Simgesel Hesap

Giriş

Matematiğin kullanıldığı tüm bilim ve mühendislik dallarında çeşitli denklemlerin oluşturulması ve çözülmesi gerekir: Lineer denklemler, polinom denklemleri, diferansiyel ve integral denklemler, tansör denklemleri, vs. Birçok durumda bu denklemlerin elle çözümü uzun, karmaşık ve hataya açık işlemler gerektirir. Bu işlemler asla insan eliyle gerçekleştirilemeyecek derecede kapsamlı olabilir. Uygulanabildiği durumlarda matematiksel problemlerin bilgisayar yardımıyla yaklaşık çözümlerini bulmak için sayısal yöntemler geliştirilmiştir. Sayısal yöntemler belli bir hata payı içerir. Sayısal analizin temel problemlerinden biri bu hatayı kontrol altında tutmaktır.

Simgesel hesaplama, matematiksel işlemlerin bilgisayar yardımıyla tam ve hatasız olarak gerçekleştirilmesiyle uğraşır. Bunun için öncelikle ilgili matematiksel alandaki nesnelerin tam olarak gösterilebilmesi, daha sonra da bu alandaki problemlerin çözümü için kullanılan yöntemlerin bilgisayar yardımıyla işletilebilecek algoritmalara dönüştürülmesi gerekmektedir. Burada matematiksel anlamda en basit nesneler olan tamsayı ve reel sayıların bile bilgisayar donanımı tarafından matematiksel anlamda tam olarak gösterilmediğini hatırlatalım.

Bilgisayar üzerinde ilk simgesel hesaplama çalışmaları 1953 yılında başlamıştır. Fakat düşünce daha eskiye dayanır. Ada Lovelace, 1842 yılında Charles Babbage'ın Analitik Makina'sı hakkındaki notlarında şu satırları yazmıştır:

Matematiğe aşina olmayan kişiler, makinanın sonuçları sayısal olarak vermesinden dolayı yaptığı işin cebirsel ya da analitik olmaktan ziyade aritmetik olduğunu düşünebilirler. Bu yanlıştır. Makina sayısal değerleri sanki harflermiş ya da başka simgelermiş gibi düzenleyip birleştirebilir. Hatta, uygun düzenlemeler yapıldığı takdirde sonuçları cebirsel notasyon kullanarak çıkarabilir.

"İlk programcı" olarak bilinen Ada Lovelace'in öngörüsü büyük ölçüde gerçekleşmiştir. Günümüzde Reduce, Maple, Macsyma, Mathematica, Axiom, MuPad gibi genel amaçlı, CoCoA, GAP, Magma, Macaulay, Singular, Schoonship gibi özel amaçlı pek çok simgesel hesap sistemi yaygın olarak kullanılmaktadır.

Simgesel hesabın tarihi

Simgesel hesabın tarihini incelerken bilgisayarlardan daha geriye gitmemiz gerekiyor. Bunu yaparken matematik tarihinin büyük bir bölümünü işin içine katma tehlikesiyle karşılaşıyoruz. Çünkü matematik problemlerinin çözümü için iyi tanımlı mekanik yöntemlerin, yani algoritmaların bulunması her zaman matematiksel yaratımın temel itici

güçlerinden birisi olmuştur.Burada M.Ö. 3. yüzyılda Euklides tarafından bulunan tamsayıların en büyük ortak bölenlerin bulunması algoritmasının ve genelleştirmelerinin bugün kullanılan simgesel hesap sistemlerinin en temel algoritmaları arasında yer aldığını belitmeliyiz. Polinom denklemlerinin köklerinin aritmetik ve kök alma işlemleri kullanılarak hesaplanması problemi matematikçileri yüzyıllar boyunca meşgul etmiştir. Bu problem 19. yüzyılda Abel ve Galois tarafından olumsuz olarak sonuçlandırılmışsa da bu araştırmalar modern cebirin doğmasına yol açmıştır.

