

Analyse et implantation d'algorithmes rapides pour l'évaluation polynomiale sur les nombres flottants

Guillaume Revy Claude-Pierre Jeannerod

Équipe Arénaire (LIP-ENS-Lyon)

26/06/2006



Contexte

Problème : évaluer une fonction élémentaire (log, exp, cos, ...) le plus rapidement possible, à une certaine précision (projet CR-Libm).

Les processeurs actuels :

- ▶ peuvent effectuer : +, ×
 - ⇒ évaluer une fonction élémentaire $\approx$ évaluer un polynôme d'approximation : $\sum_{i=0}^n a_i x^i$
- ▶ fonctionnent en virgule flottante
 - ⇒ être conscient de la notion de précision d'évaluation d'un polynôme
- ▶ sont pipelinés ou superscalaires (plusieurs opérations parallèles)
 - ⇒ évaluer rapidement les polynômes en tenant compte des architectures disponibles

État de l'art

Soit $a(x) = a_3x^3 + a_2x^2 + a_1x + a_0$. Quelle schéma utiliser pour évaluer ce polynôme ?

- ▶ **Schéma de Horner** : $((a_3x + a_2)x + a_1)x + a_0$

- n multiplications / n additions
- méthode précise mais séquentielle

- ▶ **Schéma de Estrin** : $(a_3x + a_2)x^2 + (a_1x + a_0)$

- $n + \log(n + 1) - 1$ multiplications / n additions
- plus de multiplications que Horner mais évaluation parallélisée
- utilisée sur le processeur Itanium d'Intel depuis 2000

Problématique

Mais il existe des méthodes plus rapides en nombres d'opérations.

- ⇒ nécessitent une reformulation du polynôme : **préconditionnement**
- ⇒ effectuée une seule fois avant l'évaluation

- ▶ **Schéma de Knuth & Eve** : $\frac{n}{2}$ multiplications / n additions
- ▶ **Schéma de Paterson & Stockmeyer** : $\frac{n}{2} + \log(n)$ multiplications / $\frac{3n}{2}$ additions

La littérature présente ces schémas de manière théorique. Qu'en est-il en termes de précision d'évaluation ? Pourquoi ne pas utiliser ces méthodes rapides ? Peut-on être plus rapide que Horner et aussi précis ?

Plan de la soutenance

Étude de l'algorithme de Knuth & Eve

Description de l'algorithme de Paterson & Stockmeyer

Application à CR-Libm : utilisation de schémas hybrides

Conclusion & Perspectives

Plan de la soutenance

Étude de l'algorithme de Knuth & Eve

Description de l'algorithme de Paterson & Stockmeyer

Application à CR-Libm : utilisation de schémas hybrides

Conclusion & Perspectives

Phase de préconditionnement

Entrées : $a_n, a_{n-1}, \dots, a_1, a_0$
Sortie : $\alpha_1, \dots, \alpha_m, \beta_0, \dots, \beta_m$

1. On découpe a en une partie paire g et une partie impaire h

$$a(x) = g(y) + x \cdot h(y), \text{ avec } y = x^2 \text{ et } m = \deg(h)$$

2. On suppose que h n'a que des racines réelles : $\alpha_1, \dots, \alpha_m$

$$a(x) = g(y) + a_n x \prod_{i=1}^m (y - \alpha_i) = g(y) + a_n x (y - \alpha_1) \dots (y - \alpha_m)$$

→ idée : choisir une racine α_i et mettre $(y - \alpha_i)$ en facteur

→ β_0 = reste de la division de g par $(y - \alpha_i)$

$$a(x) = (g^{(1)}(y) + a_n x \prod_{i=2}^m (y - \alpha_i))(y - \alpha_1) + \beta_0$$

Phase de préconditionnement

Entrées : $a_n, a_{n-1}, \dots, a_1, a_0$
Sortie : $\alpha_1, \dots, \alpha_m, \beta_0, \dots, \beta_m$

1. On découpe a en une partie paire g et une partie impaire h

$$a(x) = g(y) + x \cdot h(y), \text{ avec } y = x^2 \text{ et } m = \deg(h)$$

2. On suppose que h n'a que des racines réelles : $\alpha_1, \dots, \alpha_m$

$$a(x) = g(y) + a_n x \prod_{i=1}^m (y - \alpha_i) = g(y) + a_n x (y - \alpha_1) \dots (y - \alpha_m)$$

