On evaluating multivariate Taylor polynomials

Andreas Griewank[†‡], Jan Riehme[‡], Jean Utke[⋆]
[†] Matheon Research Center Berlin
[‡] Humboldt Universität zu Berlin,
[⋆] Argonne National Laboratory, USA

Matheon Workshop in honor of the 60th birthday of Richard Brent
Computing by the Numbers: Algorithms, Precision, and Complexity
July 20 - 21, 2006, Berlin

Table of Contents

Table of Contents

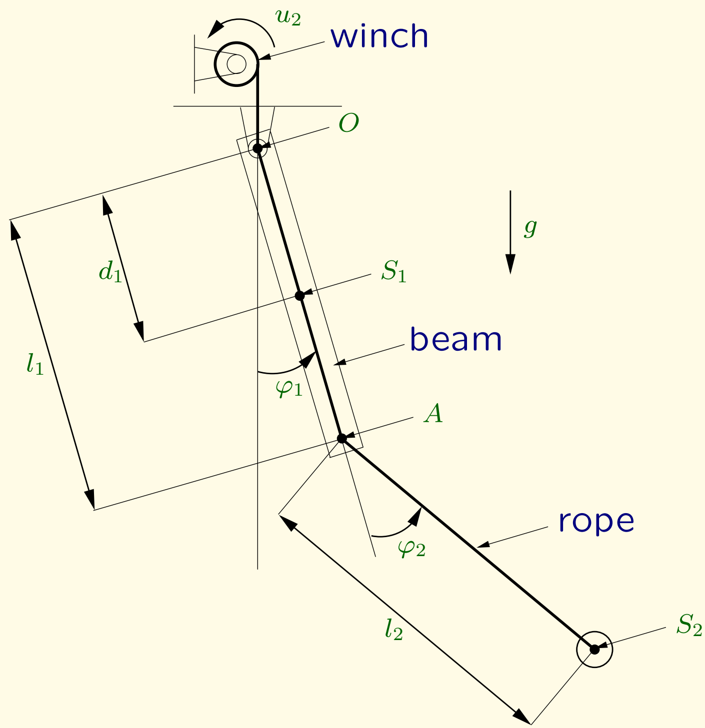
Table of Contents

Table of Contents

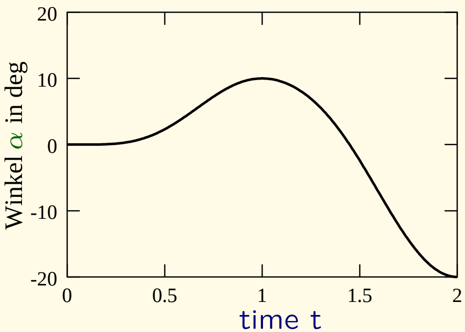
Table of Contents

Numerical System Inversion

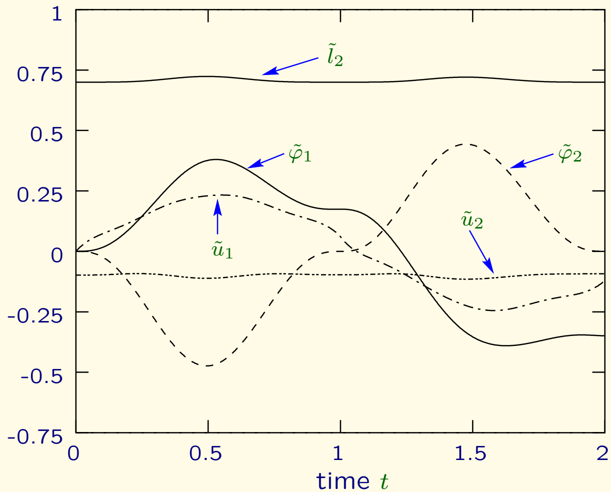
Physical model of a manipulator:



Goal: Compute input for prescribed output!



$\overset{?}{\Longrightarrow}$



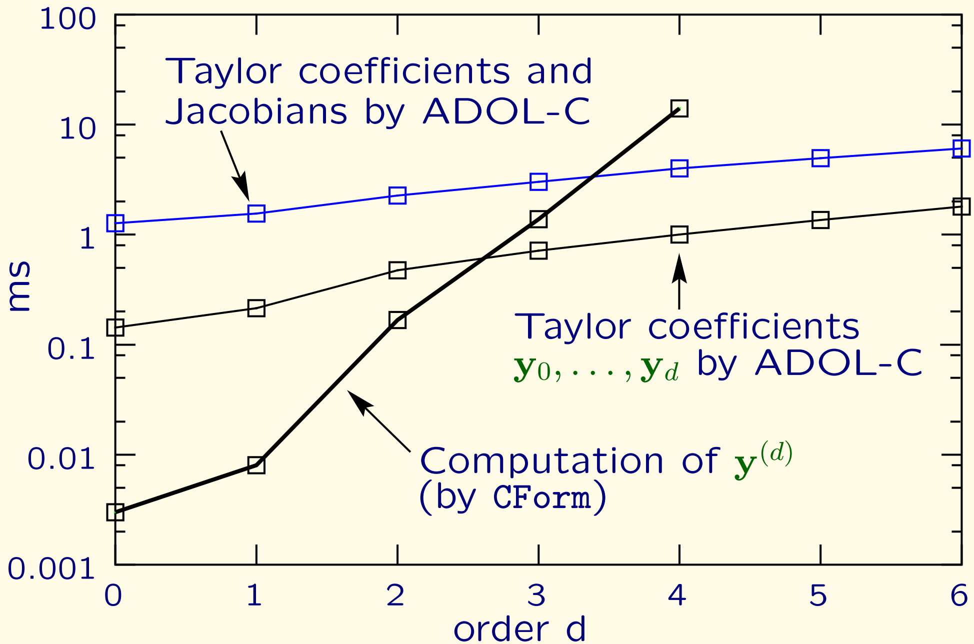
$$\begin{aligned} \tilde{y}_1(t) &= \sin\alpha(t) \\ \tilde{y}_2(t) &= \cos\alpha(t) \end{aligned}$$

ADOL-C vs. Mathematica

C code generated by Mathematica (`CForm`):

derivative order	lines of C code	
d	$y_1^{(d)}$	$y_2^{(d)}$
0	1	1
1	2	2
2	49	49
3	416	421
4	4149	4164
5	56619	57027

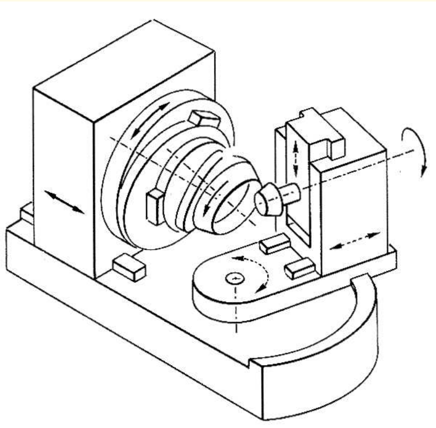
Runtimes:



