Three laws for good optional expression transformation

David R. Stoutemyer
dstout at hawaii dot edu

Abstract

*Default simplificatio*n means what a computer-algebra system does to a standard mathematical expression when the user presses ENTER, using factory-default mode settings, without enclosing the expression in an optional transformational function such as expand(...), factor(...), or simplify(...). The default simplification of most computer algebra systems typically does only transformations that are always fast, don't significantly increase the size of any sub-expression, and are acceptable to most users most of the time. Examples include sorting factors and terms, collecting similar factors and terms, doing arithmetic, exploiting symmetry or anti-symmetry such as $\sin(-x) \rightarrow \sin x$, and applying identities such as $u + 0 \rightarrow u$, $1u \rightarrow u$ and $u^1 \rightarrow u$.

These transformations alone are not sufficient to guarantee that even mere polynomial expressions don't have superfluous variables or misleadingly large apparent degree is some of their variables. That is why such systems usually have optional simplify(...) functions that tentatively try more transformations. Moreover, users often want a specific form of a result. Consequently, most systems typically provide numerous optional control variables and transformation functions with various optional arguments for expanding, factoring, reduction over a common denominator, various trigonometric transformations, etc. There is usually a steep learning curve in mastering the use of all these functions, and they never seem to be sufficient to provide the sort of fine control that would often be preferred, such as expanding with respect to some variables, but having the coefficients of those variables collected and factored over a common denominator with respect to other variables, or *vice versa*.

This article describes the idea of an interactive general-purpose wizard for organizing optional transformations and allowing even novices such fine grain control over the desired form of the resulting expression.

1 Introduction

An earlier article [19] provides ten goals for the design and improvement of *default simplification*. The most important of these goals is that default simplification should return a *candid* result, meaning a result that visibly manifests the simplest class to which the function specified by the expression belongs. For example, the result should contain no superfluous variables; and if it is equivalent to a polynomial, then it should simplify to one whose degree is as low as possible in every variable. For reasons of speed and not rudely transforming the input drastically more than requested, default simplification should ideally transform the input as little as is necessary for this goal and associated goals, such as not contracting the domain of definition. Algorithms were described that accomplish such flexible but candid simplification at least for rational expressions, returning a result that

is somewhere in the spectrum between fully factored over a common denominator and complete multivariate partial fractions. Additional heuristics were provided to strive for candidness with expressions containing fractional powers and elementary functions.

In contrast, this sequel describes three design principles for organizing *optional transformations* in easily-used but powerful and flexible combinations.[1]

Sections 2.1 through 2.6 describe useful categories of alternative forms. Section 3 describes a dialog-driven wizard that makes it easy for even novice users to request combinations of the various transformations. Section 4 describes the idea of having default simplification automatically return several alternative forms as it computes them.

2 Alternate forms for expressions

2.1 Alternate forms for the rational aspect of expressions

Definition: A *functional form* is an expression of the form

$$f\left(expression_1, expression_2, \ldots\right)$$

where f is any function name.

Definition: A *generalized variable* is either a variable or a functional form,

Although specific functional forms often have special transformation rules, such as

$$\begin{aligned} \sin\left(\pi\right) &\rightarrow 0, \\ \sin\left(x\right)^2 + \cos\left(x\right)^2 &\rightarrow 1. \end{aligned}$$

Functional forms that survive such transformations can be treated similar to variables for purposes such as combining similar terms and factors. For example, $2\sin\left(x\right) + 3\sin\left(x\right) \rightarrow 5\sin\left(x\right)$. This is true not only for functional forms, but also for their arguments, which recursively might contain other functional forms. This subsection discusses only rational expressions, but the ideas also apply recursively to rational expressions of generalized variables and of fractional powers of generalized variables.

2.1.1 The spectrum from fully factored through complete partial fractions

The following observations enable us to unify factoring, common denominators, polynomial expansion and partial fraction expansion into a helpful single spectrum:

1. Making a common denominator can be regarded as factoring out the least common denominator.

2. Polynomial expansion can be regarded as a special case of expanding the polynomial part of a partial fraction expansion.

[1] The title was inspired by the original three *laws* of mechanics, thermodynamics and robotics (there are now more than three of each). However, "design *principles*" is a more accurate phrase.

3. For factorizations there are commonly named variants for some points on the spectrum, such as

   (a) primitive, which implies a common denominator because the gcd of two ratios is the gcd of their numerators divided by the least common multiple of their denominators,

   (b) square-free,

   (c) distinct-degree,

   (d) over the integers,

   (e) over the Gaussian integers,

   (f) over algebraic extensions, and

   (g) over the floating-point complex numbers.

4. The same adjectives can be used to specify the amount of denominator factorization for partial-fraction expansion.

2.1.2 Polynomial shifts

Merely re-expressing a polynomial in terms of shifted variables can greatly reduce the number of terms and/or the size of coefficients. For example,

$$\left(y^2 - 4y + 4\right) x^3 + \left(3y^2 - 12y + 12\right) x^2 + \left(3y^2 - 12y + 12\right) x + y^2 - 4y + 1 \\ \rightarrow (y-2)^2 (x+1)^3 + 8$$

References [8] and [9] give algorithms for computing optimal shifts.

2.1.3 Polynomial decomposition

Complementary to such shift decompositions, Kozen and Landau [12] describe an efficient algorithm for completely decomposing a univariate polynomial $p(x)$ into nested polynomials

$$p_1 \left(p_2 \left(\dots \left(p_m \left(x\right)\right) \cdots\right)\right)$$

with each $p_k\left(t\right)$ of degree at least 2 in t. For example, the irreducible polynomial

$$\begin{aligned} P(x) &:= x^{12} + 4x^{10} + x^9 + 6x^8 + 3x^7 + 4x^6 + 3x^5 + x^4 + x^2 + 7 \\ &\rightarrow \left(x^3 + x\right)^4 + \left(x^3 + x\right)^3 + 7. \end{aligned}$$

Their algorithm also applies recursively to multivariate polynomials represented recursively. There are also algorithms for other kinds of univariate and multivariate polynomial decompositions, as described, for example, by Klüners [11] or von zer Gathen and Weiss [24].

Polynomials rewritten in these ways can reveal significant structure, help precondition an expression for efficient repeated numeric evaluation or reduced rounding error, and facilitate solutions of higher-degree polynomial equations or systems of equations. For example, with the above decomposition, masochists could apply the quartic formula then the cubic formula to express the zeros of $P(x)$ in terms of radicals.

The motivation for seeking such decompositions are similar to those for seeking optimal shifts. Therefore these two types of special forms could be combined into a single optional transformation.