- idée : choisir une racine α_i et mettre $(y - \alpha_i)$ en facteur
- β_0 = reste de la division de g par $(y - \alpha_i)$

$$a(x) = (g^{(1)}(y) + a_n x \prod_{i=2}^m (y - \alpha_i))(y - \alpha_1) + \beta_0$$

Phase de préconditionnement

Entrées : $a_n, a_{n-1}, \dots, a_1, a_0$
Sortie : $\alpha_1, \dots, \alpha_m, \beta_0, \dots, \beta_m$

3. Par divisions successives de g par $(y - \alpha_i)$, on obtient les β_i .

4. Finalement, avec $y = x^2$ et $\beta_m = a_n x + a_{n-1}$, la nouvelle formule de $a(x)$ est :

$$a(x) = (\dots ((\beta_m(y - \alpha_m) + \beta_{m-1})(y - \alpha_{m-1}) + \beta_{m-2}) \dots)(y - \alpha_1) + \beta_0$$

⇒ préconditionnement non rationnel

- les α_i sont obtenus en valeurs approchées
- les β_i sont par conséquent valeurs approchées

Phase de préconditionnement

Entrées : $a_n, a_{n-1}, \dots, a_1, a_0$
Sortie : $\alpha_1, \dots, \alpha_m, \beta_0, \dots, \beta_m$

3. Par divisions successives de g par $(y - \alpha_i)$, on obtient les β_i .
4. Finalement, avec $y = x^2$ et $\beta_m = a_n x + a_{n-1}$, la nouvelle formule de $a(x)$ est :

$$a(x) = (\dots ((\beta_m(y - \alpha_m) + \beta_{m-1})(y - \alpha_{m-1}) + \beta_{m-2}) \dots)(y - \alpha_1) + \beta_0$$

⇒ préconditionnement non rationnel

- les α_i sont obtenus en valeurs approchées
- les β_i sont par conséquent valeurs approchées

Phase de préconditionnement

Entrées : $a_n, a_{n-1}, \dots, a_1, a_0$
Sortie : $\alpha_1, \dots, \alpha_m, \beta_0, \dots, \beta_m$

3. Par divisions successives de g par $(y - \alpha_i)$, on obtient les β_i .
4. Finalement, avec $y = x^2$ et $\beta_m = a_n x + a_{n-1}$, la nouvelle formule de $a(x)$ est :

$$a(x) = \left(\dots ((\beta_m(y - \alpha_m) + \beta_{m-1})(y - \alpha_{m-1}) + \beta_{m-2}) \dots \right) (y - \alpha_1) + \beta_0$$

⇒ préconditionnement non rationnel

- les α_i sont obtenus en valeurs approchées
- les β_i sont par conséquent valeurs approchés

Cas général (avec $n \geq 3$)

- ▶ Cette méthode se généralise au cas n pair
- ▶ Si h n'a pas que des racines réelles ?
 - ⇒ Si on décale suffisamment le polynôme a , h n'aura que des racines réelles (Théorème de Eve).
 - 1. détermination d'un décalage c et décalage du polynôme : $f(x) = a(x - c)$
 - 2. appliquer l'algorithme de préconditionnement au polynôme décalé $f(x)$
 - 3. évaluation en $x + c$
 - ⇒ deux types de décalages (Eve / Knuth)

Cas général (avec $n \geq 3$)

- ▶ Cette méthode se généralise au cas n pair
- ▶ Si h n'a pas que des racines réelles ?
 - ⇒ Si on décale suffisamment le polynôme a , h n'aura que des racines réelles (Théorème de Eve).
 1. détermination d'un décalage c et décalage du polynôme : $f(x) = a(x - c)$
 2. appliquer l'algorithme de préconditionnement au polynôme décalé $f(x)$
 3. évaluation en $x + c$
 - ⇒ deux types de décalages (Eve / Knuth)

