Computing modular Galois representations - the modulo $p$ approach (after Jinxiang Zeng)

Maarten Derickx

Universiteit Leiden
and
Université Bordeaux 1

Sage Days 51
22-26 July 2013

1Original slides by Jinxiang Zeng, modified by D.
Computing Coefficients of modular forms

1. Introduction/Main Results
   - How fast can $\tau(p)$ be computed?
   - An algorithm work with finite fields
   - Complexity analysis
   - A lower bound on the number of generators of $m \subset \mathbb{T}$

2. A First Description of the Algorithm
   - Congruence of Modular Forms
   - Galois Representations and Modular Forms
   - Computing The Ramanujan subspace

3. Future work
The discriminant modular form

Let \( q := e^{2\pi i z} \), the discriminant modular form is defined by

\[
\Delta(q) = q \prod_{n=1}^{\infty} (1 - q^n)^24 = \sum_{n=1}^{\infty} \tau(n)q^n \in S_{12}(\text{SL}_2(\mathbb{Z}))
\]

where \( \tau : \mathbb{Z} \to \mathbb{Z} \) is called Ramanujan tau function.

\( \Delta(q) \) plays a crucial role during the developments of theory of modular forms. In this lecture we focus on the computational aspects of \( \Delta(q) \).
The discriminant modular form

**Arithmetic of the Ramanujan tau function**

- \( \tau(mn) = \tau(m)\tau(n) \) for any integers satisfying \((m, n) = 1\).
- \( \tau(p^{n+1}) = \tau(p)\tau(p^n) - p^{11}\tau(p^{n-1}) \) for any prime \( p, n \geq 1 \).
- \( |\tau(p)| \leq 2p^{11/2} \), Deligne’s bound.
- \( \tau(p) \equiv p(1 + p^9) \mod 25, \tau(p) \equiv p(1 + p^3) \mod 7, \tau(p) \equiv 1 + p^{11} \mod 691 \)

**Lehmer’s Conjecture**

- \( \tau(n) \neq 0 \) for any \( n \geq 1 \).

Serre: if \( \tau(p) = 0 \) then \( p = hM - 1 \) with \( M = 2^{14}3^75^3691 \), \( \left(\frac{h+1}{23}\right) = 1 \) and some \( h \mod 49 \in \{0, 30, 48\} \).
How fast can \( \tau(p) \) be computed?

A question that Schoof asked to Edixhoven in 1995
Can we compute \( \tau(p) \) for prime \( p \) in time polynomial in \( \log p \)?

Theorem (Edixhoven, Couveignes, etc.)
For prime \( p \), there exist algorithms to compute \( \tau(p) \) in time polynomial in \( \log p \).
- work with complex number field, using numerical approximation.
- work with finite fields, using CRT.

\[ |\tau(p)| \leq 2p^{11/2} \] so \( \tau(p) \) can be computed by computing \( \tau(p) \mod \ell \) for sufficiently many small primes \( \ell \) (where small means \( O(\log p) \).)
How fast can $\tau(p)$ be computed?

**Generalization and explicit calculation**

- Bruin generalized the methods to modular forms for the groups of the form $\Gamma_1(n)$.
- Bosman implemented an algorithm using numerical approximation $\mathbb{C}$ and computed

$$\rho_l^{proj} : \text{Gal}\overline{\mathbb{Q}}/\mathbb{Q} \to \text{PGL}(V_l)$$

for $l \in \{13, 17, 19\}$. This allows one to calculate $\pm \tau(p) \mod l$
which he used to prove

$$\tau(n) \not= 0, \forall n < 2 \cdot 10^{19}.$$
A probabilistic algorithm

Algorithm (Zeng 2012)

Following Couveignes’s idea, working with finite fields, we give a probabilistic algorithm, which is rather simple and well suited for implementation.

The following calculation was done using a personal computer.