2.1.4 Linear combinations of powers

At least since Pythagoras people have been interested in representing numbers and non-numeric expressions as sums, differences, or general linear combinations of powers of other expressions. For example, if an expression can be rewritten as a sum of positive even powers of real expressions, then the expression is thereby proven to be nonnegative for all real values of all variables therein.[2] As another example, students are often asked to transform an equation such as

$$8x^2 + 128x + 15y^2 - 300y + 1772 = 0$$

into standard form, which turns out to be

$$\frac{1}{30}(x+8)^2 + \frac{1}{16}(y-10)^2 = 1.$$

This transformation usefully reveals that the equation describes an ellipse centered at $(-8, 10)$ with semi-major axis length $\sqrt{30}$ and semi-minor axis length 4, parallel to the x and y axes respectively.

Also, powers often have meaningful physical interpretations, such as kinetic, potential, capacitive or inductive energy components.

Examples of algorithms can be found in, for example, in Delzell [7], Reznick [17] or the aptly named Powers and Wörmann [16]. The special case of sums of even powers is associated with semi-definite programming [23]. The literature on quadratic forms is also relevant.

2.1.5 Expression in terms of symmetric or orthogonal polynomials

A change of basis from ordinary monomials to symmetric polynomials or orthogonal polynomials can

- yield more concise results,
- yield results less subject to magnified rounding errors,
- reveal patterns that otherwise wouldn't be apparent, or
- suggest truncated approximations.

Sturmfels [21] contains algorithms for transforming to and from a basis of symmetric polynomials. Transformation from a linear combination of orthogonal polynomials can be done by mere substitution or more efficiently using well-known two-term recurrences. Transformation from monomials to orthogonal polynomials can be done by solving systems of linear equations for unknown coefficients.

2.1.6 Rational Decomposition

Decomposition of univariate and multivariate rational expressions into nested rational expressions is discussed in [3, 10, 28]. As an example from the first of these articles, but using recursive form

[2] Although such a decomposition is *sufficient* for non-negativity, it isn't *necessary*, because $z^6+x^4y^2+x^2y^4-3x^2y^2z^2$ is nonnegative over the reals but can't be represented as a sum of squares of polynomials. [18]

and primitive normalization, the ratio

$$\frac{\left(y^2+2z^2y+z^4-81\right)x^2-2y\cdot\left(y^5+z^2y^4+225z\right)x+y^2\left(y^8-625z^2\right)}{\left(y^2+2z^2y+z^4-162\right)x^2-2y\cdot\left(y^5+z^2y^4+450z\right)x+y^2\left(y^8-1250z^2\right)}$$

$$\rightarrow\begin{cases}1 & \text{if } 9x+25zy=0,\\ \dfrac{\left(\dfrac{\left(y+z^2\right)x-y^5}{9x+25zy}\right)^2-1}{\left(\dfrac{\left(y+z^2\right)x-y^5}{9x+25zy}\right)^2-2} & \text{otherwise.}\end{cases}$$

This example also illustrates that rational decomposition can introduce new removable singularities that aren't removed in the nested form, such as on the manifold $9x+25zy=0$ for this example. We can avoid this by clearing the nested denominators but preserving the nested polynomial components thereof to obtain correlated polynomial decompositions of the numerator and denominator:

$$\frac{\left(\dfrac{\left(y+z^2\right)x-y^5}{9x+25zy}\right)^2-1}{\left(\dfrac{\left(y+z^2\right)x-y^5}{9x+25zy}\right)^2-2}\rightarrow\frac{\left(\left(y+z^2\right)x-y^5\right)^2-\left(9x+25zy\right)^2}{\left(\left(y+z^2\right)x-y^5\right)^2-2\left(9x+25zy\right)^2}.$$

If desired, for this example we can then further factor the difference in two squares in the numerator and denominator to obtain the algebraic factorization

$$\frac{\left(\left(y+z^2+9\right)x-y^5+25zy\right)\left(\left(y+z^2-9\right)x-y^5-25zy\right)}{\left(\left(y+z^2+9\sqrt{2}\right)x-y^5+25\sqrt{2}zy\right)\left(\left(y+z^2-9\sqrt{2}\right)x-y^5-25\sqrt{2}zy\right)}.$$

The numerator factorization could easily have been computed from the original numerator. However, the required $\sqrt{2}$ algebraic extension necessary to factor the multivariate denominator would be more difficult to determine without the intervening rational decomposition.

2.1.7 Continued fractions

Continued fractions are another type of compound-fraction representation for rational expressions. Acton [1] lists three different variants of continued fractions together with algorithms for converting between them and a reduced ratio. Cuyt and Verdonk [6] review methods for multivariate continued fractions. Here is an example of one kind of continued fraction expansion that reveals a simple pattern:

$$\frac{\left(z^4-105z^2+945\right)z}{15\left(z^4-28z^2+63\right)}\mid z^2\notin\left\{63,\frac{45}{2},\frac{3}{2}\left(35\pm\sqrt{805}\right)\right\}\rightarrow\cfrac{z}{1-\cfrac{z^2}{3-\cfrac{z^2}{5-\cfrac{z^2}{7-\cfrac{z^2}{9}}}}}.$$

Here a constraint was appended to the input to avoid the appearance of contracting the domain of definition because of removable singularities introduced by the continued fraction. If instead we

used a piecewise result, then it would have 9 pieces, 8 of which are constants that can be determined by substituting the two square roots of each element in the constraint set into the original expression.

Actually, if the computer algebra automatically handles unsigned zeros and infinities correctly, then with exact computation the continued fraction form evaluates to the correct finite values even at these removable singularities. However, unlike the original expression, the continued fraction form might be subject to catastrophic cancellation near those introduced singularities. Therefore it is worth alerting the user to these singularities by either appending an input constraint or producing a piecewise result.

2.1.8 Hornerized forms

In its simplest form, Horner's rule is the factoring out of the least power of a variable from successive terms. For example,

$$-1234321x^7 - 1234321x^5 + 2468642x^4 + 7x^3 + 14x - 21 \\ \rightarrow ((((-1234321x^2 - 1234321)x + 2458642)x + 7)x^2 + 14)x - 21.$$

It is also worth partially factoring out units and numeric content to the extent that it reduces bulk or the number of operations. For example,

$$-1234321x^7 - 1234321x^5 + 2468642x^4 + 7x^3 + 14x - 21 \\ \rightarrow ((-1234321((x^2 + 1)x - 2)x + 7)x^2 + 14)x - 21.$$

Horner's rule often leads to faster evaluation when substituting numbers for variables, which is particularly important in situations such as plotting, where substitution is done many times for different values. Horner's rule also often improves accuracy for approximate arithmetic, because the operands of a catastrophic cancellation are closer to the input numbers, hence less contaminated with rounding error.