Newton'un 17. yüzyılda diferansiyel hesabı geliştirmesinden beri diferansiyel denklemler matematiğin en önemli dallarından birini oluşturur. Temel fonksiyonların türev ve integrallerinin hesaplanması matematikle ilgili herkesin edinmesi gereken temel beceriler arasında yer alır. Diferansiyel hesap öğrenen her öğrencinin gözlemlediği gibi, türev alma işleminin sınırlı sayıda temel kural kullanılarak her durumda gerçekleştirilebilmesine rağmen, bunun tersi olan belirsiz integral hesaplama işlemi için kullanım alanı sınırlı birçok değişik yöntem öğretilir. Sonuç almak için bu yöntemlerin hangilerinin ne şekilde ve hangi sırayla uygulanacağının bulunması birçok durumda kişinin beceri ve deneyimine kalmıştır; deneme ve yanılma yoluna gidilmesi gerekebilir. Ayrıca, örneğin $\int \frac{\sin x}{x} dx$ gibi, çözümü temel fonksiyonlar kullanılarak ifade edilemeyen problemler vardır. Bu yöntemlerle bir çözüme ulaşamadığımız zaman gerçekten çözümü olmayan bir problemle mi uğraştığımızı, yoksa sadece uygun bir dönüşümler zincirini mi bulamadığımızı kestiremeyiz.

Tanım gereği, bir algoritmanın işletilmesinin uygulayıcının kullanılan temel işlemleri gerçekleştirebilmesinden öte bir beceri ya da deneyimine bağlı olmaması gerekir. Bu özelliğin herhangi bir yöntemin bilgisayar tarafından uygulanabilmesi için gerekli olduğu açıktır.

Belirsiz integrallerin hesaplanması için standart olarak öğretilen yöntemlerin bir algoritma oluşturmadığını gördük. Bu problemin algoritmik çözümü olup olmadığı başlangıcından itibaren matematikçilerin kafasını kurcalamıştır. İlk önemli adım 1703 yılında Johan Bernoulli tarafından "kısmi kesirler" yönteminin bulunmasıyla atılmıştır. Kısmi kesirler yönteminin sonucu olarak her rasyonel fonksiyonun belirsiz integralinin rasyonel fonksiyonlar ve rasyonel fonksiyonların logaritmalarının bir toplamı olarak ifade edilebileceğini görüyoruz. 1833-35 yıllarında Liouville bu tür bir gösterimin daha genel fonksiyon sınıflarında da geçerli olacağını göstermiştir. Bu gösterimin algoritmik olarak nasıl elde edilebileceği uzun süre araştırılmış, sonunda 1948 yılında Ritt tarafından geliştirilen türevli halkalar ve cisimler teorisi çözüm için gerekli temeli oluşturmuştur.

Kısmi kesirler yöntemi paydadaki polinomun çarpanlarına ayrılmasını gerektirir. Polinomları çarpanlara ayırma problemi de uzun bir tarihe sahiptir. Tamsayılar üzerinde tek değişkenli polinomların çarpanlara ayrılması için ilk algoritma 1793 yılında Schubert tarafından bulunmuştur. 1882 yılında bu algoritma Kronecker tarafından yeniden bulunmuş ve cebirsel katsayılı çok değişkenli polinomlara genişletilmiştir.

Bu algoritmalardan faydalanılabilmesi için bilgisayarların sahneye çıkması gerekecektir. Bilgisayarla ilk simgesel hesap uygulamaları 1953 yılında Kahrimanian ve Nolan tarafından yazılan türev hesaplama programları ile başlamıştır. 1950lerin sonunda liste işleme dillerinin geliştirildiği görüyoruz. Bu diller, özellikle de en yaygın ve uzun ömürlü olanı, Lisp, simgesel hesap çalışmalarında çok önemli rol oynamıştır. Simgesel integral hesaplanmasıyla ilgili ilk program, 1961 yılında Slagle tarafından yazılmıştır. Lisp ile yazılmış olan bu program yukarıda sözettiğimiz “ders kitabı” yöntemlerini kullanıyordu. O yıllarda simgesel matematik problemlerinin “yapay zeka” ile ilişkilendirildiğini düşünebiliriz.

Bilgisayarla simgesel matematik problemlerinin çözülebileceğinin anlaşılması üzerine alanın 1960lı yıllardan itibaren hızlı bir gelişme içine girdiğini görüyoruz:

Polinomların çarpanlara ayrılması için başlangıçta Schubert ve Kronecker algoritmaları kullanıldı ve bilgisayar üzerinde bile oldukça yavaş oldukları görüldü. 1967 yılında Berlekamp tarafından sonlu cisimler üzerindeki polinomların çarpanlara ayrılması için hızlı bir algoritma bulunması bu probleme yeni bir yön verdi. 1969 yılında Zassenhaus, Berlekamp algoritması ile elde edilen çarpanlardan tamsayılar üzerindeki çarpanların elde edilebileceğini gösterdi. 1975-76 yıllarında Musser, Wang ve Rothschild benzer yöntemleri çok değişkenli ve cebirsel katsayılı polinomlar için geliştirdiler.

