

---

# Intégration numérique avec erreur bornée en précision arbitraire

Laurent Fousse

laurent@komite.net

LORIA – Université Henri Poincaré Nancy I



# Plan de l'exposé

---

1. Motivations et enjeux
2. État de l'art et contributions
3. Conclusion et perspectives



# Plan de l'exposé

---

1. Motivations et enjeux
  - (a) Fiabilité
  - (b) Exemples d'intégrales
  - (c) Exemples logiciels
  - (d) Modèle de calcul flottant
  - (e) Enjeux
2. État de l'art et contributions
3. Conclusion et perspectives



# Fiabilité

Nous avons besoin d'arithmétique **fiable**.

Exemple : échec d'un missile Patriot (25 février 1991), ou « petite erreur deviendra grosse ».

- horloge interne en  $1/10$  de seconde,
- erreur de troncation sur la constante  $1/10$  en binaire,
- erreur totale de  $0,34s$ , pour une vitesse de  $1600m/s$ ,
- 28 morts.



# Fiabilité

## Explosion d'Ariane 5 (4 juin 1996)

- Conversion d'un nombre flottant 64 bits vers un entier 16 bits,
- Échec de conversion (constante trop grande),
- Panne du Système de Référence Inertielle,
- Perte totale d'information de guidage,
- Explosion (500 millions de dollars).



---

## 1. Motivations et enjeux

(a) Fiabilité

(b) Exemples d'intégrales

(c) Exemples logiciels

(d) Modèle de calcul flottant

(e) Enjeux

## 2. État de l'art et contributions

## 3. Conclusion et perspectives



# Calcul d'intégrales

---

Le calcul d'intégrales joue un rôle important dans les calculs scientifiques en général, avec des applications dans divers domaines :

- physique (cristallographie [Jézéquel Chesneaux 2004])

$$g(a) = \int_0^{+\infty} [(\exp(x) + \exp(-x))^a - \exp(ax) - \exp(-ax)] dx$$

où  $0 < a < 2$  (énergie électronique totale de cristaux).

- probabilités [Trefethen, Kern 2002]

$$2 = \frac{1}{\pi} \int_{-\pi}^{\pi} \frac{d\phi}{\sqrt{3 - 4 \cos \phi + \cos^2 \phi + 16\epsilon^2}}$$

(marche aléatoire paramétrée par  $\epsilon$  dans  $\mathbb{Z}^2$ ).



# Calcul d'intégrales

Interpolation asymptotique [Pauls Frisch 2006]

$$\partial_t u(t, x) + u(t, x) \partial_x u(t, x) = 0$$

$$u(t, x) = \sum_{k=-\infty}^{\infty} e^{ikx} \hat{u}_k(t)$$

$$\hat{u}_k(t) = -\frac{1}{2i\pi kt} \int_0^{2\pi} e^{-ik(a - \frac{1}{2}t \sin(a))} da.$$

→ besoin de calculer des milliers d'intégrales avec une centaine de bits de précision.



---

## 1. Motivations et enjeux

(a) Fiabilité

(b) Exemples d'intégrales

(c) Exemples logiciels

(d) Modèle de calcul flottant

(e) Enjeux

## 2. État de l'art et contributions

## 3. Conclusion et perspectives



# Intégration fiable

---

Les erreurs d'arrondi entachent le résultat et s'accroissent.

Exemple avec Pari/GP 2.3.0 :  $\int_0^{10} x^2 \sin(x^3) dx$

```
? default(realprecision, 115); intnum(x=0, 10, x^2 * sin(x^3))  
%2 = 16.28927300548418203118991830[...]
```

```
? default(realprecision, 116); intnum(x=0, 10, x^2 * sin(x^3))  
%2 = 0.145873643233630725025779820[...]
```

Quel résultat inspire le plus confiance ?



# Intégration fiable

Les erreurs d'arrondi entachent le résultat et s'accroissent.

Exemple avec Pari/GP 2.3.0 :  $\int_0^{10} x^2 \sin(x^3) dx$

```
? default(realprecision, 115); intnum(x=0, 10, x^2 * sin(x^3))
%2 = 16.28927300548418203118991830[...]
```

```
? default(realprecision, 116); intnum(x=0, 10, x^2 * sin(x^3))
%2 = 0.145873643233630725025779820[...]
```

Quel résultat inspire le plus confiance ?

$$\int_0^{10} x^2 \sin(x^3) dx = \frac{1}{3} (1 - \cos(1000))$$
$$\approx 0,1458736432336$$



# Intégration (non ?) fiable

---

Un système célèbre de calcul formel :

```
|\~/|      Maple 10 (IBM INTEL LINUX)
._|\|     |/|_. Copyright (c) Maplesoft, a division of Waterloo Maple Inc. 2005
 \  MAPLE / All rights reserved. Maple is a trademark of
 <-----> Waterloo Maple Inc.
      |      Type ? for help.
> evalf(Int(exp(-x^2)*ln(x), x=17..42));
                                     -126
                                0.2604007480 10
```



# Intégration (non ?) fiable

Un système célèbre de calcul formel :

```
|\~/|      Maple 10 (IBM INTEL LINUX)
._|\|     |/|_. Copyright (c) Maplesoft, a division of Waterloo Maple Inc. 2005
 \ MAPLE / All rights reserved. Maple is a trademark of
 <-----> Waterloo Maple Inc.
      |      Type ? for help.
> evalf(Int(exp(-x^2)*ln(x), x=17..42));
                                     -126
                                0.2604007480 10
> Digits:=20: evalf(Int(exp(-x^2)*ln(x), x=17..42));
                                     -126
                                0.34288028340847034512 10
```



# Intégration (non ?) fiable

Un système célèbre de calcul formel :

```
|\~/|      Maple 10 (IBM INTEL LINUX)
._|\|    |/|_. Copyright (c) Maplesoft, a division of Waterloo Maple Inc. 2005
 \  MAPLE / All rights reserved. Maple is a trademark of
 <-----> Waterloo Maple Inc.
      |      Type ? for help.
> evalf(Int(exp(-x^2)*ln(x), x=17..42));
                                     -126
                                0.2604007480 10
> Digits:=20: evalf(Int(exp(-x^2)*ln(x), x=17..42));
                                     -126
                                0.34288028340847034512 10
> Digits:=50: evalf(Int(exp(-x^2)*ln(x), x=17..42));
                                                                 -128
                                0.49076783443012876473973482836733778547443399549250 10
```



# Intégration (non ?) fiable

Un système célèbre de calcul formel :

```
|\~/|      Maple 10 (IBM INTEL LINUX)
._|\\|    |/|_ . Copyright (c) Maplesoft, a division of Waterloo Maple Inc. 2005
 \  MAPLE / All rights reserved. Maple is a trademark of
 <_____> Waterloo Maple Inc.
   |      Type ? for help.
> evalf(Int(exp(-x^2)*ln(x), x=17..42));
                                     -126
                                0.2604007480 10
> Digits:=20: evalf(Int(exp(-x^2)*ln(x), x=17..42));
                                     -126
                                0.34288028340847034512 10
> Digits:=50: evalf(Int(exp(-x^2)*ln(x), x=17..42));
                                     -128
                                0.49076783443012876473973482836733778547443399549250 10
> Digits:=100: evalf(Int(exp(-x^2)*ln(x), x=17..42));
                                     -128
                                0.49076783443012876473973482836733778547443399549250[...] 10
```



# Intégration (non ?) fiable

$$\int_{17}^{42} \exp(-x^2) \log x dx \approx 0,2565728500 \times 10^{-126}$$

Précision demandée	Réponse Maple	Chiffres corrects
10	$0,2604007480 \times 10^{-126}$	1,8
20	$0,3428802834 \times 10^{-126}$	0,6
50	$0,4907678344 \times 10^{-128}$	-1,7
100	$0,4907678344 \times 10^{-128}$	-1,7



# Intégration (non ?) fiable

$$\int_{17}^{42} \exp(-x^2) \log x dx \approx 0,2565728500 \times 10^{-126}$$

Précision demandée	Réponse Maple	Chiffres corrects
10	$0,2604007480 \times 10^{-126}$	1,8
20	$0,3428802834 \times 10^{-126}$	0,6
50	$0,4907678344 \times 10^{-128}$	-1,7
100	$0,4907678344 \times 10^{-128}$	-1,7

→ augmenter la précision nous a conforté dans notre erreur !



---

## 1. Motivations et enjeux

(a) Fiabilité

(b) Exemples d'intégrales

(c) Exemples logiciels

(d) Modèle de calcul flottant

(e) Enjeux

## 2. État de l'art et contributions

## 3. Conclusion et perspectives



# Arithmétique flottante – Arrondi

---

## Précision finie

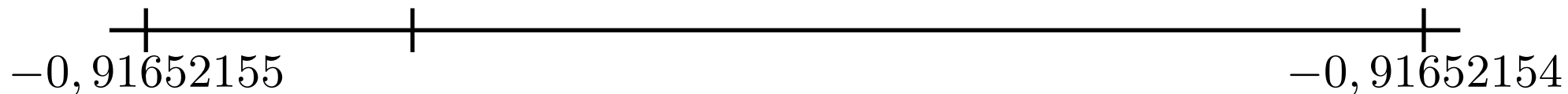
$$\sin(42) = -0,916521547915633785 \dots$$

$$\approx -0,91652154 ?$$

$$\approx -0,91652155 ?$$

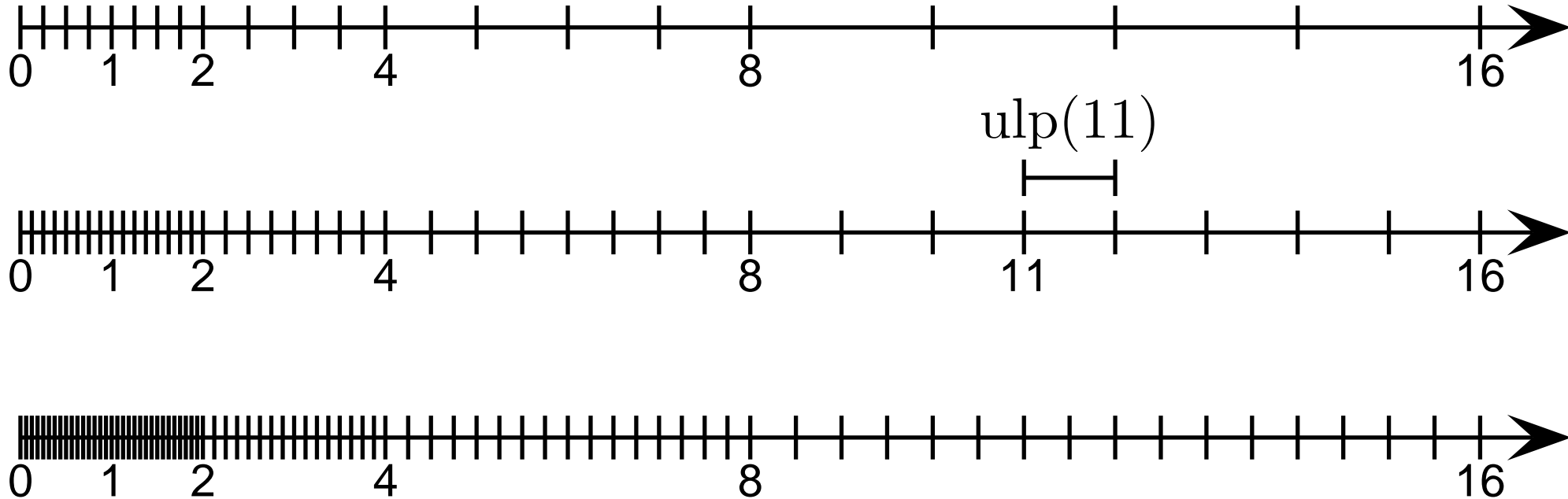
$$\circ(\sin(42)) = -0,91652155 \quad [\text{arrondi au plus proche}]$$

$\sin(42)$



# Arithmétique flottante – Précision arbitraire

---



$$x = 0,91652155$$

$$\text{ulp}(x) = 0,00000001$$



# Enjeux

---

- Contrôler **toutes** les sources d'**erreurs** :
  - borne explicite ;
  - borne fine.



