hours I stopped the kernel)

– `https://ask.sagemath.org/question/52653`

Problem 2: efficiency

N = 1/16(44*(7*sqrt(2)-10)*sqrt(sqrt(2)+2)*sqrt(-17*sqrt(2)+26)*

+ 2(11*(7*sqrt(2)-10)*sqrt(sqrt(2)+2)*sqrt(-17*sqrt(2)+26)-...*

(...this goes on for several screens...)

> *I have to check if this value is equal to (...). Sadly it keeps loading for hours (at 6 hours I stopped the kernel)*
>
> – https://ask.sagemath.org/question/52653

Calcium 0.3:

```
> build/examples/huge_expr
Evaluating N... (...)    Equal = T_TRUE
Total: cpu/wall(s): 8.462 8.464
```

Idea: automatically constructing subfields of $\mathbb{C}$

Field elements: $z \in K, \qquad K = \mathbb{Q}(a_1, \ldots, a_n)$

Extension numbers $a_k \in \mathbb{C}$:

- $\sqrt{2}$, i, ...
- π, $e^{2\sqrt{2}+\pi i}$, $\log(2\pi)$, ...
- Black box computable numbers (todo)

Need algorithms for:

- Arithmetic
- Choosing and simplifying extension numbers
- Predicates (example: $z = 0$?) - not decidable, but can have partial algorithms (e.g. for $\overline{\mathbb{Q}}$)

Previous work and inspiration

Implementations of $\overline{\mathbb{Q}}$:

- SageMath - hybrid representation
- Magma (by Allan Steel) - multivariate representation

Transcendental numbers:

- Theoretical work (Richardson's algorithm, D-finite numbers)
- Mathematica, Maple, ...

Dependencies:

- Flint (polynomials - credits to Bill Hart & Daniel Schultz)
- Arb (ball arithmetic)
- Antic (number fields)

Field structure

The trivial field $K = \mathbb{Q}$

Field structure

The trivial field $K = \mathbb{Q}$

Transcendental number fields

$$K = \mathbb{Q}(a_1, \ldots, a_n) \;\cong\; \mathbb{Q}(X_1, \ldots, X_n),$$
$$a_1, \ldots, a_n \text{ algebraically independent over } \mathbb{Q}$$

Field structure

$$\frac{\pi^2 - 9}{\pi + 3} = \pi - 3$$

```
>>> (pi**2 - 9) / (pi + 3)
0.141593 {a-3 where a = 3.14159 [Pi]}

>>> (pi**2 - 9) / (pi + 3) - (pi - 3)
0

>>> (pi**2 - 9) / (pi + 3) == pi - 3
True
```

Field structure

The trivial field $K = \mathbb{Q}$

Transcendental number fields

$$K = \mathbb{Q}(a_1, \ldots, a_n) \;\cong\; \mathbb{Q}(X_1, \ldots, X_n),$$
$$a_1, \ldots, a_n \text{ algebraically independent over } \mathbb{Q}$$

Algebraic number fields

$$K = \mathbb{Q}(a) \;\cong\; \mathbb{Q}[X]/\langle f(X)\rangle$$

Field structure

$$\frac{\varphi^{100} - (1-\varphi)^{100}}{\sqrt{5}} = F_{100}$$

```
>>> phi = (sqrt(5)+1)/2
>>> phi
1.61803 {(a+1)/2 where a = 2.23607 [a^2-5=0]}

>>> (phi**100 - (1-phi)**100)/sqrt(5)
3.54225e+20 {354224848179261915075}
```

Field structure

The trivial field $K = \mathbb{Q}$

Transcendental number fields

$$K = \mathbb{Q}(a_1, \ldots, a_n) \;\cong\; \mathbb{Q}(X_1, \ldots, X_n),$$
$$a_1, \ldots, a_n \text{ algebraically independent over } \mathbb{Q}$$

