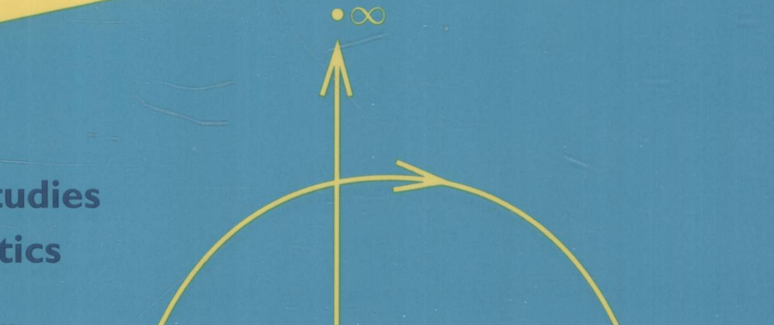
# Modular Forms, a Computational Approach

**William Stein**



**Graduate Studies in Mathematics**

**Volume 79**

**American Mathematical Society**

O156.4
S819

# Modular Forms, a Computational Approach

William Stein

*with an appendix by*
*Paul E. Gunnells*

Graduate Studies
in Mathematics

Volume 79

[illegible]大学
图书馆藏书

E2007002225

American Mathematical Society
Providence, Rhode Island

**Editorial Board**

David Cox
Walter Craig
Nikolai Ivanov
Steven G. Krantz
David Saltman (Chair)

This edition is published by the American Mathematical Society
under license from the author.

2000 *Mathematics Subject Classification.* Primary 11F11, 11Y16, 11F67, 11F55, 11F75.

For additional information and updates on this book, visit
**www.ams.org/bookpages/gsm-79**

**Library of Congress Cataloging-in-Publication Data**

Stein, William, 1974–
Modular forms, a computational approach / William Stein ; with an appendix by Paul E. Gunnells.
p. cm. — (Graduate studies in mathematics ; v. 79)
Includes bibliograpical references and index.
ISBN-13: 978-0-8218-3960-7 (alk. paper)
1. Forms, Modular—Data processing—Textbooks. 2. Algebraic spaces—Data processing—Textbooks. I. Title.

QA243.S74 2007
512.7′3—dc22 2006047950

**Copying and reprinting.** Individual readers of this publication, and nonprofit libraries acting for them, are permitted to make fair use of the material, such as to copy a chapter for use in teaching or research. Permission is granted to quote brief passages from this publication in reviews, provided the customary acknowledgment of the source is given.

Republication, systematic copying, or multiple reproduction of any material in this publication is permitted only under license from the American Mathematical Society. Requests for such permission should be addressed to the Acquisitions Department, American Mathematical Society, 201 Charles Street, Providence, Rhode Island 02904-2294, USA. Requests can also be made by e-mail to reprint-permission@ams.org.

© 2007 by the author. All rights reserved.
© 2007 for the Appendix by Paul E. Gunnells. All rights reserved.
Printed in the United States of America.

∞ The paper used in this book is acid-free and falls within the guidelines
established to ensure permanence and durability.
Visit the AMS home page at http://www.ams.org/

10 9 8 7 6 5 4 3 2 1 12 11 10 09 08 07

# Modular Forms, a Computational Approach

To my grandmother, Annette Maurer

试读结束，需要全本PDF请购买 www.ertongbook.com

# Preface

This is a graduate-level textbook about algorithms for computing with modular forms. It is nontraditional in that the primary focus is not on underlying theory; instead, it answers the question *"how do you use a computer to explicitly compute spaces of modular forms?"*

This book emerged from notes for a course the author taught at Harvard University in 2004, a course at UC San Diego in 2005, and a course at the University of Washington in 2006.