<table>
<thead>
<tr>
<th>level (\ell)</th>
<th>time (projective representation)</th>
<th>time (entire representation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\ell = 13)</td>
<td>several minutes</td>
<td>one hour</td>
</tr>
<tr>
<td>(\ell = 17)</td>
<td>several hours</td>
<td>one day</td>
</tr>
<tr>
<td>(\ell = 19)</td>
<td>several days</td>
<td>less than four days</td>
</tr>
<tr>
<td>(\ell = 29)</td>
<td>waiting</td>
<td>waiting</td>
</tr>
<tr>
<td>(\ell = 31)</td>
<td>several days</td>
<td>several days</td>
</tr>
</tbody>
</table>
A probabilistic algorithm

Exact value of $\tau(p)$ mod $\ell$

Since we can compute the entire representation, the exact values of $\tau(p) \mod \ell$ for $\ell \in \{13, 17, 19\}$ can be computed.

Nonvanishing of tau function

Since we can compute the projective representation for $\ell = 31$, we can prove\(^a\)

$$\tau(n) \neq 0, \text{ for all } n < 982149821766199295999 \approx 9 \cdot 10^{20}$$

\(^a\)Bosman proved the nonvanishing holds for $n < 22798241520242687999 \approx 2 \cdot 10^{19}$
Complexity of the algorithm

Theorem (Zeng 2012)

For prime $p$, $\tau(p)$ can be computed in time $O(\log^{6+2\omega+\delta+\epsilon} p)$.

- $\omega$ is a constant in $[2,4]$, refers to that addition in Jacobian can be done in time $O(g^\omega)$,
- $\delta$ is a constant, measuring the heights of the points of the Ramanujan subspace $V_\ell$,
- $\epsilon$ is any real positive number.

$\omega$ depends on the complexity of calculations in $J_1(l)(\mathbb{F}_{p^e})$. Using Khuri-Makdisi’s algorithm, the constant $\omega$ is 2.376. Our computation suggests $\delta \approx 3$, although this is based on a very small sample ($l = 13, 17, 19$)
On the generators of the maximal ideal

**Theorem (Zeng 2012)**

If $\ell \geq 13$ is prime and $m = (l, T_1 - \tau(1), T_2 - \tau(2), T_3 - \tau(3), \ldots) \subset \mathbb{T}$, then $m$ can be generated by $\ell$ and $T_n - \tau(n)$ with $n \leq \frac{2\ell+1}{12}$.

**Remarks**

- It makes the algorithm faster. The previous known upper-bound was $(\ell^2 - 1)/6$, making step 5 very slow.
- In practice the upper bound is even much better.
  - $m = (\ell, T_2 - \tau(2))$ for $\ell \in \{13, 17, 19, 29, 37, 41, 43\}$
  - $m = (\ell, T_3 - \tau(3))$ for $\ell = 31$
Theorem (Mazur, Ribet, Gross, Edixhoven etc.)

Let $n, k \in \mathbb{Z}_+$, $\mathbb{F}/\mathbb{F}_\ell$ finite extension, and $f : \mathbb{T}(n, k) \to \mathbb{F}$ a surjective ring morphism. Assume $2 < k \leq \ell + 1$ and the associated Galois representation $\rho_f : \text{Gal}(\overline{\mathbb{Q}}/\mathbb{Q}) \to \text{GL}_2(\mathbb{F})$ is absolutely irreducible. Then there is a unique ring morphism $f_2 : \mathbb{T}(n\ell, 2) \to \mathbb{F}$ such that:

- $f_2$ is surjective, $f_2(T_i) = f(T_i)$, $f_2(<a>) = f(<a>)a^{k-2}$ for all $i \geq 1$ and any $a$ satisfying $(a, n\ell) = 1$.
- $V_f := J_1(n\ell)[\ker f_2]$ realizes $\rho_f$.