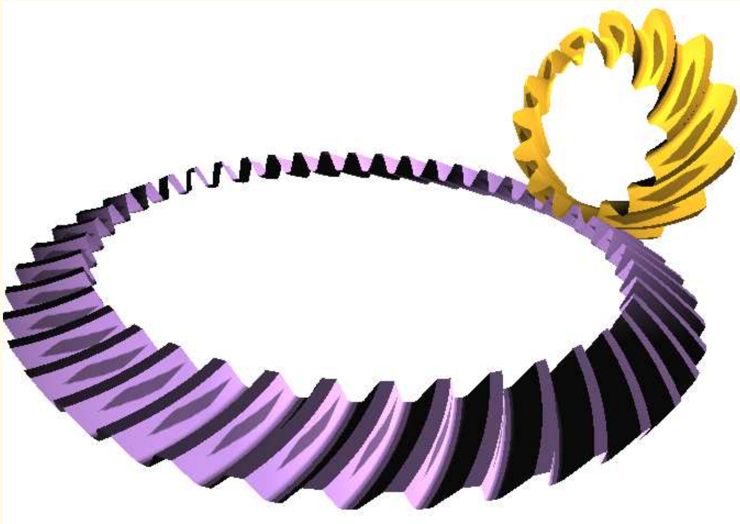
100
10
1
0.1
0.01
0.001
ms
0
1
2
3
4
5
6
order d
Taylor coefficients and
Jacobians by ADOL-C
Taylor coefficients
$\mathbf{y}_0, \ldots, \mathbf{y}_d$ by ADOL-C
Computation of $\mathbf{y}^{(d)}$
(by CForm)

Gear Tooth Contact Analysis and Optimization

Virtual Machine Tool



$\overset{?}{\Longrightarrow}$



$$F : \mathbb{R}^3 \times \mathbb{R}^{n_p} \to \mathbb{R}^3 :$$
$$(w, p) \mapsto x = F(w, p)$$

Evaluation of Restricted Higher Derivative Tensors

$$\nabla_S^k F(u) \equiv \frac{\partial^k}{\partial v^k} F(u + Sv)\Big|_{v=0} \in \mathbb{R}^{3\times q^k}, \qquad u \equiv (w,p) \in \mathbb{R}^{3+n_p}$$

$$S \in \mathbb{R}^{(3+n_p)\times q}$$

q	Maple	ADOL-C	Ratio
1	$0.12\ ms$	$0.20\ ms$	0.60
2	$0.13\ ms$	$0.45\ ms$	0.29
3	$0.14\ ms$	$0.99\ ms$	0.14

Table 1: $n_p = 0$

q	Maple	ADOL-C	Ratio
1	$1.02\ ms$	$0.25\ ms$	4.1
2	$1.20\ ms$	$0.60\ ms$	2.0
3	—	$1.28\ ms$	—

Table 2: $n_p = 6$

q	Maple	ADOL-C	Ratio
1	—	$0.28\ ms$	—
2	—	$0.67\ ms$	—
3	—	$1.49\ ms$	—

Table 3: $n_p = 12$

q	Maple	ADOL-C	Ratio
1	—	$0.32\ ms$	—
2	—	$0.82\ ms$	—
3	—	$1.49\ ms$	—

Table 4: $n_p = 18$

Taylor Coefficients and Jacobians

Suppose F is d times continuously differentiable on some neighborhood of a point $x_0 \in \mathbb{R}^n$. Then we have

$$y(t) \;=\; F\left(\sum_{i=0}^{d-1} x_i t^i\right) \;=\; \sum_{i=0}^{d-1} y_i t^i + o(t^{d-1})$$

and $y_i = F_i(x_0, \ldots x_i)$ has the Jacobian

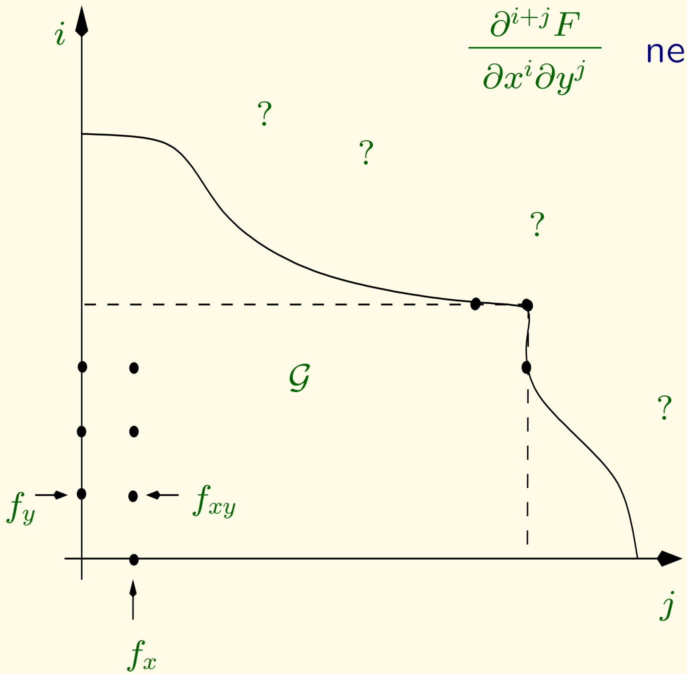
$$\frac{\partial y_j}{\partial x_i} = \frac{\partial F_j}{\partial x_i} \equiv A_{j-i}\,(x_0, \ldots, x_{j-i}) \quad \text{for all} \quad 0 \le i \le j < d\,,$$

with $A_i \equiv F_i'(x_0, \ldots, x_i)$ the ith Taylor coefficient of the Jacobian of F at $x(t)$, i.e.

$$F'(x(t)) = \sum_{i=0}^{d-1} A_i t^i + o(t^{d-1})\;.$$

Cartesian Derivative Structure for $n = 2$

$$\frac{\partial^{i+j} F}{\partial x^i \partial y^j} \quad \text{needed} \iff (i,j) \in \mathcal{G} \supset \mathbb{N} \times \mathbb{N}$$