The author has spent years trying to find good practical ways to compute with classical modular forms for congruence subgroups of $\mathrm{SL}_2(\mathbb{Z})$ and has implemented most of these algorithms several times, first in C++ [**Ste99b**], then in MAGMA [**BCP97**], and as part of the free open source computer algebra system SAGE (see [**Ste06**]). Much of this work has involved turning formulas and constructions buried in obscure research papers into precise computational recipes then testing these and eliminating inaccuracies.

The author is aware of no other textbooks on computing with modular forms, the closest work being Cremona's book [**Cre97a**], which is about computing with elliptic curves, and Cohen's book [**Coh93**] about algebraic number theory.

In this book we focus on how to compute *in practice* the spaces $M_k(N, \varepsilon)$ of modular forms, where $k \geq 2$ is an integer and $\varepsilon$ is a Dirichlet character of modulus $N$ (the appendix treats modular forms for higher rank groups). We spend the most effort explaining the general algorithms that appear so far to be the best (in practice!) for such computations. We will not discuss in any detail computing with quaternion algebras, half-integral weight forms, weight 1 forms, forms for noncongruence subgroups or groups other than $GL_2$, Hilbert and Siegel modular forms, trace formulas, $p$-adic modular forms, and modular abelian varieties, all of which are topics for additional books. We also rarely analyze the complexity of the algorithms, but instead settle for occasional remarks about their practical efficiency.

For most of this book we assume the reader has some prior exposure to modular forms (e.g., [**DS05**]), though we recall many of the basic definitions. We cite standard books for proofs of the fundamental results about modular forms that we will use. The reader should also be familiar with basic algebraic number theory, linear algebra, complex analysis (at the level of [**Ahl78**]), and algorithms (e.g., know what an algorithm is and what big oh notation means). In some of the examples and applications we assume that the reader knows about elliptic curves at the level of [**Sil92**].

Chapter 1 is foundational for the rest of this book. It introduces congruence subgroups of $SL_2(\mathbb{Z})$ and modular forms as functions on the complex upper half plane. We discuss $q$-expansions, which provide an important computational handle on modular forms. We also study an algorithm for computing with congruence subgroups. The chapter ends with a list of applications of modular forms throughout mathematics.

In Chapter 2 we discuss level 1 modular forms in much more detail. In particular, we introduce Eisenstein series and the cusp form $\Delta$ and describe their $q$-expansions and basic properties. Then we prove a structure theorem for level 1 modular forms and use it to deduce dimension formulas and give an algorithm for explicitly computing a basis. We next introduce Hecke operators on level 1 modular forms, prove several results about them, and deduce multiplicativity of the Ramanujan $\tau$ function as an application. We also discuss explicit computation of Hecke operators. In Section 2.6 we make some brief remarks on recent work on asymptotically fast computation of values of $\tau$. Finally, we describe computation of constant terms of Eisenstein series using an analytic algorithm. We generalize many of the constructions in this chapter to higher level in subsequent chapters.

In Chapter 3 we turn to modular forms of higher level but restrict for simplicity to weight 2 since much is clearer in this case. (We remove the weight restriction later in Chapter 8.) We describe a geometric way of viewing cuspidal modular forms as differentials on modular curves, which leads to modular symbols, which are an explicit way to present a certain homology group. This chapter closes with methods for explicitly computing cusp forms of weight 2 using modular symbols, which we generalize in Chapter 9.

In Chapter 4 we introduce Dirichlet characters, which are important both in explicit construction of Eisenstein series (in Chapter 5) and in decomposing spaces of modular forms as direct sums of simpler spaces. The main focus of this chapter is a detailed study of how to explicitly represent and compute with Dirichlet characters.

Chapter 5 is about how to explicitly construct the Eisenstein subspace of modular forms. First we define generalized Bernoulli numbers attached to a Dirichlet character and an integer then explain a new analytic algorithm for computing them (which generalizes the algorithm in Chapter 2). Finally we give without proof an explicit description of a basis of Eisenstein series, explain how to compute it, and give some examples.

Chapter 6 records a wide range of dimension formulas for spaces of modular forms, along with a few remarks about where they come from and how to compute them.

