submitted to *J. Symbolic Computation*

An object oriented concept for elliptic curves

MARC CONRAD AND SUSANNE SCHMITT

Universität des Saarlandes, Saarbrücken

(Received 06 July 1999)

We describe objects and methods for an object oriented concept of elliptic curves. In order to achieve this, the representation of elliptic curves in the computer algebra system SIMATH is analyzed. This leads to the objects *models* (objects which realize Weierstraß equations) and *elliptic curves* (objects which realize the part of elliptic curves which is invariant under isomorphism). The distinction between methods defined over arbitrary fields and methods defined over concrete fields involves the idea of a polymorphic field. We consider elliptic curves over finite fields and over number fields, but the concept may also be extended for elliptic curves over any other field.

1. Introduction

In the last years computations with elliptic curves have been a main field of research. One reason is the increasing importance of elliptic curves in cryptography (see e.g. (Araki, Satoh, Miura, 1998), (Koblitz, 1998), (Blake, Seroussi, Smart, 1999)). But there has also been progress in number theory itself, as the computation of rank or torsion points of the Mordell-Weil group of an elliptic curve, or the search for integral points (e.g. (Zimmer, 1996), (Cremona, 1997), (Gebel, Pethő, Zimmer, 1997), (Smart, Stephens, 1997)). The aim of this paper is to develop a concept which proposes the way elliptic curves should be implemented in software.

In (Papanikolaou, 1997) the *demands* on software for computations in number theory in the context of a computer algebra system are formulated (efficiency, reuseability, adaptability, extensibility, portability, simplicity and reliability). We give three examples how these demands affect computations with elliptic curves.

- The demand for *reuseability* implies the possibility to reuse code written for elliptic curves over one field (e.g. $\mathbf{F}_p$) also for other fields (e.g. $\mathbf{Q}$). So we need a clear distinction between algorithms available for arbitrary fields (e.g. the computing of the j-invariant) and algorithms which depend on the basic field (e.g. the counting of points of the Mordell Weil group over $\mathbf{F}_p$).
- The idea of *adaptability* implies an independence of the elliptic curve package to the basic arithmetic and the memory manager.
- *Simplicity* implies that the design has to follow the mathematical intuition (e.g. elliptic curves may be interpreted as a class of isomorphic equations).

0747–7171/90/000000 + 00 \$03.00/0

Only an object oriented approach can follow these demands. The existing software packages for elliptic curves (e.g. Apecs, mwrank, or SIMATH) do not implement such a concept. However, in order to get a better understanding which methods and data structures are appropriate we will investigate in Section 2 the implementation of elliptic curves in SIMATH. Although SIMATH is, as a C library, a priori not object oriented this will lead us to the idea how a strictly object oriented implementation should be set up.

Using the ideas of SIMATH we develop in Section 3 object hierarchies concerning elliptic curves and describe the methods of the involved objects. In this paper there is no preference on a special programming language (as C++, Java, ...) or on a special basic arithmetic (as LiDIA, NTL, ...). However, we mention here the project CAGE which aims to provide a platform for a convenient object oriented implementation of mathematical structures in C++.

We assume in the following that the reader is familiar with the arithmetic of elliptic curves. For details see e.g. (Silverman, 1986).

2. The representation of elliptic curves in SIMATH

For elliptic curves over algebraic number fields it is often convenient to consider at least three different isomorphic equations.

The first one is the equation which was originally given to define the curve, the *actual model*. The second one is a model in short Weierstraß normal form with integral coefficients. This is required for example for the computation of the torsion group of an elliptic curve over the rational numbers, using the criterion of Nagell, Lutz, and Cassels.

Moreover there exist minimal models at all places. These models are required for many higher algorithms, as e.g. the computation of the reduction type, the L-series of the curve, or the computation of the conductor.

The idea to use different models for different computations is implemented in the computer algebra system SIMATH for elliptic curves over $\mathbf{Q}$ and over quadratic number fields. This concept has been successfully used for the search for integral points over $\mathbf{Q}$ (Gebel, Pethő, Zimmer, 1997) and for the computation of the Mordell Weil group over quadratic number fields in (Cremona, Serf, 1999) and (Schmitt, 1999).

Note that SIMATH contains also algorithms for elliptic curves over number fields of higher degree and for elliptic curves over finite fields. Their implementation does not follow the concept described in this section.