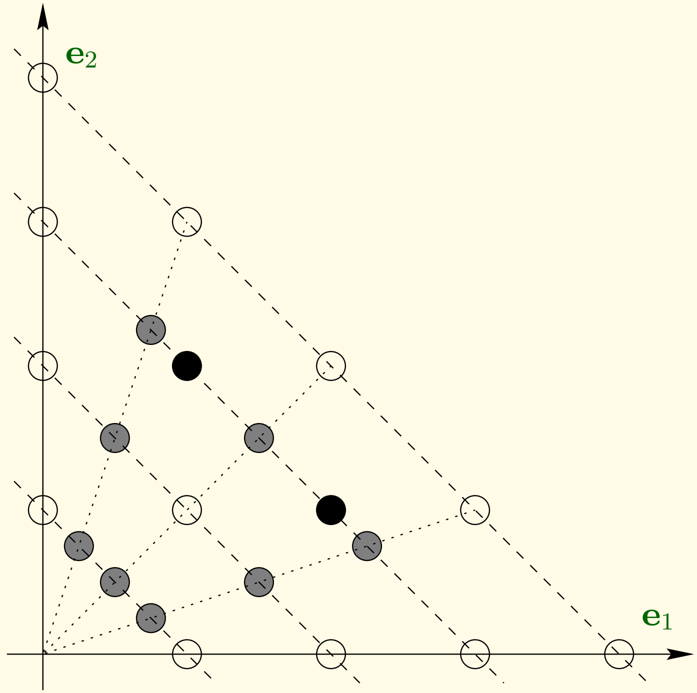


$$0 \leq (\tilde{\imath}, \tilde{\jmath}) \leq (i,j) \in \mathcal{G} \Longrightarrow (\tilde{\imath}, \tilde{\jmath}) \in \mathcal{G}$$

Practical Worries

- Storage $\longleftrightarrow$ Access Pattern
- Number of Arithmetic Operations

Conversion to Family of univariate Taylor Polynomials



Computed and desired values for $p = 2$ and $d = 4$

Proposition:

Taylor to Tensor Conversion

Let $F : \mathbb{R}^n \mapsto \mathbb{R}^m$ be at least d-times continuously differentiable at some point $\mathbf{x} \in \mathbb{R}^n$ and denote by $F_r(x,s)$ the rth Taylor coefficient of the curve $F(x+ts)$ at $t=0$ for some direction $s \in \mathbb{R}^n$. Then we have for any seed matrix $S = [s_j]_{j=1}^p \in \mathbb{R}^{n\times p}$ and any multi-index $\mathbf{i} \in \mathbb{N}^p$ with $|\mathbf{i}| \le d$ the identity

$$\left.\frac{\partial^{|\mathbf{i}|} F(x + z_1 s_1 + z_2 s_2 + \cdots + z_p s_p)}{\partial z_1^{\mathbf{i}_1} \partial z_2^{\mathbf{i}_2} \ldots \partial z_p^{\mathbf{i}_p}}\right|_{z=0} = \sum_{|\mathbf{j}|=d} \gamma_{\mathbf{ij}}\, F_{|\mathbf{i}|}(\mathbf{x}, S\,\mathbf{j}) \ ,$$

where the constant coefficients $\gamma_{\mathbf{ij}}$ are given by the finite sums

$$\gamma_{\mathbf{ij}} \equiv \sum_{\mathbf{0}<\mathbf{k}\le\mathbf{i}} (-1)^{|\mathbf{i}-\mathbf{k}|} \binom{\mathbf{i}}{\mathbf{k}} \binom{d\mathbf{k}/|\mathbf{k}|}{\mathbf{j}} \left(\frac{|\mathbf{k}|}{d}\right)^{|\mathbf{i}|} .$$

Multvariate $\longleftrightarrow$ Uni–univariate

Example

$$F(x, y) = \sum_{i \geq 0 \leq j} c_{ij}\, x^i y^j$$

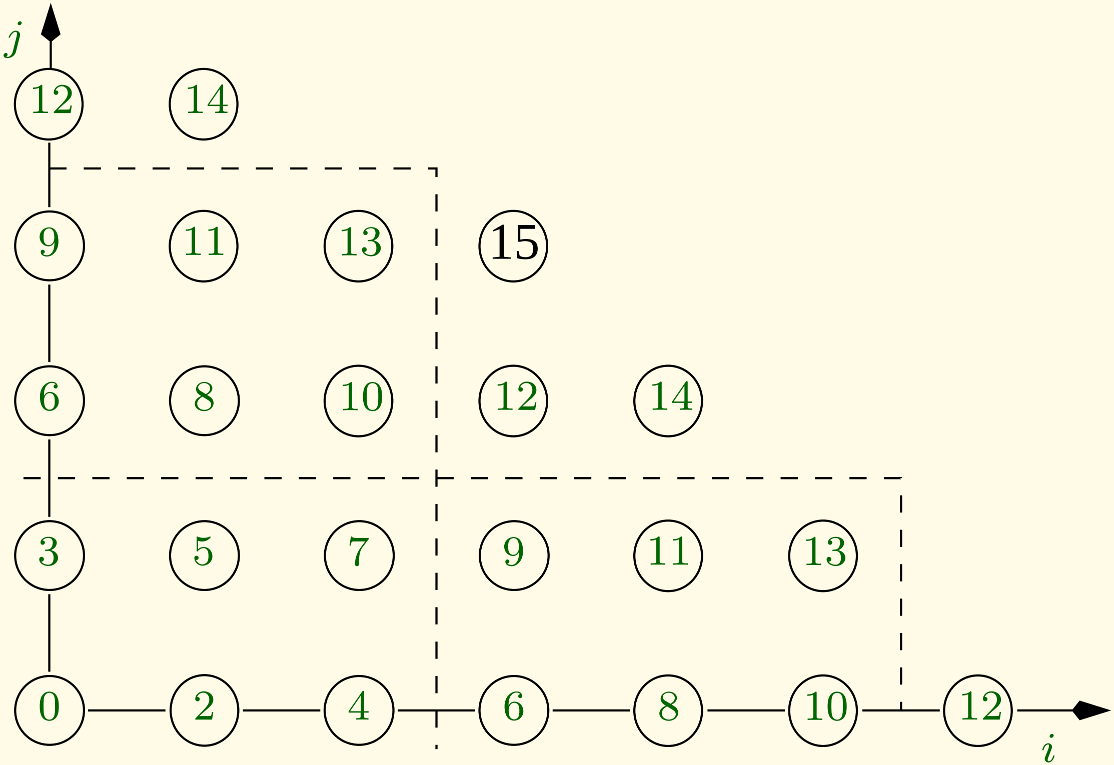
Substitution

$$z(t) = F(t^2, t^3) = \sum_{i \geq 0 \leq j} c_{ij}\, t^{2i+3j}$$

$$= c_{00} + c_{10}t^2 + c_{01}t^3 + c_{20}t^4 + c_{11}\, t^5 + ?\, t^6 + c_{21}\, t^7 + O(t^6)$$