1968-70 yıllarında Risch üstel, logaritmik, trigonometrik ve rasyonel fonksiyonlari içeren bir temel fonksiyonlar sınıfı için belirsiz integral probleminin algoritmik çözümünü oluşturdu. Risch algoritmasının daha genel fonksiyon sınıflarına genelleştilmesi için yapılan çalışmalar günümüzde de sürmektedir.

Integral hesaplama ile benzerlik gösteren bir problem de $\sum_{k=1}^{n} f(k)$ şeklindeki serilerin toplamlarının kapalı olarak gösterilmesidir. 1978 yılında Gosper ardışık terimlerinin oranı k cinsinden rasyonel bir fonksiyon olan seriler için bu problemin çözümünü buldu. Daha genel bir algoritma 1990 yılında Zeilberger tarafından geliştirildi.

Gröbner tabanları polinom halkaları teorisinde kullanılan çok önemli bir araçtır. Gröbner tabanları teorik olarak 1964 yılında Hironaka, daha

önce de 1899 yılında Gordan tarafından kullanılmıştı. 1965 yılında Buchberger Gröbner tabanlarınının hesaplanmasını sağlayan bir algoritma buldu. Buchberger algoritmasının polinom sistemleri açısından önemi lineer cebirdeki Gauss yoketme yöntemiyle karşılaştırılabilir. Bu algoritmanın polinom denklemlerinin köklerinin bulunması, bir polinomun verilen başka polinomlar cinsinden ifade edilmesi gibi pek çok uygulaması vardır.

Yine bu yıllarda ilk genel amaçlı simgesel hesap sistemlerinin ortaya çıktığını görüyoruz: 1968 yılında Reduce, 1970'de Macsyma ve Reduce 2, 1971'de Scratchpad.

Bilgisayar teknolojisinin matematiksel algoritmalara etkisini en iyi şekilde gösteren örneği tamsayılar üzerindeki aritmetik işlemler oluşturur: Bilgisayar donanımı sınırlı tamsayılar üzerinde çalışır. Bu sınırın üzerinde tamsayıları bilgisayarda çok basamaklı sayılar olarak göstermek ve aritmetik işlemleri yazılım yoluyla yapmak gerekir. İlkokulda öğretilen toplama, çıkarma ve bölme yöntemleri bu iş için kullanılabilir. Örneğin N basamaklı iki sayıyı çarpmak istediğimizde, günlük hayatta kullandığımız yöntemi kullanırsak, birinci sayının her basamağını ikinci sayının her basamağıyla çarpmamız gerekir. Dolayısıyla bu işlem N^2 ile orantılı bir zaman alacaktır. Çarpma işleminin daha hızlı yapılabileceği ilk defa 1963 yılında Karatsuba tarafından gösterildi. Karatsuba yöntemi $N^{\log_2 3} \approx N^{1.59}$ ile orantılı zaman alır. 1971 yılında ise Schönhage ve Strassen tarafından $N \log N \log \log N$ ile orantılı zaman alan çok daha hızlı bir çarpma algoritması bulundu. Bu algoritma yine 1960larda bulunan Hızlı Fourier Dönüşümü algoritmasıyla ilişkilidir.

Simgesel hesabın kullanımı

Simgesel hesabın en önemli getirisi anlaşılabilirliktir. Bir problemin simgesel kapalı çözümü insan için sayısal bir çözümün vereceği herhangi bir sayılar dizisinden çok daha fazla şey ifade edebilir. Eğer bu çözüm başka problemlerin girdisi olarak kullanılacaksa simgesel çözüm açıkça sayısal bir çözüme göre daha üstündür. Sayısal değerler gerektiği zaman bunlar her zaman simgesel çözümden elde edilebilir.