We fix some notations. Let K be the field of rationals $\mathbf{Q}$ or a quadratic number field $\mathbf{Q}(\sqrt{D})$. For any $x \in K$ we denote by $\mathcal{F}x$ the factorization of x, i.e. a list of prime numbers when $K = \mathbf{Q}$ and prime ideals for $K = \mathbf{Q}(\sqrt{D})$. For $x \in \mathbf{Q}(\sqrt{D})$ let $\mathcal{N}(x)$ denote the norm of x.

In SIMATH an elliptic curve $E|K$ is represented as a list $\mathcal{L} = (\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4)$ for $K = \mathbf{Q}$ and $\mathcal{L} = (\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4, \mathcal{L}_5)$ for $K = \mathbf{Q}(\sqrt{D})$, where the lists $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ represent several isomorphic models for the curve, the list $\mathcal{L}_4$ contains constants which are independent of the model, and $\mathcal{L}_5$ contains information about the quadratic number field. We describe now the lists $\mathcal{L}_1, \ldots, \mathcal{L}_5$ in detail.

The list

$$\mathcal{L}_1 = (\mathcal{L}_{11}, \mathcal{L}_{12}, \mathcal{L}_{13}, \mathcal{L}_{14}, \mathcal{L}_{15}, \mathcal{L}_{16}, \mathcal{L}_{17})$$

contains as elements seven sublists. These lists are given as follows:

$\mathcal{L}_{11}$ $= (a_1, a_2, a_3, a_4, a_6)$ contains the coefficients of the actual model of $E|K$,
$\mathcal{L}_{12}$ $= (b_2, b_4, b_6, b_8)$ and
$\mathcal{L}_{13}$ $= (c_4, c_6)$ contain the Tate values for the actual model,
$\mathcal{L}_{14}$ $= (\Delta, \mathcal{F}\Delta)$ for $K = \mathbf{Q}$ and $\mathcal{L}_{14} = (\Delta, \mathcal{F}\Delta, \mathcal{N}\Delta, \mathcal{F}\mathcal{N}\Delta)$ for $K = \mathbf{Q}(\sqrt{D})$, where Δ is the discriminant of the actual model,
$\mathcal{L}_{15}$ $= (P, G)$, where P is a list of the torsion points of the actual model and G is a list of generators of the torsion group of the actual model,
$\mathcal{L}_{16}$ $= (T_m, T_s)$ (or $\mathcal{L}_{16} = ((T_{m_1}, \ldots, T_{m_k}), T_s)$) contains the birational transformation from the actual model to the minimal model (T_m) or to the k minimal models (T_{m_i}), and from the actual model to the model in short Weierstraß normal form (T_s); the birational transformations are given as a list of the parameters of the transformation,
$\mathcal{L}_{17}$ $= B$ contains the basis of the Mordell Weil group of the actual model.

If there exists a global minimal model for $E|K$ we have

$$\mathcal{L}_2 = (\mathcal{L}_{21}, \mathcal{L}_{22}, \mathcal{L}_{23}, \mathcal{L}_{24}, \mathcal{L}_{25}, \mathcal{L}_{26}, \mathcal{L}_{27}),$$

where

$\mathcal{L}_{21}$ $= (a_1^m, a_2^m, a_3^m, a_4^m, a_6^m)$,
$\mathcal{L}_{22}$ $= (b_2^m, b_4^m, b_6^m, b_8^m)$,
$\mathcal{L}_{23}$ $= (c_4^m, c_6^m)$,
$\mathcal{L}_{24}$ $= (\Delta^m, \mathcal{F}\Delta^m, \delta^m)$ for $K = \mathbf{Q}$ and $\mathcal{L}_{24} = (\Delta^m, \mathcal{F}\Delta^m, \mathcal{N}\Delta^m, \mathcal{F}\mathcal{N}\Delta^m, \delta^m)$ for $K = \mathbf{Q}(\sqrt{D})$,
$\mathcal{L}_{25}$ $= (P^m, G^m)$,
$\mathcal{L}_{26}$ $= (T_a^m, T_s^m)$, and
$\mathcal{L}_{27}$ $= B^m$

are analogous to the lists $\mathcal{L}_{12}, \ldots, \mathcal{L}_{17}$ with the minimal model instead of the actual model. In addition there is an estimate for the difference between the Weil height and the Néron-Tate height on the minimal model (δ^m in list $\mathcal{L}_{24}$). The transformations in $\mathcal{L}_{26}$ are from the minimal model to the actual model (T_a^m) and from the minimal model to the model in short Weierstraß normal form (T_s^m).