$\Longrightarrow$ First 8 univariate coefficients of $z(t)$

yield 6 multivariate coefficients of $f(x, y)$


j
12
14
9
11
13
15
6
8
10
12
14
3
5
7
9
11
13
0
2
4
6
8
10
12
i

Generally by Chinese Remainder

$d \;\equiv\;$ maximal degree of differentiation
$n \;\equiv\;$ number of independent variables

$$d < p_j \quad \text{for} \quad 1 \le j \le n\,, \qquad gcd(p_i, p_j) = 1$$

$$m \equiv p_1 \cdot p_2 \; \ldots \; p_n \gtrapprox d^n\,, \qquad m_i \equiv m/p_i = \prod_{j \neq i} p_j$$

Univariate polynomial

$$x(t) \equiv (t^{m_1}, t^{m_2}, \ldots, t^{m_n}) : \mathbb{R} \to \mathbb{R}^n$$

with

$$F(x(t)) = \sum_{j=0}^{m} y_j\, t^j + o(t^{nm})$$

identifies all derivatives

$$\frac{\partial^{(i)} F}{\partial_{x_1}^{i_1} \cdots \partial_{x_n}^{i_n}} \quad \text{with} \quad \sum_{k=1}^{n} i_k \le d$$

Memory is increased by roughly $n!$ if $d \gg n$.

Computational Complexity of Multiplication

Truncated multiplication of one pair of n-variate polynomials of degree d involves

$$\binom{2n+d}{n} \quad \approx \quad \frac{d^{2n}}{(2n)!} \quad \text{if} \quad n \ll d$$

real multiplications. Other nonlinear elementaries incur similar costs.
Multiplication of family of pairs of univariate polynomials of maximal degree d costs

$$\binom{n+d-1}{n}\binom{d+2}{2} \quad \approx \quad \frac{d^{n+1}}{2\;n!} \quad \text{if} \quad n \ll d$$

Multiplication of one pair of univariate polynomials of maximal degree $m \approx d^n$

$$0.5d^{2n} \quad \text{or} \quad c\,d^n\,n\log^2(d)$$

when basic or fast convolution methods are used, respectively.

On closer expection one finds that the single univariate approach is numerically equivalent to the multivariate approach (almost).

Real analytic $\varphi : \mathbb{R} \to \mathbb{R}$ has extension

$$\varphi\left(\sum u_i t^i\right) = \sum_{k=1}^{d} \frac{1}{k!}\varphi^{(k)}(u_0)\left(\sum_{i=1}^{d-1} u_i t^i\right)^k$$

Linear ODE on reals

$$\varphi'(u_0) = g(u_0) + m(u_0) \cdot \varphi(u_0)$$

allows recursive evaluation of $v = \varphi(u) = \sum v_i \in \mathcal{R}$

$$v_i = \frac{1}{i}\sum_{j=0}^{i-1} a_j \cdot \tilde{u}_{i-j} \quad \text{with} \quad a_i = g_i + \sum_{j=0}^{i} m_j \cdot v_{i-j}$$

where $\tilde{u}_k \equiv k \cdot u_k$.

Cost ≈ 2 Convolutions

Most Important Cases

Reciprocal:

$$v_i = -v_0 \left(\sum_{j=0}^{i-1} v_j u_{i-j} \right) \quad \text{for} \quad i = 1 \ldots d$$

Logarithm:

$$v_i = \frac{1}{u_0} \left(u_i - \sum_{j=1}^{i-1} \frac{j}{i} v_j \; u_{i-j} \right) \quad \text{for} \quad i = 1 \ldots d \quad .$$

Exponential:

$$v_i = \frac{1}{i} \sum_{j=0}^{i-1} v_j \, \tilde{u}_{i-j} \quad \text{with} \quad \tilde{u}_j = j \, u_j \quad \text{for} \quad i = 1 \ldots d$$

Power:

$$v_i = \frac{1}{i \, u_0} \sum_{j=0}^{i-1} v_j \, u_{i-j} \, [c - j(c+1)/i] \quad \text{for} \quad i = 1 \ldots d$$

Numerical Example

$$f = \prod_{i=1}^{n} \sin(x_i)$$

For $n = 3$ and $d = 3$, we choose $(p_1, p_2, p_3) = (4, 5, 7)$ and so that $m = 140 = 4 * 5 * 7$ and get as result for $\mathbf{x} = (1.3, 1.6, 1.9)$

index	location	entry
(0,0,3)	60	0.311375
(0,1,2)	68	0.0266246
(0,2,1)	76	0.311375
(0,3,0)	84	0.0266246
(1,0,2)	75	-0.253026
(1,1,1)	83	0.00252516
(1,2,0)	91	-0.253026
(2,0,1)	90	0.311375
(2,1,0)	98	0.0266246
(3,0,0)	105	-0.253026

Maximum relative error vs the directly computed entries 4.38778e-16.

Alternatively we propagate 10 uivariate directions of degree 3 and get the same result this time with maximum relative error of 1.83766e-14.

Fast Evaluations á la Brent et al.

Reciprocal:

$v = 1/u \equiv \mathrm{rec}(u)$ solves $f(v) \equiv 1/v - u = 0$ yielding by Newton

$$N(v) = v + (1/v - u) * v * v = v + (1 - u * v) * v$$

Fast Evaluations á la Brent et al.

Reciprocal:

$v = 1/u \equiv \mathrm{rec}(u)$ solves $f(v) \equiv 1/v - u = 0$ yielding by Newton

$$N(v) = v + (1/v - u) * v * v = v + (1 - u * v) * v$$

Logarithm:

$v = \log(u)$ satisfies $\dot{v} = \dot{u} * (1/u)$ where $\dot{w}_j = (j + 1)w_{j+1}$ for $w = u, v$

Fast Evaluations á la Brent et al.

Reciprocal:

$v = 1/u \equiv \mathrm{rec}(u)$ solves $f(v) \equiv 1/v - u = 0$ yielding by Newton

$$N(v) = v + (1/v - u) * v * v = v + (1 - u * v) * v$$

Logarithm:

$v = \log(u)$ satisfies $\dot{v} = \dot{u} * (1/u)$ where $\dot{w}_j = (j+1)w_{j+1}$ for $w = u, v$

Exponential:

$v = \exp(u)$ solves $f(u) \equiv \log(v) - u = 0$ yielding by Newton

$$N(v) = v * (1 + u - \log v)$$

Fast Evaluations á la Brent et al.