Schéma d'évaluation

- ▶ Avec $y = (x + c)^2$, le schéma d'évaluation est :

$$a(x) = \left(\dots ((\beta_m(y - \alpha_m) + \beta_{m-1})(y - \alpha_{m-1}) + \beta_{m-2}) \dots \right) (y - \alpha_1) + \beta_0$$

→ schéma semblable à Horner

- ▶ Complexité de l'évaluation : $\lfloor \frac{n}{2} \rfloor + 2$ multiplications et n additions

→ toujours plus rapide que Horner

- ▶ Dans le *modèle flottant standard* : algorithme stable

→ *backward stable* : résultat obtenu = résultat exact d'un problème proche

→ erreur majorée par $\gamma_{3n} \approx 3n \times 2^{-p}$ (p : précision)

→ si $c = 0$: erreur majorée par γ_{2n} (comme Horner)

Résultats numériques et observations

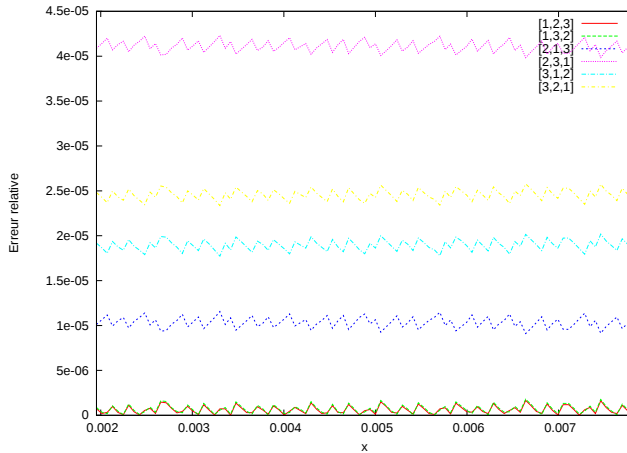
Observation générale sur des polynômes aléatoires

- ▶ Malgré un décalage nul : erreur relative par Knuth & Eve plus élevée que pour Horner (≈ 100 fois).
 - valeur approchée des racines de h

Effet de la permutation

- ▶ Importance de l'ordre de traitement des racines
 - effet sur la valeur des β_i
 - variation de l'erreur relative en fonction de la permutation

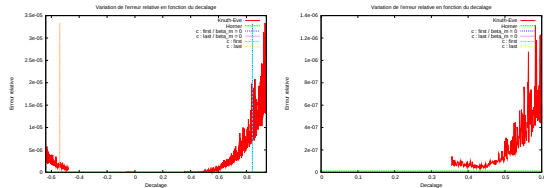
Résultats numériques et observations



Résultats numériques et observations

Effet du décalage

- De manière générale, plus le décalage est élevé, plus l'erreur relative est importante

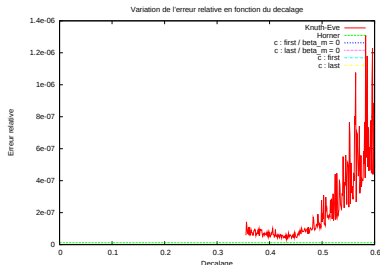


- Décalages de Eve et Knuth : non optimaux / pessimistes

Résultats numériques et observations

Effet du décalage

- Dans certains cas : effet positif du décalage (degré 7 / $[-2^{-7}, 2^{-7}]$)

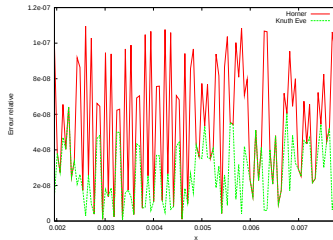


- $c = 0 \rightarrow e \approx 4.95 \times 10^{-8}$
- c de Knuth/Eve $\approx 8.43 \times 10^{-1} \rightarrow e \approx 9.86 \times 10^{-6}$
- meilleur $c \approx 4.33 \times 10^{-1} \rightarrow e \approx 3.23 \times 10^{-8}$ (Horner : $e \approx 3.1 \times 10^{-8}$)

Résultats numériques et observations

Effet du décalage nul

- ▶ analyse d'erreur : $c = 0 \rightarrow$ même borne d'erreur théorique que Horner
 - $\rightarrow$ dans certains cas, comparable à Horner
 - $\rightarrow$ dans certains cas, meilleur que Horner
- ▶ évaluation d'un polynôme de degré 12
 - $\rightarrow [2^{-9}, 2^{-7}]$
 - $\rightarrow c = 0$