# Enjeux

---

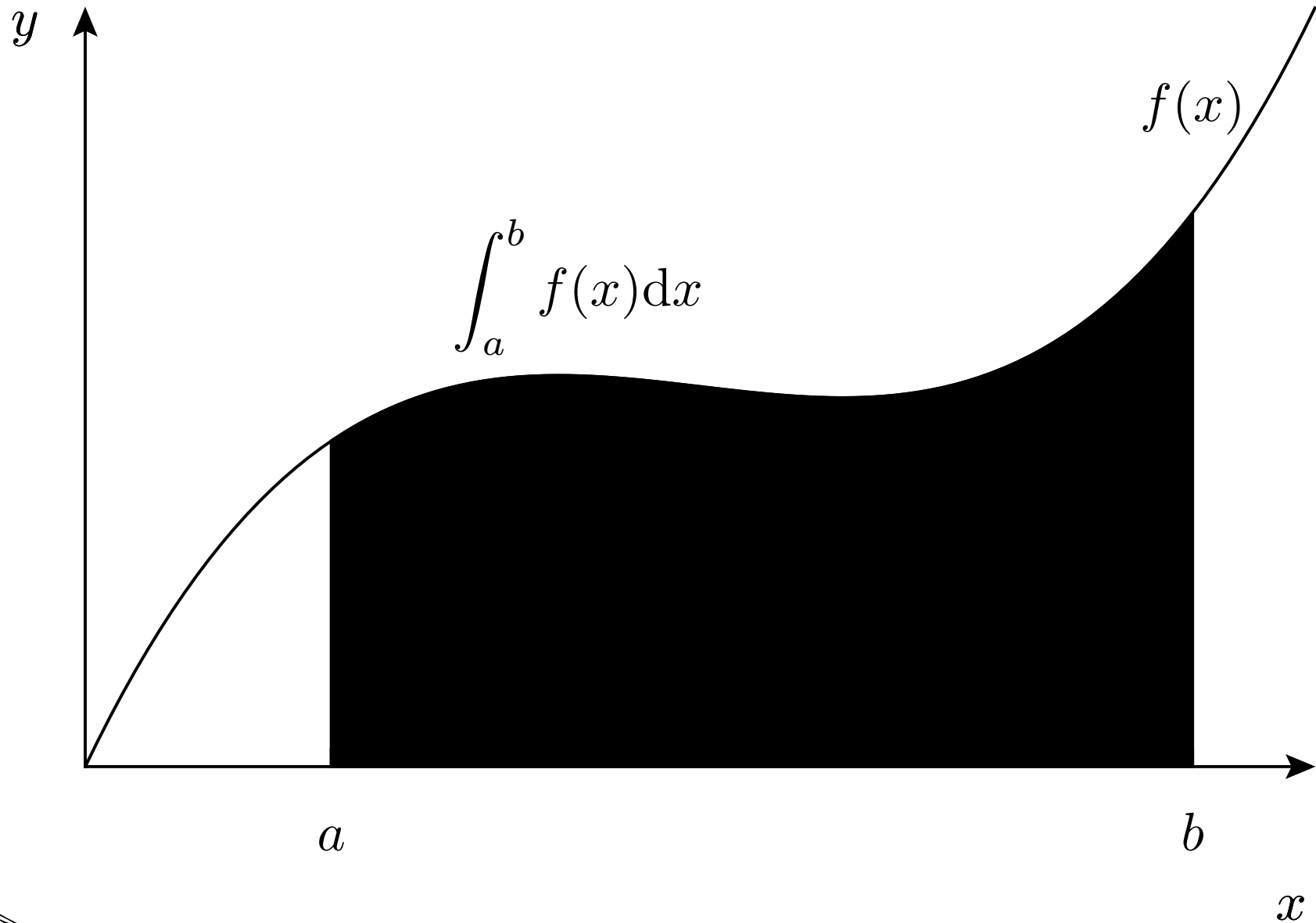
- Contrôler **toutes** les sources d'**erreurs** :
  - borne explicite ;
  - borne fine.
- Être **rapide** en grande précision.



- 
1. Motivations et enjeux
  2. État de l'art et contributions
    - (a) Problématique
    - (b) État de l'art
    - (c) Méthodes par interpolation
    - (d) Méthode de Newton-Cotes
    - (e) Méthode de Gauss-Legendre
    - (f) Résultats expérimentaux
  3. Conclusion et perspectives



# Problématique



# Problématique

---

$$f : [a, b] \rightarrow \mathbb{R} \text{ de classe } C^\infty$$

Calculer

$$I = \int_a^b f(x) dx$$

en **précision arbitraire** et avec **erreur bornée**.

Hypothèses : ●  $f$  connue en précision arbitraire,  
● on sait borner  $|f^{(k)}(x)|$  sur  $[a, b]$  pour  $k \geq 1$ .

Par exemple :  $f$  fonction entière.



# État de l'art

---

- Méthodes par interpolation :
  - Méthode de Newton-Cotes,
  - Méthodes de Gauss (Gauss-Legendre, Gauss-Laguerre, Gauss-Tchebychev, ...) ou assimilées (Gauss-Kronrod),
  - Méthode de Clenshaw-Curtis ;
- Méthode de Romberg ;
- Méthode de Monte-Carlo ;
- Méthode doublement exponentielle.

Gestion d'erreur : ● par analyse directe ;

- par arithmétique d'intervalles (pour l'erreur d'arrondi) ;
- par estimation dynamique ;
- par arithmétique stochastique.



- 
1. Motivations et enjeux
  2. État de l'art et contributions
    - (a) Problématique
    - (b) État de l'art
    - (c) Méthodes par interpolation
    - (d) Méthode de Newton-Cotes
    - (e) Méthode de Gauss-Legendre
    - (f) Résultats expérimentaux
  3. Conclusion et perspectives



# Algorithme d'intégration par interpolation

---

$$\int_a^b f(x)dx \approx \int_a^b P(x)dx$$
$$P(x) = \sum_{i=0}^{n-1} f(x_i) \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$$



# Algorithme d'intégration par interpolation

---

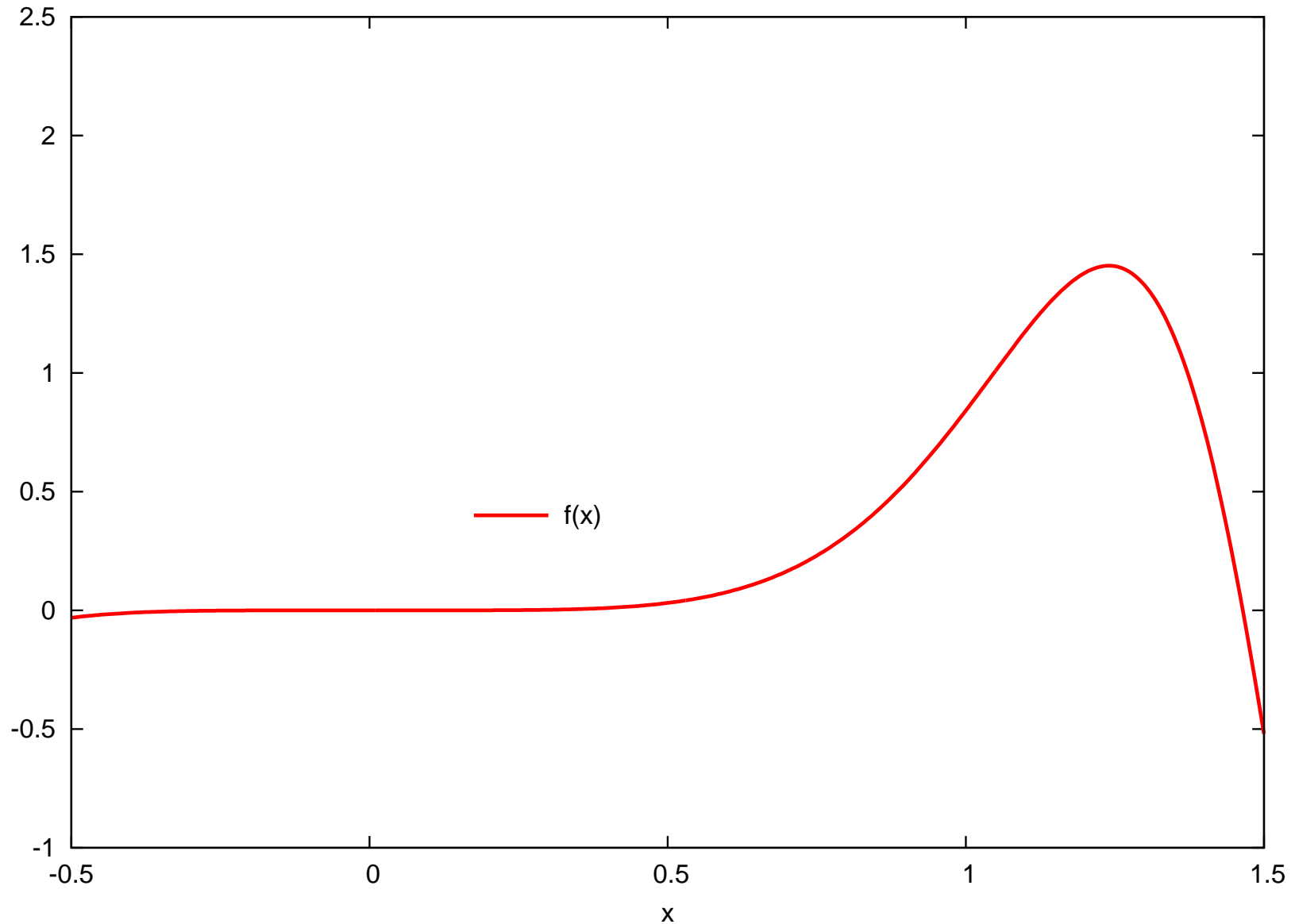
$$\int_a^b f(x)dx \approx \int_a^b P(x)dx$$
$$P(x) = \sum_{i=0}^{n-1} f(x_i) \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$$

$$\int_a^b P(x)dx = \sum_{i=0}^{n-1} w_i f(x_i)$$
$$w_i = \int_a^b \prod_{j \neq i} \frac{x - x_j}{x_i - x_j} dx$$

