Remarks on the 3D Lattice Sieve

Pavol Zajac

Department of Applied Informatics and IT
Slovak University of Technology

CECC'09

Outline

Number Field Sieve

NFS Number Field Sieve

- Subexponential method for factoring and finding discrete logarithms.

NFS Main principle

- Find enough equations to construct a specific linear system.
- Find enough smooth algebraic numbers in specific number fields.
- Smooth numbers are identified via sieving a predefined space (lattice).

Number Field Sieve

NFS Number Field Sieve

- Subexponential method for factoring and finding discrete logarithms.

NFS Main principle

- Find enough equations to construct a specific linear system.
- Find enough smooth algebraic numbers in specific number fields.
- Smooth numbers are identified via sieving a predefined space (lattice).

Number Field Sieve

NFS Number Field Sieve

- Subexponential method for factoring and finding discrete logarithms.

NFS Main principle

- Find enough equations to construct a specific linear system.
- Find enough smooth algebraic numbers in specific number fields.
- Smooth numbers are identified via sieving a predefined space (lattice).

Algebraic factor base

- Different views in NFS: algebraic numbers, ideals vs. points in space, memory cells.
- Factor base: A set of ideals with the norm $< B$.
- Prime ideal — a set of algebraic numbers:
  - $\mathfrak{p} = 7\mathbb{Z}[\alpha] + (3+\alpha)\mathbb{Z}[\alpha]$
  - a lattice

$$\begin{pmatrix} 7 & 0 \\ 3 & 1 \end{pmatrix}$$

  - every number has norm divisible by 7, main ideal divisible by $\mathfrak{p}$

Algebraic factor base

- Different views in NFS: algebraic numbers, ideals vs. points in space, memory cells.
- Factor base: A set of ideals with the norm $< B$.
- Prime ideal — a set of algebraic numbers:
  - $\mathfrak{p} = 7\mathbb{Z}[\alpha] + (3 + \alpha)\mathbb{Z}[\alpha]$
  - a lattice

$$\begin{pmatrix} 7 & 0 \\ 3 & 1 \end{pmatrix}$$

  - every number has norm divisible by 7, main ideal divisible by $\mathfrak{p}$

Algebraic factor base

- Different views in NFS: algebraic numbers, ideals vs. points in space, memory cells.
- Factor base: A set of ideals with the norm $< B$.
- Prime ideal — a set of algebraic numbers:
  - $\mathfrak{p} = 7\mathbb{Z}[\alpha] + (3 + \alpha)\mathbb{Z}[\alpha]$
  - a lattice
$$\begin{pmatrix} 7 & 0 \\ 3 & 1 \end{pmatrix}$$
  - every number has norm divisible by 7, main ideal divisible by $\mathfrak{p}$

Sieving

- Sieving region:
  - A set of points representing algebraic integers in 2 (or more) number fields.
  - Rectangle, lattice, box, ...
  - Memory cells with precomputed (approximations of) norms...
- Sieving:
  - For each prime ideal in the factor base, mark corresponding points in the sieve region.
  - Check which points are "marked enough"
- Main implementation problem: Large sieve region = large memory requirements.

