

# Une nouvelle interface pour la bibliothèque Sollya

Christoph Lauter

Équipe PEQUAN, LIP6, UPMC Paris 6

`christoph.lauter@lip6.fr`

Journées ANR TaMaDi, Lyon, 13 décembre 2011



# Qu'est-ce que Sollya ?

- Sollya :
  - un **outil intégré** pour la conception d'algorithmes numériques
  - basé sur des algorithmes multiprécision
  - avec une **garantie pour les résultats**
    - ▶ évaluation en **arrondi fidèle** avec adaptation de la précision,
    - ▶ algorithmes validés : normes sup, nombres de racines de polynômes etc.

# Qu'est-ce que Sollya ?

- Sollya :
  - un **outil intégré** pour la conception d'algorithmes numériques
  - basé sur des algorithmes multiprécision
  - avec une **garantie pour les résultats**
    - ▶ évaluation en **arrondi fidèle** avec adaptation de la précision,
    - ▶ algorithmes validés : normes sup, nombres de racines de polynômes etc.
  - Un petit noyau pour les *manipulations formelles*
    - ▶ manipulation d'expressions de fonctions,
    - ▶ calcul sur polynômes,
    - ▶ chaînes, listes, structures, pattern-matching.

# Qu'est-ce que Sollya ?

- Sollya :
  - un **outil intégré** pour la conception d'algorithmes numériques
  - basé sur des algorithmes multiprécision
  - avec une **garantie pour les résultats**
    - ▶ évaluation en **arrondi fidèle** avec adaptation de la précision,
    - ▶ algorithmes validés : normes sup, nombres de racines de polynômes etc.
  - Un petit noyau pour les *manipulations formelles*
    - ▶ manipulation d'expressions de fonctions,
    - ▶ calcul sur polynômes,
    - ▶ chaînes, listes, structures, pattern-matching.
  - Du support pour *l'arithmétique flottante*
    - ▶ binary16 à binary128 tous supportés,
    - ▶ arrondis garantis d'expressions,
    - ▶ support pour la génération de code.

# Qu'est-ce que Sollya ?

- Sollya :
  - un **outil intégré** pour la conception d'algorithmes numériques
  - basé sur des algorithmes multiprécision
  - avec une **garantie pour les résultats**
    - ▶ évaluation en **arrondi fidèle** avec adaptation de la précision,
    - ▶ algorithmes validés : normes sup, nombres de racines de polynômes etc.
  - Un petit noyau pour les *manipulations formelles*
    - ▶ manipulation d'expressions de fonctions,
    - ▶ calcul sur polynômes,
    - ▶ chaînes, listes, structures, pattern-matching.
  - Du support pour *l'arithmétique flottante*
    - ▶ binary16 à binary128 tous supportés,
    - ▶ arrondis garantis d'expressions,
    - ▶ support pour la génération de code.
  - URL : `http://sollya.gforge.inria.fr/`

# Quel intérêt pour TaMaDi ?

- L'intérêt pour TaMaDi :
  - Approximation
    - ▶ La fonction analysée est remplacée par des polynômes.
    - ▶ Sollya intègre un algorithme rapide d'approximation minimax.
  - Choix du degré du polynôme approximateur
    - ▶ Le degré du polynôme doit être choisi pour une précision cible.
    - ▶ Une boucle bête *minimax - norme sup* est lente.
    - ▶ Sollya propose la procédure *guessdegree* optimisée.
  - Bornes sur l'erreur d'approximation commise
    - ▶ Un polynôme approximateur n'est pas la fonction même.
    - ▶ L'erreur d'approximation  $\varepsilon = p/f - 1$  doit être bornée.
    - ▶ Pour TaMaDi, la correction de cette borne est primordiale.
    - ▶ Le groupe *Coqapprox* détient la vérité absolue.
    - ▶ Sollya intègre des algorithmes de norme sup rapides et validés.
  - C'est du logiciel libre, on peut patcher tout bogue ensemble