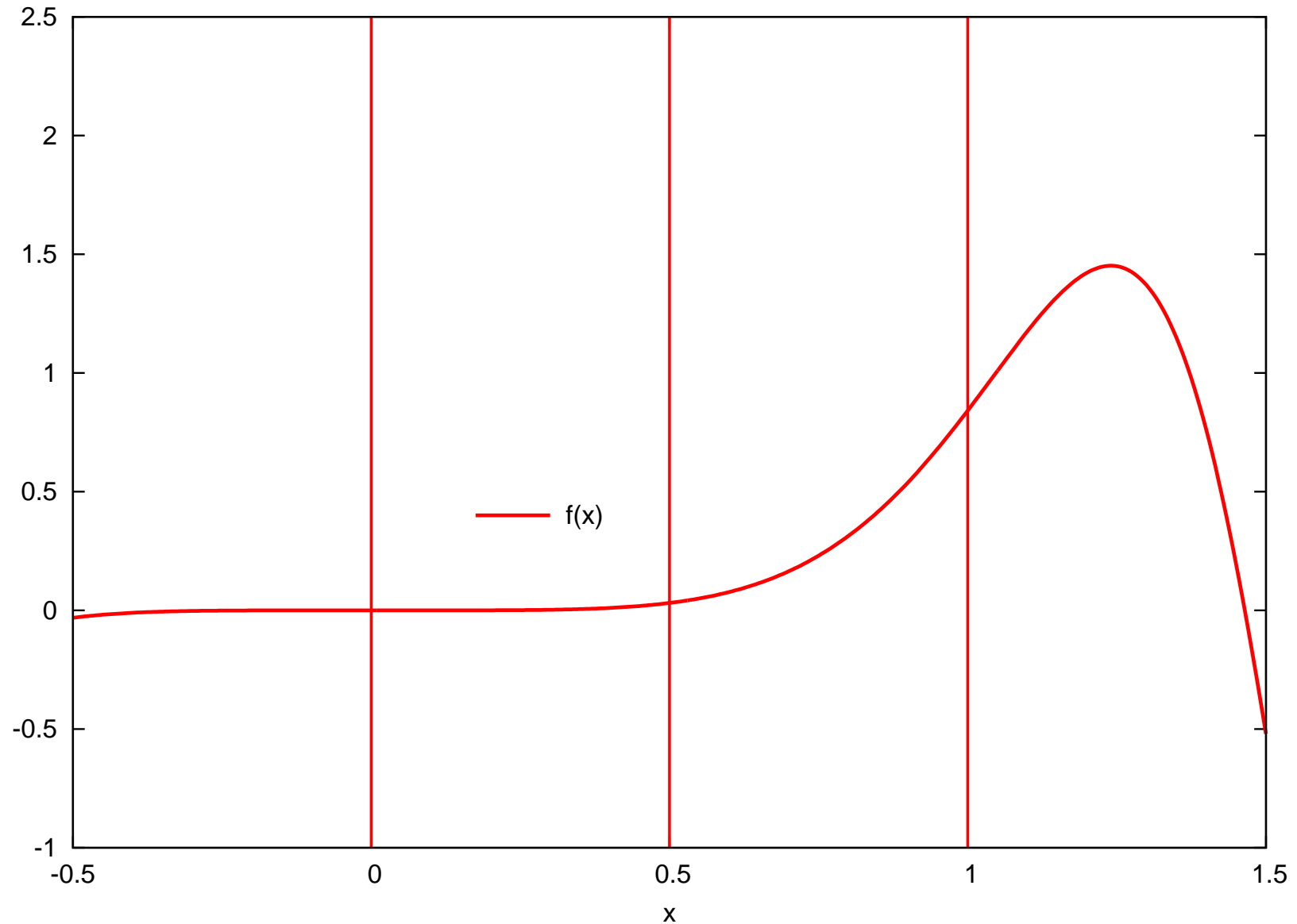


# Algorithme d'intégration par interpolation

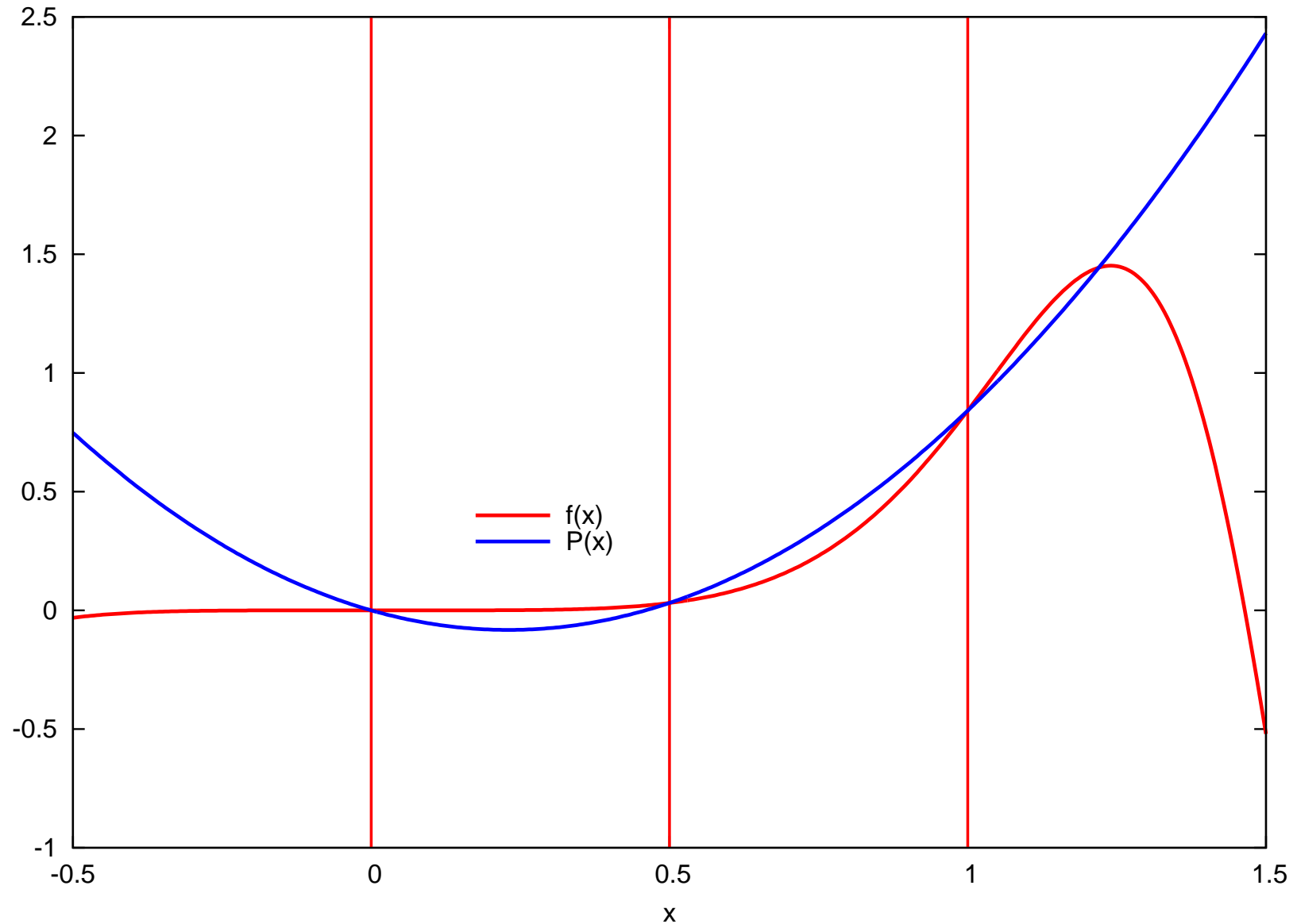
---



# Algorithme d'intégration par interpolation



# Algorithme d'intégration par interpolation



# Enjeux

---

- Contrôler **toutes** les sources d'**erreur** :
  - la formule d'intégration n'est qu'une approximation (**erreur mathématique**),

$$\int_a^b f(x)dx \approx \int_a^b P(x)dx$$

# Enjeux

---

- Contrôler **toutes** les sources d'**erreur** :
  - la formule d'intégration n'est qu'une approximation (**erreur mathématique**),
  - le calcul des  $(x_i)$  et des poids  $(w_i)$  est inexact pour la plupart des méthodes d'intégration,

$$\int_a^b P(x)dx = \sum_{i=0}^{n-1} w_i f(x_i)$$

# Enjeux

---

- Contrôler **toutes** les sources d'**erreur** :
  - la formule d'intégration n'est qu'une approximation (**erreur mathématique**),
  - le calcul des  $(x_i)$  et des poids  $(w_i)$  est inexact pour la plupart des méthodes d'intégration,
  - le calcul de  $f(x_i)$  sera probablement inexact, même pour des fonctions simples,

$$\int_a^b P(x)dx = \sum_{i=0}^{n-1} w_i f(x_i)$$

# Enjeux

---

- Contrôler **toutes** les sources d'**erreur** :
  - la formule d'intégration n'est qu'une approximation (**erreur mathématique**),
  - le calcul des  $(x_i)$  et des poids  $(w_i)$  est inexact pour la plupart des méthodes d'intégration,
  - le calcul de  $f(x_i)$  sera probablement inexact, même pour des fonctions simples,
  - la formule d'intégration en elle-même demande beaucoup de calculs, les **erreurs d'arrondi** s'accumulent.

$$\int_a^b P(x)dx = \sum_{i=0}^{n-1} w_i \times f(x_i)$$



# Enjeux

---

- Contrôler **toutes** les sources d'**erreur** :
  - la formule d'intégration n'est qu'une approximation (**erreur mathématique**),
  - le calcul des  $(x_i)$  et des poids  $(w_i)$  est inexact pour la plupart des méthodes d'intégration,
  - le calcul de  $f(x_i)$  sera probablement inexact, même pour des fonctions simples,
  - la formule d'intégration en elle-même demande beaucoup de calculs, les **erreurs d'arrondi** s'accumulent.
- Être **rapide** en grande précision.



- 
1. Motivations et enjeux
  2. État de l'art et contributions
    - (a) Problématique
    - (b) État de l'art
    - (c) Méthodes par interpolation
    - (d) Méthode de Newton-Cotes
    - (e) Méthode de Gauss-Legendre
    - (f) Résultats expérimentaux
  3. Conclusion et perspectives



# Méthode de Newton-Cotes

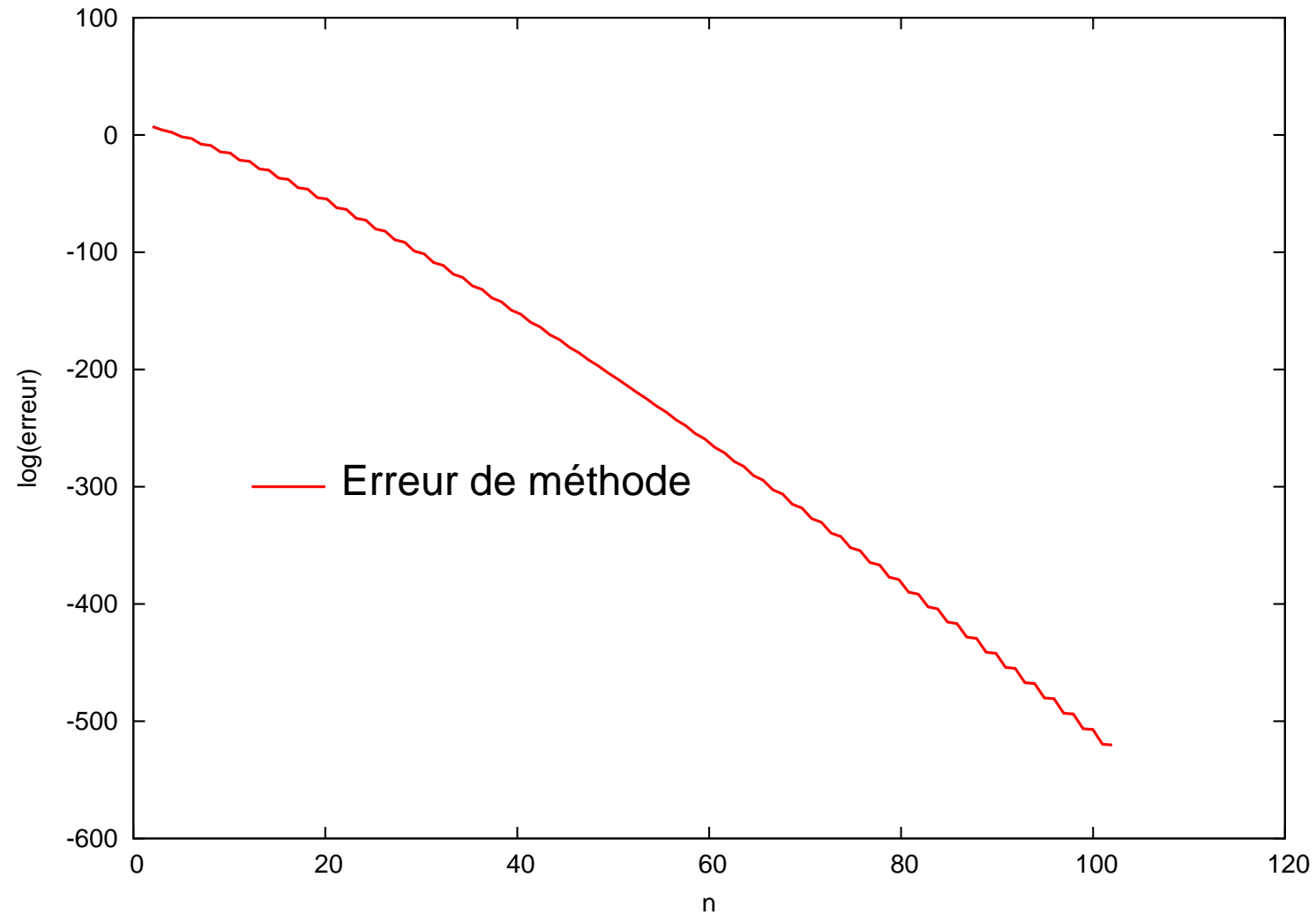
---

- $x_i = a + \frac{i(b-a)}{n-1}$  (points équi-répartis),
- Poids  $w_i = \int_a^b \prod_{j \neq i} \frac{x-x_j}{x_i-x_j} dx$  rationnels, ne dépendant pas de l'intervalle  $[a, b]$ ,
- l'erreur mathématique décroît lorsque le nombre de points croît, on la borne par