Chapter 7 is about linear algebra over exact fields, mainly the rational numbers. This chapter can be read independently of the others and does not require any background in modular forms. Nonetheless, this chapter occupies a central position in this book, because the algorithms in this chapter are of crucial importance to any actual implementation of algorithms for computing with modular forms.

Chapter 8 is the most important chapter in this book; it generalizes Chapter 3 to higher weight and general level. The modular symbols formulation described here is central to general algorithms for computing with modular forms.

Chapter 9 applies the algorithms from Chapter 8 to the problem of computing with modular forms. First we discuss decomposing spaces of modular forms using Dirichlet characters, and then explain how to compute a basis of Hecke eigenforms for each subspace using several approaches. We also discuss congruences between modular forms and bounds needed to provably generate the Hecke algebra.

Chapter 10 is about computing analytic invariants of modular forms. It discusses tricks for speeding convergence of certain infinite series and sketches how to compute every elliptic curve over $\mathbb{Q}$ with given conductor.

Chapter 11 contains detailed solutions to most of the exercises in this book. (Many of these were written by students in a course taught at the University of Washington.)

Appendix A deals with computational techniques for working with generalizations of modular forms to more general groups than $\mathrm{SL}_2(\mathbb{Z})$, such as $\mathrm{SL}_n(\mathbb{Z})$ for $n \geq 3$. Some of this material requires more prerequisites than the rest of the book. Nonetheless, seeing a natural generalization of the material in the rest of this book helps to clarify the key ideas. The topics in the appendix are directly related to the main themes of this book: modular symbols, Manin symbols, cohomology of subgroups of $\mathrm{SL}_2(\mathbb{Z})$ with various coefficients, explicit computation of modular forms, etc.

**Software.** We use SAGE, Software for Algebra and Geometry Experimentation (see [**Ste06**]), to illustrate how to do many of the examples. SAGE is completely free and packages together a wide range of open source mathematics software for doing much more than just computing with modular forms. SAGE can be downloaded and run on your computer or can be used via a web browser over the Internet. The reader is encouraged to experiment with many of the objects in this book using SAGE. We do not describe the basics of using SAGE in this book; the reader should read the SAGE tutorial (and other documentation) available at the SAGE website [**Ste06**]. All examples in this book have been automatically tested and should work exactly as indicated in SAGE version at least 1.5.

**Acknowledgements.** David Joyner and Gabor Wiese carefully read the book and provided a huge number of helpful comments.

John Cremona and Kevin Buzzard both made many helpful remarks that were important in the development of the algorithms in this book. Much of the mathematics (and some of the writing) in Chapter 10 is joint work with Helena Verrill.

Noam Elkies made remarks about Chapters 1 and 2. Sándor Kovács provided interesting comments on Chapter 1. Allan Steel provided helpful feedback on Chapter 7. Jordi Quer made useful remarks about Chapter 4 and Chapter 6.

The students in the courses that I taught on this material at Harvard, San Diego, and Washington provided substantial feedback: in particular, Abhinav Kumar made numerous observations about computing widths of cusps (see Section 1.4.1) and Thomas James Barnet-Lamb made helpful remarks about how to represent Dirichlet characters. James Merryfield made helpful remarks about complex analytic issues and about convergence in Stirling's formula. Robert Bradshaw, Andrew Crites (who wrote Exercise 7.5), Michael Goff, Dustin Moody, and Koopa Koo wrote most of the solutions included in Chapter 11 and found numerous typos throughout the book. Dustin Moody also carefully read through the book and provided feedback.

H. Stark suggested using Stirling's formula in Section 2.7.1, and Mark Watkins and Lynn Walling made comments on Chapter 3.

Parts of Chapter 1 follow Serre's beautiful introduction to modular forms [**Ser73**, Ch. VII] closely, though we adjust the notation, definitions, and order of presentation to be consistent with the rest of this book.