Reciprocal:

$v = 1/u \equiv \mathrm{rec}(u)$ solves $f(v) \equiv 1/v - u = 0$ yielding by Newton

$$N(v) = v + (1/v - u) * v * v = v + (1 - u * v) * v$$

Logarithm:

$v = \log(u)$ satisfies $\dot{v} = \dot{u} * (1/u)$ where $\dot{w}_j = (j+1) w_{j+1}$ for $w = u, v$

Exponential:

$v = \exp(u)$ solves $f(u) \equiv \log(v) - u = 0$ yielding by Newton

$$N(v) = v * (1 + u - \log v)$$

$\Longrightarrow$ Quadratic convergence = doubling of correct terms

Complexity $\sim$ Cost (Workhorse Convolution)

Mean Value Form in Taylor Calculus

$$\varphi(\check{u} + \Delta u) \approx \varphi(\check{u}) + \varphi'(\check{u}) * \Delta u$$

where

$$\mathcal{R} \ni \varphi'(\check{u}) \quad \cong \quad \text{Töplitz matrix} \quad \in \mathbb{R}^{d\times d}$$

corresponds to pre-accumulation of local Jacobian.

Mean value from

$$v = \varphi(u) \subset \varphi(\check{u}) + \varphi'(u) * (u - \check{u})$$

with interval radius

$$\|\varphi'(\check{u})\|_1 \, \rho(u) \approx \rho(v) \leq \|\varphi'(u)\|_1 \, \rho(u) \quad .$$

Codependence and Postiteration

While naive convolution is single usage and thus optimal

Reciprocal:

$$v = 1/u \quad \text{implies} \quad v_2 = -v_0(v_0 u_2 + v_1 u_1) = (u_1^2 - (v_0 u_2)/u_0^3$$

Codependence and Postiteration

While naive convolution is single usage and thus optimal

Reciprocal:

$$v = 1/u \quad \text{implies} \quad v_2 = -v_0(v_0 u_2 + v_1 u_1) = (u_1^2 - (v_0 u_2)/u_0^3$$

Meanvalue:

$$M(v) = 1/\check{u} - v * v * (u - \check{u})$$

Codependence and Postiteration

While naive convolution is single usage and thus optimal

Reciprocal:

$$v = 1/u \quad \text{implies} \quad v_2 = -v_0(v_0 u_2 + v_1 u_1) = (u_1^2 - (v_0 u_2)/u_0^3$$

Meanvalue:

$$M(v) = 1/\check{u} - v * v * (u - \check{u})$$

Newton:

$$N(v) = \check{v} + (1 - u * \check{v}) * \check{v} + 2(1 - \check{u} * v)(v - \check{v}) \quad .$$

Codependence and Postiteration

While naive convolution is single usage and thus optimal

Reciprocal:

$$v = 1/u \quad \text{implies} \quad v_2 = -v_0(v_0 u_2 + v_1 u_1) = (u_1^2 - (v_0 u_2)/u_0^3$$

Meanvalue:

$$M(v) = 1/\check{u} - v * v * (u - \check{u})$$

Newton:

$$N(v) = \check{v} + (1 - u * \check{v}) * \check{v} + 2(1 - \check{u} * v)(v - \check{v}) \quad .$$

Initialization

$I(u)$ based on classical recurrence.

General Procedure

$$v := I(u); \quad v := v \cap M(v)$$
$$v := v \cap N(v) \text{ until } v \subseteq N(v)$$

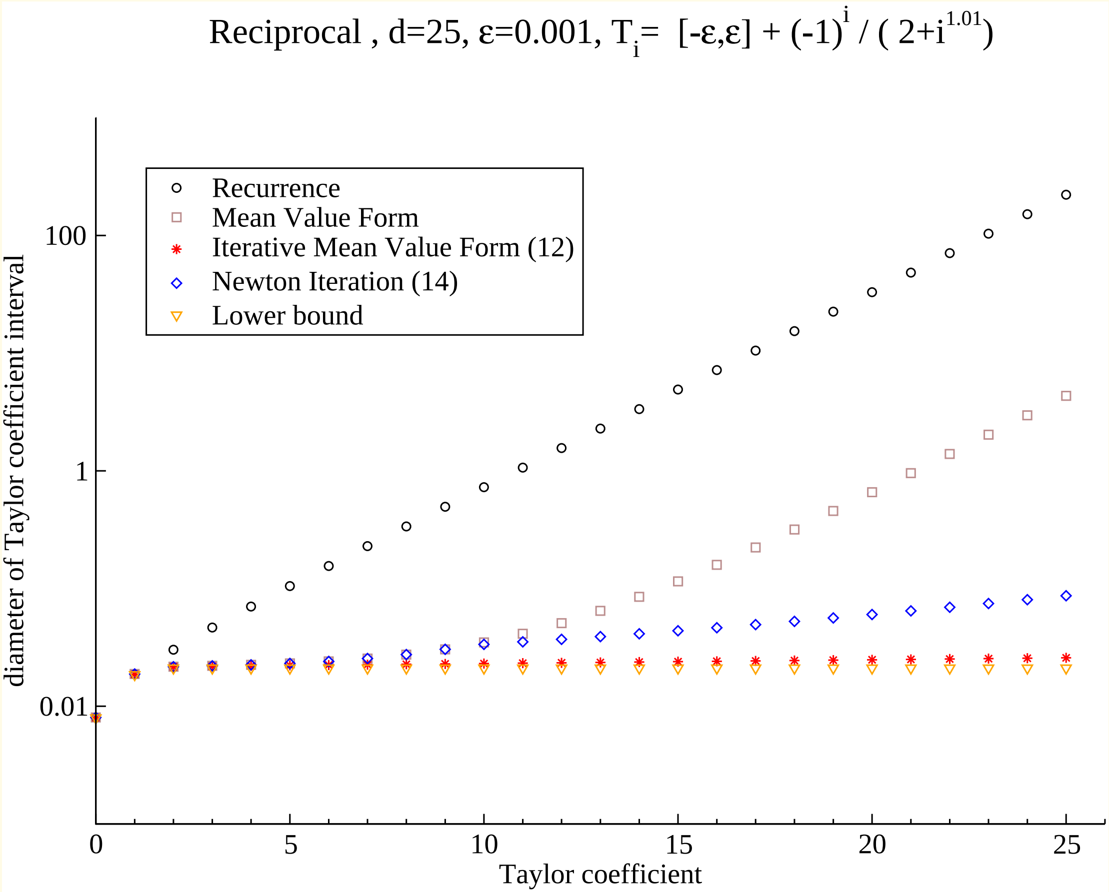
Other cases:

