Risch algorithm

From Wikipedia, the free encyclopedia

The **Risch algorithm**, named after Robert Henry Risch, is an algorithm for the calculus operation of indefinite integration (i.e., finding antiderivatives). The algorithm transforms the problem of integration into a problem in algebra. It is based on the form of the function being integrated and on methods for integrating rational functions, radicals, logarithms, and exponential functions. Risch, who developed the algorithm in 1968, called it a decision procedure, because it is a method for deciding *whether* a function has an elementary function as an indefinite integral; and also, if it does, determining it. The Risch algorithm is summarized in *Algorithms for Computer Algebra* by Keith O. Geddes, Stephen R. Czapor and George Labahn. The Risch–Norman algorithm (after A. C. Norman), a faster but less powerful technique, was developed in 1976.

Contents [hide]

Description [edit]

The Risch algorithm is used to integrate elementary functions. These are functions obtained by composing exponentials, logarithms, radicals, trigonometric functions, and the four arithmetic operations (+ − × ÷). Laplace solved this problem for the case of rational functions, as he showed that the indefinite integral of a rational function is a rational function and a finite number of constant multiples of logarithms of rational functions. The algorithm suggested by Laplace is usually described in calculus textbooks; as a computer program, it was finally implemented in the 1960s.

Liouville formulated the problem that is solved by the Risch algorithm. Liouville proved by analytical means that if there is an elementary solution g to the equation $g' = f$ then for constants α_i and elementary functions u_i and v the solution is of the form

$$g = v + \sum_{i<n} \alpha_i \ln(u_i)$$

Risch developed a method that allows one to consider only a finite set of elementary functions of Liouville's form.

The intuition for the Risch algorithm comes from the behavior of the exponential and logarithm functions under differentiation. For the function $f\,e^g$, where f and g are differentiable functions, we have

$$(f \cdot e^g)' = (f' + f \cdot g') \cdot e^g,$$

so if e^g were in the result of an indefinite integration, it should be expected to be inside the integral. Also, as

$$(f \cdot (\ln g)^n)' = f'(\ln g)^n + n f \frac{g'}{g} (\ln g)^{n-1}$$

then if $(\ln g)^n$ were in the result of an integration, then only a few powers of the logarithm should be expected.

Problem examples [edit]

Finding an elementary antiderivative is very sensitive to details. For instance, the following algebraic function[1] has an elementary antiderivative:

$$f(x) = \frac{x}{\sqrt{x^4 + 10x^2 - 96x - 71}},$$

namely:

$$F(x) = -\frac{1}{8}\ln\Big((x^6+15x^4-80x^3+27x^2-528x+781)\sqrt{x^4+10x^2-96x-71} \\ -(x^8+20x^6-128x^5+54x^4-1408x^3+3124x^2+10001)\Big)+C.$$

But if the coefficient 71 is changed to 72, it is not possible to represent the antiderivative in terms of elementary functions. (Some computer algebra systems may here return an antiderivative in terms of *non-elementary* functions (i.e. elliptic integrals), which however are outside the scope of the Risch algorithm.)

The following is a more complex example[2] that involves both algebraic and transcendental functions:

$$f(x) = \frac{x^2+2x+1+(3x+1)\sqrt{x+\ln x}}{x\,\sqrt{x+\ln x}(x+\sqrt{x+\ln x})}.$$

In fact, the antiderivative of this function has a fairly short form:

$$F(x) = 2(\sqrt{x+\ln x}+\ln(x+\sqrt{x+\ln x}))+C.$$

Implementation [edit]

Transforming Risch's theoretical algorithm into an algorithm that can be effectively executed by a computer was a complex task which took a long time.

