Fast Library for Number Theory: An Introduction

William B. Hart

Mathematics Institute, Warwick University, Coventry, United Kingdom

Abstract. We discuss FLINT (Fast Library for Number Theory), a library to support computations in number theory, including highly optimised routines for polynomial arithmetic and linear algebra in exact rings.

1 Introduction

The Fast Library for Number Theory (FLINT) [6] is a software library, written in highly optimised C, to support computations in number theory. Its initial scope is to cover the polynomial arithmetic and linear algebra functionality of a library like Victor Shoup's Number Theory Library (NTL), [16]. However, the eventual aim of FLINT will be to provide an alternative to the Pari library [2], with a focus on higher level computations in Algebraic Number Theory.

The design motivations for FLINT are that it be:

- Written entirely in C (some assembly optimisations)
- Threadsafe design
- Implement asymptotically fast algorithms where available
- As fast or faster than other Open Source and Proprietary options
- Completely Open Source (GPL licensed)

FLINT is constructed as a set of modules, each based around a given type, e.g. the fmpz_poly module, which is based around a type for polynomials with multiple precision integer coefficients (the FLINT fmpz type).

2 Basic Integer Arithmetic

FLINT supports integer arithmetic in three ways. Firstly, the fast polynomial multiplication code (see the next section) is used to provide very fast integer multiplication for operands above about 2000 limbs. This implementation is often faster than GMP [5] (which is used for smaller multiplications), by as much as 30%.

Secondly, FLINT offers a highly optimised multiple polynomial quadratic sieve, for factoring integers. This is efficient up to about 70 decimal digits and still much faster than Pari for larger factorisations.

K. Fukuda et al. (Eds.): ICMS 2010, LNCS 6327, pp. 88–91, 2010.

Thirdly, FLINT's ulong_extras module provides fast code for operations involving C integers, i.e. long int's. This includes modular arithmetic, gcd, primality testing, efficient factorisation, via numerous optimised factoring routines. One innovation here is the One Line Factor algorithm, which is competitive with SQUFOF (also implemented). See [10] for details.

3 Polynomial Arithmetic

The bulk of code in FLINT supports polynomial arithmetic for multiprecision integer coefficients and for coefficients in $\mathbb{Z}/n\mathbb{Z}$ for up to machine word sized moduli.

FLINT implements numerous integer polynomial multiplication routines, including the classical and Karatsuba routines, Kronecker Segmentation and the Schoënhage-Strassen algorithm. The latter is based on highly optimised Fast Fourier Transform code, the final version of which was developed by David Harvey, based on the ideas presented in his paper [12].

Division of polynomials is achieved using a modified version of Mulders' algorithm (see [14], [11]), which is competitive with the usual middle product approach, but simpler to implement.

The polynomial modules also offer routines for power series operations, GCD, resultant, evaluation and composition. The latter is achieved with an algorithm implemented by Andy Novocin and the author [7].

The $\mathbb{Z}/n\mathbb{Z}$ module in the FLINT 1 series makes use of David Harvey's zn_poly library. This uses his fast Kronecker Segmentation variations [13] to achieve up to a 40% improvement over standard Kronecker Segmentation, and a highly optimised Schoënhage-Nussbaumer FFT implementation.

The $\mathbb{Z}/n\mathbb{Z}[x]$ module offers factorisation based on the Berlekamp and Cantor-Zassenhaus algorithms. The $\mathbb{Z}[x]$ module has a first implementation of the new factorisation algorithm of Novocin and van Hoeij [8] which improves on the original algorithms of van Hoeij, Belabas and others.

4 Linear Algebra

The linear algebra component of FLINT is still relatively young, providing some basic types and a FLINT rewrite of Damien Stehlé's fpLLL [15]. It also offers Hermite Normal Form and basic operations including matrix multiplication.

5 FLINT 2

The FLINT 2 series is a complete rewrite of FLINT 1 from scratch. Its focus is on very clean code and even better performance than FLINT 1. It also offers modules for multivariate polynomial arithmetic, polynomials over $\mathbb{Z}/n\mathbb{Z}$ for multiprecision n and optimised linear algebra over $\mathbb{Z}/nZ$ for word sized moduli. It should be released by the end of 2010.