Remark

For the rest of this talk: $f = \Delta(q) \mod \ell$, so $\mathbb{F} = \mathbb{F}_\ell$, $\ker f_2 = <\ell, T_i - \tau(i) : i \geq 1 >$ and $V_{\ell} := V_{\Delta, \ell} = J_1(\ell)[\ker f_2]$. 
Let \( \rho_\ell \) be the Galois representation associated to the newform \( \Delta(q) \)

\[
\rho_\ell : \text{Gal}(\overline{\mathbb{Q}}/\mathbb{Q}) \to \text{GL}_2(\mathbb{F}_\ell)
\]

then

- For prime \( p \neq \ell \):
  \[
  \text{Tr}(\rho_\ell(\text{Frob}_p)) \equiv \tau(p) \mod \ell \quad \text{and} \quad \text{det}(\rho_\ell(\text{Frob}_p)) \equiv p^{11} \mod \ell.
  \]
- The representation space (called Ramanujan subspace denoted by \( V_\ell \)) is
  \[
  V_\ell = \bigcap_{1 \leq k \leq \frac{\ell^2 - 1}{6}} \ker(T_k - \tau(k), J_1(\ell)[\ell])
  \]
Computing $V_\ell \mod p$: the strategy

1) Find an $e$ s.t. $V_\ell(\overline{\mathbb{F}}_p) = V_\ell(\mathbb{F}_{p^e})$

2) Compute $n := \# J_1(\ell)(\mathbb{F}_{p^e})$

3) Pick $P \in J_1(\ell)(\mathbb{F}_{p^e})$ random.

4) Multiply $P$ by $n\ell^{-v_\ell(n)}$, and then repeatedly by $\ell$ until $P \in J_1(\ell)[\ell]$

5) Compute $Q := f(P)$ for some surjection $J_1(\ell)[\ell] \rightarrow V_\ell$.

6) Repeat 3), 4) and 5) till you find linearly independent $Q_1, Q_2 \in V_\ell$. 

Computing modular Galois representations
Step 1: find $e$ s.t.: $V_\ell(\overline{\mathbb{F}}_p) = V_\ell(\mathbb{F}_{p^e})$

The characteristic polynomial of $\text{Frob}_p$ on $V_\ell$ is $X^2 - \tau(p)X + p^{11}$

We need $\text{Frob}_p = \text{Id}_{V_\ell}$ so we can take:

$$e := \min\{t \mid t \geq 1, X^t = 1 \in \mathbb{F}_\ell[X]/(X^2 - \tau(p)X + p^{11})\}$$

Remark

Step 4 is very expensive if $e$ is big. So we only compute $V_\ell \mod p$ for the $p$ s.t. $e$ is small.
Step 5: Computing the surjection $J_1(\ell)[\ell] \rightarrow V_\ell$

Let $S \subset \mathbb{N}$ s.t. $m$ is generated by $\ell$ and $T_n - \tau(n)$ for $n \in S$. Let $A_n(X)$ be the characteristic polynomial of $T_n$ on $S_2(\Gamma_1(\ell))$. Write $A_n(X) \equiv B_n(X) \cdot (X - \tau(n))^{e_n} \mod \ell$, with $e_n \geq 1$ and $A_n(\tau(n)) \not\equiv 0 \mod \ell$. 

Let $\pi_S := \prod_{n \in S} B_n(T_n)$, then for all $P \in J_1(\ell)[\ell]$ and all $n \in S$:

$$(T_n - \tau(n))^{e_n} \pi_S(P) = 0.$$ 

If $\pi_S(P) \neq 0$ then there are $d_n < e_n$ s.t.

$$Q := \left( \prod_{n \in S} (T_n - \tau(n))^{d_n} \right) \pi_S(P)$$

is a nonzero point in $V_\ell = J_1(\ell)[\ell] \cap \bigcap_{n \in S} \ker T_n - \tau(n)$. 

Computing modular Galois representations
In step 4 we have to multiply a $P \in J_1(\ell)(\mathbb{F}_{p^e})$ by a huge integer ($\approx p^{eg}$). But in fact $J_1(\ell)$ is isogenous to $\prod_f A_f$ where $f$ runs through Galois conj. classes of newforms of $S_2(\Gamma_1(\ell))$ and $A_f \subset J_1(\ell)$ is the factor corresponding to $f$.

