Computing weight 3/2 mock Eisenstein series

For any Gram matrix S (symmetric integer with even diagonal) of signature $1 \bmod 4$ one can find an associated vector-valued mock Eisenstein series of weight 3/2 taking values in $\mathbb{C}[A]$, where A is the group $A = S^{-1}\mathbb{Z}^n/\mathbb{Z}^n$ (with n being the number of rows of S). This is the holomorphic part of a harmonic Maass form. Letting $Q(x) = \frac{1}{2}x^T Sx$ be the associated quadratic form (which induces a quadratic form $Q : A \to \mathbb{Q}/\mathbb{Z}$), this series takes the form

$$E^*_{3/2}(\tau) = \sum_{\gamma \in A} \sum_{n \in \mathbb{Z}-Q(\gamma)} c(n,\gamma)q^n \mathfrak{e}_\gamma + \frac{1}{\sqrt{y}} \sum_{\gamma \in A} \sum_{n \in \mathbb{Z}+Q(\gamma)} a(n,\gamma)\beta(4\pi ny)q^{-n}\mathfrak{e}_\gamma,$$

where $\beta(x)$ is the special function

$$\beta(x) = \frac{1}{16\pi}\int_1^\infty u^{-3/2}e^{-xu}\,\mathrm{d}u.$$

The coefficients $c(n,\gamma)$ are always rational and the coefficients $a(n,\gamma)$ always lie in $\sqrt{|\det(S)|/2} \cdot \mathbb{Q}$. It is known that the series

$$\vartheta(\tau) = \sum_{\gamma \in A} \sum_{n \in \mathbb{Z}+Q(\gamma)} a(n,\gamma)q^n \mathfrak{e}_\gamma$$

is a modular form of weight 1/2 (for the Weil representation attached to the Gram matrix $-S$).

This worksheet gives functions MockEisenstein(g,n,S) and Shadow(g,n,S) to compute $c(n,g)$ and $a(n,g)$, respectively. Here, S should be a Gram matrix of signature $1 \bmod 4$ and $g \in \mathbb{Q}^n$ should be a vector such that Sg is integral. (The function DiscriminantGroup(S) computes a list of representatives g of $S^{-1}\mathbb{Z}^n/\mathbb{Z}^n$. Using ReducedDiscriminantGroup(S) instead computes a list of representatives of pairs $\pm g$; this is useful, because the coefficients $c(n,g) = c(n,-g)$ and $a(n,g) = a(n,-g)$ are equal.)

Example 1. Zagier's Eisenstein series normalized to have constant term 1 appears as the mock Eisenstein series attached to the matrix $S = \begin{pmatrix} 2 \end{pmatrix}$. To compute its coefficients up to the term q^{10} you can use the commands

```
S = matrix([[2]])
for g in ReducedDiscriminantGroup(S):
    offset = frac(g*S*g/2)
    for n in range(10):
        print [g,n-offset,MockEisenstein(g,n-offset,S)]
```

and

```
S = matrix([[2]])
for g in ReducedDiscriminantGroup(S):
    offset = frac(g*S*g/2)
    for n in range(10):
        print [g,n+offset,Shadow(g,n+offset,S)]
```