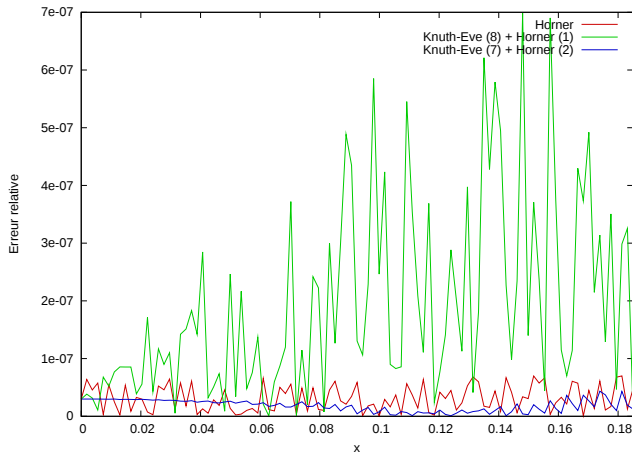


- ▶ évaluation par Knuth & Eve :
 $e \approx 6.4 \times 10^{-8}$
- ▶ évaluation par Horner :
 $e \approx 1.1 \times 10^{-7}$

Conclusions l'algorithme de Knuth & Eve

- ▶ comparaison par rapport à Horner
 - toujours plus rapide que Horner (moins d'opérations)
 - dans certains cas : aussi précis que Horner
- ▶ plusieurs préconditionnements possibles
 - difficulté de déterminer un meilleur préconditionnement
- ▶ efficacité des schémas hybrides
 - combinaison Knuth & Eve / Horner
 - compenser avec Horner les pertes de précision de Knuth & Eve

Conclusions l'algorithme de Knuth & Eve



Plan de la soutenance

Étude de l'algorithme de Knuth & Eve

Description de l'algorithme de Paterson & Stockmeyer

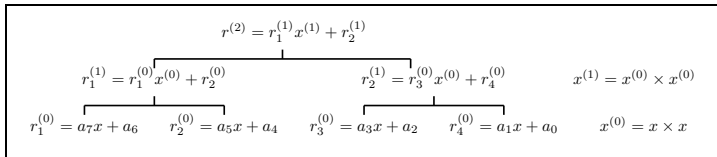
Application à CR-Libm : utilisation de schémas hybrides

Conclusion & Perspectives

Rappel sur la méthode de Estrin

- Le schéma de Estrin pour un polynôme de degré 7 est :

$$a(x) = ((a_7 \cdot x + a_6)x^2 + (a_5 \cdot x + a_4))x^4 + (a_3 \cdot x + a_2)x^2 + (a_1 \cdot x + a_0)$$



→ stratégie de « diviser pour régner »

⇒ Paterson & Stockmeyer : schéma similaire à celui de Estrin

Idée générale de l'algorithme de Paterson & Stockmeyer

► Idées :

- supprimer les multiplications au niveau des feuilles de l'arbre : diminuer le nombre total de multiplications
- travailler uniquement sur des polynômes **unitaires** (coefficient de plus haut degré = 1)

⇒ décalage des polynômes au moment de la division

► Complexité de l'évaluation :

- $\frac{n+1}{2} + p - 1$ multiplications / $\frac{3n-1}{2}$ additions ($p = \log(n+1)$)
- plus d'additions que par le schéma de Horner mais évaluation parallèle (comme Estrin)

► Observations pratiques :

- l'évaluation par Paterson & Stockmeyer peut se comporter comme Horner ou Estrin
- préconditionnement unique

Idée générale de l'algorithme de Paterson & Stockmeyer

► Idées :

- supprimer les multiplications au niveau des feuilles de l'arbre : diminuer le nombre total de multiplications
- travailler uniquement sur des polynômes **unitaires** (coefficient de plus haut degré = 1)

⇒ décalage des polynômes au moment de la division

► Complexité de l'évaluation :

- $\frac{n+1}{2} + p - 1$ multiplications / $\frac{3n-1}{2}$ additions ($p = \log(n+1)$)
- plus d'additions que par le schéma de Horner mais évaluation parallèle (comme Estrin)