If there exists no global minimal model, then we have

$$\mathcal{L}_2 = ((P_1, \mathcal{L}_2^1), \ldots, (P_k, \mathcal{L}_2^k)),$$

where the P_i are the prime ideals of bad reduction and the $\mathcal{L}_2^i$ contain the minimal model for the prime ideal P_i. The lists $\mathcal{L}_2^i$ are of an analogous form as the list $\mathcal{L}_2$ for a global minimal model.

We have

$$\mathcal{L}_3 = (\mathcal{L}_{31}, \mathcal{L}_{32}, \mathcal{L}_{33}, \mathcal{L}_{34}, \mathcal{L}_{35}, \mathcal{L}_{36}, \mathcal{L}_{37}),$$

where

$\mathcal{L}_{31}$ $= (a_4^s, a_6^s)$ are the coefficients of a model in short Weierstraß normal form of $E|K$,
$\mathcal{L}_{32}$ $= (b_2^s, b_4^s, b_6^s, b_8^s)$,
$\mathcal{L}_{33}$ $= (c_4^s, c_6^s)$,
$\mathcal{L}_{34}$ $= (\Delta^s, \mathcal{F}\Delta^s, \delta^s)$ for $K = \mathbf{Q}$ and $\mathcal{L}_{34} = (\Delta^s, \mathcal{F}\Delta^s, \mathcal{N}\Delta^s, \mathcal{F}\mathcal{N}\Delta^s, \delta^s)$ for $K = \mathbf{Q}(\sqrt{D})$,

$\mathcal{L}_{35} = (P^s, G^s)$,
$\mathcal{L}_{36} = (T_a^s, T_m^s)$ or $(T_a^s, (T_{m_1}^s, \ldots, T_{m_k}^s))$, and
$\mathcal{L}_{37} = B^s$.

These lists are analogous to the lists $\mathcal{L}_{22}, \ldots, \mathcal{L}_{27}$ with a model in short Weierstraß normal form instead of the minimal model. Here the transformation T_a^s is from the model in short Weierstraß normal form to the actual model and the transformation T_m^s (or the k transformations $T_{m_1}^s, \ldots, T_{m_k}^s$) are from the model in short Weierstraß normal form to the global minimal model (or to the k different minimal models).

The list

$$\mathcal{L}_4 = (\mathcal{L}_{41}, \mathcal{L}_{42}, \mathcal{L}_{43}, \mathcal{L}_{44}, \mathcal{L}_{45}, \mathcal{L}_{46})$$

contains data of the elliptic curve, which is invariant of the current model:

$\mathcal{L}_{41} = (j, \mathcal{F}d_j)$ contains the j-invariant and the factorization of the denominator d_j of j,
$\mathcal{L}_{42} = (N, \mathcal{F}N, N_{rt}, L_{c_p})$ for $K = \mathbf{Q}$ and $\mathcal{L}_{42} = (N, \mathcal{F}N, \mathcal{N}N, N_{rt}, L_{c_P})$ for $K = \mathbf{Q}(\sqrt{D})$, where N is the conductor of the curve, N_{rt} is the list of the places where E has bad reduction together with their reduction type, and L_{c_P} is the list of the c_P-values which are used in the conjecture of Birch and Swinnerton-Dyer,
$\mathcal{L}_{43} = (o_T, S_T)$ are the order (o_T) and the structure (S_T) of the torsion group of $E|K$,
$\mathcal{L}_{44} = (\varepsilon, r, L_r, R, \#\,Ш, \chi, l_1)$ for $K = \mathbf{Q}$ and $\mathcal{L}_{44} = (\varepsilon, r, L_r, R, \#\,Ш)$ for $K = \mathbf{Q}(\sqrt{D})$ contains the sign of the functional equation (ε), the rank r of the curve, the value of the r-th derivation of the L-series at 1 (L_r), the regulator R, the order $\#\,Ш$ of the Tate Shafarevich group and in addition for $K = \mathbf{Q}$ the characteristic polynomial of the regulator matrix χ, and the least eigenvalue of the regulator matrix l_1,
$\mathcal{L}_{45} = (p_{\omega_1}, \omega_1, p_{\omega_2}, \omega_2, p_\tau, \tau)$ for $K = \mathbf{Q}$, where ω_1 is the real period, ω_2 is the complex period and $\tau = \pm\frac{\omega_1}{\omega_2}$ such that $\mathrm{Im}(\tau) > 0$; for $x \in \{\omega_1, \omega_2, \tau\}$ the integer p_x denotes the precision with which the value x has been computed; for $K = \mathbf{Q}(\sqrt{D})$ the list $\mathcal{L}_{45} = (\omega)$ contains only the real period,
$\mathcal{L}_{46} = (N', n_{th}, b_2', b_4', b_6', b_8')$ are constants which are used during the computation of the archimedian local Néron-Tate height; for $K = \mathbf{Q}(\sqrt{D})$ the list $\mathcal{L}_{46}$ is omitted.