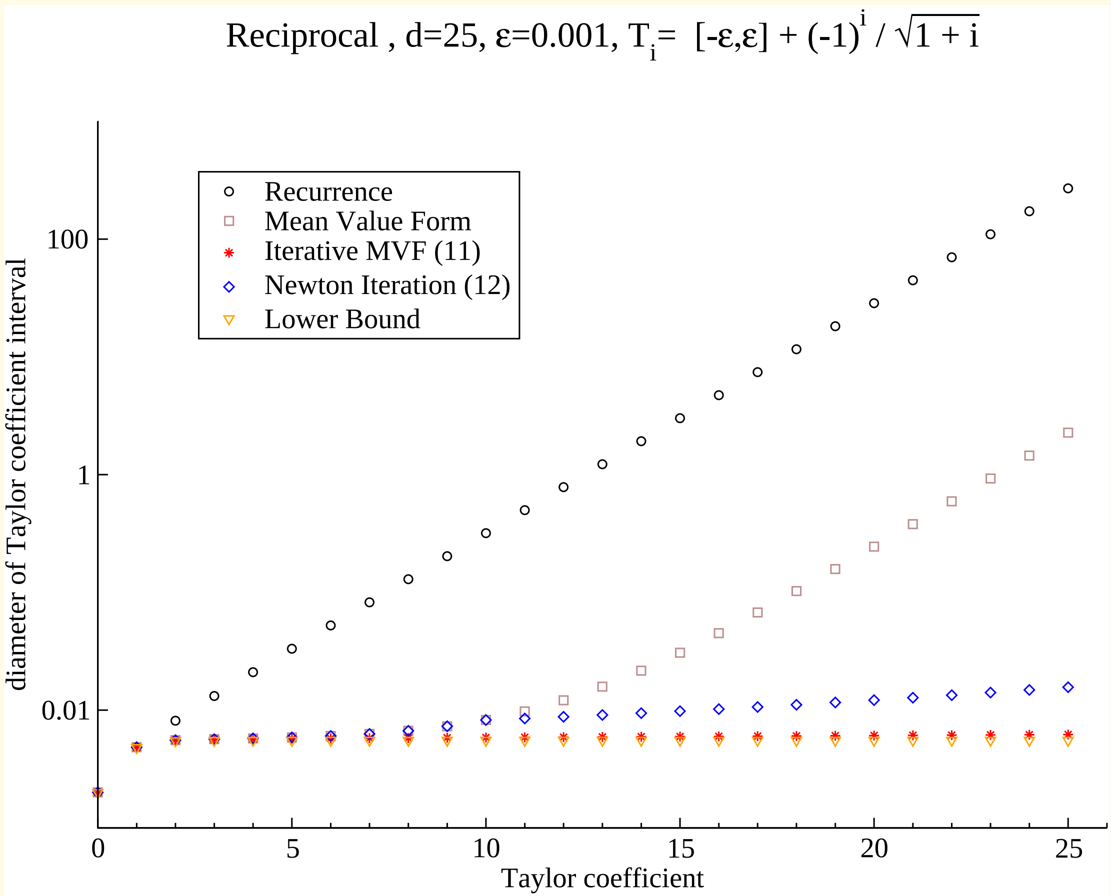
$v = \log(u) \quad \Longrightarrow$

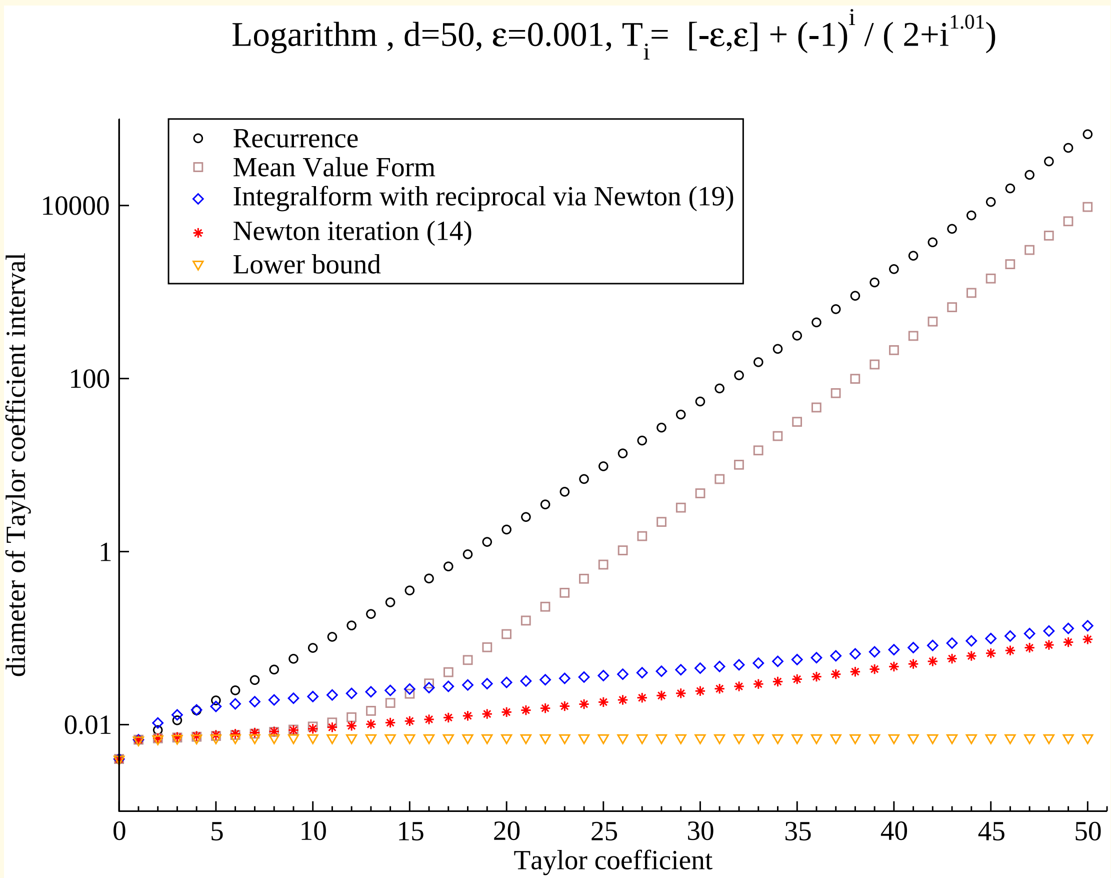
$$I(u) = \text{ lifted } \dot{v} = \mathrm{rec}(u)$$
$$M(v) = \log(\check{u}) + \mathrm{rec}(u) * (u - \check{u})$$
$$N(v) = \check{v} + (u * \exp(-\check{v}) - 1) + (1 - \check{u} * \exp(-v)) * (u - \check{u})$$