Sieving

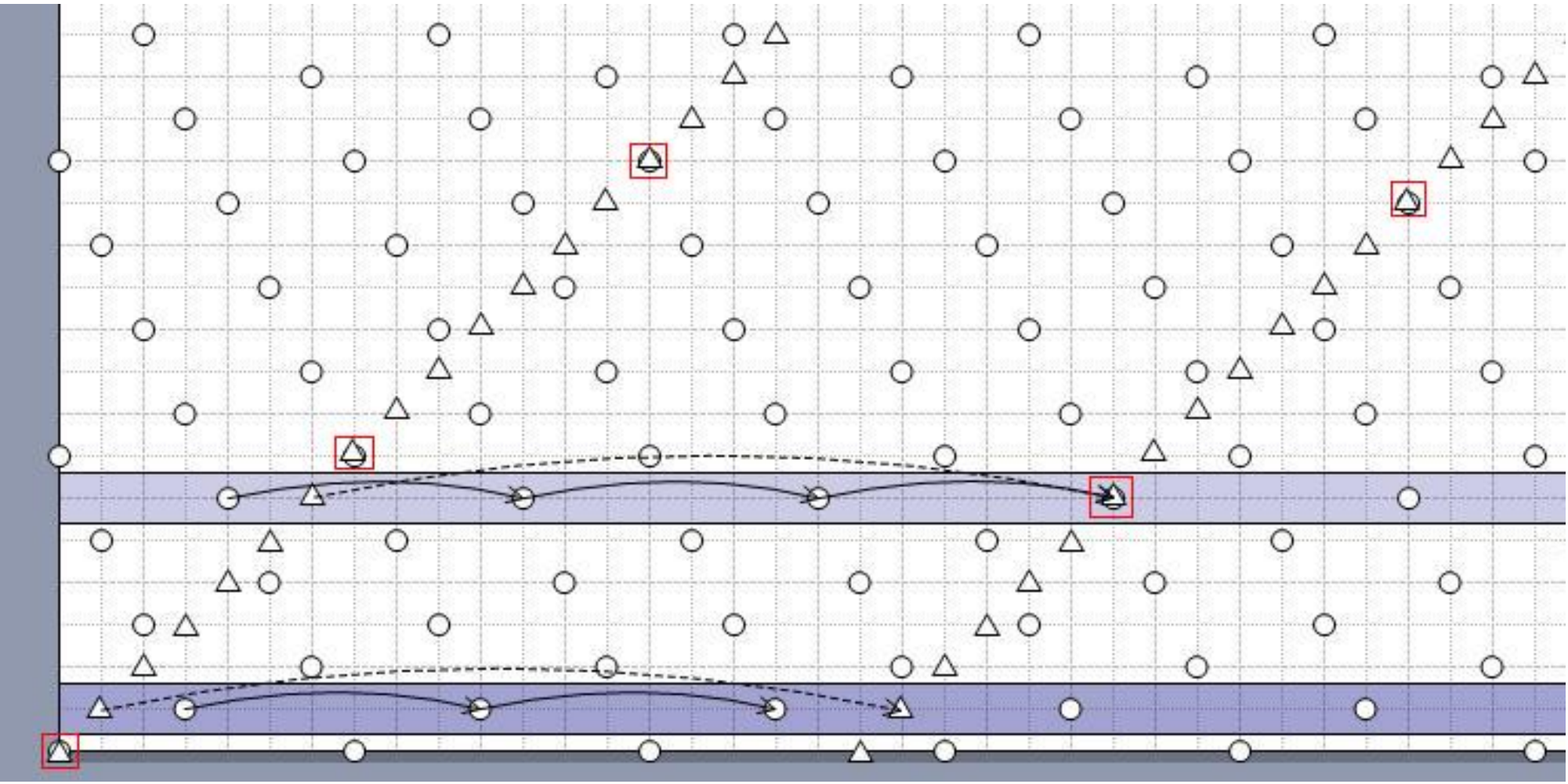
- Sieving region:
  - A set of points representing algebraic integers in 2 (or more) number fields.
  - Rectangle, lattice, box, ...
  - Memory cells with precomputed (approximations of) norms...
- Sieving:
  - For each prime ideal in the factor base, mark corresponding points in the sieve region.
  - Check which points are "marked enough"
- Main implementation problem: Large sieve region = large memory requirements.

Sieving

- Sieving region:
  - A set of points representing algebraic integers in 2 (or more) number fields.
  - Rectangle, lattice, box, ...
  - Memory cells with precomputed (approximations of) norms...
- Sieving:
  - For each prime ideal in the factor base, mark corresponding points in the sieve region.
  - Check which points are "marked enough"
- Main implementation problem: Large sieve region = large memory requirements.