Algebraic number fields

$$K = \mathbb{Q}(a) \;\cong\; \mathbb{Q}[X]/\langle f(X) \rangle$$

Mixed fields

$$\text{Example: } K = \mathbb{Q}(\log(i), \pi, i) \;\cong\; \mathrm{Frac}(\mathbb{Q}[X_1, X_2, X_3]/I)$$
$$\text{where } I = \langle 2X_1 - X_2X_3,\; X_3^2 + 1 \rangle$$

General framework

- $K = \mathbb{Q}(a_1, \ldots, a_n), \quad a_1, \ldots, a_n \in \mathbb{C}$
  Field generated by extension numbers a_k

General framework

- $K = \mathbb{Q}(a_1, \ldots, a_n), \quad a_1, \ldots, a_n \in \mathbb{C}$
  Field generated by extension numbers a_k
- $R = \mathbb{Q}[X_1, \ldots, X_n]$
  Polynomial ring

General framework

- $K = \mathbb{Q}(a_1, \ldots, a_n), \quad a_1, \ldots, a_n \in \mathbb{C}$
  Field generated by extension numbers a_k
- $R = \mathbb{Q}[X_1, \ldots, X_n]$
  Polynomial ring
- $\mu : R \to \mathbb{C}, \quad X_k \mapsto a_k$
  Numerical embedding (evaluation homomorphism)

General framework

- $K = \mathbb{Q}(a_1, \ldots, a_n), \quad a_1, \ldots, a_n \in \mathbb{C}$
  Field generated by extension numbers a_k
- $R = \mathbb{Q}[X_1, \ldots, X_n]$
  Polynomial ring
- $\mu : R \to \mathbb{C}, \quad X_k \mapsto a_k$
  Numerical embedding (evaluation homomorphism)
- $I = \ker \mu = \{f \in R : f(a_1, \ldots, a_n) = 0\}$
  The ideal of all algebraic relations among $a_1, \ldots, a_n$ over $\mathbb{Q}$

General framework

- $K = \mathbb{Q}(a_1, \ldots, a_n), \quad a_1, \ldots, a_n \in \mathbb{C}$
  Field generated by extension numbers a_k
- $R = \mathbb{Q}[X_1, \ldots, X_n]$
  Polynomial ring
- $\mu : R \to \mathbb{C}, \quad X_k \mapsto a_k$
  Numerical embedding (evaluation homomorphism)
- $I = \ker \mu = \{f \in R : f(a_1, \ldots, a_n) = 0\}$
  The ideal of all algebraic relations among $a_1, \ldots, a_n$ over $\mathbb{Q}$
- $K_{\text{formal}} = \text{Frac}(R/I)$
  Formal field

General framework

- $K = \mathbb{Q}(a_1, \ldots, a_n), \quad a_1, \ldots, a_n \in \mathbb{C}$
  Field generated by extension numbers a_k
- $R = \mathbb{Q}[X_1, \ldots, X_n]$
  Polynomial ring
- $\mu : R \to \mathbb{C}, \quad X_k \mapsto a_k$
  Numerical embedding (evaluation homomorphism)
- $I = \ker \mu = \{f \in R : f(a_1, \ldots, a_n) = 0\}$
  The ideal of all algebraic relations among $a_1, \ldots, a_n$ over $\mathbb{Q}$
- $K_{\text{formal}} = \text{Frac}(R/I)$
  Formal field

Theorem: $K \cong K_{\text{formal}}$

General framework

- $K = \mathbb{Q}(a_1, \ldots, a_n), \quad a_1, \ldots, a_n \in \mathbb{C}$
  Field generated by extension numbers a_k
- $R = \mathbb{Q}[X_1, \ldots, X_n]$
  Polynomial ring
- $\mu : R \to \mathbb{C}, \quad X_k \mapsto a_k$
  Numerical embedding (evaluation homomorphism)
- $I = \ker \mu = \{f \in R : f(a_1, \ldots, a_n) = 0\}$
  The ideal of all algebraic relations among $a_1, \ldots, a_n$ over $\mathbb{Q}$
- $K_{\text{formal}} = \text{Frac}(R/I)$
  Formal field