$v = \exp(u) \quad \Longrightarrow$

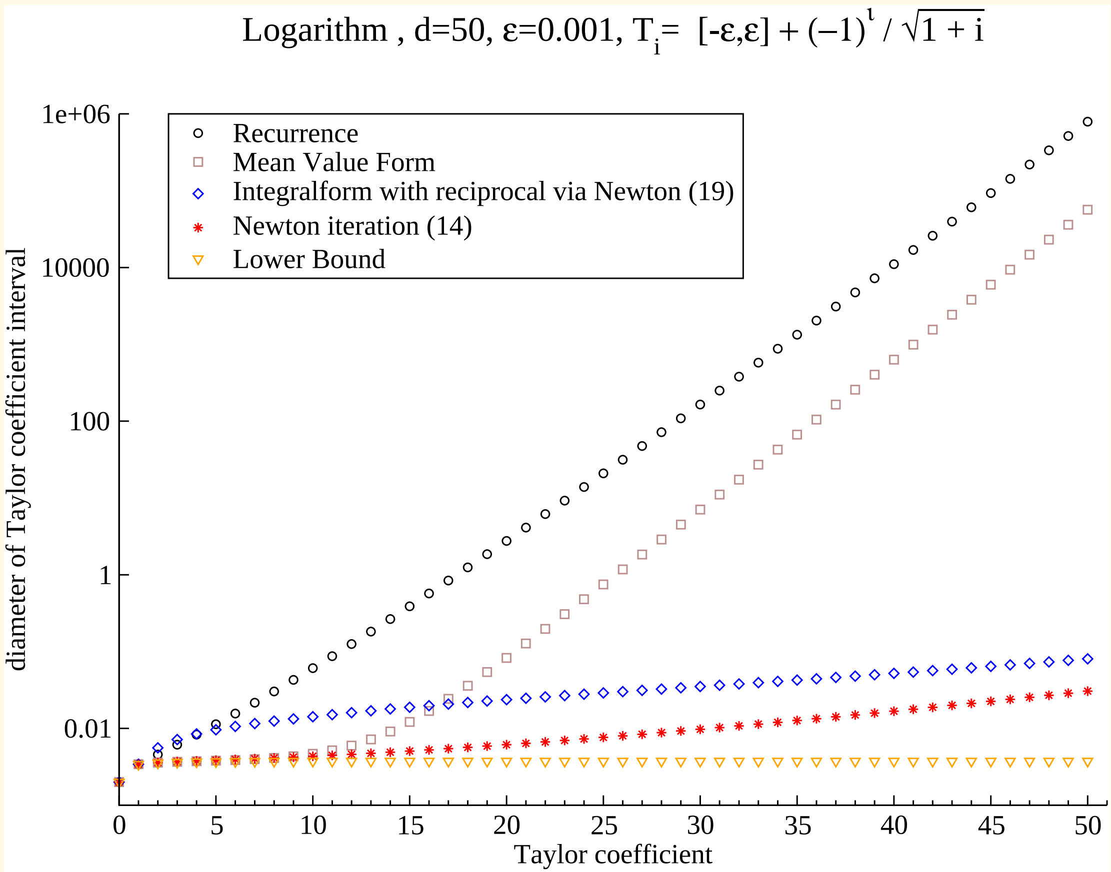
$$I(u) = \text{ classical recurrence}$$
$$M(v) = \exp(\check{u}) + v * (u - \check{u})$$
$$N(v) = \check{v} + (u - \log(\check{v})) * \check{v} + (\check{u} - \log(v)) * (u - \check{u})$$


Reciprocal , d=25, ε=0.001, T_i= [-ε,ε] + $(-1)^i$ / ($2+i^{1.01}$)
Recurrence
Mean Value Form
Iterative Mean Value Form (12)
Newton Iteration (14)
Lower bound
100
1
0.01
diameter of Taylor coefficient interval
0
5
10
15
20
25
Taylor coefficient

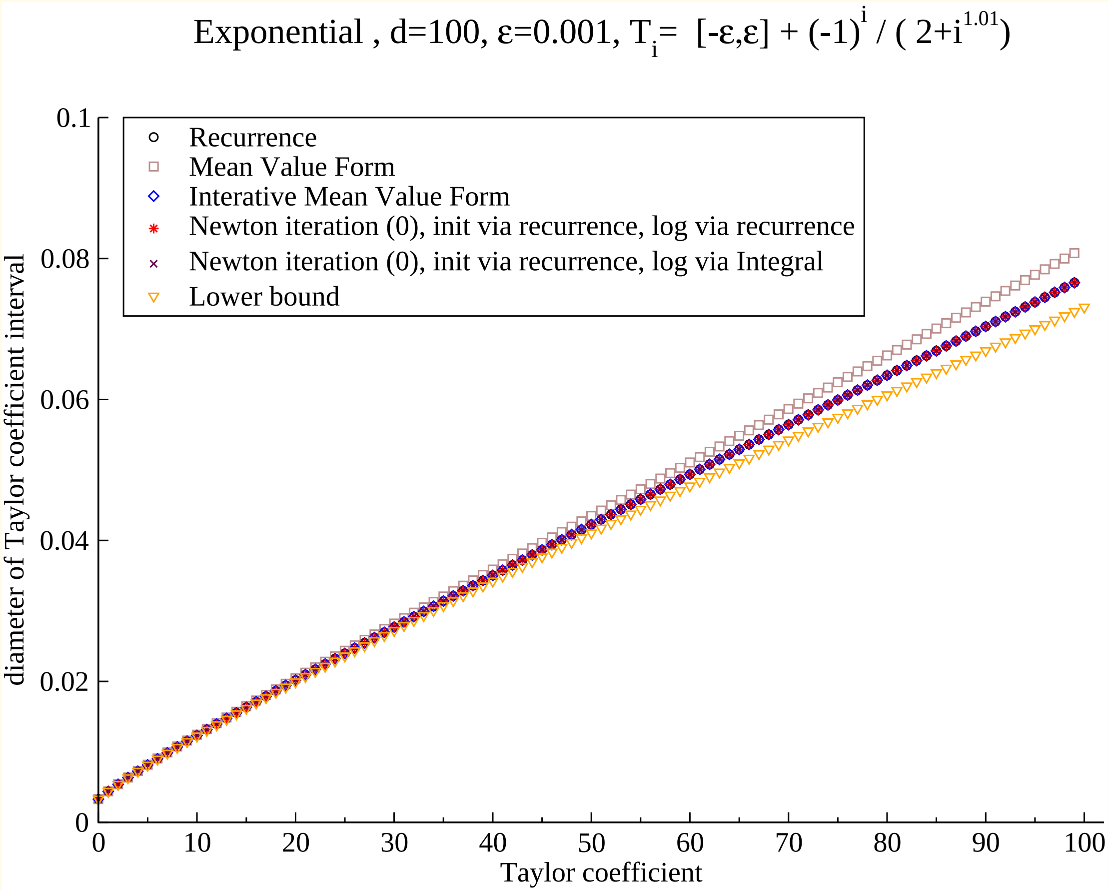

Reciprocal , d=25, ε=0.001, Ti= [-ε,ε] + (-1)i / √1 + i
Recurrence
Mean Value Form
Iterative MVF (11)
Newton Iteration (12)
Lower Bound
diameter of Taylor coefficient interval
100
1
0.01
0
5
10
15
20
25
Taylor coefficient