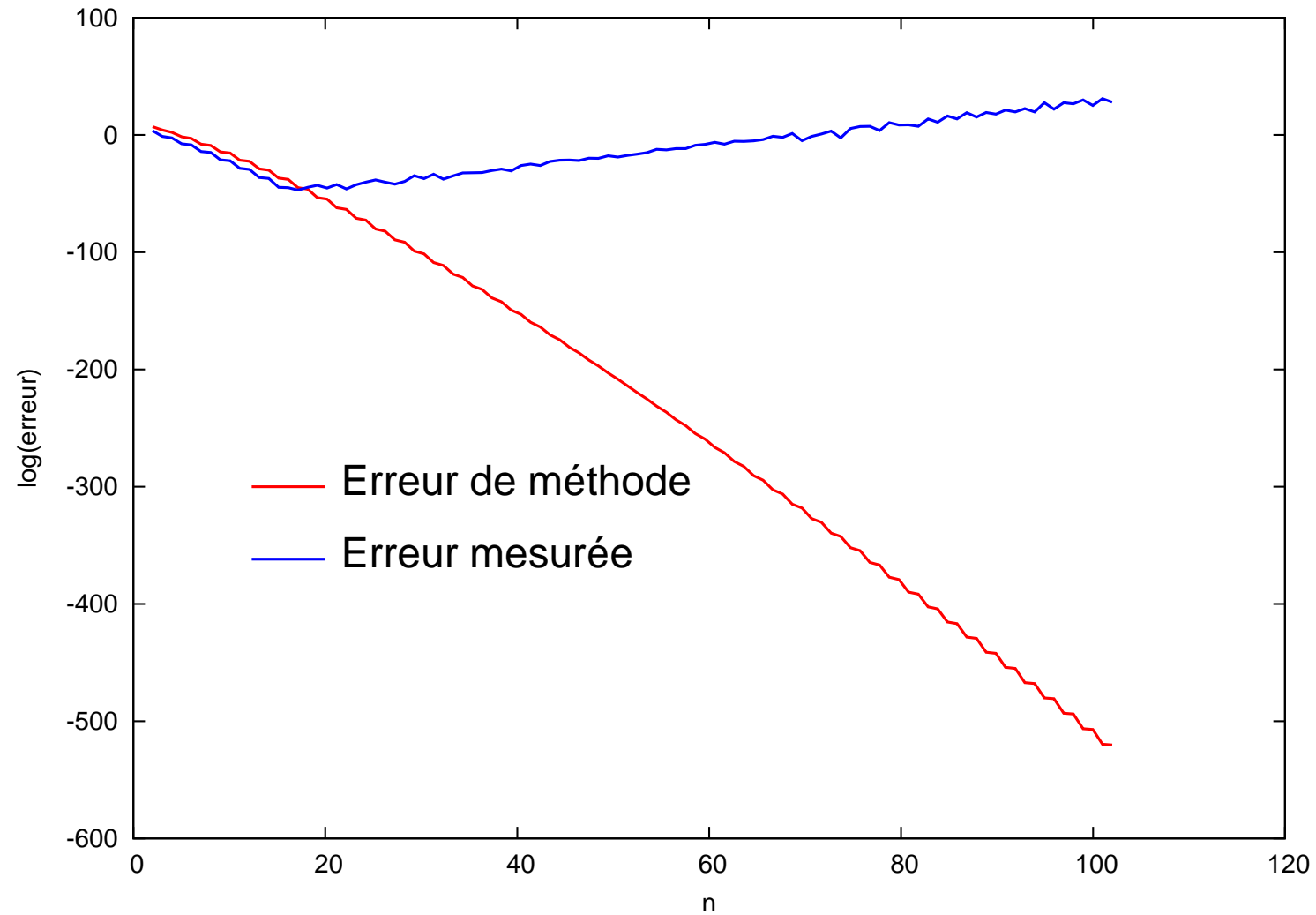
$$E_{\text{math}} \leq \left( \frac{b-a}{n-1} \right)^{n+1} \max_{[a,b]} \left| f^{(n)} \right|$$



# Méthode de Newton-Cotes



# Méthode de Newton-Cotes



# Calcul des poids de Newton-Cotes

ENTRÉE :  $n$  nombre de points de la méthode.

SORTIE :  $(v_0, v_1, \dots, v_{\lfloor \frac{n-1}{2} \rfloor}, d)$  tels que  $w_i = \frac{v_i}{d}$ .

- 1:  $\delta \leftarrow \text{ppcm}(2, 3, \dots, n)$   $\mathcal{O}(n^2 \log n)$
- 2:  $l_0^* \leftarrow (x-1)(x-2)\dots(x-(n-1))\delta$   $\mathcal{O}(n^3 \log^2 n)$
- 3: **for**  $i \leftarrow 0$  to  $n-1$  **do**
- 4:      $L_i(n-1) \leftarrow \int_0^{n-1} l_i^*(x) dx$   $\mathcal{O}(n^3 \log^2 n)$
- 5:      $l_{i+1}^* \leftarrow \frac{x-i}{x-(i+1)} l_i^*$   $\mathcal{O}(n^3 \log^2 n)$
- 6:      $v_i \leftarrow (-1)^{n-1-i} \binom{n-1}{i} L_i(n-1)$   $\mathcal{O}(n^2 \log^2 n)$
- 7: **end for**
- 8: **return**  $(v_0, v_1, \dots, v_{\lfloor \frac{n-1}{2} \rfloor}, d = \delta \cdot (n-1)!)$   $\mathcal{O}(n^3 \log^2 n)$



# Newton-Cotes rapide

Problème : calculer

$$\begin{aligned}w_i &= \int_a^b \prod_{j \neq i} \frac{x - x_j}{x_i - x_j} dx \\ &= \frac{(-1)^{n-1-i}}{i!(n-1-i)!} \int_0^{n-1} \prod_{j \neq i} (x - j) dx.\end{aligned}$$

Idée : les calculer simultanément.

$$\begin{aligned}p(x) &= x(x-1) \dots (x-(n-1)) \\ q_i(x) &= \frac{p(x)}{x-i} \\ \hat{l}_i &= \int_0^{n-1} q_i(x) dx\end{aligned}$$



# Newton-Cotes rapide

---

On note  $p(x) = x^n + a_{n-1}x^{n-1} + \dots + a_1x$  :

$$\begin{aligned}q_i(x) &= x^{n-1} + (i + a_{n-1})x^{n-2} + (i^2 + a_{n-1}i + a_{n-2})x^{n-3} \\ &+ \dots \\ &+ (i^{n-1} + a_{n-1}i^{n-2} + \dots + a_2i + a_1).\end{aligned}$$

Par intégration :

$$\begin{aligned}\widehat{l}_i &= \frac{(n-1)^n}{n} + (i + a_{n-1})\frac{(n-1)^{n-1}}{n-1} \\ &+ (i^2 + a_{n-1}i + a_{n-2})\frac{(n-1)^{n-2}}{n-2} + \dots \\ &+ (i^{n-1} + a_{n-1}i^{n-2} + \dots + a_2i + a_1)(n-1).\end{aligned}$$



# Newton-Cotes rapide

---

On a réussi à exprimer  $\widehat{l}_i$  comme un polynôme en  $i$  :  $\widehat{l}_i = r(i)$  avec

$$\begin{aligned} r(y) &= (n-1)y^{n-1} + [(n-1)a_{n-1} + \frac{(n-1)^2}{2}]y^{n-2} \\ &+ [(n-1)a_{n-2} + \frac{(n-1)^2}{2}a_{n-1} + \frac{(n-1)^3}{6}]y^{n-3} + \dots \\ &+ [(n-1)a_1 + \frac{(n-1)^2}{2}a_2 + \dots + \frac{(n-1)^{n-1}}{n-1}a_{n-1} + \frac{(n-1)^n}{n}]. \end{aligned}$$

De plus  $r(x)$  s'écrit  $r(x) = p(x)s(x) \text{ div } x^n$

$$s(x) = (n-1)x^{n-1} + \frac{(n-1)^2}{2}x^{n-2} + \dots + \frac{(n-1)^n}{n}.$$



# Newton-Cotes rapide

---

ENTRÉE :  $n$  nombre de points de la méthode.

SORTIE :  $(w_0, w_1, \dots, w_{n-1})$  les poids de la méthode.

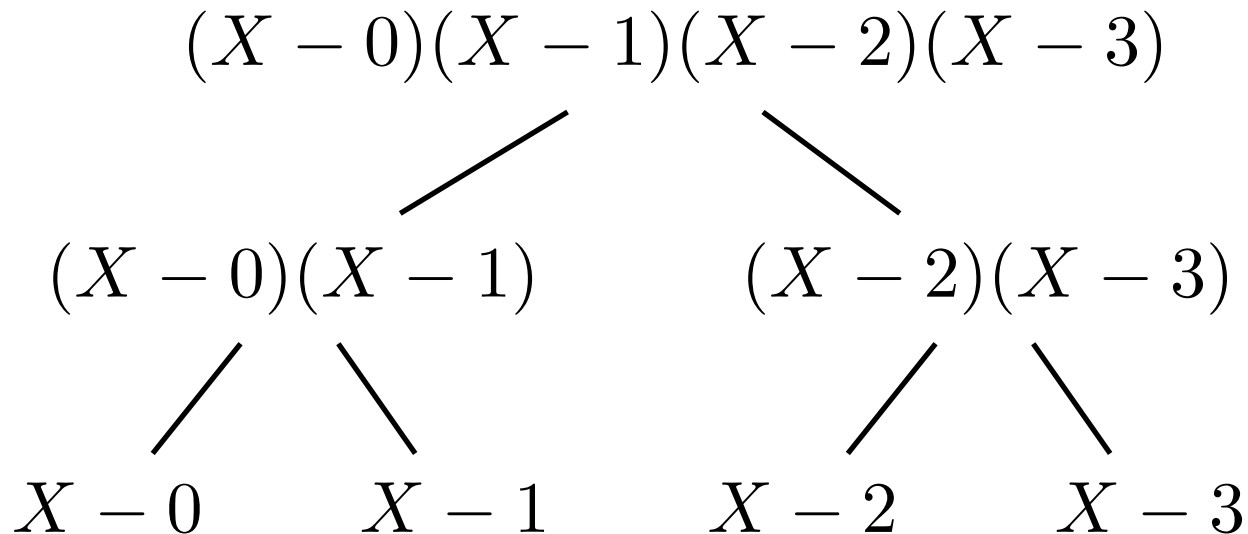
- 1:  $p(x) \leftarrow x(x-1)\dots(x-(n-1))$  ▷ arbre des produits
- 2:  $s(x) \leftarrow (n-1)x^{n-1} + \frac{(n-1)^2}{2}x^{n-2} + \dots + \frac{(n-1)^n}{n}$
- 3:  $r(x) \leftarrow \text{quo}(p(x)s(x), x^n)$  ▷ produit haut
- 4: **return**  $(r(0), r(1), \dots, r(n-1))$  ▷ évaluation multi-points



# Newton-Cotes rapide – Arbre des produits

---

Exemple :  $n = 4$ .