For elliptic curves over quadratic number fields, there is a fifth list which contains data of the field. We have

$$\mathcal{L}_5 = (D, d_4),$$

where D is the discriminant of the field K and $d_4 = \begin{cases} 1 & \text{if } D \equiv 1 \pmod 4, \\ 0 & \text{else.} \end{cases}$

An elliptic curve over K in SIMATH is initialized with the values $a_1, \ldots, a_6$ which are stored in $\mathcal{L}_{11}$. The other lists are empty after the initialization. When some of the data is needed the first time it will be computed and stored at the right place. When the data will be needed again, it will be simply read out.

3. The object oriented representation of elliptic curves

For the object oriented representation of an elliptic curve we follow two principles.

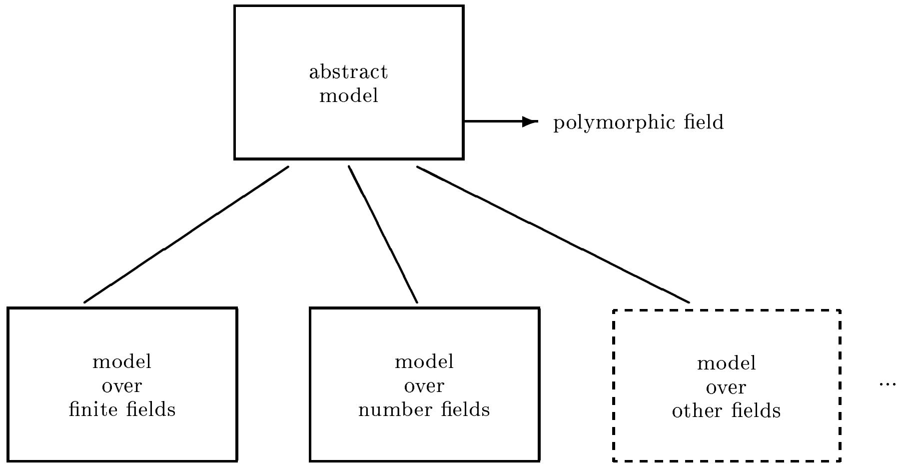

Figure 1. Object hierarchy of models of an elliptic curve.

- An elliptic curve over an arbitrary field is given by an equation of genus 1 which can always be transformed to a Weierstraß normal form. We call such an equation a *model* of the elliptic curve. As we have seen in Section 2, there are data which depend on the equation ($\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$), and data which are invariant under birational transformations ($\mathcal{L}_4$). In an object oriented environment the idea of having data leads to methods which are bound to certain objects. As objects we introduce here *models* with methods dependent on a specific equation and *elliptic curves* with methods which are invariant under birational transformations.
- Many basic algorithms (e.g. the computation of Tate values or discriminant) can be performed for elliptic curves over arbitrary fields. For this we use a polymorphic implementation of a field. For a polymorphic field we may assume that all standard field operations ($+$, $-$, $\cdot$, $\ldots$) are available. In an application the polymorphic mechanism will find the appropriate realization of operations depending on the concrete field during the execution. The idea of operating in a polymorphic field leads to the distinction between abstract objects and concrete specifications.

We describe now the objects *model* and *elliptic curve* on the abstract level. After that we show how these abstract objects should be extended (by inheritance) to objects over number fields and finite fields. Figures 1 and 2 give an overview of these objects. We focus here only on algebraic number fields and finite fields. However, the abstract model can (and should) also be used to define models over other fields, e.g. function fields.

We assume that an elliptic curve is given by a Weierstraß normal form. For elliptic curves defined by a more general equation (which can always be transformed to a Weierstraß normal form), the described objects have to be extended by appropriate methods. As done in SIMATH, it is recommended to store computed values in order to avoid recomputation.