Horner's rule can be viewed as factoring out term content term by term, starting with the highest-degree terms at each level. With this viewpoint, we can apply it to multinomials throughout an expression in all of the above special forms. Ceberio and Kreinovich [5] discuss greedy algorithms for computing efficient multivariate Hornerized forms.

Another transformation that can enable faster evaluation when substituting numbers for variables is to factor out an integer common divisor of the exponents from a product of powers. For example, if the powers are done with the help of repeated squaring, then

$$(y + 3)^6 x^4 \rightarrow \left((y + 3)^3 x^2\right)^2$$

uses only five multiplications rather than six.

2.2 Alternate forms for irrational expressions

As *examples* of alternate forms for irrational expressions, this section discusses only fractional powers and trigonometric functions.

2.2.1 Alternate forms for fractional powers

Good default simplification automatically does transformations of fractional powers that users almost always want, such as denesting radicals – at least those that match simple patterns and therefore don't require extensive time to test for applicability, such as the rewrite rule

$$\sqrt{u \pm \sqrt{v}} \mid u > 0 \ \wedge \ u^2 - v > 0 \ \wedge \ u^2 - v \text{ is a perfect square } \rightarrow \sqrt{\frac{u + \sqrt{u^2 - v}}{2}} \pm \sqrt{\frac{u - \sqrt{u^2 - v}}{2}}.$$

For example,

$$\begin{aligned} \sqrt{5 + 2\sqrt{6}} \ &\rightarrow \ \sqrt{\frac{5 + \sqrt{5^2 - 4 \times 6}}{2}} + \sqrt{\frac{5 - \sqrt{5^2 - 4 \times 6}}{2}} \\ &\rightarrow \ \sqrt{3} + \sqrt{2}. \end{aligned}$$

However, that still leaves numerous alternative forms for fractional powers, depending on personal taste and the purpose of the result. For example, Albert Rich has graciously provided Table 1 containing some alternate forms of a particular number having the form p^q, where p and q are rational numbers with p positive and not a perfect power.[3] (We can always avoid p being a perfect power by $(p^n)^{\hat{q}} \rightarrow p^{n\hat{q}} \rightarrow p^q$).

So many alternatives is an embarrassment of riches, but if users could optionally see and compare all of the good alternatives relevant to their number, then they might strongly prefer one that wouldn't have occurred to them. Of course in many cases, such as $1/\sqrt{2}$, there are significantly fewer distinct alternatives.

The choice of which alternative to display by default is primarily an aesthetic issue, but perhaps most people would most often prefer one of the alternatives that entails only a single fractional power. Of the three such alternatives in Table 1, the last row is probably most often preferred because the denominator is rationalized and the one radicand is the smallest possible such integer.

Good default simplification should simplify the difference of any of these two alternatives to 0, but many systems don't. For example, do your computer algebra systems automatically simplify

$$2\,\frac{98^{1/3}}{15^{2/3}} - \frac{2}{15}\,1470^{1/3} \ \rightarrow \ 0?$$

The easiest way to implement this 0 recognition is to transform all irrational absurd numbers to a canonical form *internally*. The prime decomposition has the advantage that subsequent multiplication requires only adding exponents of similar primes. Various co-prime representations have the advantage of requiring only gcds rather than more costly integer factorizations. The *Derive* internal form $rational_1^{rational_2}$ has the advantage that rational numbers are a special case, with $rational_2$ being an implicit 1. Addition and subtraction is easier when the rational part is factored out, which requires only verifying that the irrational factors are identical, then adding the rational factors.

[3] He calls such numbers *absurd numbers* because *ab* means "from" in Latin, and because of the tradition started with whimsical nomenclature such as imaginary numbers, radicals, irrational numbers, and surreal numbers.

Table 1: Some alternative forms for $\left(\frac{784}{225}\right)^{1/3}$

Ratio	Product form	Description of the result form
		One or more factors not in $\mathbb{Q}$ or $\mathbb{Z}$
$\frac{2^{4/3}\,7^{2/3}}{3^{2/3}\,5^{2/3}}$	$2^{4/3}\,3^{-2/3}\,5^{-2/3}\,7^{2/3}$	primes raised to rationals
$\frac{2^{4/3}\,7^{2/3}}{15^{2/3}}$	$2^{4/3}\,7^{2/3}\,15^{-2/3}$	*co-prime integers* raised to *distinct* rationals
	$2^{4/3}\left(\frac{7}{15}\right)^{2/3}$	*co-prime rationals* raised to *distinct positive* rationals
$\frac{784^{1/3}}{15^{2/3}}$	$15^{-2/3}\,784^{1/3}$	ratio of two *co-prime integers* raised to *positive* rationals
	$\left(\frac{784}{225}\right)^{1/3}$	$rational_1^{rational_2}$, which is how *Derive* stores it *internally*
		Integer times 1 or more factors not in $\mathbb{Q}$ or $\mathbb{Z}$
$\frac{2\cdot 2^{1/3}\,7^{2/3}}{3^{2/3}\,5^{2/3}}$	$2\cdot 2^{1/3}\,3^{-2/3}\,5^{-2/3}\,7^{2/3}$	times *primes* raised to fractions in $(-1, 1)$
$\frac{2\cdot 2^{1/3}\,7^{2/3}}{15^{2/3}}$	$2\cdot 2^{1/3}\,7^{2/3}\,15^{-2/3}$	times *co-prime integers* raised to *distinct* fractions in (-1, 1)
	$2\cdot 2^{1/3}\left(\frac{7}{15}\right)^{2/3}$	[4]times *co-prime rationals* raised to *distinct* fractions in $(0, 1)$
$\frac{2\cdot 98^{1/3}}{15^{2/3}}$	$2\cdot 15^{-2/3}\,98^{1/3}$	times ratio of *co-prime integers* raised to fraction in $(0, 1)$
	$2\left(\frac{98}{225}\right)^{1/3}$	times fraction in $(0, 1)$ raised to fraction in $(0, 1)$
		Reciprocal times 1 or more factors not in $\mathbb{Q}$ or $\mathbb{Z}$
	$\frac{1}{15}\,2^{4/3}\,3^{1/3}\,5^{1/3}\,7^{2/3}$	times *primes* raised to *positive* rationals
	$\frac{1}{15}\,2^{4/3}\,7^{2/3}\,15^{1/3}$	times *co-prime integers* raised to *distinct* positive rationals
	$\frac{1}{15}\,11760^{1/3}$	times *integer* raised to *positive* rational
		Fraction times one or more factors not in $\mathbb{Q}$ or $\mathbb{Z}$
	$\frac{2}{15}\,2^{1/3}\,3^{1/3}\,5^{1/3}\,7^{2/3}$	times *primes* raised to fractions in $(0, 1)$
	$\frac{2}{15}\,7^{2/3}\,30^{1/3}$	times *co-prime integers* raised to *distinct* fractions in $(0, 1)$
	$\frac{2}{15}\,1470^{1/3}$	times integer raised to fraction in $(0, 1)$, as *Derive displays*