# Quel intérêt pour TaMaDi ?

- L'intérêt pour TaMaDi :
  - Approximation
    - ▶ La fonction analysée est remplacée par des polynômes.
    - ▶ Sollya intègre un algorithme rapide d'approximation minimax.
  - Choix du degré du polynôme approximateur
    - ▶ Le degré du polynôme doit être choisi pour une précision cible.
    - ▶ Une boucle bête *minimax - norme sup* est lente.
    - ▶ Sollya propose la procédure *guessdegree* optimisée.
  - Bornes sur l'erreur d'approximation commise
    - ▶ Un polynôme approximateur n'est pas la fonction même.
    - ▶ L'erreur d'approximation  $\varepsilon = p/f - 1$  doit être bornée.
    - ▶ Pour TaMaDi, la correction de cette borne est primordiale.
    - ▶ Le groupe *Coqapprox* détient la vérité absolue.
    - ▶ Sollya intègre des algorithmes de norme sup rapides et validés.
  - C'est du logiciel libre, on peut patcher tout bogue ensemble
- Sollya est aussi pratique pour vite tester quelques idées.

## Sous quelle forme ça se présente ?

- Sollya est avant tout un outil interactif :
  - L'outil vient avec un langage de script
  - Le langage est très faiblement typé
  - La plupart des commandes sont polymorphes :
    - ▶ `evaluate(exp(x),5) ;`
    - ▶ `evaluate(exp(x),[-4 ;4]) ;`



# Sous quelle forme ça se présente ?

- Sollya est avant tout un outil interactif :
  - L'outil vient avec un langage de script
  - Le langage est très faiblement typé
  - La plupart des commandes sont polymorphes :
    - ▶ `evaluate(exp(x),5) ;`
    - ▶ `evaluate(exp(x),[-4 ;4]) ;`
- Sollya peut être aussi utilisé en mode bibliothèque
  - La `libsollya` a été créée en 2008...
  - ... en ouvrant le ventre de Sollya et en exposant ses tripes.
    - ▶ `int evaluateFaithfulWithCutOffFast(mpfr_t y,  
node *f, node *d, mpfr_t x, mpfr_t c, mp_prec_t p) ;`
  - La bibliothèque n'était pas officiellement supportée

# Sous quelle forme ça se présente ?

- Sollya est avant tout un outil interactif :
  - L'outil vient avec un langage de script
  - Le langage est très faiblement typé
  - La plupart des commandes sont polymorphes :
    - ▶ `evaluate(exp(x),5) ;`
    - ▶ `evaluate(exp(x),[-4 ;4]) ;`
- Sollya peut être aussi utilisé en mode bibliothèque
  - La `libsollya` a été créée en 2008...
  - ... en ouvrant le ventre de Sollya et en exposant ses tripes.
    - ▶ `int evaluateFaithfulWithCutOffFast(mpfr_t y, node *f, node *d, mpfr_t x, mpfr_t c, mp_prec_t p) ;`
  - La bibliothèque n'était pas officiellement supportée
  - Pour l'efficacité, il faut un mode bibliothèque raisonnable
    - ▶ avec une interface stable dans le temps
    - ▶ de la documentation et
    - ▶ un support raisonnable.

# Une nouvelle interface pour libsollya

- Idée pour une nouvelle interface pour la bibliothèque Sollya :
  - Reproduire le polymorphisme du mode interactif en C
  - Toutes les fonctions manipulent un même type polymorphe
    - ▶ `typedef something sollya_obj_t ;`
  - Elles prennent les mêmes arguments qu'en interactif.
  - Pour les commandes variadiques, elles sont variadiques en C :
    - ▶ `sollya_obj_t sollya_lib_min(sollya_obj_t, ...) ;`

