Implantation de SLZ

Rappels sur l'architecture de l'algorithme

Trois parties "indépendantes" :

- Génération des polynômes ;

Rappels sur l'architecture de l'algorithme

Trois parties "indépendantes" :

- Génération des polynômes ;
- Construction d'un réseau et recherche d'un petit vecteur (LLL) ;

Rappels sur l'architecture de l'algorithme

Trois parties "indépendantes" :

- Génération des polynômes ;
- Construction d'un réseau et recherche d'un petit vecteur (LLL) ;
- Résolution d'un système polynomial (2 équations, 2 inconnues).

Rappels sur l'architecture de l'algorithme

Trois parties "indépendantes" :

- Génération des polynômes ;
- Construction d'un réseau et recherche d'un petit vecteur (LLL) ;
- Résolution d'un système polynomial (2 équations, 2 inconnues).

... plus une surcouche de l'ensemble chargée

- d'adapter les paramètres ;

Rappels sur l'architecture de l'algorithme

Trois parties "indépendantes" :

- Génération des polynômes ;
- Construction d'un réseau et recherche d'un petit vecteur (LLL) ;
- Résolution d'un système polynomial (2 équations, 2 inconnues).

... plus une surcouche de l'ensemble chargée

- d'adapter les paramètres ;
- éventuellement, de choisir les stratégies (de façon adaptative) ;

Rappels sur l'architecture de l'algorithme

Trois parties "indépendantes" :

- Génération des polynômes ;
- Construction d'un réseau et recherche d'un petit vecteur (LLL) ;
- Résolution d'un système polynomial (2 équations, 2 inconnues).

... plus une surcouche de l'ensemble chargée

- d'adapter les paramètres ;
- éventuellement, de choisir les stratégies (de façon adaptative) ;
- et de distribuer les intervalles.

Rappels sur l'architecture de l'algorithme

Trois parties "indépendantes" :

- Génération des polynômes ;
- Construction d'un réseau et recherche d'un petit vecteur (LLL) ;
- Résolution d'un système polynomial (2 équations, 2 inconnues).

... plus une surcouche de l'ensemble chargée

- d'adapter les paramètres ;
- éventuellement, de choisir les stratégies (de façon adaptative) ;
- et de distribuer les intervalles.
- éventuellement, vérifie les candidats pires cas à la volée.

Carrosserie

État courant : à faire (en dernier).

- Langage de haut niveau ;

Carrosserie

État courant : à faire (en dernier).

- Langage de haut niveau ;
- Gérer la distribution du calcul.

Phase 1

État : TODO.

- En première approximation, Taylor ;

Phase 1

État : TODO.

- En première approximation, Taylor ;
- Avec coefficients arrondis (comment ?) ;

Phase 1

État : TODO.

- En première approximation, Taylor ;
- Avec coefficients arrondis (comment ?) ;
- ... tout en contrôlant l'erreur (pour qu'elle reste petite devant la taille des pires cas cherchés) ;

Phase 1

État : TODO.

- En première approximation, Taylor ;
- Avec coefficients arrondis (comment ?) ;
- ... tout en contrôlant l'erreur (pour qu'elle reste petite devant la taille des pires cas cherchés) ;
- Perspectives : choisir de meilleurs polynômes. Est-ce que ca vaut le coup ?

Phase 2

État : prototype écrit par Andy.

- Phase a priori dominante de l'algorithme (LLL) ;

Phase 2

État : prototype écrit par Andy.

- Phase a priori dominante de l'algorithme (LLL) ;
- Renvoie FAIL ou deux vecteurs courts ;

Phase 2

État : prototype écrit par Andy.

- Phase a priori dominante de l'algorithme (LLL) ;
- Renvoie FAIL ou deux vecteurs courts ;
- Perspectives :
  - tester,

Phase 2

État : prototype écrit par Andy.

- Phase a priori dominante de l'algorithme (LLL) ;
- Renvoie FAIL ou deux vecteurs courts ;
- Perspectives :
  - tester,
  - comprendre la dépendance du temps de calcul en le choix des paramètres

Phase 2

État : prototype écrit par Andy.

- Phase a priori dominante de l'algorithme (LLL) ;
- Renvoie FAIL ou deux vecteurs courts ;
- Perspectives :
  - tester,
  - comprendre la dépendance du temps de calcul en le choix des paramètres
  - tester des améliorations algorithmiques naives,

Phase 2

État : prototype écrit par Andy.

- Phase a priori dominante de l'algorithme (LLL) ;
- Renvoie FAIL ou deux vecteurs courts ;
- Perspectives :
  - tester,
  - comprendre la dépendance du temps de calcul en le choix des paramètres
  - tester des améliorations algorithmiques naives,
  - comprendre si et comment les travaux récents autour de LLL peuvent aider.

Phase 3

État courant : (à peu près) opérationnel.

- Environ 500 lignes de code C (bcp de code dupliqué) ;

Phase 3

État courant : (à peu près) opérationnel.

- Environ 500 lignes de code C (bcp de code dupliqué) ;
- Renvoie FAIL ou un ensemble de candidats pires cas.

Phase 3

État courant : (à peu près) opérationnel.

- Environ 500 lignes de code C (bcp de code dupliqué) ;
- Renvoie FAIL ou un ensemble de candidats pires cas.
- Très efficace en pratique, encore un peu optimisable.

Code de vérification de certificat

- Devra être redéveloppé intégralement ;

Code de vérification de certificat

- Devra être redéveloppé intégralement ;
- Calcule les combinaisons linéaires P_1, P_2 de polynômes décrite par les deux vecteurs renvoyés par LLL (stockés) ;

Code de vérification de certificat

- Devra être redéveloppé intégralement ;
- Calcule les combinaisons linéaires P_1, P_2 de polynômes décrite par les deux vecteurs renvoyés par LLL (stockés) ;
- Vérifie que P_1, P_2 ont petite norme 1 ;

Code de vérification de certificat

- Devra être redéveloppé intégralement ;
- Calcule les combinaisons linéaires P_1, P_2 de polynômes décrite par les deux vecteurs renvoyés par LLL (stockés) ;
- Vérifie que P_1, P_2 ont petite norme 1 ;
- Compte les solutions du système $P_1(X, Y) = P_2(X, Y) \bmod p$;

Code de vérification de certificat

- Devra être redéveloppé intégralement ;
- Calcule les combinaisons linéaires P_1, P_2 de polynômes décrite par les deux vecteurs renvoyés par LLL (stockés) ;
- Vérifie que P_1, P_2 ont petite norme 1 ;
- Compte les solutions du système $P_1(X, Y) = P_2(X, Y) \bmod p$;
- Vérifie qu'elles sont simples ;

Code de vérification de certificat

- Devra être redéveloppé intégralement ;
- Calcule les combinaisons linéaires P_1, P_2 de polynômes décrite par les deux vecteurs renvoyés par LLL (stockés) ;
- Vérifie que P_1, P_2 ont petite norme 1 ;
- Compte les solutions du système $P_1(X, Y) = P_2(X, Y) \bmod p$;
- Vérifie qu'elles sont simples ;
- Vérifie que les racines modulo p^k (stockées) sont bien des racines et qu'il y en a le bon nombre.