O halde neden her problemin çözümü için simgesel hesap kullanmıyoruz? Birinci ve en temel neden bazı problemlerin simgesel yöntemlerle çözümünün bilinmemesidir. İkinci neden simgesel yöntemlerin genel olarak sayısal yöntemlerden daha yavaş olmasıdır. Dolayısıyla zamanın önemli olduğu, örneğin robot kontrolü gibi konularda, sayısal hesapları simgesel hesaplara tercih etmemiz gerekebilmektedir. Üçüncü neden de çoğu zaman çözülmesi gereken problemin deneysel, kesin olmayan parametreler içermesidir. Bu durumda çözüm de zorunlıu olarak yaklaşık olmak zorundadır. Bunların dışında sonuçların sayısal olarak elde edilmesinin temel amaç olduğu

problemler olabilir. Meteorolojide karşılaşılan denklemler tüm bu durumlara iyi bir örnek oluşturur..

Sonunda sayısal yöntemleri kullanmak zorunda kaldığımız durumlarda bile, simgesel hesaptan problemi sayısal yöntemlerle çözmek için en uygun hale getirmek için faydalanabiliriz. Simgesel ve sayısal yöntemler birbirinin alternatifi değil,.matematiksel problemlerin çözümünde kullanabileceğimiz yöntemlerin, birlikte ya da ayrı ayrı kullanılabilecek iki altkümesidir.

Sorunlar ve sınırlar

Çok basit bazı integrallerin temel fonksiyonlar cinsinden ifade edilemediğini biliyoruz. Aynı şekilde çözümleri temel fonksiyonlar cinsinden ifade edilemeyen pek çok diferansiyel denklem vardır. Bunlarla karşılaşıldığında simgesel hesap yöntemleri bize sonucu veremeyecektir. Bu durumda matematikte kullanılmış olan yolu izleyerek simgesel hesap yöntemlerini özel fonksiyonlarla çalışacak şekilde geliştirmek gerekir: Eliptik fonksiyonlar, hipergeometrik fonksiyonlar, Bessel fonksiyonları, vb. Fakat bunu yapmanın tek ve en iyi bir yöntemi olduğu kuşkuludur.

Matematiğin genel olarak algoritmik hale getirilmesi projesi 19. yüzyılın sonlarında başlamıştır ve günümüzde “Hilbert programı” olarak bilinir. Bunun bütünüyle gerçekleştirilemeyeceği 1930larda Gödel tarafından elde edilen çözülemezlik sonuçlarıyla ortaya çıkmıştır. Daha sonra Matijasevic tarafından Hilbert’in 10uncu probleminin, yani polinom denklemlerinin tamsayı çözümlerinin bulunmasının genel olarak çözülemez olduğunun gösterilmesi bizim açımızdan daha ilginçtir. 20. yüzyılda matematiğin birçok alanında algoritmik olarak çözülemezlik sonuçları ispatlanmıştır. Simgesel hesap açısından önemli olan sadeleştirme problemi bunlardan biridir.

Sadeleştirme problemi simgesel hesapta her noktada karşımıza çıkar: İki tamsayıyı ya da polinomu birbirine böldüğümüz zaman ortak çarpanları yokederek sonucu kısaltmamız gerekir. Ya da örneğin karmaşık bir fonksiyonun türevini aldığımızda benzer terimleri bir araya getirerek sonucu en kısa hale getirmek isteriz. Fakat sadeleştirme problemi ilk bakışta görünebileceği kadar basit bir problem değildir. Şu üç ifadeyi ele alalım:

$$(x+1)^{10} - (x-1)^{10} =$$

$$20x^9 + 240x^7 + 504x^5 + 240x^3 + 20x =$$

$$4x(x^4 + 10x^2 + 5)(5x^4 + 10x^2 + 1)$$

Bunlardan hangini polinomun en “sade” halidir sorusunun cevabı çözmeye calıştığımız probleme göre değişir. En iyi durumda bunların herbirini diğerinden elde edebilmeyi isteriz. En azından bunlardan

herhangi ikisiyle karşılaştığımızda eşit olduklarını, yani aralarındaki farkın sıfıra eşit olduğunu gösterebilmeliyiz.

Polinomlar için bu kolayca gösterilebilir. Daha genel ifadeler için sıfıra eşitlik probleminin çözümü daha zor olabilmektedir. Sıfıra eşitlik probleminin rasyonel sayılar, π, sin ve mutlak değer fonksiyonlarını içeren ve toplama ve çarpma altında kapalı olan fonksiyon sınıfı için algoritmik olarak çözülemez olduğu, diğer bir deyişle bu sınıf için bir kanonik sadeleştirici yazmanın olanaksızlığı1970 yılında Caviness, Richardson ve Matijasevic tarafından gösterilmiştir.

