MATH 470: Concrete computations using PARI/GP

Frédéric Cai

24-02-2025

We will present how to use the software PARI/GP to compute different results on modular forms and Hecke operators we obtained the previous weeks.

1 Warm-up

1.1 Recall

We recall some notions from Valentin's last talk.
We will be looking at how Hecke operators $T(n)$ acts on the space of modular forms of weight k, M_k. In fact, for $f \in M_k$, with Fourier expansion

$$f(z) = \sum_{n=1}^{\infty} a_n(f) q^n,$$

we have

$$T(n)f(z) = \sum_{m \in \mathbb{Z}} a_m(T(n)f(z)) q^m, \quad a_m(T(n)f(z)) = \sum_{\substack{d|(n,m) \\ d \geq i}} d^{k-1} a_{mn\ a^2}(f).$$

From Ludovic's talk, M_k has finite dimension (we even had a formula), and has basis

$$\{E_4^a E_6^b : 4a + 6b = k\}.$$

Thus $T(n)$ can be seen as a matrix acting on those basis. We say f is an eigenform if $T(n)f = \lambda(n)f$ for some $\lambda(n) \in \mathbb{C}$ for all $n \geq 1$, i.e. simultaneously an eigenvector for all Hecke operators. Also, it said to be normalized if $a_1(f) = 1$.
We had the result that if f is a normalized eigenform, $\lambda(n) = a_n(f)$. Thus, all non-zero eigenspaces would be of dimension one.

1.2 Orthogonal decomposition

Consider f, g two cusp forms of weight k, we can define a $\mathrm{SL}_2(\mathbb{Z})$ invariant measure

$$\mu(f, g) = f(z)\overline{g(z)} y^{k-2} dx dy, \quad x = Re(z), y = Im(z).$$

One can verify the invariance, and bounded on $\mathfrak{H}/\mathrm{SL}_2(\mathbb{Z})$. Then, we form the *Petersson product* :

$$< f, g >= \int_{\mathfrak{H}/\mathrm{SL}_2(\mathbb{Z})} \mu(f, g) = \int_{\mathfrak{H}/\mathrm{SL}_2(\mathbb{Z})} f(z)\overline{g(z)} y^{k-2} dx dy.$$

One can check that this is a Hermitian, positive and non degenerate inner product on S_k. One should check

$$< T(n)f, g >=< f, T(n)g > .$$

i.e. $T(n)$ is self adjoint on $< .,. >$. Using spectral theorem and knowing that all Hecke operators commute with each other, we see that they can be simultaneously diagonalizable, with real eigenvalues.
Especially, we can find an orthogonal basis of S_k made of eigenvectors of $T(n)$.
Unfortunately, the Petersson product is not bounded on the whole modular forms space, and so the integral does not converge. Thus, we don"t have the self-adjointness of Hecke operators. But we have that $M_k/S_k = \mathbb{C}E_k$, the Eisenstein series of weight k, and so

$$M_k = \mathbb{C}E_k \oplus S(k).$$

This complete diagonalization of M_k motivates our search of eigenforms. Actually, we already have one.

Lemma 1.1. *E_k is an eigenform.*

Proof. Consider the normalized Eisenstein series

$$\tilde{E_k} = c(k) + \sum_{n=1}^{\infty} \sigma_{k-1}(n) q^n,$$

with $\sigma_k(n) = \sum_{d|n} d^k$ and some constant $c(k) \in \mathbb{C}$. Then one can prove that

$$a_m(T(n)f) = \sum_{\substack{d|(n,m) \\ d \geq i}} d^{k-1} \sigma_{k-1}(\frac{mn}{d^2}) = \sigma_{k-1}(m)\sigma_{k-1}(n).$$

For $(m,n) = 1$, we see that $a_m(T(n)f) = \sigma_{k-1}(mn) = \sigma_{k-1}(m)\sigma_{k-1}(n)$ using multiplicativity of $\sigma_k(n)$ ∎

Thus the plan to find the eingeforms is to express $T(n)$ for some $n \in \mathbb{N}$ acting on M_k in matrix form such that it fully diagonalizes (usually $n = 2$ works), and then write the eigenvectors as linear combinations of our chosen basis of M_k.

2 Let's use PARI/GP to find the eigenforms

For help, use ? command, it is very useful to understand how a specific function works.
I recommand in order to learn to write out each function each line at a time We will apply the method said above, and use PARI/GP to helpfor our computations. We will limit ourselves to linear algebra functions. Full code in a block can be find in appendix A. I will try to break the code in steps, and explain it. We want to buiild a function that depends on k which spits out, the eigenvalue, eigenvectors of $T(2)$ and the Fourier Series of the eigenforms.