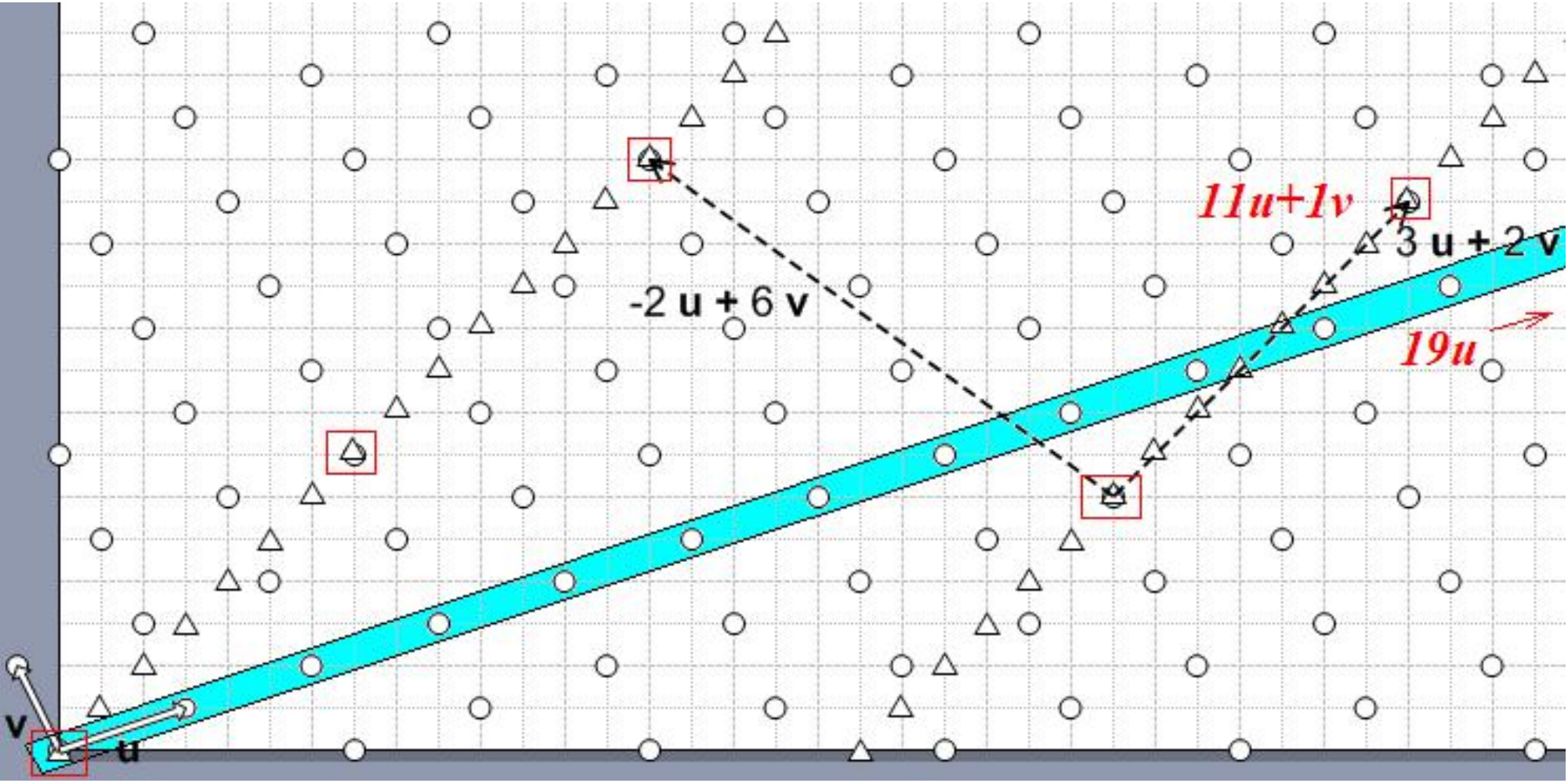
Line sieve

- Main principle: Sieve the region along one-dimensional lines.
- Additional optimizations can be used.



Special-q sieve

- Also: Lattice sieve.
- Main principle: Take a sub-lattice corresponding to a chosen ideal (special-$\mathfrak{q}$) and sieve this smaller region.
- If the region is large enough, can combine with line sieve.
- If the region is small to fit in memory, points of ideals can be enumerated.


11u+1v
3 u + 2 v
-2 u + 6 v
19u
v
u

Lattice sieve

- Main advantage — smaller number of points to examine with a guaranteed factor, smaller work-factor.
- Disadvantages:
  - Time to recompute bases of factor base ideals.
  - Faster growth of the norms (strong limit on sieve bounds).
  - Collisions (requires post-filtering).