Good default simplification should also automatically simplify *non-real* numbers such as

$$\frac{(-1)^{1/8}\sqrt{1+i}}{2^{3/4}} + ie^{i\pi/2} \quad \to \quad -\frac{1}{2} + \frac{i}{2}.$$

However, many systems don't fully simplify this example because they don't automatically resolve mixtures of rectangular form $a + bi$, unit polar form $r\,(-1)^{m/n}$, and exponential polar form $re^{i\theta}$. (Try this example on your computer algebra systems.)

Which of rectangular, exponential polar and unit polar form is most attractive depends on the particular number. For example, compare

$$\begin{array}{rcccc} \mathbf{7+5}\boldsymbol{i} & = & (-1)^{\arctan(5/7)/\pi} & = & \sqrt{74}e^{\arctan(5/7)i}, \\ -2\sin\left(\frac{3\pi}{14}\right) + 2\cos\left(\frac{3\pi}{14}\right)i & = & \mathbf{2(-1)^{5/7}} & = & 2e^{5\pi i/7}, \\ 2\cos(1) + 2\sin(1)\,i & = & 2\,(-1)^{1/\pi} & = & \mathbf{2}\boldsymbol{e}^{\boldsymbol{i}}. \end{array}$$

Default simplification should probably use the form that is most concise for each non-real number in a result. However, optional transformations should conveniently offer all three alternatives.

The alternative forms for p^q exemplified in Table 1 are also relevant when p is a non-numeric rational expression and a factor of p always nonnegative. For example, we can safely distribute a fractional power over two factors if at least one of the factors is always nonnegative. However, we often won't be able to determine such nonnegativity. Moreover, we must take care not to introduce a removable singularity, thus making the expression undefined where the input was defined. For example, the transformation

$$\frac{1}{\sqrt{u}} \quad \stackrel{?}{\to} \quad \frac{\sqrt{u}}{u}$$

makes the right side undefined at $u = 0$, whereas the left side is complex infinity at $u = 0$. Thus we can't be as cavalier about operations such as rationalizing denominators when those denominators might not be positive. In fact, simplifying expressions of the form $w^\alpha \left(w^\beta\right)^\gamma$ with rational α, β and γ is currently handled poorly and inconsistently by most widely-used computer algebra systems, as reported in [20].

2.2.2 Alternate forms for trigonometric functions

Good default simplification automatically transforms products and ratios of integer powers of expressions of the form $\sin u$, $\cos u$, $\tan u$. $\cot u$, $\csc u$ and $\sec u$ to a good form. For example,

$$\frac{\sin z}{\tan z} \to \cos z,$$

which in this case has the extra benefit of removing the removable singularity at $z = 0$.[5] However, users sometimes need a different explicit form. Therefore it is helpful to offer optional transforms between these functions.[6] Other generally-useful transformations of trigonometric sub-expressions include

[5] To indicate the enlargement of the domain of definition for those who care, we could optionally have the simplified result be $\cos z \mid z \neq 0$, but most computer algebra systems don't currently offer that option.

[6] Tables of csc, sec and cot usefully replaced a division with a multiplication in the days of manual computation. However, those days are long gone, and most students quickly forget even whether csc is the reciprocal of cos or sin. Therefore we should stop wasting valuable curriculum time torturing students with these secondary trigonometric functions and their hyperbolic counterparts, just as we no longer torture students with other trivially related trigonometric functions such the haversine, which was convenient for manual celestial navigation computations.

1. multiple-angle and angle sum expansion,

2. transformation of integer powers and products of sinusoids to sinusoids of integer linear combinations of angles (the inverse of transformation 1),

3. transformation of sums and differences of of power products of trigonometric functions into factored form with each factor containing only one trigonometric function, such as the *Mathematica* example

   $\mathsf{In}[1] := \texttt{TrigFactor}[\texttt{Cos[x]}^3 + 3\,\texttt{Cos[x]}^2\,\texttt{Sin[x]} + 3\,\texttt{Cos[x]}\,\texttt{Sin[x]}^2 + \texttt{Sin[x]}^3]$

   $\mathsf{Out}[1] = 2\sqrt{2}\,\texttt{Sin}[\frac{\pi}{4} + \texttt{x}]^3,$

   which is useful for computing limits and solving equations.

4. transformation of sinusoids to complex exponentials,

5. transformation to tangents of half angles or cotangents of half angles,[7]

6. transformation of sine and cosine of half angles.[8]

7. transformation of trigonometric functions of rational multiples of π to (perhaps nested) radicals of real numbers to the extent that it is possible, as described in [25, 26].[9]

Principle 1 (useful optional transformations)**.** Optional transformations should include a broad variety of useful candid forms such as named points on the spectrum from fully factored over a common denominator through complete multivariate partial fractions, decompositions, continued fractions, linear combinations of perfect powers, most useful transformations for irrational sub-expressions, and Hornerized forms.

2.3 Concise candid form

Carette [4] uses the theory of minimum description length to define a measure that permits us to decide which of several equivalent expressions are most concise.

As another optional transformation, it would be nice to have an algorithm that could determine at acceptable cost the most concise candid forms.

The *Mathematica* Simplify [...] and FullSimplify [...] functions have this goal, as does the Maple simplify (...) function.

In principle, for each possible ordering of variables one could produce a complete partial fraction expansion with respect all variables, then try combining all combinations of fractions, with fully factored numerators and denominators and all polynomially-expanded combinations thereof, to

[7] If this introduces new removable singularities in the default or declared intervals of the variables, then corresponding constraints should be appended to the input.

[8] If this requires a $\operatorname{sign}(\sin(\ldots/2))$ or $\operatorname{sign}(\cos(\ldots/2))$ factor on account of insufficiently constrained angles, then that factor should be included, which of course makes the transformation rather useless because the result then contains the input as a proper sub-expression.