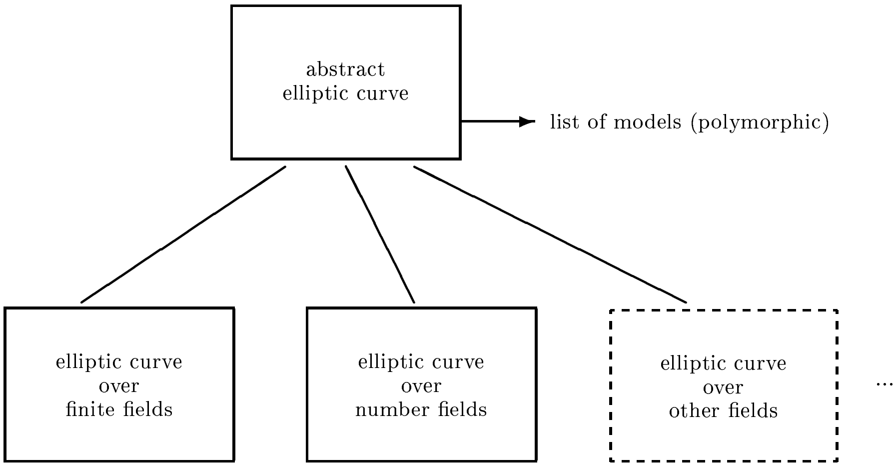

Figure 2. Object hierarchy of elliptic curves.

3.1. THE ABSTRACT SITUATION

The abstract model ...

... is constructed by the values a_1, a_2, a_3, a_4, a_6, representing the equation $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$; or by giving a birational transformation to another existing model.

... contains a reference to the elliptic curve which it belongs to. So the model will always be in the context of an elliptic curve. This can be achieved in two ways:

> If the construction of the model is called by an elliptic curve the reference will be set to this curve.
>
> Otherwise the model will construct an elliptic curve, putting itself in the list of models as the "actual model".

... has methods for the computation of the Tate values and the discriminant. Note that the j-invariant is independent of the model and will be computed by a method of the elliptic curve.

... can check if a point with given coordinates lies on the model.

... provides the addition of two points and the multiplication with $n \in \mathbf{Z}$ of a point on the model.

... has a reference to a polymorphic field.

All methods described above can be implemented in the abstract model using the reference to the polymorphic field. Of course, this does not exclude the possibility to overload certain methods in a concrete specification.

Without going into details we note that we may derive an object “abstract model in short Weierstraß normal form” from the abstract model in order to increase the performance by overloading some of the methods of the abstract model.

An elliptic curve can be considered as an isomorphism class of models. So, the object “abstract elliptic curve” contains a list of polymorphic references to different, isomorphic models. Here, ‘list’ stands for any appropriate aggregate type as e.g. arrays or mappings.

A typical problem in the context of elliptic curves is “Given the curve $E : y^2 = x^3 + 3x + 4$ over $\mathbf{Q}$. Compute a basis of the Mordell-Weil group”. So, questions about elliptic curves are often related to a specific model. Therefore we stress one of the models as the *actual model*: the model which has been used to define the curve. We describe now the abstract elliptic curve in detail. The abstract elliptic curve ...

... is constructed by a model which will then be the actual model.

... has facilities to manage the list of models. That means to add a model if it is isomorphic to the actual model, to return a specified model etc. It must also be possible to clearly distinguish the different models, e.g. by giving them names.

... provides for two models a birational transformation between these models.

... computes the j-invariant.

... analyses the structure of the Mordell-Weil group. We note that this method would be too difficult (perhaps impossible) to be implemented independent from the field where the curve is defined. It is therefore recommended to declare the method as virtual or to accept "unknown" as a return value and to overload this method in the specification. Because the *question* for the structure of the Mordell-Weil group is common to all fields, we think that this method should not be omitted on the abstract level.

... has a convenient and simple mechanism to call the methods of a model, e.g. the computation of the Tate values, the addition of points, etc. This can be implemented by defining as methods of the elliptic curve the methods of the model with an optional additional parameter. This additional parameter is the model where the method is actually performed. If the parameter is not given the actual model is chosen.

3.2. Elliptic curves over number fields

The investigation of SIMATH in Section 2 shows which data and functions are necessary in doing computations with elliptic curves over number fields. We do not make here extra considerations for the field $\mathbf{Q}$ because it can be seen as a special case of an algebraic number field. However, depending on the intended application, elliptic curves and/or models over $\mathbf{Q}$ may also be implemented as objects on their own, possibly derived from the objects over algebraic number fields.

A model over number fields is derived from the abstract model and in addition ...

... computes the norm of the discriminant, the factorization of the discriminant (in prime ideals), and the factorization of the norm of the discriminant.