Theorem: $K \cong K_{\text{formal}}$

Theorem: if $I = \langle f_1, \ldots, f_r \rangle$ is known, K is an effective field
(proof: Gröbner bases)

Ideally:

$$\mathbb{Q}(a_1, \ldots, a_n) \;\cong\; \mathrm{Frac}(\mathbb{Q}[X_1, \ldots, X_n]/I)$$

Ideally:

$$\mathbb{Q}(a_1, \ldots, a_n) \;\cong\; \mathrm{Frac}(\mathbb{Q}[X_1, \ldots, X_n]/I)$$

FATAL PROBLEM: we may not know the ideal I

Ideally:

$$\mathbb{Q}(a_1, \ldots, a_n) \;\cong\; \mathrm{Frac}(\mathbb{Q}[X_1, \ldots, X_n]/I)$$

FATAL PROBLEM: we may not know the ideal I

Theoretical reasons:

- $\mathbb{Q}(\pi) \cong \mathbb{Q}(X_1)$
- $\mathbb{Q}(e) \cong \mathbb{Q}(X_2)$
- Is $\mathbb{Q}(\pi, e) \cong \mathbb{Q}(X_1, X_2)$?
  (Open problem: Schanuel's conjecture.)

Ideally:

$$\mathbb{Q}(a_1, \ldots, a_n) \;\cong\; \mathrm{Frac}(\mathbb{Q}[X_1, \ldots, X_n]/I)$$

FATAL PROBLEM: we may not know the ideal I

Theoretical reasons:

- $\mathbb{Q}(\pi) \cong \mathbb{Q}(X_1)$
- $\mathbb{Q}(e) \cong \mathbb{Q}(X_2)$
- Is $\mathbb{Q}(\pi, e) \cong \mathbb{Q}(X_1, X_2)$?
  (Open problem: Schanuel's conjecture.)

Efficiency reasons:

- $\mathbb{Q}(a_1, \ldots, a_n)$ with many algebraic $a_k \to$ many, HUGE polynomials in I

Working with an incomplete ideal

Instead of computing I, compute some *reduction ideal* $I_{\text{red}} \subseteq I$:

$$\mathbb{Q}(a_1, \ldots, a_n) \overset{?}{\cong} \text{Frac}(\mathbb{Q}[X_1, \ldots, X_n]/I_{\text{red})}$$

Can use the map μ (numerical evaluation) as certificate of nonvanishing for given $z \in K$

Asymmetric zero test

Algorithm: test if $z = 0$ where $z \cong p/q$

1. [**Algebraic** $z = 0$ **test**]
   If $p \equiv 0 \bmod I_{\text{red}}$, return True.

2. [**Algebraic** $z \neq 0$ **test**]
   If $I_{\text{red}} = I$, return False.

3. [**Numerical** $z \neq 0$ **test**]
   Using ball arithmetic, compute an enclosure E with $\mu(p) \in E$.
   If $0 \notin E$, return False.

4. [**Iterate**]
   Attempt to find a new set of relations J with $J \subseteq I$, and set $I_{\text{red}} \leftarrow I_{\text{red}} \cup J$. Increase precision. Goto 1.

Failing gracefully

Successful $z = 0$ test:

```
>>> A = ca_mat([[pi, pi**2], [pi**3, pi**4]])
>>> A.det() == 0
True
```

Failing gracefully

Successful $z = 0$ test:

```
>>> A = ca_mat([[pi, pi**2], [pi**3, pi**4]])
>>> A.det() == 0
True
```

Successful $z \neq 0$ test:

```
>>> (A + (1 - exp(exp(-1000)))).det() == 0
False
```

Failing gracefully

Successful $z = 0$ test:

```
>>> A = ca_mat([[pi, pi**2], [pi**3, pi**4]])
>>> A.det() == 0
True
```

Successful $z \neq 0$ test:

```
>>> (A + (1 - exp(exp(-1000)))).det() == 0
False
```

Limits exceeded:

```
>>> (A + (1 - exp(exp(-10000)))).det() == 0
   ...
NotImplementedError: unable to decide predicate: equal
```