[9] For most systems, default simplification automatically does this in at least *some* of the cases where the result has only one or two unnested radicals, such as $\sin(\pi/3) \to \sqrt{3}/2$ and perhaps also $\sin(\pi/12) \to \left(\sqrt{6} - \sqrt{2}\right)/4$.

fully explore the fully-factored to fully-expanded spectrum. Fractional powers and functional forms would entail additional alternative forms. In principle, for each of these alternatives one could also explore all combinations of all implemented optional shifts, decompositions, continued fractions, and Hornerizations, etc. However, the computing time is likely to be prohibitive except for rather small expressions.

A more practical approximation to this exhaustive search would be to compute the fully expanded and fully factored extremes of the spectrum, then use greedy local heuristics to combine fractions and to expand factors only when it seems likely to give a more concise result, backtracking where it doesn't. The best forms thus obtained along the main spectrum could then be checked for shifts, decompositions, continued fractions, Hornerizations, etc.

As examples of such local greedy heuristics:

- Expanding integer powers of multinomials almost always increases the number of terms and coefficient sizes.
- Distributing multinomials across each other increases the number of terms and coefficient sizes if the numeric coefficients therein all have the same sign.
- Distributing multinomials across each other increases the number of terms if the multinomials have disjoint sets of variables.

Principle 2 (A concise-result capability)**.** Optional transformations should include a facility that attempts to produce a particularly concise form.

2.4 Series and other approximations

Closed-form exact results aren't always obtainable. Even when they are, the results might be too bulky to convey needed insight or to permit fast enough well-conditioned evaluation for numerous floating-point values of the variables therein. Therefore, various kinds of approximation are useful transformations. For example,

- Quadrature can often be used to determine a single approximate number for a definite integral.
- Approximate equation solving is often applicable when compact explicit exact solutions are unobtainable.
- Truncated power series expanded about 0 and ∞ can be used to display quickly and concisely the lowest degree and highest degree terms of a polynomial whose expanded form is excessive.
- Generalized infinite or truncated series (allowing, for example, logarithmic factors) can approximate non-polynomial expression within their region of convergence.
- Padé approximations often often have a larger region of convergence and greater computational efficiency than power series.
- Truncated Fourier or wavelet series are often more appropriate than expansions about a point.

2.5 Control over the order of indeterminates

"*Order is the shape upon which beauty depends*"
– Pearl S. Buck

Users often feel strongly about the order of indeterminates within monomials and the order of monomials within multinomials.

Often indeterminates in an expression partition into categories, such as:

1. indeterminates of major interest, such as integration variables or solution variables,
2. indeterminates representing known numeric constants, such as c for the speed of light, and
3. indeterminates representing coefficients or other such parameters for which values can later be substituted, such as a, b and c in the expression $ax^2 + bx + c$.

Moses [15] discusses the ordering conventions used in mathematics when indeterminates have no particular physical meaning: Alphabetic order is often preferred within monomials. The order of monomials within multinomials is often descending-degree lexicographic,[10] with the indeterminates of major interest considered before other indeterminates, alphabetically within these two categories. Mathematicians often use letters near the end of the alphabet for indeterminates of major interest and letters near the beginning of the alphabet for all other indeterminates. This induces a *rotated* alphabet for the lexicographic aspect of ordering monomials within multinomials:

$$\cdots \succ x \succ y \succ z \succ a \succ b \succ c \succ \cdots .$$

For example, $ax^2 + bxy + cy^3 + a^9$ complies with these ordering rules. The boundary between the indeterminates of major interest and other indeterminates is necessarily flexible over mathematics literature.

The *Derive*® computer algebra system attempts to partially accommodate the conventions of mathematics by rotating the alphabet between w and x. The computer algebra in the Texas Instruments products instead rotates the alphabet between q and r.

Although better than pure alphabetical order or order based on less transparent causes such as hashing or order of first occurrence during a session, no such rotated alphabet with or without a fixed boundary can satisfy all mathematicians all of the time. Consequently, when an input to the TI computer algebra system implies that a particular variable is of particular interest, then that variable is made to be the most main variable. For example,

$$\begin{aligned} \int ax^2 + \boldsymbol{b}x \, d\boldsymbol{b} &\rightarrow \frac{\boldsymbol{b^2}x}{2} + \boldsymbol{b}x^2 a, \qquad \text{with } b \succ x \succ a; \\ \text{expand}\left((x+y+z+1)^2, \boldsymbol{y}\right) &\rightarrow \boldsymbol{y^2} + 2\boldsymbol{y}\,(x+z+1) + (x+z+1)^2 \qquad \text{with } y \succ x \succ z. \end{aligned} \tag{1}$$

However, such local ordering is volatile. For example, if we assign

$$\text{expand}\left((x+y+z+1)^2, \boldsymbol{y}\right)$$

[10] The next most common order outside the Gröbner basis literature is probably total degree with ties broken lexicographically. Ascending degree is more common for power series variables.

to a variable w then enter $w-(x+z+1)^2$, we obtain

$$2xy+y^2+2y\,(z+1)\,,$$

in which the order of indeterminates is the default $x \succ y \succ z$ rather than the temporary $y \succ x \succ z$ displayed in (1).

In *Derive* this temporary ordering more generally applies to a sequence of variables, such as for

$$\text{expand}\left(\left(x+y+z+1\right)^2, \boldsymbol{y}, \boldsymbol{z}\right) \rightarrow \boldsymbol{y^2}+2\boldsymbol{yz}+\boldsymbol{y}\,(2x+2)+\boldsymbol{z^2}+\boldsymbol{z}\,(2x+2)+x^2+2x+1,$$

in which the temporary order of indeterminates is $y \succ z \succ x$.

In comparison to pure alphabetical order for both factors and terms, the default rotated alphabetical ordering for terms, overridden by making indeterminates that are arguments of certain functions be more main than any others produces results that are more often closer to user's preferences.

However, user's input often contains no function that indicates certain variables are of particular interest, and there are many exceptions to the above rules for mathematical expressions when the variables have no particular physical meaning. For example, anyone familiar with Einstein's energy-mass equivalence equation or Newton's definition of force is likely to find the right sides of the equations

$$\begin{aligned} E &= c^2 m, \\ f &= am \end{aligned}$$

visually quite disturbing despite their compliance with the usual mathematics ordering convention. Consequently, optional transformations should include control over the ordering of indeterminates.

