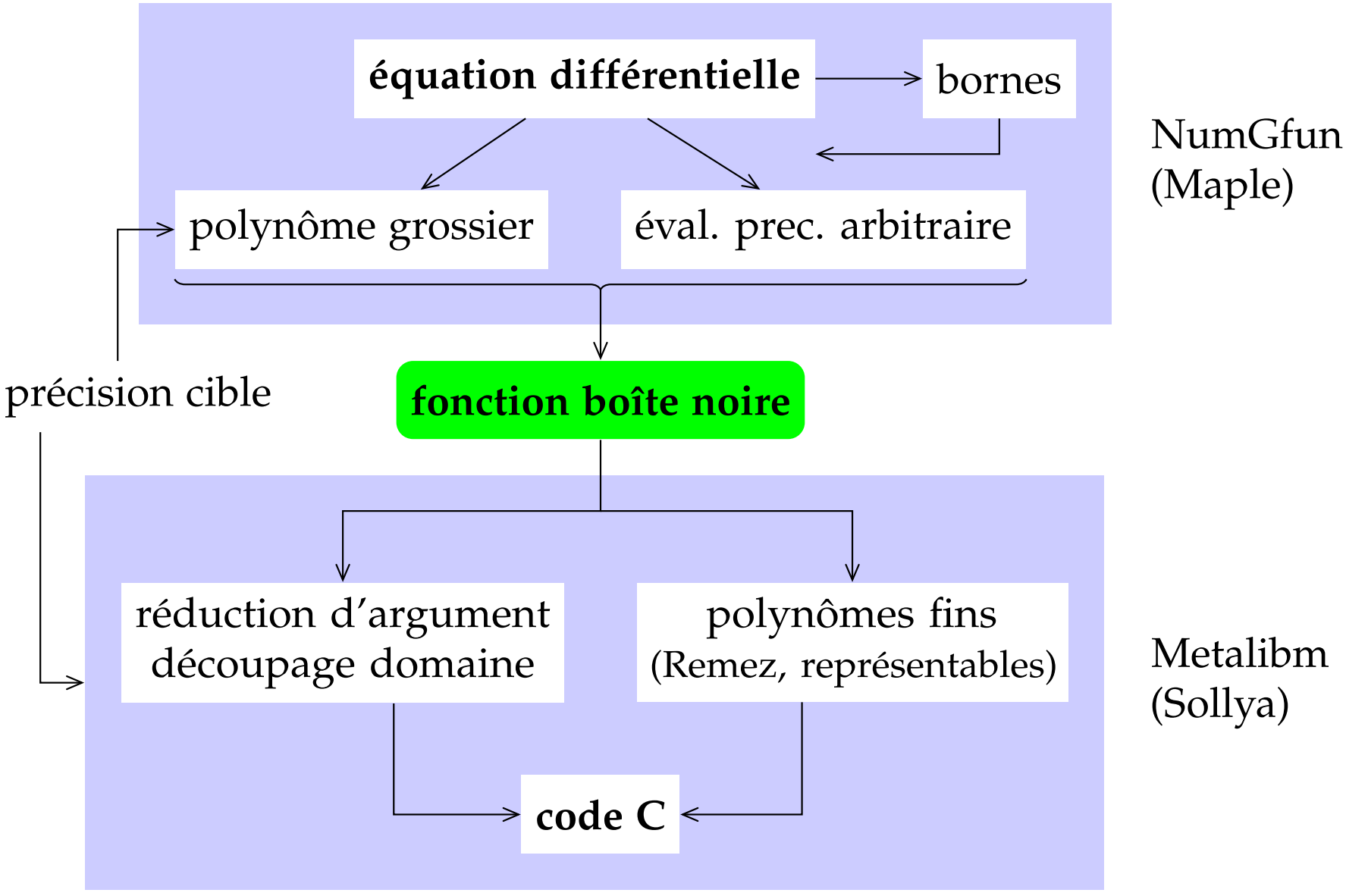
FRANKENSTEIN

Implémentation automatique de fonctions données par des équations différentielles

Christoph Lauter Marc Mezzarobba

Metalibm kick-off meeting, 2013-01-22

Ce qu'on a à ce stade



Metalibm à la Parisienne

- Metalibm version UPMC/LIP6
- Commencé en 2007/08 à Lyon, encore en thèse
- Code écrit en langage Sollya
- Sollya comme moteur de calcul
- Synchronisation entre développement Metalibm et Sollya
- C'est ce Metalibm qu'on essaie vendre à RedHat (libc).