Lattice sieve

- Main advantage — smaller number of points to examine with a guaranteed factor, smaller work-factor.
- Disadvantages:
  - Time to recompute bases of factor base ideals.
  - Faster growth of the norms (strong limit on sieve bounds).
  - Collisions (requires post-filtering).

3D sieve

- Required for the NFS for DLP in degree 6 fields.
- Extend the sieve region to a box, recompute 3D bases of ideals, e.g.

$$\begin{pmatrix} 7 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & 3 & 1 \end{pmatrix}$$

- More difficult to estimate the norm.
- Problems with the lattice sieve become more profound.

3D sieve

- Required for the NFS for DLP in degree 6 fields.
- Extend the sieve region to a box, recompute 3D bases of ideals, e.g.

$$\begin{pmatrix} 7 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & 3 & 1 \end{pmatrix}$$

- More difficult to estimate the norm.
- Problems with the lattice sieve become more profound.

3D sieve

- Required for the NFS for DLP in degree 6 fields.
- Extend the sieve region to a box, recompute 3D bases of ideals, e.g.

$$\begin{pmatrix} 7 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & 3 & 1 \end{pmatrix}$$

- More difficult to estimate the norm.
- Problems with the lattice sieve become more profound.

3D sieve

- Required for the NFS for DLP in degree 6 fields.
- Extend the sieve region to a box, recompute 3D bases of ideals, e.g.

$$\begin{pmatrix} 7 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & 3 & 1 \end{pmatrix}$$

- More difficult to estimate the norm.
- Problems with the lattice sieve become more profound.

Line sieving

- Parameters:
  - Region size: $2^{19} \times 2^{13} \times 2^{10}$
  - Factor base size: 893707 ideals ($B = 6500000$)
  - Number of equations: 1077984
  - Total time: 2 087 070 s
- Only 89987 equations with more than 2 factors above 400000-th ideal, ordered by frequency of the occurence in smooth equations.

Line sieving

- Parameters:
  - Region size: $2^{19} \times 2^{13} \times 2^{10}$
  - Factor base size: 893707 ideals ($B = 6500000$)
  - Number of equations: 1077984
  - Total time: 2 087 070 s
- Only 89987 equations with more than 2 factors above 400000-th ideal, ordered by frequency of the occurence in smooth equations.

Line sieving with large factors

- After sieve reduction to first 400000 ideals ($B' = 2757229$):
  - B'-smooth equations: 119053 (11%)
  - B-smooth equations (single large factor): 306198 (28%)
  - Total time: 232358 s (11%, but faster computers...)
- Possible problem with detection:
  - In general: Higher tolerance = longer total time (due to factoring).
  - Different behaviour with large factors.

Line sieving with large factors

- After sieve reduction to first 400000 ideals ($B' = 2757229$):
  - B'-smooth equations: 119053 (11%)
  - B-smooth equations (single large factor): 306198 (28%)
  - Total time: 232358 s (11%, but faster computers...)
- Possible problem with detection:
  - In general: Higher tolerance = longer total time (due to factoring).
  - Different behaviour with large factors.