2.6 Different forms for different generalized variables

"*To each his own*"
– Cicero

"*I got different strokes for different folks*"
– Muhammad Ali

Often we want certain transformations such as expansion or factoring only with respect to certain indeterminates. For example:

- To compute the integral of $\left(x^5+(c+1)^{999}\,x+1\right)^2$ with respect to x, it is helpful to expand with respect to x, but foolish to expand with respect to c.
- To solve
$$\left(c^{999}-1\right)\left(z^2-1\right)=0 \mid c^{999} \neq 1$$
it is helpful to factor with respect to z, but foolish to factor with respect to c.

In these cases we would prefer mere default simplification with respect to c.

Moreover, when we expand with respect to a proper subset of the indeterminates, it is often helpful to collect monomials that are similar in those indeterminates, forming coefficients that are generally polynomials in the remaining indeterminates. For example, if a user requests expansion of

$$(cxy + cx + x + c + 1)^2 \tag{2}$$

with respect to x and y with $x \succ y$, then the user would probably prefer

$$c^2x^2y^2 + 2c\,(c+1)\,x^2y + \left(c^2 + 2c + 1\right)x^2 + 2c\,(c+1)\,xy + 2\,(c+1)^2\,x + c^2 + 2c + 1 \tag{3}$$

to the fully expanded

$$c^2x^2y^2 + 2c^2x^2y + 2cx^2y + c^2x^2 + 2cx^2 + x^2 + 2c^2xy + 2cxy + 2c^2x + 4cx + 2x + c^2 + 2c + 1.$$

The coefficients that are polynomials in c in result (3) are the happenstance result of default simplification in *Derive*. For consistency and conciseness, the user might prefer to request that expression (2) be expanded with respect to x and y, with coefficients that are factored over $\mathbb{Z}$ with respect to c, giving

$$c^2x^2y^2 + 2c\,(c+1)\,x^2y + (c+1)^2\,x^2 + 2c\,(c+1)\,xy + 2\,(c+1)^2\,x + (c+1)^2\,.$$

This can be regarded as an expanded distributed representation with respect to the two most main variables, but with a factored recursive representation of the coefficients, which are polynomials in c, factored over $\mathbb{Z}$. However, even though expanding with respect to both x and y, the user might also prefer a recursive representation for them too for greater conciseness and more efficient substitution of numbers:

$$(cy + c + 1)^2\,x^2 + 2(c+1)(cy + c + 1)x + (c+1)^2\,.$$

Thus for maximum flexibility:

1. The user should be able to choose the order of variables.
2. For each variable the user should be able to choose default simplification, a concise form, or a named point on the fully factored over a common denominator through complete partial fraction expansion, or a truncated series approximation.
3. For each consequent ratio the user should be able to choose rational decomposition or various continued fraction forms if desired.
4. For each consequent multinomial the user should be able to choose optimal shifts, decomposition, a linear combination of perfect powers, transformation to symmetric or orthogonal polynomials, approximation, and Hornerization if desired.

3 A transformation wizard and function

Most computer algebra systems provide optional transformations via specific functions named, for example, factor, expand, decompose, Hornerize, etc. Some computer algebra systems also provide global control variables whose settings control what optional transformations are done by default. For example, Maxima has both a trigexpand function and a trigexpand variable.

An advantage of a control variable is that several complementary transformations can be requested to occur at every opportunity, which can be impossible to request in full generality by finitely nesting function invocations. For example, with trigreduce(factor(...)), there might be additional opportunities for factoring after trigreduce is finished. Thus to be more thorough we should use factor(trigreduce(factor (...))), but that outer factoring might expose additional opportunities for trigreduce, etc. The more different transformations we want to do at every opportunity, the more awkward and less thorough nesting function invocations becomes. Also, it is inefficient to traverse the entire expression multiple times.

On the other hand, control variable settings can easily contradict each other, causing infinite recursion. Moreover, user's tend to forget that they have made certain settings earlier in a session that are inconsistent with their current desires. If available, dynamic scoping of control variables can mitigate some of the hazards for programmers, but not for end users who operate only at the top level.

Neither specific transformation functions nor control variables are well suited to letting users request different treatment with respect to different indeterminates, such as expanded with respect to c as the most main variable, with coefficients that are square-free factored then decomposed with respect to x as the next most main variable, etc.

Moreover, consider how many functions and/or control variables are necessary to provide all of the different transformations described in Section 2 and analogous ones for all other common types of irrational functions. Indeed, most major computer algebra systems tend to have more than a thousand user-level functions, averaging several arguments each, including optional ones that each might have several alternative keyword options such as "square-free", "integerCoefficients", etc. Often the more specialized functions are in packages that must be loaded by a command, and their documentation is not fully integrated into the data base for functions that are preloaded.

At any one time, most computer algebra users are amateurs. Therefore quite often a user isn't aware that a helpful mathematical transformation exists, or else the user isn't aware of its availability in their system via a function or control variable setting – or more obscurely via a particular setting of a perhaps-optional argument.

How can we provide users with intuitive detailed control over so many combinations of choices?

One partial way is to provide some direct manipulation capability wherein the user can drag and drop factors and terms to rearrange, distribute or factor them out. This idea was pioneered in the Milo and Graphing Calculator programs [2] and in the Theorist program [22], now known as LiveMath. This technique permits intuitive fine control of transformations such as moving terms or factors from one side of an equation or inequality to the other, reordering factors or terms, and distributing factors over sums. However, direct manipulation is less suitable for other transformations, such as canceling polynomial gcds, factoring polynomials or doing partial fraction expansion. Also, this technique is awkward and tedious for large expressions. Therefore, although excellent for some purposes, it alone does not suffice.

Another complementary way is to allow the user to interactively select sub-expressions and apply

transformation functions only to those sub-expressions. However, this alone doesn't conveniently provide for a mixture of transformations, such as expanding with respect to the main variable, but factoring its collected coefficients that contain all of the other variables.

In contrast, whether applied to selected sub-expressions or an entire expression, a context-dependent *transformation wizard* can help a user select wisely from among relevant options without requiring the user to know numerous transformation function names together with elaborate argument sequences, argument syntax and keywords. The Newton [13, 14] front end for Maple helped pioneer the use of such wizards.

What we want is that when a user presses a transformation button or selects it from a drop down menu, a dialog box listing the user's generalized variables and relevant transformations appears. These transformations can be indicated with patterns and/or natural language phrases. The offered transformations for the highlighted sub-expression should be context dependent to avoid intimidating, distracting and annoying users with irrelevant alternatives. Quick relevance tests could include, for example:

- whether the selected generalized variable occurs in an expandable sub-expression,
- whether the generalized variable occurs in a denominator sum,
- whether the generalized variable occurs to degree greater than 1, and
- whether the generalized variable is a logarithm that could be combined with other logarithms.