# Une nouvelle interface pour libsollya

- Idée pour une nouvelle interface pour la bibliothèque Sollya :
  - Reproduire le polymorphisme du mode interactif en C
  - Toutes les fonctions manipulent un même type polymorphe
    - ▶ `typedef something sollya_obj_t ;`
  - Elles prennent les mêmes arguments qu'en interactif.
  - Pour les commandes variadiques, elles sont variadiques en C :
    - ▶ `sollya_obj_t sollya_lib_min(sollya_obj_t, ...) ;`
- Des **accesseurs** pour les objets `sollya_obj_t`
  - Construction de `sollya_obj_t` à partir de `mpfr_t`, `mpfi_t`, `double`, `int`, `char*`...
  - Affichage de `sollya_obj_t`
  - Accesseurs pour le contenu des `sollya_obj_t`

# Une nouvelle interface pour libsollya

- Idée pour une nouvelle interface pour la bibliothèque Sollya :
  - Reproduire le polymorphisme du mode interactif en C
  - Toutes les fonctions manipulent un même type polymorphe
    - ▶ `typedef something sollya_obj_t ;`
  - Elles prennent les mêmes arguments qu'en interactif.
  - Pour les commandes variadiques, elles sont variadiques en C :
    - ▶ `sollya_obj_t sollya_lib_min(sollya_obj_t, ...) ;`
- Des **accesseurs** pour les objets `sollya_obj_t`
  - Construction de `sollya_obj_t` à partir de `mpfr_t`, `mpfi_t`, `double`, `int`, `char*`...
  - Affichage de `sollya_obj_t`
  - Accesseurs pour le contenu des `sollya_obj_t`
- Des **convenience functions** pour faciliter la programmation
  - Évaluation de fonctions en des points ou sur des intervalles
  - *Parsing* de chaînes de caractères
  - Construction d'expressions de fonctions

## ● Pérennité

- L'interface en mode interactif est et reste stable.
- La nouvelle bibliothèque est calée sur le mode interactif.
- Elle est suffisamment abstraite pour permettre des extensions.
- Votre investissement à coder en `libsollya` ne sera pas perdu.

## ● Documentation

- Sollya vient avec une documentation compréhensive.
- On étendra cette doc pour les spécificités de la bibliothèque.
- Cela ne sera pas difficile à faire :
  - ▶ Interactif : `evaluate({fonction}, {constant})` :  
`(fonction, constant) -> constant`
  - ▶ Bibliothèque : `sollya_obj_t`  
`sollya_lib_evaluate(sollya_obj_t, sollya_obj_t)` ;

- Une bibliothèque conçue *a posteriori*
  - Sollya a d'abord été un outil interactif,
  - le code de l'interface bibliothèque sera d'abord inefficace,
  - mais on pourra y remédier au long terme.
  
- Une bibliothèque non-réentrante
  - le cœur de Sollya n'a jamais été conçu pour le parallélisme
  - même si ça représenterait un vrai intérêt
  - ⇒ travaux futurs.

# Problèmes déjà connus

- Une bibliothèque conçue *a posteriori*
  - Sollya a d'abord été un outil interactif,
  - le code de l'interface bibliothèque sera d'abord inefficace,
  - mais on pourra y remédier au long terme.
  
- Une bibliothèque non-réentrante
  - le cœur de Sollya n'a jamais été conçu pour le parallélisme
  - même si ça représenterait un vrai intérêt
  - $\Rightarrow$  travaux futurs.
  
- Trouvez-vous encore quelque chose ?



# Appel à participation

- État actuel de l'interface de la bibliothèque
  - L'interface est définie et l'entête `.h` est implémentée
  - On a déjà fait quelques consultations avec nos utilisateurs
    - ▶ Ceci a permis de régler quelques oublis
  - L'implémentation est censée être finie fin Janvier
  - On envisage une release fin Février.
- On ne pourra pas faire sans vous !
  - Le développement actuel répond à un réel besoin.
    - ▶ travaux sur l'approximation polynomiale pour la L-méthode
  - Représentée, la bibliothèque ne sera que péniblement modifiable.
  - Donc un appel à tous :
    - ▶ Faites des check-outs du SVN
    - ▶ Envoyez-nous des commentaires
    - ▶ Appelez
    - ▶ Discutons

# Discussion

Merci pour votre attention !

Des questions ?