Line sieving with large factors

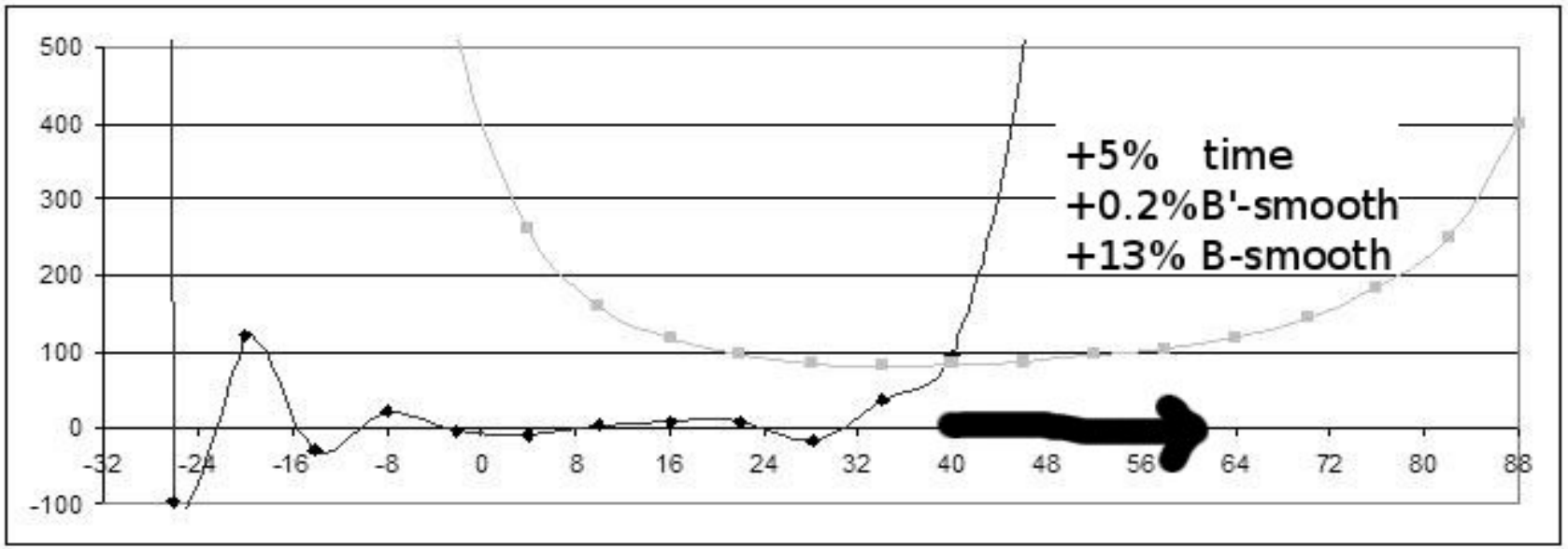


Figure: Additional time to find one smooth equation and average sieving time per smooth equation with a given sieve tolerance.

Line sieving with large factors

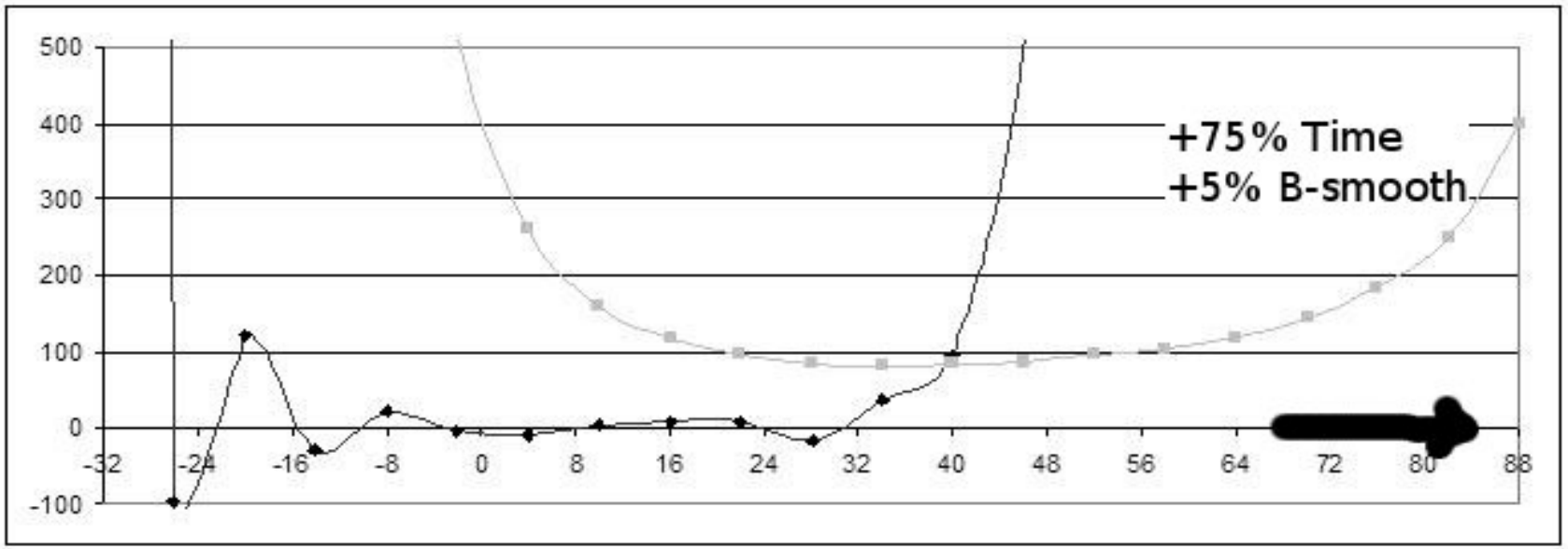


Figure: Additional time to find one smooth equation and average sieving time per smooth equation with a given sieve tolerance.