I would like to acknowledge the partial support of NSF Grant DMS 05-55776. Gunnells was supported in part by NSF Grants DMS 02-45580 and DMS 04-01525.

**Notation and Conventions.** We denote canonical isomorphisms by $\cong$ and noncanonical isomorphisms by $\approx$. If $V$ is a vector space and $s$ denotes some sort of construction involving $V$, we let $V_s$ denote the corresponding subspace and $V^s$ the quotient space. E.g., if $\iota$ is an involution of $V$, then $V_+$ is $\mathrm{Ker}(\iota - 1)$ and $V^+ = V/\mathrm{Im}(\iota - 1)$. If $A$ is a finite abelian group, then $A_{\mathrm{tor}}$ denotes the torsion subgroup and $A/\mathrm{tor}$ denotes the quotient $A/A_{\mathrm{tor}}$. We denote right group actions using exponential notation. Everywhere in this book, $N$ is a positive integer and $k$ is an integer.

If $N$ is an integer, a *divisor* $t$ of $N$ is a *positive* integer such that $N/t$ is an integer.

# Contents

试读结束，需要全本PDF请购买 www.ertongbook.com

*Chapter 1*

# Modular Forms

This chapter introduces modular forms and congruence subgroups, which are central objects in this book. We first introduce the upper half plane and the group $\mathrm{SL}_2(\mathbb{Z})$ then recall some definitions from complex analysis. Next we define modular forms of level 1 followed by modular forms of general level. In Section 1.4 we discuss congruence subgroups and explain a simple way to compute generators for them and determine element membership. Section 1.5 lists applications of modular forms.

We assume familiarity with basic number theory, group theory, and complex analysis. For a deeper understanding of modular forms, the reader is urged to consult the standard books in the field, e.g., [**Lan95, Ser73, DI95, Miy89, Shi94, Kob84**]. See also [**DS05**], which is an excellent first introduction to the theoretical foundations of modular forms.

## 1.1. Basic Definitions

The group

$$\mathrm{SL}_2(\mathbb{R}) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} : ad - bc = 1 \text{ and } a,b,c,d \in \mathbb{R} \right\}$$

acts on the *complex upper half plane*

$$\mathfrak{h} = \{z \in \mathbb{C} : \mathrm{Im}(z) > 0\}$$

by *linear fractional transformations*, as follows. If $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{SL}_2(\mathbb{R})$, then for any $z \in \mathfrak{h}$ we let

$$\gamma(z) = \frac{az+b}{cz+d} \in \mathfrak{h}. \tag{1.1.1}$$

Since the determinant of $\gamma$ is 1, we have

$$\left(\frac{d}{dz}\gamma\right)(z) = \frac{1}{(cz+d)^2}.$$

**Definition 1.1** (Modular Group). The *modular group* is the group of all matrices $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ with $a, b, c, d \in \mathbb{Z}$ and $ad - bc = 1$.

For example, the matrices

$$S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \tag{1.1.2}$$

are both elements of $\mathrm{SL}_2(\mathbb{Z})$; the matrix $S$ induces the function $z \mapsto -1/z$ on $\mathfrak{h}$, and $T$ induces the function $z \mapsto z + 1$.

**Theorem 1.2.** *The group* $\mathrm{SL}_2(\mathbb{Z})$ *is generated by* $S$ *and* $T$.

**Proof.** See e.g. [**Ser73**, §VII.1]. □

In SAGE we compute the group $\mathrm{SL}_2(\mathbb{Z})$ and its generators as follows:

```
sage: G = SL(2,ZZ); G
Modular Group SL(2,Z)
sage: S, T = G.gens()
sage: S
[ 0 -1]
[ 1  0]
sage: T
[1 1]
[0 1]
```