# Newton-Cotes rapide – Arbre des produits

---

Complexité binaire :  $d$  degré du polynôme,  $c$  borne sur la taille binaire de ses coefficients,  $k$  étage de l'arbre.

$$M(d, c) = M(dc)$$

$$T(k + 1) = 2T(k) + M(2^{2k}(\log_2 n + \mathcal{O}(1)))$$

$$\begin{aligned} T(\log_2 n) &= M(n^2 \log_2 n) + 2M\left(\frac{n^2}{4} \log_2 n\right) + \dots + nM(\log_2 n) \\ &= \mathcal{O}(M(n^2 \log n)) \end{aligned}$$



# Newton-Cotes rapide – Évaluation multi-points

---

$$r(X) = 3X^3 - \frac{27}{2}X^2 + 15X - \frac{9}{4}$$

$$3X^3 - \frac{27}{2}X^2 + 15X - \frac{9}{4} \bmod (X - 0)(X - 1)(X - 2)(X - 3)$$



# Newton-Cotes rapide – Évaluation multi-points

---

$$r(X) = 3X^3 - \frac{27}{2}X^2 + 15X - \frac{9}{4}$$

$$3X^3 - \frac{27}{2}X^2 + 15X - \frac{9}{4} \bmod (X - 0)(X - 1)(X - 2)(X - 3)$$

$$\frac{9}{2}X - \frac{9}{4} \bmod (X - 0)(X - 1)$$



# Newton-Cotes rapide – Évaluation multi-points

---

$$r(X) = 3X^3 - \frac{27}{2}X^2 + 15X - \frac{9}{4}$$

$$3X^3 - \frac{27}{2}X^2 + 15X - \frac{9}{4} \bmod (X - 0)(X - 1)(X - 2)(X - 3)$$

$$\frac{9}{2}X - \frac{9}{4} \bmod (X - 0)(X - 1)$$

$$-\frac{9}{4} \bmod X - 0$$



# Newton-Cotes rapide – Évaluation multi-points

---

$$r(X) = 3X^3 - \frac{27}{2}X^2 + 15X - \frac{9}{4}$$

$$3X^3 - \frac{27}{2}X^2 + 15X - \frac{9}{4} \bmod (X - 0)(X - 1)(X - 2)(X - 3)$$

$$\frac{9}{2}X - \frac{9}{4} \bmod (X - 0)(X - 1)$$

$$-\frac{9}{4} \bmod X - 0 \quad \frac{9}{4} \bmod X - 1$$



# Newton-Cotes rapide – Évaluation multi-points

---

$$r(X) = 3X^3 - \frac{27}{2}X^2 + 15X - \frac{9}{4}$$

$$3X^3 - \frac{27}{2}X^2 + 15X - \frac{9}{4} \bmod (X - 0)(X - 1)(X - 2)(X - 3)$$

$$\frac{9}{2}X - \frac{9}{4} \bmod (X - 0)(X - 1)$$

$$\frac{9}{2}X - \frac{45}{4} \bmod (X - 2)(X - 3)$$

$$-\frac{9}{4} \bmod X - 0$$

$$\frac{9}{4} \bmod X - 1$$



# Newton-Cotes rapide – Évaluation multi-points

---

$$r(X) = 3X^3 - \frac{27}{2}X^2 + 15X - \frac{9}{4}$$

$$3X^3 - \frac{27}{2}X^2 + 15X - \frac{9}{4} \bmod (X - 0)(X - 1)(X - 2)(X - 3)$$

$$\frac{9}{2}X - \frac{9}{4} \bmod (X - 0)(X - 1)$$

$$\frac{9}{2}X - \frac{45}{4} \bmod (X - 2)(X - 3)$$

$$-\frac{9}{4} \bmod X - 0$$

$$\frac{9}{4} \bmod X - 1$$

$$-\frac{9}{4} \bmod X - 2$$



# Newton-Cotes rapide – Évaluation multi-points

---

$$r(X) = 3X^3 - \frac{27}{2}X^2 + 15X - \frac{9}{4}$$

$$3X^3 - \frac{27}{2}X^2 + 15X - \frac{9}{4} \bmod (X - 0)(X - 1)(X - 2)(X - 3)$$

$$\frac{9}{2}X - \frac{9}{4} \bmod (X - 0)(X - 1)$$

$$\frac{9}{2}X - \frac{45}{4} \bmod (X - 2)(X - 3)$$

$$-\frac{9}{4} \bmod X - 0$$

$$\frac{9}{4} \bmod X - 1$$

$$-\frac{9}{4} \bmod X - 2$$

$$\frac{9}{4} \bmod X - 3$$

$$\mathcal{O}(n^2 \log^2 n)$$



# Newton-Cotes rapide – Ajustement

---

$$w_0 = \frac{(-1)^{4-1-0}}{0!(4-1-0)!} \frac{-9}{4} = \frac{3}{8}$$

$$w_1 = \frac{(-1)^{4-1-1}}{1!(4-1-1)!} \frac{9}{4} = \frac{9}{8}$$

$$w_2 = \frac{(-1)^{4-1-2}}{2!(4-1-2)!} \frac{-9}{4} = \frac{9}{8}$$

$$w_3 = \frac{(-1)^{4-1-3}}{3!(4-1-3)!} \frac{9}{4} = \frac{3}{8}$$

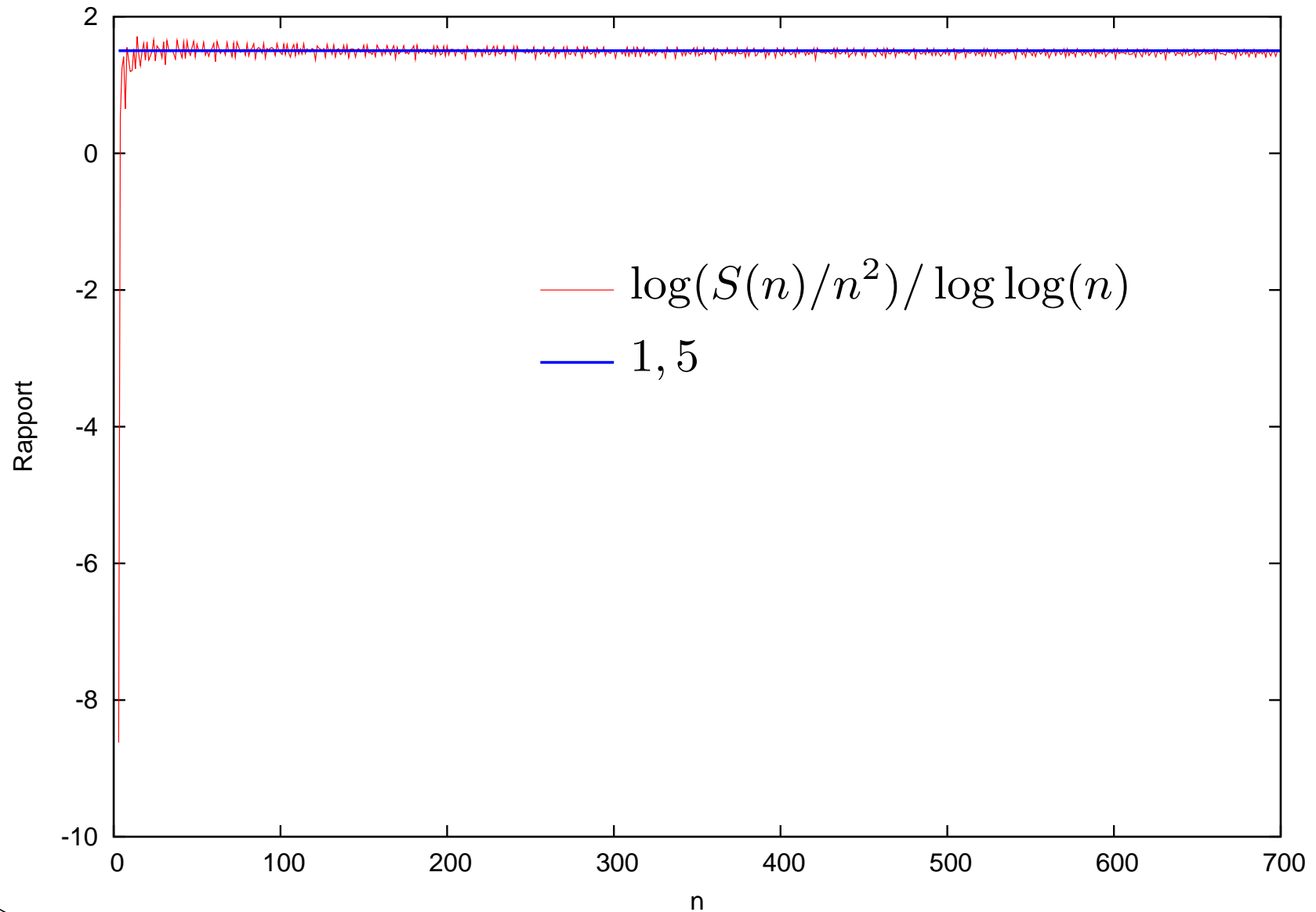


# Newton-Cotes rapide – Complexité

Opération	Coût
$p(x) \leftarrow x(x-1)\dots(x-(n-1))$ [arbre des produits]	$\mathcal{O}(n^2 \log^2 n)$
$s(x) \leftarrow (n-1)x^{n-1} + \frac{(n-1)^2}{2}x^{n-2} + \dots + \frac{(n-1)^n}{n} \pmod f$	$\mathcal{O}(n^2 \log^2 n)$
$r(x) \leftarrow \text{quo}(p(x)s(x), x^n) \pmod f$ [produit court haut]	$\mathcal{O}(n^2 \log^2 n)$
$(r(0), r(1), \dots, r(n-1)) \pmod f$ [évaluation multi-points]	$\mathcal{O}(n^2 \log^2 n)$
$(r(0), r(1), \dots, r(n-1)) \in \mathbb{Q}$ [reconstruction rationnelle]	$\mathcal{O}(n^2 \log^2 n)$
Total	$\mathcal{O}(n^2 \log^2 n)$
Naïf	$\mathcal{O}(n^3 \log^2 n)$



# Taille des coefficients (NC)



# Méthode de Newton-Cotes

---

Analyse d'erreur

# Analyse d'erreur

ENTRÉE :  $\hat{a}$  et  $\hat{b}$  les arrondis au plus proche sur  $p$  bits des bornes  $a$  et  $b$ ,  $f$ ,  $n$  le nombre de points de la méthode,  $\{v_i\}_{i \in \{0, n-1\}}$  et  $d$  tels que  $w_i = \frac{v_i}{d}$ .

SORTIE :  $\hat{I}$ .

```
1: for  $i \leftarrow 0$  to  $n - 1$  do
2:    $\hat{t} \leftarrow \circ(\hat{a} \cdot (n - i - 1))$ 
3:    $\hat{u} \leftarrow \circ(\hat{b} \cdot i)$ ,  $\hat{s} \leftarrow \circ(\hat{t} + \hat{u})$ 
4:    $\hat{x}_i \leftarrow \circ(\hat{s} / (n - 1))$ 
5:    $\hat{z}_i \leftarrow \circ(f(\hat{x}_i))$ ,  $\hat{y}_i \leftarrow \circ(\hat{z}_i \cdot v_i)$ 
6: end for
7:  $\hat{S} \leftarrow \text{somme}(\hat{y}_i, i = 0 \dots n - 1)$ 
8:  $q \leftarrow d \cdot (n - 1)$ 
9:  $\hat{U} \leftarrow \circ(\hat{S} / q)$ 
10:  $\hat{D} \leftarrow \circ(\hat{b} - \hat{a})$ 
11: return  $\circ(\hat{D}\hat{U}) = \hat{I}$ 
```

$$x_i = \frac{(n-i-1)a+ib}{n-1}$$
$$y_i = d \cdot w_i f(x_i)$$

$$S = \sum_{i=0}^{n-1} d \cdot w_i f(x_i)$$

$$I = \frac{b-a}{n-1} \sum_{i=0}^{n-1} w_i f(x_i)$$



# Analyse d'erreur

**Théorème 1.** Soit  $\delta_{\hat{y}_i} = 6|v_i| \cdot m \cdot \text{ulp}(\hat{x}_i) + \frac{5}{2}\text{ulp}(\hat{y}_i)$ .

L'erreur  $|I - \hat{I}|$  est bornée par :

$$B_{\text{totale}} = 24 \text{ulp}(\hat{I}) + \frac{1}{2}|\hat{U}| \left( \text{ulp}(\hat{a}) + \text{ulp}(\hat{b}) \right) + \frac{25\hat{D}}{4d} \max(\delta_{\hat{y}_i})$$
$$+ \begin{cases} \frac{1}{8} \left( \frac{b-a}{n-1} \right)^{n+2} M & \text{si } n \text{ est impair,} \\ \frac{1}{4} \left( \frac{b-a}{n-1} \right)^{n+1} M & \text{sinon.} \end{cases}$$



# Analyse d'erreur

---

$$\hat{x}_i = \circ \left( \frac{\circ(\circ((n-1-i) \cdot \hat{a}) + \circ(i \cdot \hat{b}))}{n-1} \right)$$

$$\hat{u} = \circ(i\hat{b})$$

$$\hat{t} = \circ((n-1-i)\hat{a})$$

$$|\circ(i \cdot \hat{b}) - ib| \leq |\circ(i \cdot \hat{b}) - i \cdot \hat{b}| + |i \cdot \hat{b} - ib|$$

$$\leq \frac{1}{2} \text{ulp}(\circ(i\hat{b})) + \frac{i}{2} \text{ulp}(\hat{b})$$

$$\leq \frac{3}{2} \text{ulp}(\circ(i\hat{b})) = \frac{3}{2} \text{ulp}(\hat{u})$$

$$|\hat{t} - t| \leq \frac{3}{2} \text{ulp}(\hat{t}).$$



# Analyse d'erreur

---

$$\hat{x}_i = \circ \left( \frac{\circ(\hat{t} + \hat{u})}{n - 1} \right)$$

$$\hat{s} = \circ(\hat{t} + \hat{u})$$

$$|\hat{s} - s| \leq \frac{1}{2} \text{ulp}(\hat{s}) + \frac{3}{2} (\text{ulp}(\hat{t}) + \text{ulp}(\hat{u}))$$

$$\leq \frac{11}{4} \text{ulp}(\hat{s})$$



# Analyse d'erreur

---

$$\begin{aligned}\hat{x}_i &= \circ \left( \frac{\hat{s}}{n-1} \right) \\ |x_i - \hat{x}_i| &\leq \frac{1}{2} \text{ulp}(\hat{x}_i) + \frac{11}{4(n-1)} \text{ulp}(\hat{s}) \\ &\leq \frac{1}{2} \text{ulp}(\hat{x}_i) + \frac{11}{2} \text{ulp}(\hat{x}_i) \\ &\leq 6 \cdot \text{ulp}(\hat{x}_i).\end{aligned}$$