Lattice sieving — large primes

- Special q-sieve, with large primes ($B' < q < B$), statistics per 20000 special-q's:
  - Equations per special prime: .90 $\rightarrow$ .63 (usable: .35 $\rightarrow$ .19)
  - Much slower due to base transformation (2-times longer than plain sieving, can be fine-tuned)
  - Not enough equations... (approx. 100000 new B'-smooth eqs., 50% missing)
- "Good prime": at least 2 equations: 23% $\rightarrow$ 14%

Lattice sieving — large primes

- Special q-sieve, with large primes ($B' < q < B$), statistics per 20000 special-q's:
  - Equations per special prime: .90 $\rightarrow$ .63 (usable: .35 $\rightarrow$ .19)
  - Much slower due to base transformation (2-times longer than plain sieving, can be fine-tuned)
  - Not enough equations... (approx. 100000 new B'-smooth eqs., 50% missing)
- "Good prime": at least 2 equations: 23% $\rightarrow$ 14%

Lattice sieving — small primes

- Special q-sieve, with 200 small primes, $128 < q < 760$:
  - 0.02% of the whole base
  - covers 80 % of all equations
- Experiment with reduced sieve region:
  - 39383 equations (3.6%) in 117057 s (5.6%)
  - clearly slower than line sieve only (100% time, at most only 64% instead of 80 % of equations)
  - 367 repeated equations (1 %), negligible cost

Lattice sieving — small primes

- Special q-sieve, with 200 small primes, $128 < q < 760$:
  - 0.02% of the whole base
  - covers 80 % of all equations
- Experiment with reduced sieve region:
  - 39383 equations (3.6%) in 117057 s (5.6%)
  - clearly slower than line sieve only (100% time, at most only 64% instead of 80 % of equations)
  - 367 repeated equations (1 %), negligible cost

Summary

- Line sieving:
  - possible speedup using partial base large factors,
  - if we can identify a-priori primes that appear frequently...
- Lattice sieving with large primes is ineffective:
  - too many base transformations
  - sieve region too slow for effective implementation
- Lattice sieving with small primes still too slow:
  - need more effective 3D base transformation

Summary

- Line sieving:
  - possible speedup using partial base large factors,
  - if we can identify a-priori primes that appear frequently...
- Lattice sieving with large primes is ineffective:
  - too many base transformations
  - sieve region too slow for effective implementation
- Lattice sieving with small primes still too slow:
  - need more effective 3D base transformation

Summary

- Line sieving:
  - possible speedup using partial base large factors,
  - if we can identify a-priori primes that appear frequently...
- Lattice sieving with large primes is ineffective:
  - too many base transformations
  - sieve region too slow for effective implementation
- Lattice sieving with small primes still too slow:
  - need more effective 3D base transformation

Summary

- Line sieving:
  - possible speedup using partial base large factors,
  - if we can identify a-priori primes that appear frequently...
- Lattice sieving with large primes is ineffective:
  - too many base transformations
  - sieve region too slow for effective implementation
- Lattice sieving with small primes still too slow:
  - need more effective 3D base transformation

Conclusions

- Direct application of the lattice sieve in 3D NFS is not effective.
- Possible speedup: More effective factoring of sets of semi-smooth numbers with a partial sieve.

`http://www.kaivt.elf.stuba.sk/kaivt/Vyskum/XTRDL`

Conclusions

- Direct application of the lattice sieve in 3D NFS is not effective.
- Possible speedup: More effective factoring of sets of semi-smooth numbers with a partial sieve.

`http://www.kaivt.elf.stuba.sk/kaivt/Vyskum/XTRDL`