Ideal construction

Heuristics to construct I_{red}:

- Direct algebraic relations: $a_k \in \overline{\mathbb{Q}}$, $a_k = \sqrt{z}$, etc.
- Log-linear relations: $m_1 \log(a_1) + \ldots + m_k \log(a_k) = 0$
  - LLL gives basis matrix of potential relations
  - Verification through recursive computations in simpler fields
- Exp-multiplicative relations: $a_1^{m_1} \cdots a_k^{m_k} = 1$
- Functional equations: $\Gamma(z+1) = z\Gamma(z)$, etc.
- Other algebraic relations: resultants, Vieta's formulas, etc.

Elementary numbers

$\mathbb{E}$ = field generated by $+, -, \cdot, /, \exp, \log$
$\mathbb{L}$ = field generated by $+, -, \cdot, /, \exp, \log$, polynomial roots

Richardson's algorithm (extremely rough explanation): assuming Schanuel's conjecture, all relations arise from some combination of:

- Log-linear relations: $\log(ab) = \log(a) + \log(b) + 2\pi ik$
- Exp-multiplicative relations: $e^{a+b} = e^a e^b$
- Identical vanishing of algebraic functions: $\sqrt{\log(2)^2} - \log(2) = 0$ because $\sqrt{x^2} - x \equiv 0$ (on the local branch)

Very far from a complete implementation...

Some neat examples

$$\sqrt{5 + 2\sqrt{6}} = \sqrt{2} + \sqrt{3}$$

```
>>> sqrt(5 + 2*sqrt(6))
3.14626 {a where a = 3.14626 [Sqrt(9.89898 {2*b+5})], b =
    2.44949 [b^2-6=0]}
>>> sqrt(2) + sqrt(3)
3.14626 {a+b where a = 1.73205 [a^2-3=0], b = 1.41421 [b
    ^2-2=0]}

>>> sqrt(5 + 2*sqrt(6)) - sqrt(2) - sqrt(3)
0e-1126 {a-c-d where a = 3.14626 [Sqrt(9.89898 {2*b+5})],
    b = 2.44949 [b^2-6=0], c = 1.73205 [c^2-3=0], d =
    1.41421 [d^2-2=0]}
>>> sqrt(5 + 2*sqrt(6)) == sqrt(2) + sqrt(3)
True
```

Some neat examples

$$\frac{\log(\sqrt{2}+\sqrt{3})}{\log(5+2\sqrt{6})} = \frac{1}{2}$$

```
>>> log(5+2*sqrt(6))
2.29243 {a where a = 2.29243 [Log(9.89898 {2*b+5})], b =
    2.44949 [b^2-6=0]}

>>> log(sqrt(2)+sqrt(3))
1.14622 {a where a = 1.14622 [Log(3.14626 {b+c})], b =
    1.73205 [b^2-3=0], c = 1.41421 [c^2-2=0]}

>>> log(sqrt(2)+sqrt(3)) / log(5+2*sqrt(6))
0.500000 {1/2}
```

Some neat examples

$$i^i = \exp\left(\frac{\pi}{\left(\left(\sqrt{-2}\right)^{\sqrt{2}}\right)^{\sqrt{2}}}\right)$$

```
>>> i**i
0.207880 {a where a = 0.207880 [Pow(1.00000*I {b},
    1.00000*I {b})], b = I [b^2+1=0]}

>>> exp(pi / (sqrt(-2)**sqrt(2))**sqrt(2))
0.207880 {a where a = 0.207880 [Exp(-1.57080 {(-b)/2})],
          b = 3.14159 [Pi]}

>>> i**i - exp(pi / (sqrt(-2)**sqrt(2))**sqrt(2))
0
```