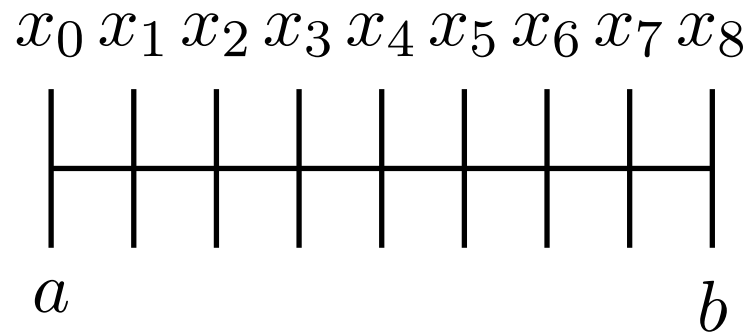
- 
1. Motivations et enjeux
  2. État de l'art et contributions
    - (a) Problématique
    - (b) État de l'art
    - (c) Méthodes par interpolation
    - (d) Méthode de Newton-Cotes
    - (e) Méthode de Gauss-Legendre
    - (f) Résultats expérimentaux
  3. Conclusion et perspectives



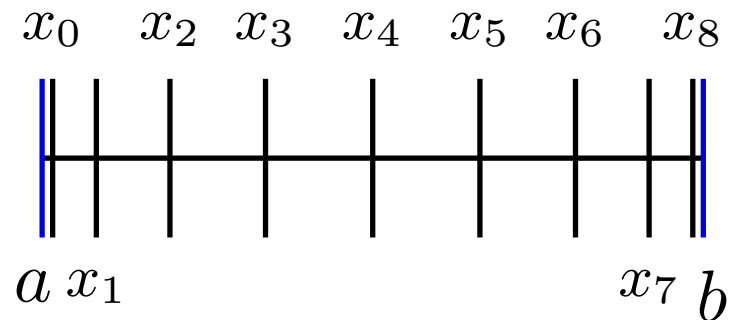
# Méthode de Gauss-Legendre

---

Newton-Cotes :



Gauss-Legendre :



# Méthode de Gauss-Legendre

---

● Famille des schémas d'intégration de Gauss, avec divers poids :

● Gauss-Tchebychev (premier ordre)

$$I(f) = \int_{-1}^1 \frac{1}{\sqrt{1-x^2}} f(x) dx,$$

● Gauss-Tchebychev (deuxième ordre)

$$I(f) = \int_{-1}^1 \sqrt{1-x^2} f(x) dx,$$

● Gauss-Legendre (poids unitaire)

$$I(f) = \int_{-1}^1 f(x) dx;$$

● ordre  $2n - 1$  pour  $n$  points d'évaluation.



# Méthode de Gauss-Legendre

---

Le produit scalaire

$$\langle f, g \rangle = \int_a^b f(x)g(x)dx$$

définit une famille de polynômes orthogonaux  $(p_i)_{i \geq 0}$  :

$$\forall i \in \mathbb{N}, \deg(p_i) = i$$

$$\forall (i, j) \in \mathbb{N}^2, \langle p_i, p_j \rangle = 1 \quad \text{si } i = j$$

$x_0 < x_1 < \dots < x_{n-1}$  les  $n$  racines distinctes de  $p_n$  dans  $]a, b[$ .

Polynômes de Legendre  $P_n$  : définis pour  $[a, b] = [-1, 1]$  et normalisés par  $P(1) = 1$ .



# Méthode de Gauss-Legendre

---

- Les  $n$  points d'évaluations sont les racines  $(x_i)$  de  $P_n$  ;
- Les poids  $w_i$  sont donnés par  $w_i = \frac{2}{(1-x_i^2)P_n'(x_i)}$ .

Algorithme (haut niveau) :

1. Calculer  $P_n$ .
2. Calculer les racines  $(x_i)$  de  $P_n$  (**isolation** puis **raffinement**).
3. Calculer les poids  $(w_i)$ .
4. Appliquer la formule d'intégration :

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n-1} w_i f(x_i)$$



# Algorithmes

---

Récurrance des polynômes de Legendre :

$$\left\{ \begin{array}{l} P_0(X) = 1 \\ P_1(X) = X \\ (n+1)P_{n+1}(X) = (2n+1)XP_n(X) - nP_{n-1}(X) \end{array} \right.$$

# Algorithmes

---

Récurrence des polynômes de Legendre :

$$\left\{ \begin{array}{l} P_0(X) = 1 \\ P_1(X) = X \\ (n+1)P_{n+1}(X) = (2n+1)XP_n(X) - nP_{n-1}(X) \end{array} \right.$$

et  $P_n$  peut s'écrire

$$P_n(X) = \begin{cases} 2^{-n} Q_n(X^2) & \text{si } n \text{ est pair} \\ 2^{-n} X Q_n(X^2) & \text{sinon.} \end{cases}$$



# Calcul de racines de polynômes réels

---

Problème : calculer les racines réelles  $(x_i)$  d'un polynôme  $P$  (trouver un flottant proche de  $x_i$  avec tous les bits corrects, à un ulp près).

Deux étapes :

- Isolation : intervalles contenant exactement une racine de  $P$ ,
- Raffinement : calculer chaque racine avec une précision aussi grande que désirée :
  - Méthode de la sécante,
  - Itération de Newton.



# Isolation (Méthode d'Uspensky)

---

Exemple : prenons pour  $P$  le polynôme de Legendre  $P_{17}$  de degré  $n = 17$ .

$$P_{17}(X) = 2^{-15} Q_{17}(X)$$

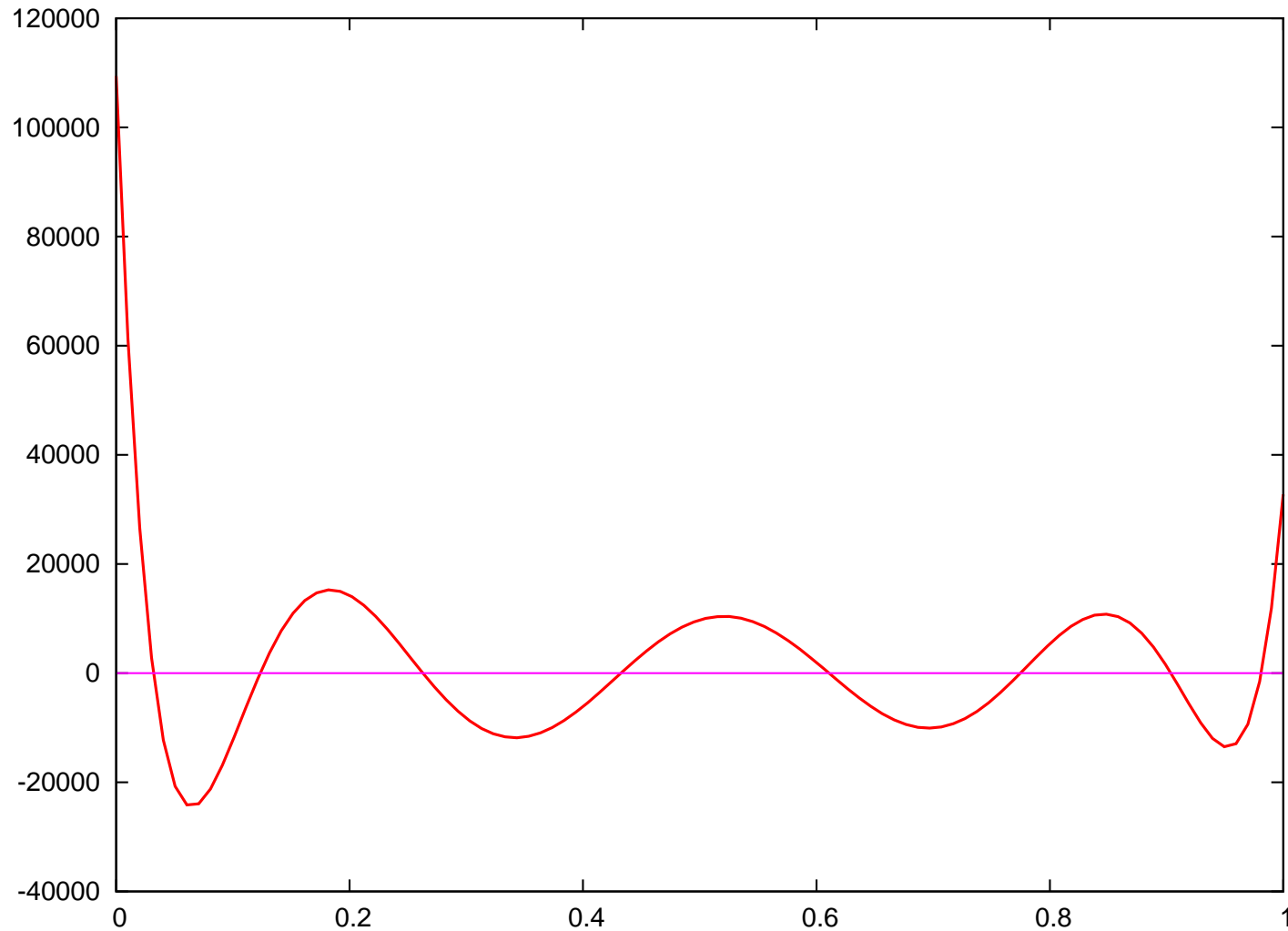
$$Q_{17}(X) = 583401555X^8 - 2404321560X^7 + 4071834900X^6 \\ - 3650610600X^5 + 1859107250X^4 - 535422888X^3 \\ + 81477396X^2 - 5542680X + 109395.$$