Daha pratik bir problem ise simgesel hesap algoritmalarının işletilmesi sırasında hesaba giren ifadeler ve çıkan sonuçlara oranla çok fazla miktarda bellek kullanımını gerektiren ara sonuçların ortaya çıkmasıdır. Ara sonuçlarin aşırı büyümesi olgusu polinom işlemleri, matrisler üzerindeki işlemler, Gröbner tabanı hesaplamaları gibi birçok simgesel algoritmada gözlenebilir: Örneğin $p(x) = x^{19} + 7x^{17} - x^7 + 2x^5 + 3$ ve $q(x) = x^{12} + 3x^8 + 2x^5 - 1$ polinomlarının en büyük ortak bölenini Euklides yönteminin düz haliyle hesapladığımız zaman bazı ara sonuçların katsayılarının 295 basamaklı sayılara kadar çıkabildiğini görürüz. Öte yandan algoritmada yapılan, bütün ara sonuçların en yüksek katsayısının 1 olacak şekilde ayarlanması gibi basit bir değişiklik sonucunda ortaya çıkan en büyük tamsayı 70 basamaklı olmaktadır. Çıkan sonuç ise 1'dir.

Bu olguyu en aza indirmek için birçok yöntem geliştirilmiştir: Modüler aritmetik yöntemleri, tamsayılar üzerindeki matrisler için Gauss-Bareiss algoritması gibi. Bareiss yöntemi Gauss yoketme algoritmasını ara sonuçları mümkün olduğunca küçük tutarak yapmaya çalışır. Sayısal analizde ise aynı algoritma yuvarlama hatalarını en aza indirecek uygulanmaya çalışılır. Bu ve benzeri gözlemlerden şu düşünceye varıyoruz: Ara sonuçların aşırı büyümesi kesin hesap yapmanin bir bedeli olarak görülebilir. Kayan noktalı sayılarla yapılan hesaplarda oluşan yuvarlama hatalarının temelde bu sayıları sınırlı bir bellek alanı kullanarak göstermemizin sonucu olması bu düşünceyi destekler.

Simgesel hesap sistemleri

Simgesel hesap sistemlerini genel ve özel olarak amaçlı ikiye ayırdık. Genel amaçlı sistemler mümkün olduğu kadar çeşitli araçlar sunarlar. Bunları kullanarak matematiğin çeşitli dallarındaki problemleri çözmeye çalışabiliriz. Bu sistemler aşağıda verilen örneklerde görüleceği gibi etkileşimli olarak kullanılabilirler. Hemen her simgesel hesap sistemi bir programlama dili içerir: Basit problemleri etkileşimli olarak çözerken bile aslında problemi ifade edebilmek için bir programlama dili kullanıyoruz. Programlanabilmeleri simgesel hesap sistemlerinin genişletilebilmelerini sağlar. Kullanıcı kendi

problemlerinin çözümünü programlama diliyle ifade edebileceği gibi matematiğin çeşitli dallarındaki problemlerin çözümü için genel amaçlı sistemler üzerinde yazılmış pek çok pakete erişebilir.

Genel amaçlı sistemlerin yeteri kadar güçlü olamadığı alanlarda bir çok özel amaçlı sistem oluşturulmuştur. Örnek olarak gruplar teorisi alanında GAP ve Magma, komütatif cebir ve cebirsel geometri çalışmaları için CoCoA, Macaulay ve Singular, yüksek enerji fiziği hesaplamalarında Schoonship, tansör analizi ve genel görelilik alanında Sheep gibi sistemleri sayabiliriz.

Örnekler

Aşağıdaki örneklerin tümü herhangi bir genel amaçlı simgesel hesap sistemiyle yapılabilecek işlemlerdir. Mümkün olduğunca değişik sistemler kullanmaya çalıştık. Kullanıcı tarafından yazılan satırlar işaretiyle başlayanlardır.