Some neat examples

$$4\,\mathrm{atan}\left(\frac{1}{5}\right) - \mathrm{atan}\left(\frac{1}{239}\right) = \frac{\pi}{4}$$

```
>>> 4*atan(ca(1)/5) - atan(ca(1)/239)
0.785398 + 0e-34*I {(a*c-4*b*c)/2 where a = 0e-35 +
    0.00836815*I [Log(0.999965 + 0.00836805*I {(239*c
    +28560)/28561})], b = 0e-34 + 0.394791*I [Log(0.923077
     + 0.384615*I {(5*c+12)/13})], c = I [c^2+1=0]}

>>> pi/4
0.785398 {(a)/4 where a = 3.14159 [Pi]}

>>> 4*atan(ca(1)/5) - atan(ca(1)/239) - pi/4
0
```

Some neat examples

$$\mathrm{erf}(e^{\pi i/3}) - \mathrm{erfc}(e^{-2\pi i/3}) = -1$$

$$\frac{\Gamma(\pi + 1)}{\Gamma(\pi)} = \pi$$

```
>>> erf(exp(pi*i/3)) - erfc(exp(-2*pi*i/3))
-1

>>> gamma(pi+1) / gamma(pi) == pi
True
```

Some neat examples

$$\log\left(\exp\begin{pmatrix}1 & 1\\1 & 2\end{pmatrix}\right) = \begin{pmatrix}1 & 1\\1 & 2\end{pmatrix}$$

```
>>> A = ca_mat([[1,1],[1,2]])
>>> A.exp()[0,0]
4.84921 {(-a*c+5*a+b*c+5*b)/10 where a = 13.7087 [Exp
    (2.61803 {(c+3)/2})], b = 1.46516 [Exp(0.381966 {(-c
    +3)/2})], c = 2.23607 [c^2-5=0]}
>>> A.exp().log()[0,0]
1
>>> A.exp().log() == A
True
```

But also limitations...

$$\log\left(\exp\begin{pmatrix}1 & -1\\ -1 & -1\end{pmatrix}\right) \stackrel{?}{=} \begin{pmatrix}1 & -1\\ -1 & -1\end{pmatrix}$$

```
>>> B = ca_mat([[1,-1],[-1,-1]])
>>> B.exp().log()[0,0]
1.00000 {(-b*d*f+b*e*f)/(2*c) where a = 1.41421 [Log
    (4.11325 {(c+d+e)/2})], b = -1.41421 [Log(0.243117 {(-
    c+d+e)/2})], c = 3.87013 [Sqrt(14.9779 {d^2+e^2-2})],
    d = 4.11325 [Exp(1.41421 {f})], e = 0.243117 [Exp
    (-1.41421 {-f})], f = 1.41421 [f^2-2=0]}
>>> B.exp().log() == B
Traceback (most recent call last):
  ...
NotImplementedError: unable to decide equality
```

Benchmark: naive DFT + zero test (times in seconds)

$$\mathbf{x} - \mathrm{DFT}^{-1}(\mathrm{DFT}(\mathbf{x})) = \mathbf{0}, \quad \mathbf{x} = (x_n)_{n=0}^{N-1}$$

x_{n-2}	N	Sage $\overline{\mathbb{Q}}$	Sage SR	SymPy	Maple	MMA	**Calcium**
n	8	0.018	0.11	1.1	0.0060	0.057	0.00016
	20	0.14	172	fail	0.13	0.96	0.00045
	100	8.2	fail	fail	9.1	> 60	0.044
$\sqrt{n}$	20	$> 10^3$	208	fail	1.1	2.3	0.064
	100	$> 10^3$	fail	fail	$> 10^3$	> 60	17
$\log(n)$	20	-	188	fail	0.74	45	0.043
	100	-	fail	fail	$> 10^3$	> 60	26
$e^{2\pi i/n}$	20	$> 10^3$	329	fail	fail	> 60	0.24
	100	$> 10^3$	fail	fail	$> 10^3$	> 60	86*
$\frac{1}{1+n\pi}$	20	-	219	fail	2.4	> 60	0.12
	100	-	fail	fail	$> 10^3$	> 60	202
$\frac{1}{1+\sqrt{n}\pi}$	8	-	0.76	22	0.074	2.6	0.072
	20	-	fail	fail	$> 10^3$	> 60	62