Before doing such tests, the default simplification should be applied to the highlighted expression to replace any bound variables with their assigned values – and to avoid, for example, offering polynomial expansion for $(x + x)^2$, which default simplification simplifies to $4x^2$.

More specifically, the wizard could present a drop-down list of all of the generalized variables occurring in the default-simplified result for which some optional transformation might be relevant, then invite the user to choose the generalized variable that they want to be most main.

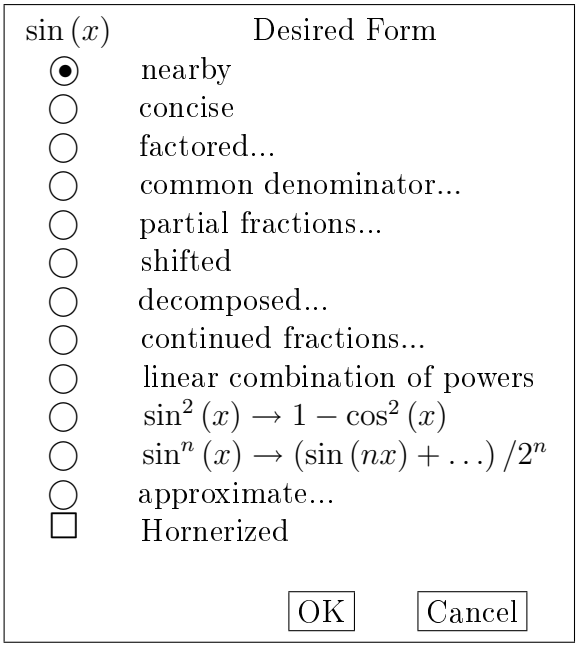
A generalized variable wouldn't be offered if no optional transformation was applicable to it. For example, if a variable x occurs only as the argument in $\sin(x)$, then it is pointless to offer x as one of the choices. However, there might be some point in offering the generalized variable $\sin(x)$, depending on how it occurs in the default-simplified result. As another example, few of the transformations being discussed here are applicable to the expression $3x^2y$.

One of the generalized variable choices would be "all remaining", meaning the next selected transformation choices should be applied to all remaining generalized variables, using their default ordering and the union of their relevant transformations.

After choosing a generalized variable or "all remaining", the user would then indicate whether they want default treatment with respect to that variable or some relevant optional transformations. This cycle would be repeated for each successively less main generalized variable until none are left or the user presses the OK button. Any remaining generalized variables would be given default simplification. Quite often the user will want specific transformations for one main variable of interest and will be content with either default or "concise" choices for the remaining variables in whatever order the system chooses.

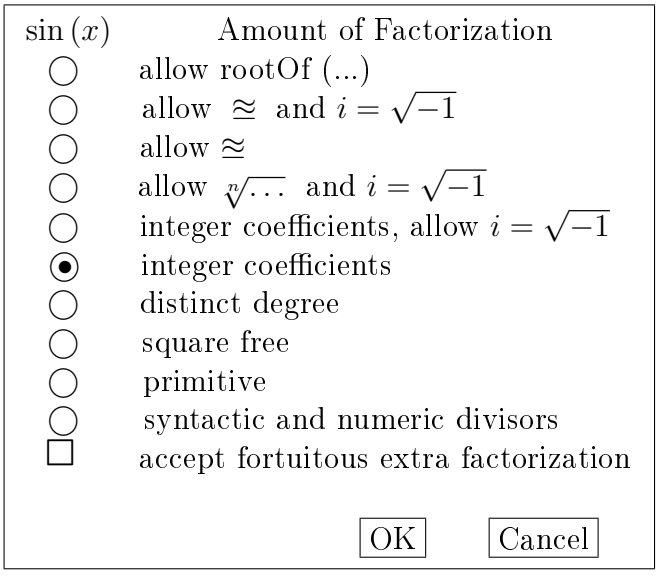
Mutually exclusive choices such as factored *versus* expanded could be controlled by radio buttons or a slider bar, whereas supplementary choices for the resulting multinomial or ratio pieces such as decomposition, continued fractions and Hornerization could be controlled by check boxes.

For example, if $\sin(x)$ occurs to a power at least 2 in a denominator in the default simplified highlighted sub-expression, then the displayed radio buttons and check boxes might be:



If limited screen space would unavoidably make the dialog box hide much of the expression or sub-expression to which the transformations will be applied, then that expression or sub-expression can also be displayed at the top of the dialog box, perhaps abbreviated with judicious use of "...".

Labels ending with an ellipsis open more detailed choices, either in place or in a dependent dialog box. For example, depending on applicability, "factored ..." could open up the choices

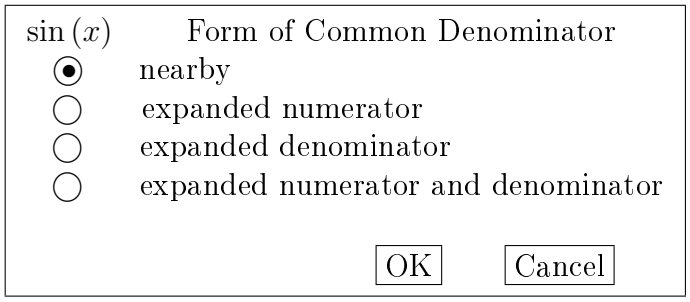


Where possible, the labels in these dialog boxes avoid terminology that might be unknown to many users, such as using i rather than "Gaussian integers". The symbol $\cong$ means that approximate coefficients are allowed if necessary. The symbol $\sqrt[n]{\ldots}$ means that fractional powers and introduced trigonometric functions of constants and of non-factorization variables are allowed.

"accept fortuitous extra factorization" means, for example, if the user selects square-free factorization but default simplification has already produced $(x+2)(x-2)(x^2-1)^3$, then it won't be gratuitously partially expanded to $(x^2-4)(x^2-1)^3$.

The choices for partial fraction expansion could include similar choices, designating the amount of denominator factorization. If applicable, there could be an additional choice

The choices for "common denominator ..." could include



Whenever the user selects "expand ..." or "partial fractions ..." for two or more variables in sequence, the consequent choices would include, if relevant, a check box

This can be done as a post-simplification process during display if we don't want to support partially-distributed forms as an internal representation.

Any choice that would otherwise reduce the domain of definition in the domain of interest should be treated by forming a piecewise result or appending a constraint to the input.