$$\{x \mid P(x) = 0\} = \{\pm\sqrt{y} \mid Q(y) = 0\} \quad (\cup \{0\})$$

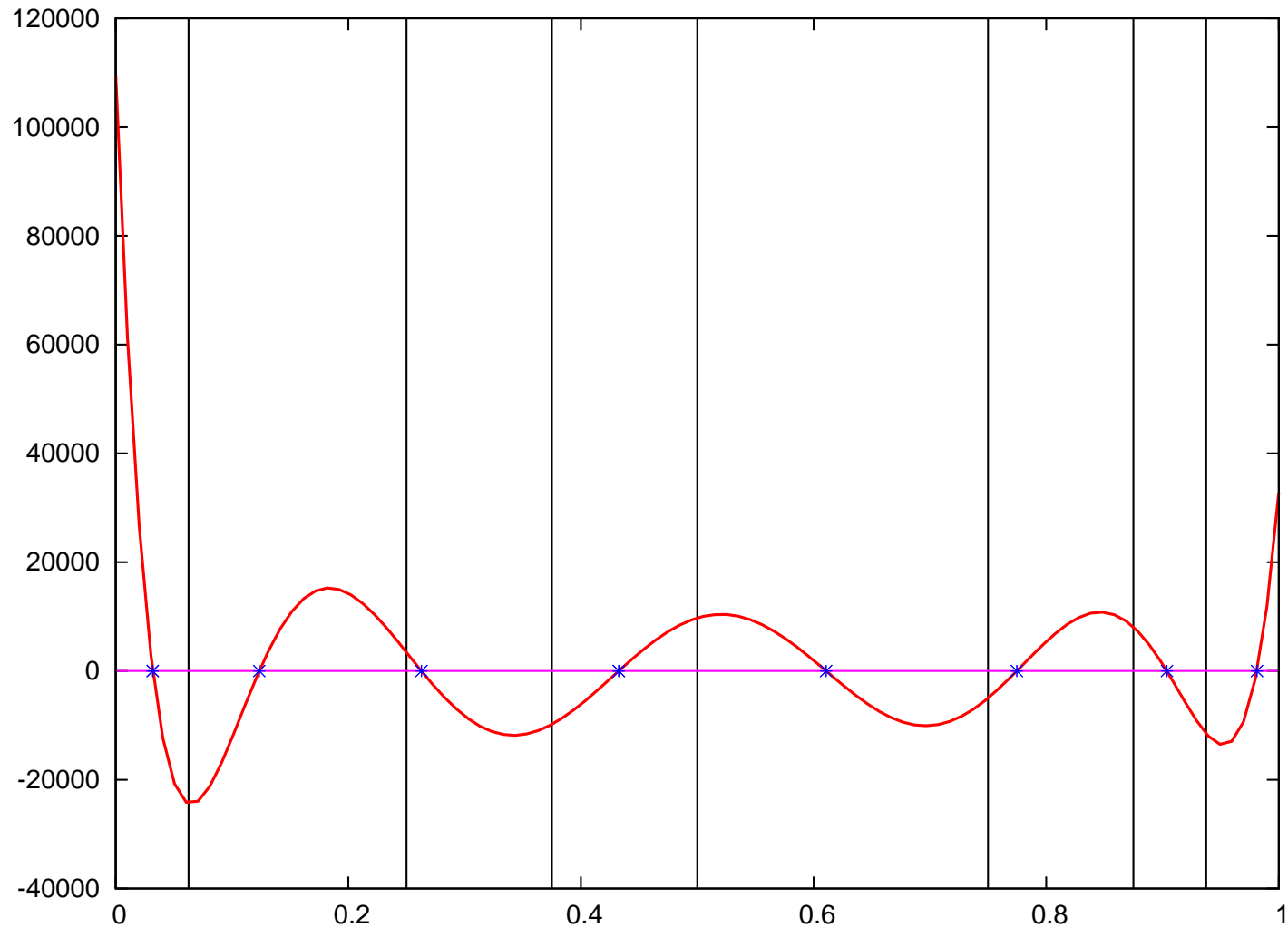
Règle des signes de Descartes : critère simple basé sur l'alternance des signes des coefficients du polynôme.



# Isolation – $Q_{17}$ sur $[0, 1]$



# Isolation – $Q_{17}$ sur $[0, 1]$



# Raffinement

---

$[x_{\min}, x_{\max}]$  connu, trouver un flottant  $y$  tel que  $|y - u| \leq \text{ulp}(y)$  sachant

$$u \in [x_{\min}, x_{\max}]$$

$$P(u) = 0$$

$$P(x) = 0 \wedge x \in [x_{\min}, x_{\max}] \Rightarrow x = u$$

- Dichotomie entière, tronquée 1 bit gagné par étape
- Méthode de la sécante ordre de convergence  $\frac{1+\sqrt{5}}{2} \approx 1,618$
- Itération de Newton (scalaire, par intervalles) ordre de convergence 2



# Méthode de la sécante

---

Récurrence d'ordre 2 :

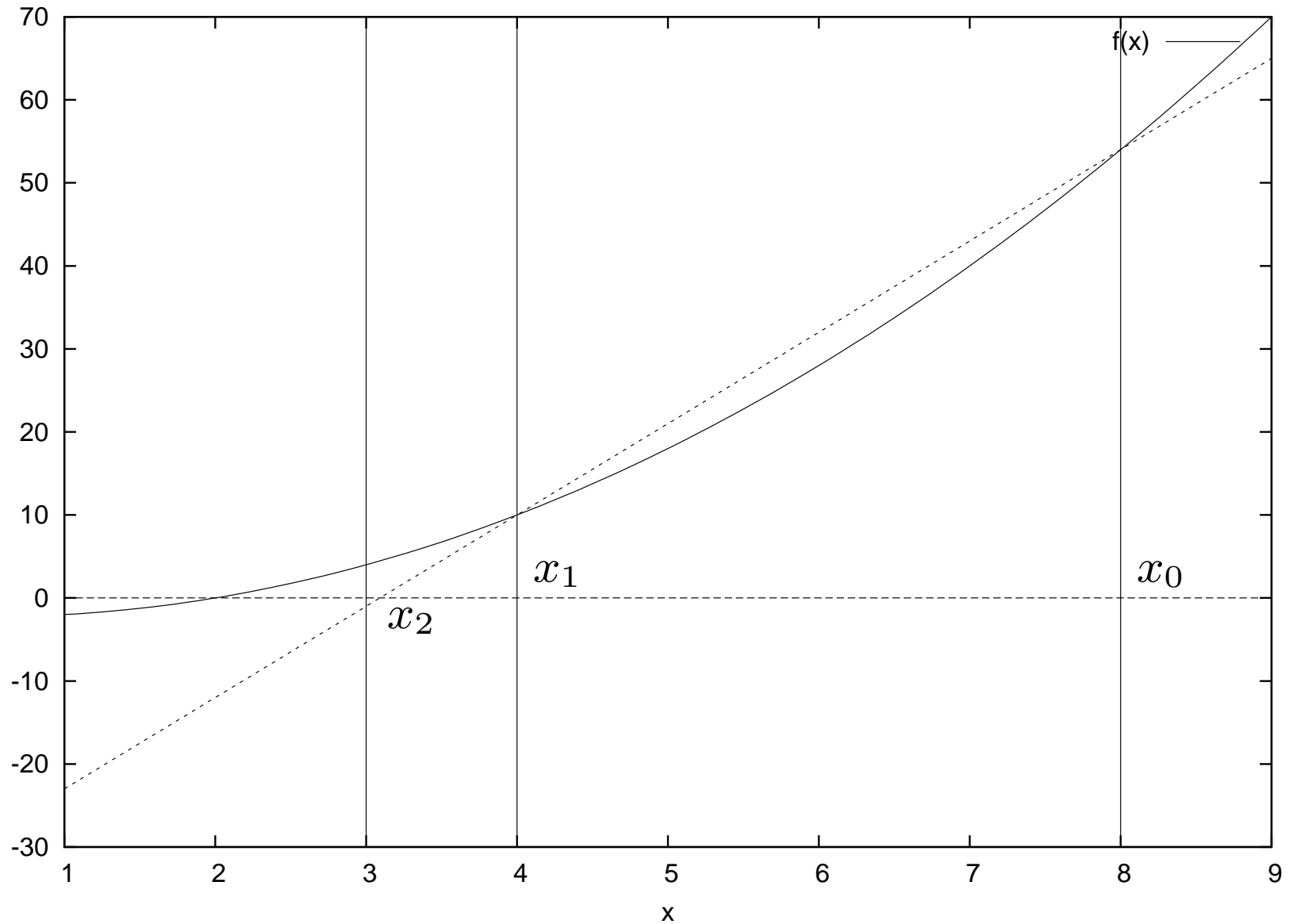
$$x_{n+2} = \frac{x_{n+1}f(x_n) - x_n f(x_{n+1})}{f(x_n) - f(x_{n+1})}$$

$$x_{n+2} - u \approx \frac{f''(u)}{2f'(u)} (x_{n+1} - u)(x_n - u)$$

$$-\log_2 |x_{n+2} - u| \approx -\log_2 |x_{n+1} - u| - \log_2 |x_n - u| + C$$

- on converge en « Fibonacci » (dans le « domaine de convergence ») ;
- on aimerait garder un intervalle d'isolation.

# Méthode de la sécante



# Méthode de la sécante

$\frac{f''(u)}{f'(u)}$	$x_n - u$	$x_{n+1} - u$	$x_{n+2} - u$
+	-	+	-
+	+	-	-
+	-	-	+
+	+	+	+
-	-	+	+
-	+	+	-
-	+	-	+
-	-	-	-

# Méthode de la sécante

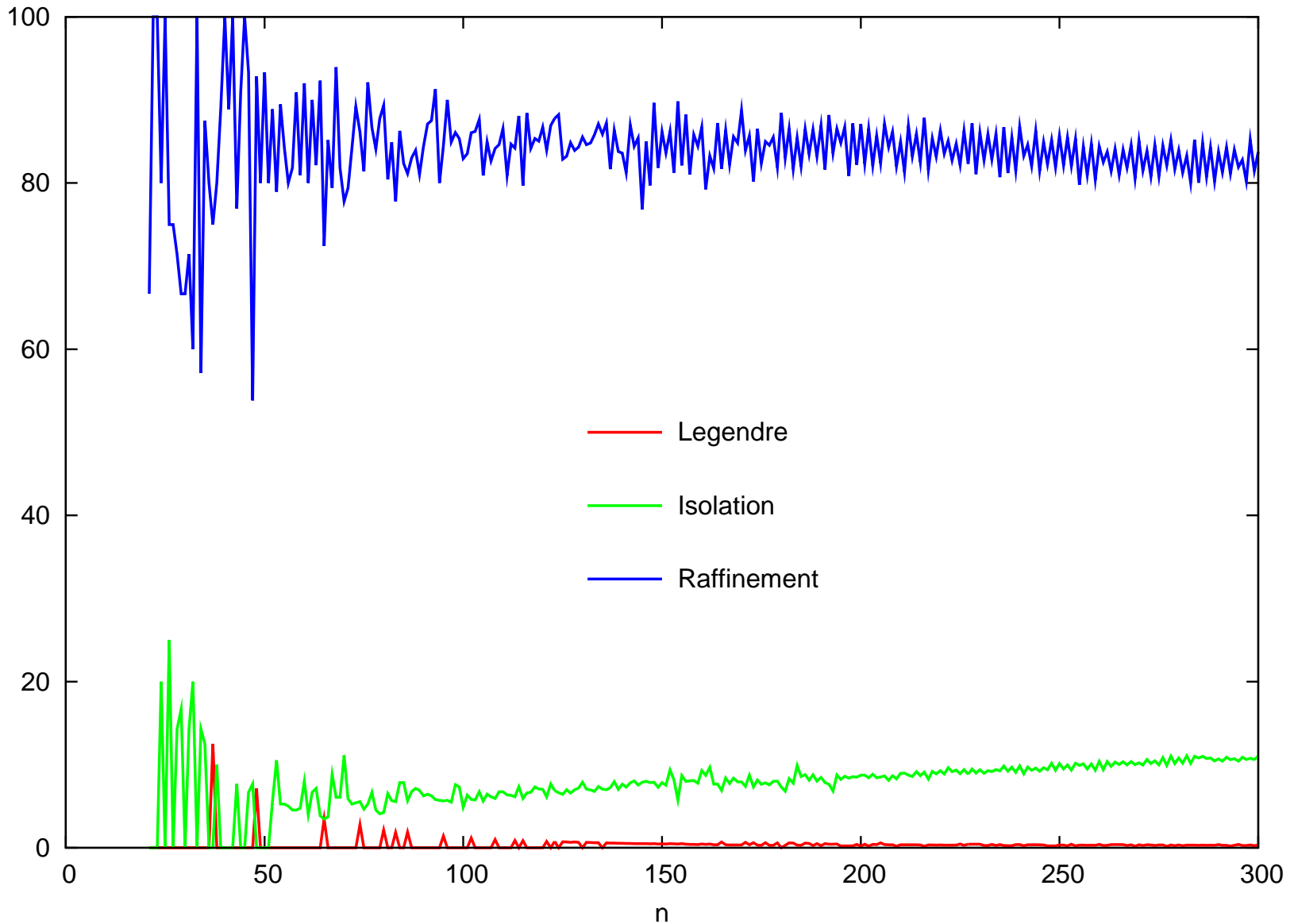
---

Optimale en temps :

- On peut estimer la valeur de la constante  $C$  ;
- Isolation des racines de  $f'$  et  $f''$  pour se ramener dans les conditions du tableau des signes ;
- Une étape de dichotomie si on perd l'intervalle d'isolation.



# Temps de calcul (GL)



# Analyse d'erreur

---

ENTRÉES :  $\widehat{a}, \widehat{b} - a, (\widehat{w}_i), f, (\widehat{v}_i), n$  où  $\widehat{v}_i = o\left(\frac{1+x_i}{2}\right)$ .