Practical implementation concerns

- Computing $I_{\text{red}} \subseteq I$: efficient algorithms, cost/benefit...
- Choosing extension numbers: $\mathbb{Q}(e^{a+b})$ vs $\mathbb{Q}(e^a, e^b)$, ...
- Ordering extension numbers: $e^\pi \succ \pi \succ i$
- Ordering monomials: lex, deglex, etc.
  - Cost of Gröbner basis computation, size of polynomials
- Normalizing fractions
  - Always remove content in $\mathbb{Q}[X_1, \ldots, X_n]$?
  - Rationalizing denominators

Non-canonical fractions

Problem: f, g reduced modulo I and coprime in $\mathbb{Q}[X_1, \ldots, X_n]$
$\quad\not\Longrightarrow \frac{f}{g}$ in canonical form

```
>>> a = exp(pi)
>>> b = exp(-pi)
>>> a*b
1
```

```
>>> a
23.1407 {a where ...}
>>> (a**3 - 2*a + b) / (a**2 + b**2 - 2)
23.1407 {(a^3-2*a+b)/(a^2+b^2-2) where ...}
```

```
>>> (a**3 - 2*a + b) / (a**2 + b**2 - 2) - a
0
```

Solutions and workarounds

- Always rationalize the denominator
  - Practical in simple cases
- Compute polynomial GCD over $\mathbb{Q}(\alpha)$ instead of $\mathbb{Q}$
  - Only applicable in some cases, potentially expensive
- General algorithm for simplifying or canonicalizing fractions modulo an ideal: Monagan and Pearce (2006)
  - Uses Gröbner bases over modules, potentially expensive
- Use algorithms that minimize divisions

Determinant of $A_{i,j} = \sqrt{i+j-1}, 1 \leq i,j \leq 5$

$$\mathbb{Q}(\sqrt{7}, \sqrt{6}, \sqrt{5}, \sqrt{3}, \sqrt{2}) \overset{?}{\cong} \text{Frac}(\mathbb{Q}[a,b,c,d,e]/\langle a^2-7, b^2-6, c^2-5, d^2-3, e^2-2, b-de \rangle)$$

Gaussian elimination:

```
(156829688*a*c*d*e-221693656*a*c*d+271638392*a*c*e-383986048*a*c
    +274164856*a*d*e-387945384*a*d+474865368*a*e-671936784*a+361353464*
    c*d*e-510531104*c*d+625886152*c*e-884270248*c+959654264*d*e
    -1358274640*d+1662163432*e-2352590040) / (18200*a*c*d*e-25732*a*c*d
    +31512*a*c*e-44565*a*c+324056*a*d*e-458284*a*d+561288*a*e-793807*a
    +847420*c*d*e-1198107*c*d+1467772*c*e-2075132*c+1068396*d*e
    -1511729*d+1850596*e-2618400)
```

Bareiss algorithm (fraction-free Gauss):

```
(-28*a*c*d*e+48*a*c*d+20*a*c*e-116*a*c+460*a*d*e-520*a*d+332*a*e-532*a
    +348*c*d*e-516*c*d-332*c*e+120*c+548*d*e-388*d+1660*e-2144) / (c*d
    -2*c+4*d*e-3*d-4)
```

Cofactor expansion or Berkowitz algorithm:

```
-4*a*c*d-20*a*c*e-24*a*c-4*a*d*e+8*a*d+136*a-28*c*d*e-116*c*d-88*c*e+64*
    c+112*d*e+164*d-60*e+244
```

Things to do

- Lots of basic implementation work
- Efficient Gröbner basis computation
- Better algorithms for dealing with fractions fields
- Better algorithms for algebraic number fields
- Implement more of Richardson's algorithm
- Better algorithms for real trigonometric functions, etc.
- Speed up integer relations
- Efficient extension $\mathbb{Q}(a_1, \ldots, a_{n-1}) \to \mathbb{Q}(a_1, \ldots, a_n)$
- Beyond elementary numbers: periods, D-finite numbers, multiple zeta values, ...

The end

No, thank *you*!