Setup We first define a few useful functions

```
sigm(n,k) = { my(t=0); for (a=1, n, if(n%a == 0 , t=t+a^k)); t;

} \\ Our sum of divisors to the power of k function


kill(q) ; \\ To assure that q is a veriable

Ek(k) = { my(T=[1]); for (a=1, 100, T=concat(T, -2*k*sigm(a,k-1)/bernfrac(k))) ;
Ser(T,q,101);

} \\ Creates Fourier Series of E_k, with 101 first coefficients



gamm(n,F,m,k)={my(t=0); for (a=1,gcd(n,m), if(gcd(n,m)%a==0,
t=t+a^(k-1)*polcoef(F,m*n/a^2))); t;

} \\ Calculates m-th coefficient of T(n)F, weight k (=a_(m)(T(n)F)
```

Now we start our function that gives what we want :

```

Eigen(k) = {

E4=Ek(4);
E6=Ek(6);
```

Basis for M_k Since $M_k =< E_4^a E_6^b : 4a + 6b = k >$, we make B our vector of d basis.

```
B=[];
for(a=0,k,for(b=0,k, if(4*a+6*b==k, B=concat(B,E4^a*E6^b)))); \\ Finds E4^aE6^b
    basis of M_k;

d=#B;
```

Matrix of Hecke operator We express $T(n)$ as a matrix acting on our basis B, in order to diagonalize it. We use here that we choose $T(2)$. If the output gives eigenvalues with multiplicity, one can change for $T(3) etc.$... Note that looking at $T(n)$'s with n prime makes the formula for $a_m(T(n)f)$ much simpler, with only 2 terms left. Also, we might want to know how many coefficients of F is needed in this. To get the m-th coefficients, we will need at least a_{mn}, so the $mn+1$-th coefficient, and since we want d coefficients, we need at least the $2n+1$ first coefficients of the basis.
We have for each $f_i \in B$, $T(2)f_i = \sum_j h_{j,i} f_j$, and write them in matrix $H = (h_{i,j})$, the matrix of how $T(2)$ acts on our basis of M_k. In order to find each $h_{i,j}$, we need to solve a system of n variables (match linear combinations of first d coefficients of $T(n)f_i$ with those of each basis).

```
C=matrix(d,d,i,j,polcoef(B[j], i-1));
TB=vector(d,i,vector(d,j,gamm(2,B[i],j-1,k))); \\ Only place where use T(2)
H=matrix(d,d,i,j, matsolve(C,mattranspose(TB[j]))[i]);
```

Eigenstuff Using PARI/GP linear algebra functions, we get :

```
EiVec=mateigen(H); \\ eigenvectors as a column of matrix
EiVal=mateigen(H,1)[1] ; \\ eigenvalues
```

Here, we write the eigenvectors as linear combinations of our basis of M_k, and then normalize it.

```
F=vector(d);
for(i=1,d,F=F+vector(d,j,B[i]*EiVec[i,j])); \\ Those are the eigenforms !

E= vector(d,i,F[i]/polcoef(F[i],1)); \\ normalised !
```

And we give our output,

```

[EiVal,EiVec, E] ;
\\ gives in order, eigenvalues, eigenvectors as columns (of T(2) on M_k) and
    eigenforms
\\ if badluck where H doesn't fully diagonalize, try T(3) etc... instead

}
```

A Full Code

Simply past the entire code in a text file, with .gp extension, and reading it in PARI with `\r directory... .gp`, or `/r directory... .gp` depending on the operating system.

```
sigm(n,k) = { my(t=0); for (a=1, n, if(n%a == 0 , t=t+a^k)); t;

} \\ Our sum divisors to the power of k function



kill(q) ;

Ek(k) = { my(T=[1]); for (a=1, 100, T=concat(T, -2*k*sigm(a,k-1)/bernfrac(k))) ;
```

```
Ser(T,q,101);

} \\ Fourier Series of E_k, with 101 first coefficients



gamm(n,F,m,k)={my(t=0); for (a=1,gcd(n,m), if(gcd(n,m)%a==0,
t=t+a^(k-1)*polcoef(F,m*n/a^2))); t;

} \\ Calculates m-th coefficient of T(n)F, weight k



Eigen(k) = {

E4=Ek(4);
E6=Ek(6);

B=[];
for(a=0,k,for(b=0,k, if(4*a+6*b==k, B=concat(B,E4^a*E6^b)))); \\ Finds E4^aE6^b
    basis of M_k;

d=#B;

C=matrix(d,d,i,j,polcoef(B[j], i-1));
TB=vector(d,i,vector(d,j,gamm(2,B[i],j-1,k))); \\ Only place where use T(2)
H=matrix(d,d,i,j, matsolve(C,mattranspose(TB[j]))[i]);

EiVec=mateigen(H); \\ eigenvectors as a column of matrix
EiVal=mateigen(H,1)[1] ; \\ eigenvalues

F=vector(d);
for(i=1,d,F=F+vector(d,j,B[i]*EiVec[i,j])); \\ Those are the eigenforms !

E= vector(d,i,F[i]/polcoef(F[i],1)); \\ normalised !


[EiVal,EiVec, E] ;
\\ gives in order, eigenvalues, eigenvectors as columns (of T(2) on M_k) and
    eigenforms
\\ if badluck where H doesn't fully diagonalize, try T(3) etc... instead

}
```