To permit convenient editing of choices, the choices for successive generalized variables could be accumulated in parallel columns, each labeled by its generalized variable. This could introduce rows of buttons or boxes that aren't relevant to all generalized variables, but those buttons or boxes could be grayed out.

A much as is practical, it is important for computer algebra systems to provide a programmatic way to accomplish anything that can be done interactively. That way programs can achieve the same flexibility non-interactively for use in algorithms or demonstration scripts. Therefore when the OK button is pressed, then before doing the transformation, the wizard would insert as the user's input a functional form such as

$$\text{transform}\left(\sin(x)\,\text{:factorOverIntegers},\ \sin^2(x)\to 1-\cos^2(x)\,,\ \text{remainingVariables}\,\text{:concise}\right).$$

Principle 3 (wizards for optional transformations)**.** Interactive wizards and corresponding automatically generated functions for optional transformations should be organized in an easily-understood comprehensive framework that helps users choose wisely.

4 Returning multiple results

"*Let them eat cake* and *bread*"
– adapted from (possibly) Marie Antoinette

The free Wolfram|alpha® web site [27] has pioneered the idea of automatically returning multiple alternative results, including 2D and 3D plots where relevant. This facility can be accessed independently or from within a *Mathematica* session.

The algorithm could display notable variants as it completes them, allowing the user to review and select favorites from this sequence of alternatives. This would be helpful for time consuming computations. The user could abort the search as soon as an acceptable form is obtained or patience is exhausted.

Either way, users are often presented with perhaps useful representations that would not have occurred to them or that they didn't have the expertise to request – or that their patience didn't suffice to choose from the alternatives offered by a wizard.

5 Summary

This article motivates and introduces the following three principles for optional transformations:

I. **(useful optional transformations)** Optional transformations should include a broad variety of useful candid forms such as fully factored over a common denominator, complete multivariate partial fractions, decompositions, continued fractions and Hornerized forms.

II. **(wizards for optional transformations)** Interactive wizards and corresponding automatically generated functions for optional transformations should be organized in an easily-understood comprehensive framework that helps the user choose wisely.

III. **(a concise-result capability)** Optional transformations should include a facility that attempts to produced a particularly concise candid form.

Acknowledgment

Thank you Albert Rich for providing the table of alternative forms for absurd numbers.

References

[1] Acton, F.S., *Numerical Methods that Work*, Chapter 11, Harper and Row, 1970 or The Mathematical Association of America, 1990.

[2] Avitzur, R., The Graphing Calculator Story, `http://www.pacifict.com/Story/`

[3] Ayad, M. and Fleischmann, P., On the decomposition of rational functions, *Journal of Symbolic Computation* 43 (4), pp. 259-274, 2008.

[4] Carette, J., Understanding expression simplification, *Proceedings of ISSAC 2004*, pp. 72-79, 2004.

[5] Ceberio, M. and Kreinovich, V., Greedy algorithms for optimizing multivariate Horner schemes, *ACM SIGSAM Bulletin* 38 (1), pp. 8-15, 2004.

[6] Cuyt, A.A.M. and Verdonk, B.M., A review of branched continued fraction theory for the construction of multivariate rational approximants, *Applied Numerical Mathematics* 4 (2-4), pp. 263-271, 2005.

[7] Delzell, C.N., A continuous, constructive solution to Hilbert's 17^{th} problem, *Inventiones Mathematicae* 76(3), pp. 365-384, 1984.

[8] Giesbrecht, M., Kaltofen, E. and Lee, Wen-shin., Algorithms for computing the sparsest shifts of polynomials via the Berlekamp/Massey algorithm, *Proceedings of ISSAC 2002*, pp. 101-108.

[9] Grigoriev, D.Y., Lakshman, Y.N., Algorithms for computing sparse shifts for multivariate polynomials, *Proceedings of ISSAC 1995*, pp. 96-103.

[10] Gutierrez, J., Rubio, R. and Sevella, D., On multivariate rational function decomposition, *Journal of Symbolic Computation* 33 (5), pp. 545-562, 2002.

[11] Klüners, J., On polynomial decompositions, *Journal of Symbolic Computation* 27 (3), pp. 261-269, 1999.

[12] Kozen, D. and Landau, S., Polynomial decomposition algorithms, *Journal of Symbolic Computation* 7 (5), pp. 445-456, 1989.

[13] Lamagna, E.A., Hayden, M.B. and Johnson, C.W., The design of a user interface to a computer algebra system for introductory calculus, *Proceedings of ISSAC 1992*, pp. 358-368.

[14] Lamagna, E.A., Hayden, M.B. and Johnson, C.W., An interactive environment for exploring mathematics, *Journal of Symbolic Computation* 25 (2), pp. 195-212, 1998.

[15] Moses, J, Algebraic simplification: a guide for the perplexed. *Proceedings of the second ACM symposium on symbolic and algebraic manipulation*, pp. 282-304, 1971.

[16] Powers, V. and Wörmann, T., An algorithm for sums of squares of real polynomials, `http://www.mathcs.emory.edu/~vicki/pub/sos.pdf`

[17] Reznick, B.E., *Sums of Even Powers of Real Linear Forms*, Memoirs of the American Mathematical Society, 1992, and `http://www.math.uiuc.edu/~reznick/memoir.html`

[18] Roy, M.F., The role of Hilbert's problems in real algebraic geometry. *Proceedings of the ninth EWM Meeting*, Loccum, Germany 1999.

[19] Stoutemyer, D.R., Ten commandments for good default expression simplification, *Journal of Symbolic Computation*, 46(7), pp. 859-887, 2011.

[20] Stoutemyer, D.R., Simplifying products of fractional powers of powers, submitted for publication.

[21] Sturmfels, B., Algorithms in invariant theory, 2nd edition, Springer 2008.

[22] Theorist, MathView, and LiveMath, `http://www.livemath.com/`

[23] Vandenberghe, L. and Boyd, S., "Semidefinite Programming", *SIAM Review* 38, pp. 49-95, March 1996.

[24] von zer Gathen, J. and Weiss, J., Homogeneous bivariate decompositions, *Journal of Symbolic Computation* 19, pp. 409-434, 1992.

[25] Weber, A., Computing radical expressions for roots of unity, *Communications in Computer Algebra* 30 (3), pp. 11-20, 1996.

[26] Weisstein, E.W., Trigonometry Angles, *MathWorld–A Wolfram Web Resource*,`http://mathworld.wolfram.com/TrigonometryAngles.html`

[27] Wolfram|alpha, `http://www.wolframalpha.com/`

[28] Zippel, R., Rational function decomposition, *Proceedings of ISSAC-91*, pp. 1-6, 1991.