Instead of computing $(\ell^{-\nu_\ell}N)P$ where $N := \#J_1(\ell)(\mathbb{F}_{p^e})$ we can instead compute $(\ell^{-\nu_\ell}N')T(P)$ where $T \in T$ s.t. $T(J_1(\ell)) \subset A_f$ and $N' := \#A_f(\mathbb{F}_{p^e}))$. Advantage: $N' \approx p^{edim A_f}$

### Comparing dimensions for $f \equiv \Delta \mod \ell$

<table>
<thead>
<tr>
<th>Level $\ell$</th>
<th>13</th>
<th>17</th>
<th>19</th>
<th>29</th>
<th>31</th>
<th>37</th>
<th>41</th>
<th>43</th>
<th>47</th>
<th>53</th>
<th>59</th>
</tr>
</thead>
<tbody>
<tr>
<td>dim $J_1(\ell)$</td>
<td>2</td>
<td>5</td>
<td>7</td>
<td>22</td>
<td>26</td>
<td>40</td>
<td>51</td>
<td>57</td>
<td>70</td>
<td>92</td>
<td>117</td>
</tr>
<tr>
<td>dim $A_f(\ell)$</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>12</td>
<td>4</td>
<td>18</td>
<td>6</td>
<td>36</td>
<td>66</td>
<td>48</td>
<td>112</td>
</tr>
</tbody>
</table>
Let $f \equiv 1 \mod \ell$ be a newform and $\chi$ be the character associated to $f$ then the characteristic polynomial of $\text{Frob}_p$ on $V_\ell$ is $X^2 - \tau(p) + \chi(p)p = X^2 - \tau(p) + p^{11}$. In other words $\chi(p) \equiv p^{10} \mod \ell$, in particular if $\ell \equiv 1 \mod 10$ then $\chi(\langle d^{(l-1)/10} \rangle) \equiv d^{(l-1)} = 1 \mod \ell$. This shows that $\langle d^{(l-1)/10} \rangle f \equiv \chi(\langle d^{(l-1)/10} \rangle) \equiv d^{(l-1)}f = f$. So $V_\ell$ can also be found in $J_H(\ell)$, the jacobian of $X_1(\ell)/\langle d^{(l-1)/10} \rangle$ with $d$ a generator of $\mathbb{F}_\ell^*$. 

### Comparing dimensions for $f \equiv \Delta \mod \ell$

<table>
<thead>
<tr>
<th>Level $\ell$</th>
<th>13</th>
<th>17</th>
<th>19</th>
<th>29</th>
<th>31</th>
<th>37</th>
<th>41</th>
<th>43</th>
<th>47</th>
<th>53</th>
<th>59</th>
</tr>
</thead>
<tbody>
<tr>
<td>dim $J_1(\ell)$</td>
<td>2</td>
<td>5</td>
<td>7</td>
<td>22</td>
<td>26</td>
<td>40</td>
<td>51</td>
<td>57</td>
<td>70</td>
<td>92</td>
<td>117</td>
</tr>
<tr>
<td>dim $A_{f_\ell}$</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>12</td>
<td>4</td>
<td>18</td>
<td>6</td>
<td>36</td>
<td>66</td>
<td>48</td>
<td>112</td>
</tr>
<tr>
<td>dim $J_H(\ell)$</td>
<td>6</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How to compute in $T_p$ in $J_1(\ell)(\mathbb{F}_q)$

Computations are $J_1(\ell)(\mathbb{F}_q)$ done using the identification:

$$J_1(\ell)(\mathbb{F}_q) = \text{Cl}^0\mathbb{F}_q(X_1(\ell))$$

and using magma’s function field+class group capabilities. There exist explicit algebraic model’s

$$\mathbb{F}_q(X_1(\ell)) \cong \mathbb{F}_q(x)[y] / (f_{\ell}(x, y))$$

that also allows you to go back and forth between zeros of $f_{\ell}(x, y)$ and pairs $(E, P)$.