**Definition 1.3** (Holomorphic and Meromorphic). Let $R$ be an open subset of $\mathbb{C}$. A function $f : R \to \mathbb{C}$ is *holomorphic* if $f$ is complex differentiable at every point $z \in R$, i.e., for each $z \in R$ the limit

$$f'(z) = \lim_{h \to 0} \frac{f(z+h) - f(z)}{h}$$

exists, where $h$ may approach 0 along any path. A function $f : R \to \mathbb{C} \cup \{\infty\}$ is *meromorphic* if it is holomorphic except (possibly) at a discrete set $S$ of points in $R$, and at each $\alpha \in S$ there is a positive integer $n$ such that $(z - \alpha)^n f(z)$ is holomorphic at $\alpha$.

The function $f(z) = e^z$ is a holomorphic function on $\mathbb{C}$; in contrast, $1/(z - i)$ is meromorphic on $\mathbb{C}$ but not holomorphic since it has a pole at $i$. The function $e^{-1/z}$ is not even meromorphic on $\mathbb{C}$.

Modular forms are holomorphic functions on $\mathfrak{h}$ that transform in a particular way under a certain subgroup of $\mathrm{SL}_2(\mathbb{Z})$. Before defining general modular forms, we define modular forms of level 1.

## 1.2. Modular Forms of Level 1

**Definition 1.4** (Weakly Modular Function). A *weakly modular function* of *weight* $k \in \mathbb{Z}$ is a meromorphic function $f$ on $\mathfrak{h}$ such that for all $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{SL}_2(\mathbb{Z})$ and all $z \in \mathfrak{h}$ we have

$$f(z) = (cz+d)^{-k} f(\gamma(z)). \tag{1.2.1}$$

The constant functions are weakly modular of weight 0. There are no nonzero weakly modular functions of odd weight (see Exercise 1.4), and it is not obvious that there are any weakly modular functions of even weight $k \geq 2$ (but there are, as we will see!). The product of two weakly modular functions of weights $k_1$ and $k_2$ is a weakly modular function of weight $k_1 + k_2$ (see Exercise 1.3).

When $k$ is even, (1.2.1) has a possibly more conceptual interpretation; namely (1.2.1) is the same as

$$f(\gamma(z))(d(\gamma(z)))^{k/2} = f(z)(dz)^{k/2}.$$

Thus (1.2.1) simply says that the weight $k$ "differential form" $f(z)(dz)^{k/2}$ is fixed under the action of every element of $\mathrm{SL}_2(\mathbb{Z})$.

By Theorem 1.2, the group $\mathrm{SL}_2(\mathbb{Z})$ is generated by the matrices $S$ and $T$ of (1.1.2), so to show that a meromorphic function $f$ on $\mathfrak{h}$ is a weakly modular function, all we have to do is show that for all $z \in \mathfrak{h}$ we have

$$f(z+1) = f(z) \qquad \text{and} \qquad f(-1/z) = z^k f(z). \tag{1.2.2}$$

Suppose $f$ is a weakly modular function of weight $k$. A *Fourier expansion* of $f$, if it exists, is a representation of $f$ as $f(z) = \sum_{n=m}^{\infty} a_n e^{2\pi i n z}$, for all $z \in \mathfrak{h}$. Let $q = q(z) = e^{2\pi i z}$, which we view as a holomorphic function on $\mathbb{C}$. Let $D'$ be the open unit disk with the origin removed, and note that $q$ defines a map $\mathfrak{h} \to D'$. By (1.2.2) we have $f(z+1) = f(z)$, so there is a function $F : D' \to \mathbb{C}$ such that $F(q(z)) = f(z)$. This function $F$ is a complex-valued function on $D'$, but it may or may not be well behaved at 0.

Suppose that $F$ is well behaved at 0, in the sense that for some $m \in \mathbb{Z}$ and all $q$ in a neighborhood of 0 we have the equality

$$F(q) = \sum_{n=m}^{\infty} a_n q^n. \tag{1.2.3}$$

试读结束，需要全本PDF请购买 www.ertongbook.com