The case of the purely transcendental functions (which do not involve roots of polynomials) is relatively easy and was implemented early in most computer algebra systems. The first implementation was done by Joel Moses in Macsyma soon after the publication of Risch's paper.[3]

The case of purely algebraic functions was solved and implemented in Reduce by James H. Davenport.[4][5]

The general case was solved and implemented in Scratchpad, a precursor of Axiom, by Manuel Bronstein.[6]

Decidability [edit]

The Risch algorithm applied to general elementary functions is not an algorithm but a semi-algorithm because it needs to check, as a part of its operation, if certain expressions are equivalent to zero (constant problem), in particular in the constant field. For expressions that involve only functions commonly taken to be elementary it is not known whether an algorithm performing such a check exists or not (current computer algebra systems use heuristics); moreover, if one adds the absolute value function to the list of elementary functions, it is known that no such algorithm exists; see Richardson's theorem.

Note that this issue also arises in the polynomial division algorithm; this algorithm will fail if it cannot correctly determine whether coefficients vanish identically.[7] Virtually every non-trivial algorithm relating to polynomials uses the polynomial division algorithm, the Risch algorithm included. If the constant field is computable, i.e., for elements not dependent on *x*, the problem of zero-equivalence is decidable, then the Risch algorithm is a complete algorithm. Examples of computable constant fields are $\mathbb{Q}$ and $\mathbb{Q}(y)$, i.e., rational numbers and rational functions in *y* with rational number coefficients, respectively, where *y* is an indeterminate that does not depend on *x*.

This is also an issue in the Gaussian elimination matrix algorithm (or any algorithm that can compute the nullspace of a matrix), which is also necessary for many parts of the Risch algorithm. Gaussian elimination will produce incorrect results if it cannot correctly determine if a pivot is identically zero[*citation needed*].

See also [edit]

- Lists of integrals
- Liouville's theorem (differential algebra)
- Symbolic integration
- Axiom (computer algebra system)
- Incomplete gamma function

References [edit]

- R. H. Risch (1969). "The problem of integration in finite terms". *Transactions of the American Mathematical Society* (American Mathematical Society) **139**: 167–189. doi:10.2307/1995313. JSTOR 1995313.
- R. H. Risch (1970). "The solution of the problem of integration in finite terms". *Bulletin of the American Mathematical Society* **76** (3): 605–608. doi:10.1090/S0002-9904-1970-12454-5.
- Maxwell Rosenlicht (1972). "Integration in finite terms". *American Mathematical Monthly* (Mathematical

Association of America) **79** (9): 963–972. doi:10.2307/2318066. JSTOR 2318066.

- Geddes, Keith O.; Czapor, Stephen R.; Labahn, George (1992). *Algorithms for computer algebra*. Boston, MA: Kluwer Academic Publishers. pp. xxii+585. doi:10.1007/b102438. ISBN 0-7923-9259-0.
- Manuel Bronstein (2005). *Symbolic Integration I*. Springer. ISBN 3-540-21493-3.
- Manuel Bronstein (1998). "Symbolic Integration Tutorial" (PDF).
- Bhatt, Bhuvanesh, "Risch Algorithm", *MathWorld*.

Notes [edit]

1. ^ This example was posted by Manuel Bronstein to the usenet forum *comp.soft-sys.math.maple* on 24 Nov. 2000.[1]
2. ^ This example comes from Manuel Bronstein's "Symbolic Integration Tutorial". See the references.
3. ^ Joel Moses (2012), "Macsyma: A personal history", *Journal of Symbolic Computation* **47**: 123–130, doi:10.1016/j.jsc.2010.08.018
4. ^ Not to be confused with his father Harold Davenport
5. ^ James H. Davenport (1981). *On the integration of algebraic functions*. Lecture notes in computer science **102**. Springer. ISBN 0-387-10290-6. ISBN 3-540-10290-6.
6. ^ Manuel Bronstein (1990), "Integration of elementary functions", *Journal of Symbolic Computation* **9** (2): 117–173, doi:10.1016/s0747-7171(08)80027-2
7. ^ "Mathematica 7 Documentation: PolynomialQuotient". *Section: Possible Issues*. Retrieved 17 July 2010.

Categories: Integral calculus | Differential algebra | Computer algebra