... has methods to compute the torsion points and generators of the torsion group.

... computes generators of the Mordell-Weil group. Although today there are only implementations available for computing the Mordell-Weil group of elliptic curves over $\mathbf{Q}$ or over quadratic number fields, there are theoretical algorithms for computing the Mordell-Weil group for elliptic curves over arbitrary number fields which may be implemented in the future. Because questions about the Mordell-Weil group are canonical appropriate methods should not be omitted, and we suggest therefore a return value "unknown" in cases where there is at present no implementation available.

... finds the integral or the S-integral points for a finite set of places S.

... computes an estimate for the difference of the Weil height and the Néron-Tate height.

An elliptic curve over number fields is derived from the abstract elliptic curve and in addition ...

... computes the factorization of the denominator of j.

... provides a model in short Weierstraß normal form.

... provides a global minimal model (if it exists), a local minimal model to a place P of bad reduction, and the reduction type of the curve at P.

... computes the conductor, the norm of the conductor, the factorization of the conductor, and the factorization of the norm of the conductor.

... has methods to compute the sign of the functional equation, the real and the complex period, the rank r, the value of the r-th derivative of the L-series at $s = 1$, and the order of the Tate-Shafarevich group.

... computes the regulator of the curve.

3.3. Elliptic curves over finite fields

A model over a finite field is derived from the abstract model and in addition ...

... has methods which compute special points of the curve e.g. points of large order or generators of the Mordell-Weil group.

... solves to a given point and an integer the discrete logarithm problem.

An elliptic curve over a finite field is derived from the abstract elliptic curve and in addition ...

... provides a model in short Weierstraß normal form, if it exists. If there is no such model (which can happen in characteristic 2 or 3) it provides some other well defined normal form.

... computes the Hasse interval.

... has a method to compute the number of points of the Mordell-Weil group.

... computes the Hasse invariant.

Because of the increasing importance in cryptography we recommend to have special objects derived from the abstract model/elliptic curve with additional and overloaded methods for $\mathbf{F}_{2^n}$ and $\mathbf{F}_p$ where p is a (large) prime.

References

K. Araki, T. Satoh, S. Miura (1998). Overview of elliptic curve cryptography. *Public key cryptography. PKC '98, Pacifico Yokohama, Japan, Lect. Notes Comput. Sci.* **1431**, 29–49.

I. Blake, G. Seroussi, N. Smart (1999). *Elliptic Curves in Cryptography.* Cambridge University Press.

J. E. Cremona (1997). *Algorithms for modular elliptic curves.* Cambridge University Press.
J. E. Cremona, P. Serf (1999). Computing the rank of elliptic curves over real quadratic number fields of class number 1. *Math. Comp.* **68**, no. 227, 1187–1200.
J. Gebel, A. Pethő, H. G. Zimmer (1997). Computing integral points on Mordell's elliptic curves. *Collect. Math.* **48**, 115–136.
N. Koblitz (1998). *Algebraic Aspects of Cryptography.* Springer.
Th. Papanikolaou (1997). *Entwurf und Entwicklung einer objektorientierten Bibliothek für algorithmische Zahlentheorie.* Dissertation an der Universität des Saarlandes, Saarbrücken.
S. Schmitt (1999). *Bestimmung der Mordell-Weil Gruppe elliptischer Kurven über algebraischen Zahlkörpern.* Dissertation an der Universität des Saarlandes, Saarbrücken.
J. Silverman (1986). *The arithmetic of elliptic curves.* Springer.
N. P. Smart, N. M. Stephens (1997). Integral points on elliptic curves over number fields, *Math. Proc. Camb. Philos. Soc.* **122.1**, 9–16.
H. G. Zimmer (1996). Basic algorithms for elliptic curves. In *Number theory (Eger 1996).* 541–595, de Gruyter.

References to Software

Apecs. *Arithmetic of plane elliptic curves*, ftp://ftp.math.mcgill.ca/pub/apecs.
CAGE. *Classes for Algebraic Geometry and Elliptic Curves*, http://cage.math.uni-sb.de.
LiDIA. *A C++ library for computational number theory*, http://www.informatik.tu-darmstadt.de/TI/LiDIA.
mwrank. *A package to compute ranks of elliptic curves over the rationals*, http://www.maths.nott.ac.uk/personal/jec/ftp/progs.
NTL. *A Library for doing Number Theory* , http://www.shoup.net/NTL.
SIMATH. *A computer algebra system for algorithmic number theory*, http://simath.math.uni-sb.de.