格基约化算法 flatter 安装及测试

作者： Mengce Zheng

日期： 2024 年 12 月 10 日

本文将在 Windows 操作系统 WSL2 的 Ubuntu 22.04 环境中安装 flatter，具体按照下述步骤进行。

安装 flatter

首先需要下载 flatter 的源代码，可以从 https://github.com/keeganryan/flatter 仓库以 ZIP 格式直接下载，或是通过 git 的方式获取。本文采用第二种方式，执行命令 `git clone https://github.com/keeganryan/flatter.git` 以获取 flatter 的官方仓库，随后在 flatter 文件夹中执行以下命令正式安装：

```
~/flatter$ sudo apt install libgmp-dev libmpfr-dev fplll-tools libfplll-dev
    libeigen3-dev
~/flatter$ mkdir build && cd ./build
~/flatter$ cmake ..
~/flatter$ make
~/flatter$ sudo make install
~/flatter$ sudo ldconfig
~/flatter$ flatter -h
```

最后一句命令将展示 flatter 的具体使用方法：

```
~/flatter$ flatter -h
Usage: flatter [-h] [-v] [-alpha ALPHA | -rhf RHF | -delta DELTA] [-logcond
    LOGCOND] [INFILE [OUTFILE]]
    INFILE -  input lattice (FPLLL format). Defaults to STDIN
    OUTFILE - output lattice (FPLLL format). Defaults to STDOUT
    -h -  help message.
    -v -  verbose output.
    -q -  do not output lattice.
    -p -  output profiles.
    Reduction quality - up to one of the following. Default to RHF 1.0219
```

```
    -alpha ALPHA -  Reduce to given parameter alpha
    -rhf RHF -  Reduce analogous to given root hermite factor
    -delta DELTA -  Reduce analogous to LLL with particular delta (
       approximate)
    -logcond LOGCOND -  Bound on condition number.
```

测试 flatter

flatter 按照 fplll 格式进行格基约化，因此先以 `latticegen` 命令生成特定格基，再以 `flatter` 命令执行格基约化：

```
~/flatter$ latticegen q 4 2 10 b | flatter
[[4 -1 1 0]
[2 10 8 2]
[1 4 -5 -13]
[4 4 -12 12]
]
```

或是

```
~/flatter$ latticegen u 5 10 | flatter
[[-45 137 -61 83 -33]
[-163 -41 133 71 101]
[170 148 185 114 45]
[192 -157 -211 427 230]
[77 -287 246 397 -446]
]
```

当然，也可以将输入格基和输出格基的相关信息打印出来：

```
~/flatter$ latticegen r 8 8 | flatter -v -p
Input lattice of rank 8 and dimension 9
Largest entry is 8 bits in length.
Skipped determining input profile, as input is not lower-triangular.
Target reduction quality alpha = 0.0625081, rhf = 1.0219
Reduction took 23 milliseconds.
Output profile:
0.792481 1.1112 1.02531 1.40012 1.11906 1.0997 1.04368 1.09199
Achieved reduction quality alpha = 0.0468503, rhf = 0.974936
[[0 0 1 0 0 -1 0 0 1]
```

```
[-1  1  0 -1 -1  0  0  0  1]
[0  1 -1  0  0  0 -1  1  1]
[-1 -1 -1  0  2 -1  0  0  0]
[1 -1  1 -1 -1  0  1  1  0]
[0  0 -2  0  0 -1  1  0  1]
[0 -1  1  0  0  1 -1 -1  1]
[1  0  0 -2  1  0 -1  0  1]
]
```

SageMath 适配

因数据格式不同，flatter 无法直接链接 SageMath 使用，需要进行一步格式转换，可按如下操作：

```python
from subprocess import check_output
from re import findall
from sage.all import *

def flatter(M):
  z = "[[" + "]\n[".join("␣".join(map(str, row)) for row in M) + "]]"
  ret = check_output(["flatter"], input=z.encode())
  return matrix(M.nrows(), M.ncols(), map(int, findall(rb"-?\d+", ret)))

L = matrix(ZZ, 5, 5)
for row in range(5):
  for col in range(5):
    L[row, col] = randint(2 ** 5, 2 ** 6)

print(f"{flatter(L)}")
```

注：本文撰写过程中参考网络资源如下：*Sage_10_3_Setup*，*flatter*，*fplll*，*flatter(M)*，如有疑惑可详阅上述文章。