► Observations pratiques :

- l'évaluation par Paterson & Stockmeyer peut se comporter comme Horner ou Estrin
- préconditionnement unique

Idée générale de l'algorithme de Paterson & Stockmeyer

- ▶ Idées :
 - ▶ supprimer les multiplications au niveau des feuilles de l'arbre : diminuer le nombre total de multiplications
 - ▶ travailler uniquement sur des polynômes **unitaires** (coefficient de plus haut degré = 1)

⇒ décalage des polynômes au moment de la division
- ▶ Complexité de l'évaluation :
 - ▶ $\frac{n+1}{2} + p - 1$ multiplications / $\frac{3n-1}{2}$ additions ($p = \log(n+1)$)
 - ▶ plus d'additions que par le schéma de Horner mais évaluation parallèle (comme Estrin)
- ▶ Observations pratiques :
 - ▶ l'évaluation par Paterson & Stockmeyer peut se comporter comme Horner ou Estrin
 - ▶ préconditionnement unique

Plan de la soutenance

Étude de l'algorithme de Knuth & Eve

Description de l'algorithme de Paterson & Stockmeyer

Application à CR-Libm : utilisation de schémas hybrides

Conclusion & Perspectives

Accélération de la fonction $\arcsin(x)$

Fonction $\arcsin(x)$ existante

- ▶ fonction approchée sur $[0, 1]$ par 10 polynômes de degré 9, 14 et 17
- ▶ tous évalués par un schéma de Horner

Solutions proposées

- ▶ schéma hybride Knuth & Eve / Estrin pour le premier polynôme
- ▶ schéma de Estrin pour les autres polynômes

⇒ gain : **73 à 83 cycles (soit plus de 20 %)** selon le compilateur

⇒ borne d'erreur d'évaluation du polynôme 1 prouvée avec Gappa

→ borne d'erreur : 2^{-60}

	CR-Libm	
	Horner	KE/Estrin
gcc - 3.4	342 cycles	259 cycles
gcc - 4.0	342 cycles	269 cycles

Accélération du logarithme népérien

Logarithme existant

- ▶ évaluation d'un polynôme de degré 7 sur $[-2^{-7}, 2^{-7}]$
 - évalué par un schéma de Estrin

Solution proposée

- ▶ schéma hybride Paterson & Stockmeyer / Estrin
- ⇒ gain : **3 à 5 cycles** selon le compilateur
- ⇒ borne d'erreur d'évaluation prouvée avec Gappa
- même borne d'erreur qu'avec Estrin : 2^{-62}

	CR-Libm	
	Estrin	PS/Estrin
gcc - 3.4	143 cycles	138 cycles
gcc - 4.0	151 cycles	148 cycles

Plan de la soutenance

Étude de l'algorithme de Knuth & Eve

Description de l'algorithme de Paterson & Stockmeyer

Application à CR-Libm : utilisation de schémas hybrides

Conclusion & Perspectives

Conclusion & Perspectives

Conclusion

- ▶ mise en évidence de l'efficacité des schémas hybrides
- ▶ accélération de plus de 20 % de la fonction $\arcsin(x)$ de CR-Libm
- ▶ preuve des résultats avec Gappa

Perspectives

- ▶ terminer la fonction $\arcsin(x)$
- ▶ détermination d'une combinaison décalage/permutation optimale pour Knuth & Eve
- ▶ efficacité des schémas hybrides : évaluation plus rapide et aussi précise qu'avec un schéma de Horner
 - automatisation la conception de ces schémas d'évaluation
- ▶ amélioration des étapes de préconditionnement
- ▶ extension à Knuth & Eve de la *méthode de Horner compensée*

Temps moyen de préconditionnement pour Knuth & Eve

Phase	Temps d'exécution
Détermination des racines de a	0.227173 s.
Détermination du décalage	0.245369 s.
Découpage paire / impaire	0.358000×10^{-3} s.
Détermination des racines de h	0.574640×10^{-1} s.
Détermination des β_i	0.257940×10^{-1} s.
Temps total	0.816209 s.

Détails sur l'algorithme de Paterson & Stockmeyer