Tamsayı işlemleri (MuPad)

```
? 40!
815915283247897734345611269596115894272000000000
? 2^(2^7) + 1
340282366920938463463374607431768211457
? ifactor(2^(2^7) + 1)
59649589127497217 5704689200685129054721
```

İntegral hesaplama (MuPad)

```
? int((2*x + 3)/(x^2 + 1), x)

               3 PI       2
3 arctan(x) - ---- + ln(x  + 1)
                2
```

Çarpanlara ayırma (MuPad)

```
? factor(poly(x^3 + 1))

      2
poly(x  - x + 1, [x]) poly(x + 1, [x])
```

Çarpanlara ayırma (Maple)

```
? p:=x^2*y^4*z - x*y^9*z^2 + x*y*z^3 + 2*x - y^6*z^4 - 2*y^5*z;
? factor(p);

  4          3            5
(y  z x + y z  + 2) (x - y  z)
```

Adi diferansiyel denklem çözümü (Reduce)

```
  load_package odesolve;
? depend y,x;
? odesolve(df(y,x,2)+4*df(y,x)+4*y-x*exp(x),y,x);

      3*x        3*x
   3*e   *x - 2*e    + 27*arbconst(1)*x + 27*arbconst(1)
{y=-----------------------------------------------------}
                              2*x
                           27*e
```

Gröbner tabanı hesaplama (CoCoA)

```
? Use R ::= Q[zyx], Lex;
? I := Ideal(x^2+y^2+z^2, xy^2+1, y^2+xz^2-2);
? ReducedGBasis(I);

[y^2 – x^3 – 1, x^4 + x + 1, z^2 + x^3 + x^2 + 1]
```

Matrislerin özdeğerlerinin hesaplanması (Mathematica)

In[39]:= $\{\lambda, \mu\} = \text{Eigenvalues}\left[A = \begin{pmatrix} 2 & 1 \\ 3 & -1 \end{pmatrix}\right]$

Out[39]= $\left\{\frac{1}{2}\left(1-\sqrt{21}\right), \frac{1}{2}\left(1+\sqrt{21}\right)\right\}$

In[47]:= $\{\lambda, \mu\} = \text{Eigenvalues}\left[A = \begin{pmatrix} a & b \\ b & -a \end{pmatrix}\right]$

Out[47]= $\left\{-\sqrt{a^2+b^2}, \sqrt{a^2+b^2}\right\}$

Lineer denklem sistemi çözümü (Maple)

$$\begin{bmatrix} 1 & 2 & 1 \\ -1 & 1 & 2 \\ 1 & 3 & -1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix}, \text{ Solution is : } \begin{bmatrix} \frac{7}{9}T \\ -\frac{1}{9}T \\ \frac{4}{9}T \end{bmatrix}$$

Sonuç

Simgesel hesap alanında elli yıllık tarihi boyunca çok önemli gelişmeler olmuş, pek çok problemin çözümü elde edilmiş, öte yandan pek çok problemin de en genel halleri henüz çözülememiş ya da algoritmik olarak çözülemezliği gösterilmiştir. Dolayısıyla gerek daha geniş problem sınıflarının çözülmesi, gerekse varolan çözümlerin daha iyileştirilmesi yönünde simgesel hesap araştırmaları yoğun şekilde sürmektedir.

Matematikle ilgili herkesin simgesel hesap yöntemlerinden yararlanması gerektiği durumlar sözkonusu olabilir. Günümüzde kişisel

bilgisayarlar sözettiğimiz simgesel hesap yöntemlerinin çoğunu gerçekleştirebilecek güçtedir, ve simgesel hesap araçları yaygın olarak ulaşılabilir hale gelmiştir. Simgesel hesabın kullanılmasının giderek daha da yaygınlaşmasını bekleyebiliriz.

Kaynaklar

M. Bronstein, Symbolic Integration I – Transcendental Functions, Springer, Berlin, 1997.

B. Buchberger, G. E. Collins, R. Loos (eds.), Computer Algebra – Symbolic and Algebraic Computation, Springer, Wien, 1983

H. Cohen, A Course in Computational Algebraic Number Theory, Springer, Berlin, 1996.

J. von zur Gathen, J. Gerhard, Modern Computer Algebra, Cambridge University Press, Cambridge, 1999.

W. Decker, Some introductory remarks on computer algebra, Proceedings of the third European Congress of Mathematics, Barcelona 2000.

D. E. Knuth, The Art of Computer Programming, vol. 2 – Seminumerical Algorithms, Addison-Wesley, Reading MA, 1969.

M. Kreuzer, L. Robbiano, Computational Commutative Algebra 1, Springer, Berlin, 2000.

R. Risch, The solution of the problem of integration in finite terms, Bull. A.M.S. **76** (1970), 605–608.