SORTIE :  $\widehat{I}$

```
1: for  $i \leftarrow 0$  to  $n - 1$  do  
2:    $t \leftarrow o((\widehat{b} - a) \cdot \widehat{v}_i)$   
3:    $\widehat{x}_i \leftarrow o(t + \widehat{a})$   
4:    $\widehat{f}_i \leftarrow o(f(\widehat{x}_i))$   
5:    $\widehat{y}_i \leftarrow o(\widehat{f}_i \cdot \widehat{w}_i)$   
6: end for  
7:  $\widehat{S} \leftarrow \text{sum}(\widehat{y}_i, i = 0 \dots n - 1)$   
8:  $\widehat{D} \leftarrow o(\widehat{b} - a)/2$   
9: return  $o(\widehat{D}\widehat{S}) = \widehat{I}$ 
```



# Analyse d'erreur

---

**Théorème 2.** Soit  $\delta_{\hat{y}_i} = \frac{11}{4} \text{ulp}(\hat{y}_i) + 6m\hat{w}_i \text{ulp}(\hat{x}_i)$ ,  $p \geq 2$  la précision de calcul,  $m$  une borne sur  $|f'|$  et  $M$  une borne sur  $|f^{(2n)}|$ . Pour  $n$  points d'intégration l'erreur  $|\hat{I} - I|$  est bornée par :

$$B_{totale} = \frac{21}{4} \text{ulp}(\hat{I}) + \frac{5n}{4} \hat{D} \cdot \max(\delta_{\hat{y}_i}) + \frac{(b-a)^{2n+1} (n!)^4}{(2n+1)[(2n)!]^3} M.$$



- 
1. Motivations et enjeux
  2. État de l'art et contributions
    - (a) Problématique
    - (b) État de l'art
    - (c) Méthodes par interpolation
    - (d) Méthode de Newton-Cotes
    - (e) Méthode de Gauss-Legendre
    - (f) Résultats expérimentaux
  3. Conclusion et perspectives



# Résultats expérimentaux

---

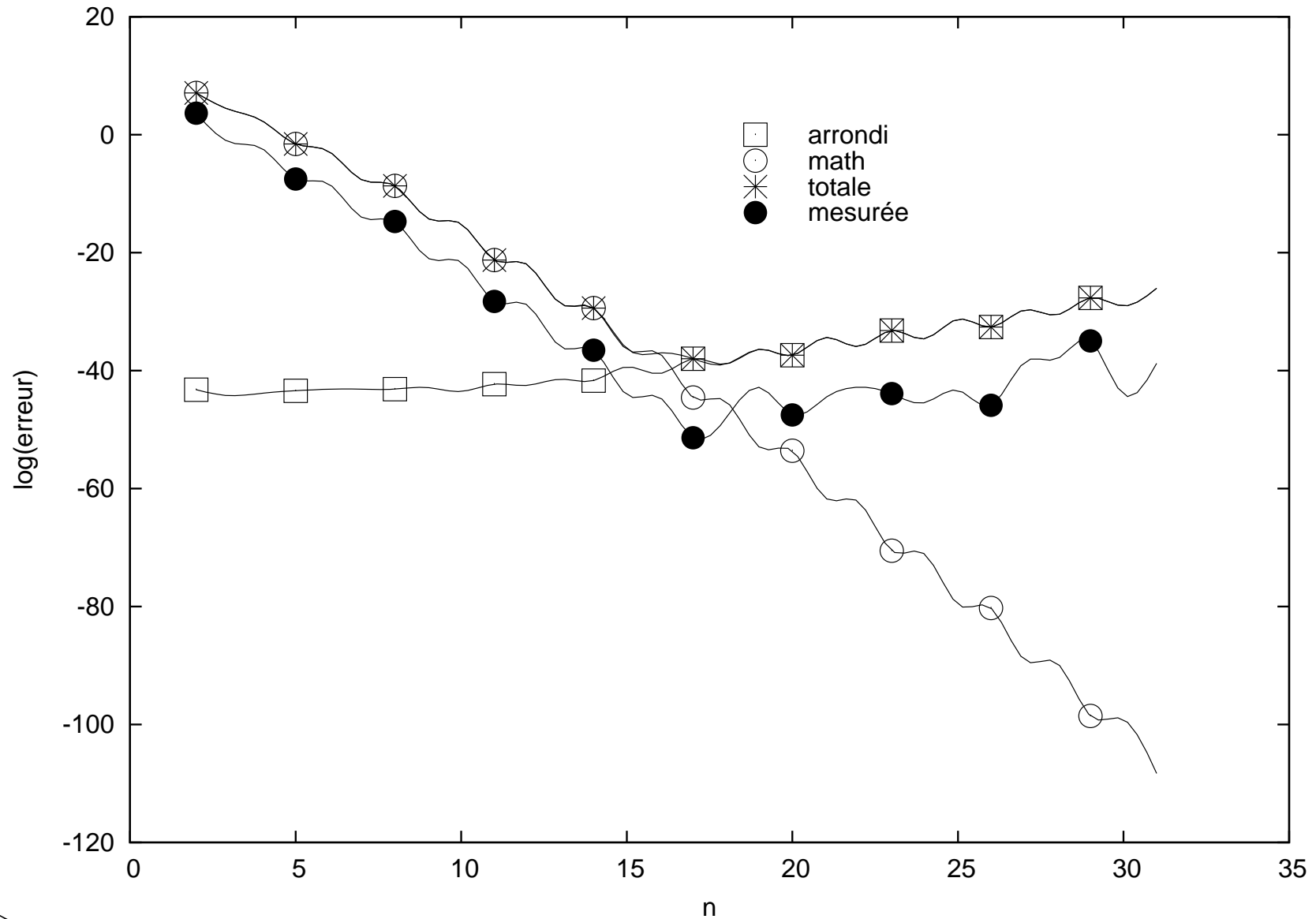
CRQ : « **C**orrectly **R**ounded **Q**uadrature »

- Bibliothèque de fonctions d'intégration en langage C,
- Comprend Newton-Cotes, Gauss-Legendre, fonctions d'isolation et de raffinement,
- > 5000 lignes de code,
- Plusieurs niveaux d'interface en fonction du besoin,
- Logiciel libre.

<http://komite.net/laurent/soft/crq/>



# Résultats expérimentaux



# Résultats expérimentaux

---

$$\int_{17}^{42} \exp(-x^2) \log x dx =$$

2,56572850056105148291735639613047859001477095540203266250504462960653767  
3604161880791363955753269531192182476023077273679855510960003686403593678  
1217907068647919804628723310428020493750490122162013404615358361319373817  
7820412122516350777255525035947116513676784199592200655526485894447669230  
5152217629197427907047695328832536058393272714140908897784950204746698754  
74902063790677460904673502379529481309977120595033841904051711412e-127

$\pm 0,5\text{ulp}$

Maple 10 :  $0,2604007480 \times 10^{-126}$ .



# Comparaison

$$I = \int_0^1 \max(\sin(x), \cos(x)) dx = \sqrt{2} - \cos 1$$

Précision		31	61	151	302	603	1506
CRQ	Temps	26ms	110ms	1s	11s	108s	3138s
	Erreur	1	2.6e-2	9.2e-1	3.6	4.2	3.5
Maple	Temps	6s	38s	371s	2618s	–	–
	Erreur	2.1e-1	1.7e-1	1.7e-1	2.9e-1	–	–
PARI/GP	Temps	20ms	80ms	480ms	3s	22s	301s
	Erreur	3.1e25	2.0e54	2.7e144	1.0e295	1.7e595	9.2e1496



# Conclusion

---

- On peut intégrer **efficacement** en grande **précision** :
  - calcul rapide des poids de Newton-Cotes (amélioration de la complexité théorique) ;
  - isolation et raffinement rapide de racines de polynômes (sécante, Newton) ;
- sans sacrifier la **qualité** des résultats :
  - raffinement avec borne d'erreur ;
  - borne d'erreur explicite pour les méthodes de Newton-Cotes et de Gauss-Legendre à tout ordre ;
- et le logiciel est disponible (CRQ).



# Perspectives

---

- Accélérer le calcul des poids de Gauss-Legendre par raffinement simultané.

# Perspectives

---

- Accélérer le calcul des poids de Gauss-Legendre par raffinement simultané.
- Étendre les résultats au reste de la famille des méthodes de Gauss.



# Perspectives

---

- Accélérer le calcul des poids de Gauss-Legendre par raffinement simultané.
- Étendre les résultats au reste de la famille des méthodes de Gauss.
- Analyser la méthode « double exponentielle ».



# Perspectives

---

- Accélérer le calcul des poids de Gauss-Legendre par raffinement simultané.
- Étendre les résultats au reste de la famille des méthodes de Gauss.
- Analyser la méthode « double exponentielle ».
- Résolution d'équations différentielles « avec erreur bornée ».



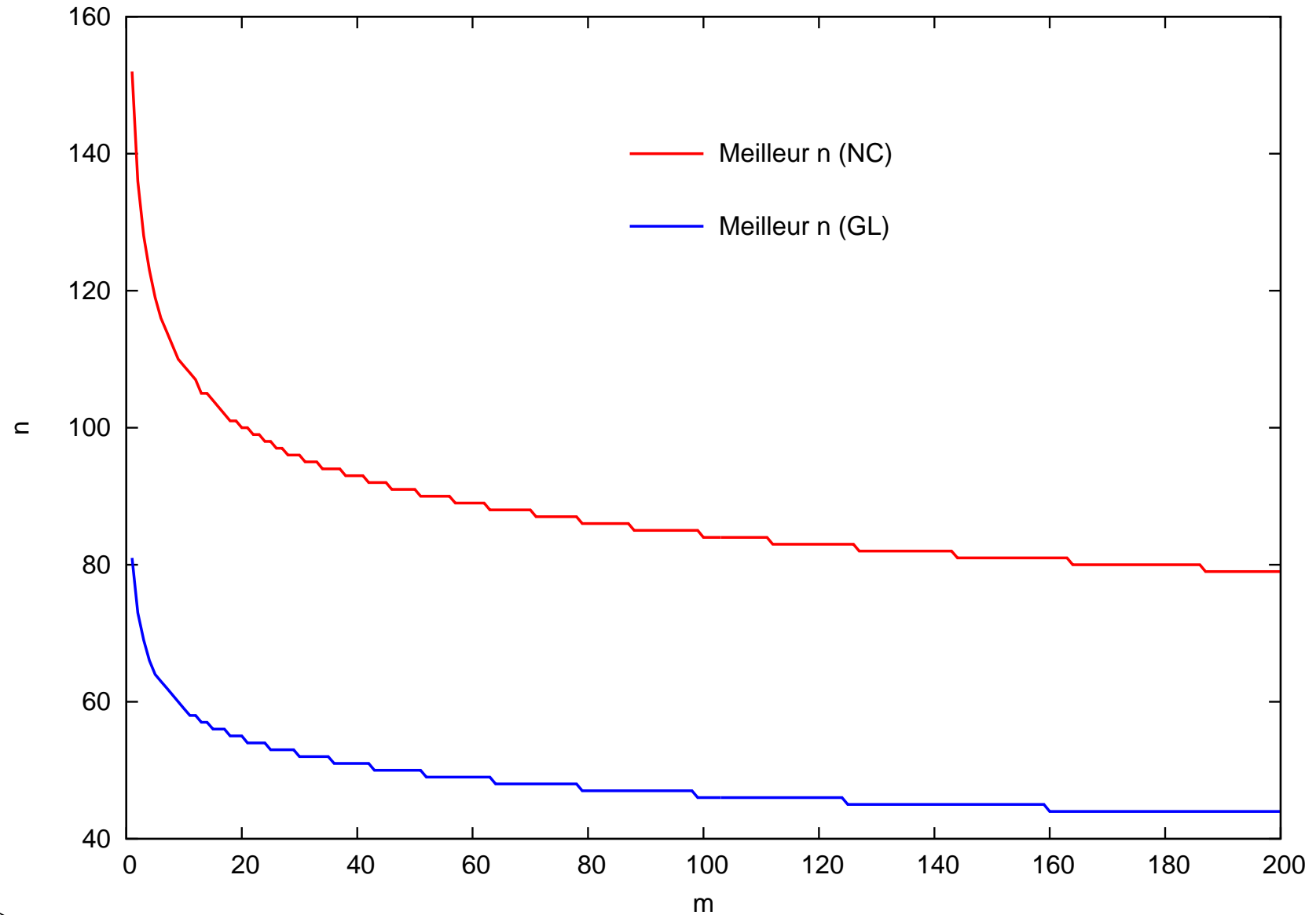
# Perspectives

---

- Accélérer le calcul des poids de Gauss-Legendre par raffinement simultané.
- Étendre les résultats au reste de la famille des méthodes de Gauss.
- Analyser la méthode « double exponentielle ».
- Résolution d'équations différentielles « avec erreur bornée ».
- Intégration numérique formellement prouvée (Coq).



# Effet de la composition



# Effet de la composition