Les polynômes au cœur de Metalibm

- Pour commencer, on a appris à Metalibm de générer du code
  - valide sur des domaines $I = [a; b]$ très petits
  - pour des fonctions ayant au plus un zéro dans l'intervalle.

- Le code consiste en l'évaluation d'un polynôme.

- Exemple:
  - $f = \sin$
  - $I = [-2^{-5}; 2^{-5}]$

```
#define f_approx_coeff_1h 9.9999999835072...e-01
#define f_approx_coeff_3h -1.666568399745...e-01

f_approx_x_0_pow2h = x * x;

f_approx_t_1_0h = f_approx_coeff_3h;
f_approx_t_2_0h = f_approx_t_1_0h * f_approx_x_0_pow2h;
f_approx_t_3_0h = f_approx_coeff_1h + f_approx_t_2_0h;
f_approx_t_4_0h = f_approx_t_3_0h * x;
*f_approx_resh = f_approx_t_4_0h;
```

Metalibm et les réduction d'argument

- Ensuite on est passé à des domaines plus larges:
  - Génération d'une réduction d'argument

$$e^x = e^{\lfloor x \rceil} \cdot e^{x - \lfloor x \rceil} = e^n \cdot e^r$$

  - Recherche de la réduction appropriée:
    - Analyse numérique de la fonction en entrée – oui, par défaut
    - Simple pattern-matching – oui, ça peut être activé
    - Annotation de la fonction avec des propriétés algébriques – non
  - Rapports "optimaux" entre tables et polynômes

Metalibm et les réduction d'argument

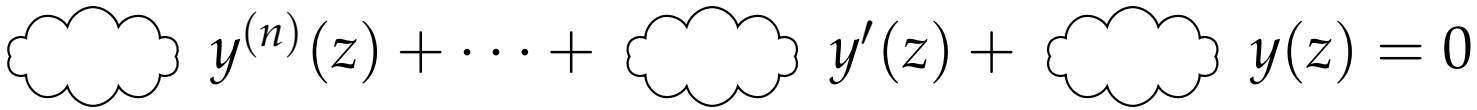
- Ensuite on est passé à des domaines plus larges:
  - Génération d'une réduction d'argument

$$e^x = e^{\lfloor x \rceil} \cdot e^{x - \lfloor x \rceil} = e^n \cdot e^r$$

  - Recherche de la réduction appropriée:
    - Analyse numérique de la fonction en entrée – oui, par défaut
    - Simple pattern-matching – oui, ça peut être activé
    - Annotation de la fonction avec des propriétés algébriques – non
  - Rapports "optimaux" entre tables et polynômes
- Metalibm: un générateur prototype fonctionnel pour
  - $f(x) = a^x$ sur le domaine des doubles
  - $f(x) = \log_a x$ sur le domaines des doubles
  - $f(x) = x^a$ avec a une constante
  - f périodique sur des domaines moyens
  - f continue sur des domaines petits/moyens
  - $f = g(h)$ avec g et h pour lesquelles ça marche
  - etc.

NumGfun

- Objets : fonctions D-finies
  (= solutions d'équa diff linéaires à coeff polynomiaux)



NumGfun

- Objets : fonctions D-finies
  (= solutions d'équa diff linéaires à coeff polynomiaux)

$$\text{poly}(z)\, y^{(n)}(z) + \cdots + \text{poly}(z)\, y'(z) + \text{poly}(z)\, y(z) = 0$$

NumGfun

- Objets : fonctions D-finies
  (= solutions d'équa diff linéaires à coeff polynomiaux)

$$\text{poly}(z)\, y^{(n)}(z) + \cdots + \text{poly}(z)\, y'(z) + \text{poly}(z)\, y(z) = 0$$

- Évaluation garantie à précision (absolue) arbitraire

```
> deq := { diff(y(z),z) = z*y(z), y(0) = 1 }:
> evaldiffeq(deq, y(z), I+1, 40);
0.54030230586813971740093660744297660373 23 +
    0.84147098480789650665250232163029899962 26 I
```

- Polynômes précalculés pour certaines évaluations
  Séries de Taylor tronquées pour garantir une précision ϵ

- Bornes sur les coefficients de séries de Taylor
  ...et autres déduites de celles-là