To compute $T_p(x)$ for $D \in \text{Cl}^0\mathbb{F}_q(X_1(\ell))$, we write $D = \sum n_i Q_i$ with $Q_i$ places of $\mathbb{F}_q(X_1(\ell))$, find the pair $(E_i, P_i)$ corresponding to each $Q_i$) and compute $T_p(E_i, P_i) = \sum_G (E_i / G, P_i \mod G)$
T. and V. Dokchitser’s method for finding Frobenius

Let $P(t) \in \mathbb{Z}[t]$ be a polynomial with splitting field $L$, denote its roots by $a_1, \ldots, a_n$. For $C \subset \text{Gal}(L/\mathbb{Q})$ a conjugacy class and $h \in \mathbb{Q}[X]$ define

$$\Gamma_C^h(t) := \prod_{\sigma \in C} (t - \sum_i h(a_i) \sigma(a_i)) \in \mathbb{Q}[X]$$

**Theorem**

- The set of $h$ with $\deg h \leq n - 1$ s.t. for all $C, C': \text{Res}(\Gamma_C^h, \Gamma_{C'}^h) \neq 0$ is open and Zarisky dense in the polynomials of $\deg \leq n - 1$.

- For $p$ not dividing any of the resultants $\text{Res}(\Gamma_C^h, \Gamma_{C'}^h)$ and also not dividing the leading coefficient of $P(t)$ one has:

$$\text{Frob}_p \in C \iff \Gamma_C(\text{Tr}_{\mathbb{F}_p[t]/(P(t))} h(t)t^p) \equiv 0 \mod p$$
An equation\(^2\) for the projective representation of \(\Delta \mod 31\):

\[
x^{32} - 4x^{31} - 155x^{28} + 713x^{27} - 2480x^{26} + 9300x^{25} - 5921x^{24} + 24707x^{23} + 127410x^{22} - 646195x^{21} + 747906x^{20} - 7527575x^{19} + 4369791x^{18} - 28954961x^{17} - 40645681x^{16} + 66421685x^{15} - 448568729x^{14} + 751001257x^{13} - 1820871490x^{12} + 2531110165x^{11} - 4120267319x^{10} + 4554764528x^9 - 5462615927x^8 + 4607500922x^7 - 4062352344x^6 + 2380573824x^5 - 1492309000x^4 + 521018178x^3 - 201167463x^2 + 20505628x - 1261963
\]

\(^2\)Thanks to Mark van Hoeij for finding this smaller equation, the equation produced by the algorithm had coefficients of 700 digits!
Operation in $J_1(\ell)(\mathbb{F}_q)$ is very slow (using Heß’s algorithm which is in magma), it would be interesting to know whether using Khuri-Makdisi’s algorithm will be faster.

Computing the points in $V_\ell$ modulo a single prime $p$ is possible if $e$ is very small using the current implementation for $\ell = 29$ and $\ell = 41$. But this takes 6 hours for $\ell = 41$ so probably something smarter is needed to reconstruct the entire polynomial. Maybe $p$-adically lifting these points will be faster than trying a lot of different primes.
How to reduce $P(t)$?

The polynomial $P(t)$ has degree $\ell^2 - 1$ and huge coefficients as well. The calculation of $\Gamma_C(t)$ for all the conjugacy classes $C \subset \text{GL}_2(\mathbb{F}_\ell)$, not only took a lot of time but also a lot of memory! Actually the coefficients of $\Gamma_C(t)$ are much bigger than those of $P(t)$. It becomes a bottleneck when dealing with higher levels. So a good algorithm for reducing the size of $P(t)$ (after we have computed it) will be useful.

The Magma code of our implementation can be downloaded from:

$$\text{http://faculty.math.tsinghua.edu.cn/~lisyin/publication.htm}$$
\[ \tau(10^{1000} + 1357) = \pm 18 \mod 31 \]

Thank you very much!