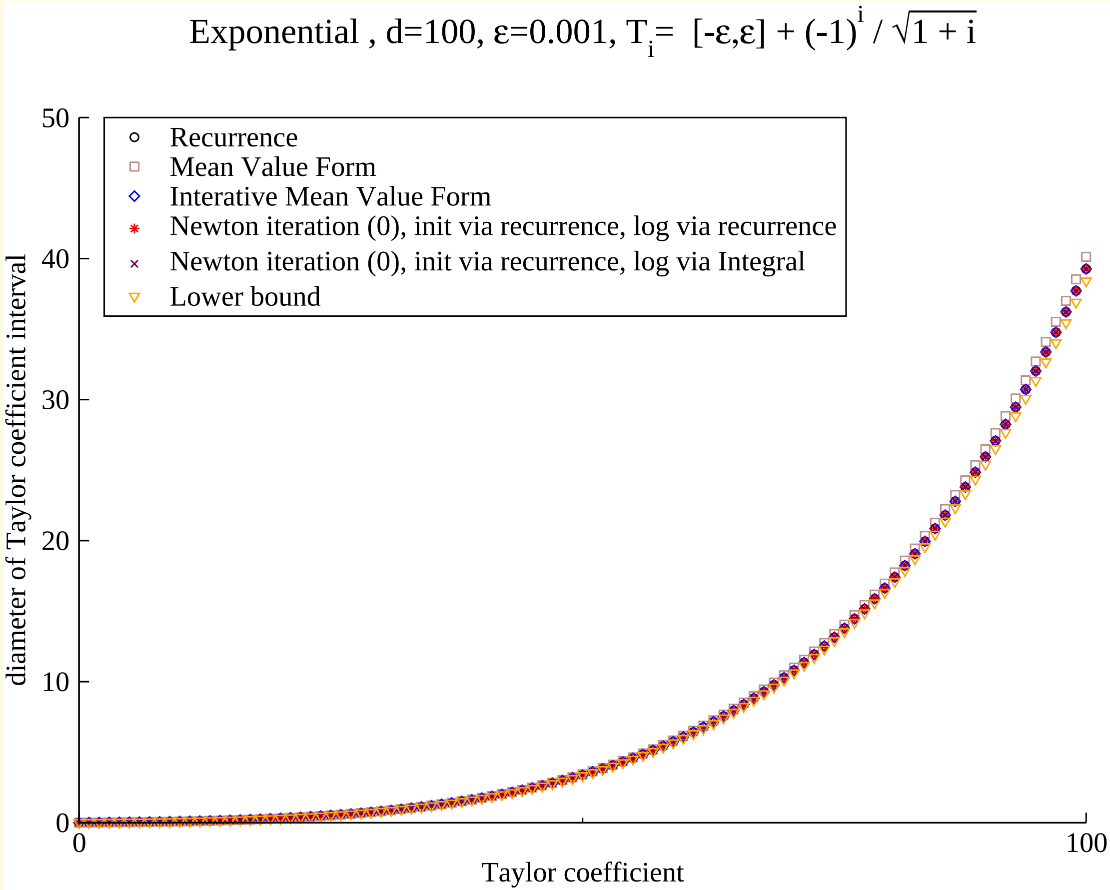


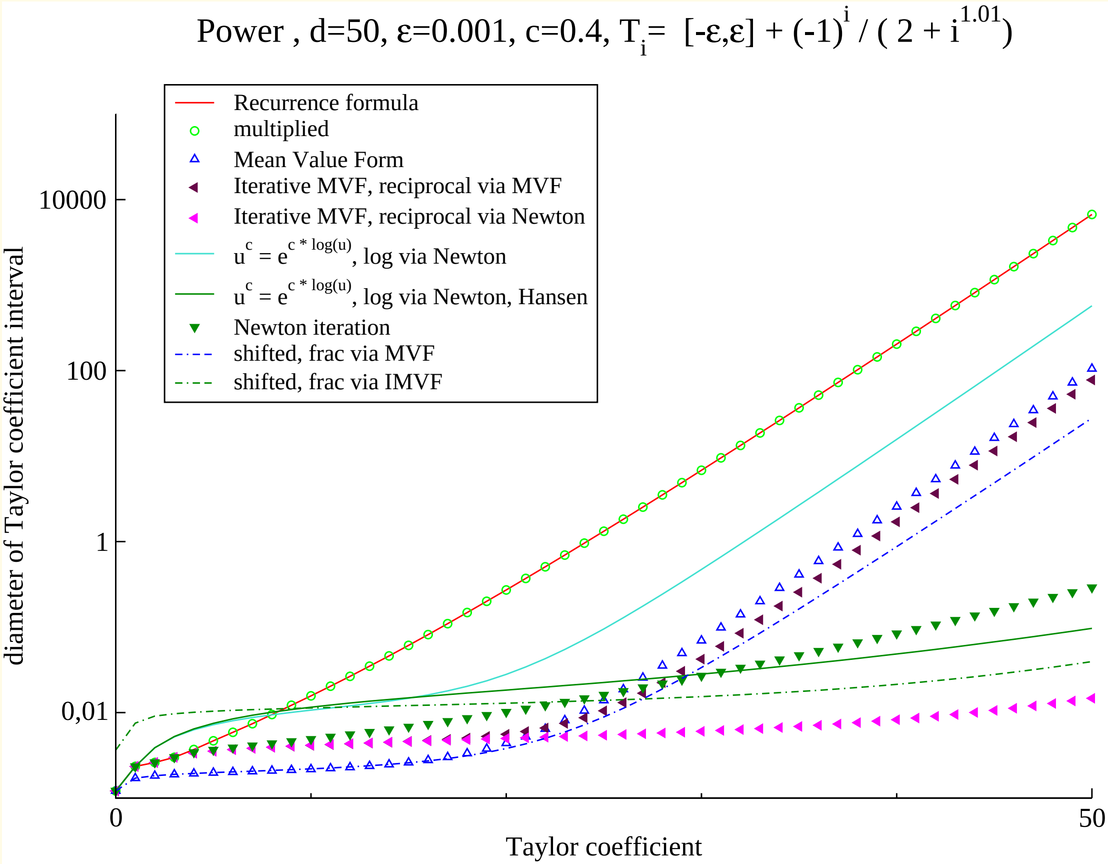
Logarithm , d=50, ε=0.001, T_i= [-ε,ε] + (-1)^i / (2+i^1.01)
Recurrence
Mean Value Form
Integralform with reciprocal via Newton (19)
Newton iteration (14)
Lower bound
diameter of Taylor coefficient interval
10000
100
1
0.01
0
5
10
15
20
25
30
35
40
45
50
Taylor coefficient



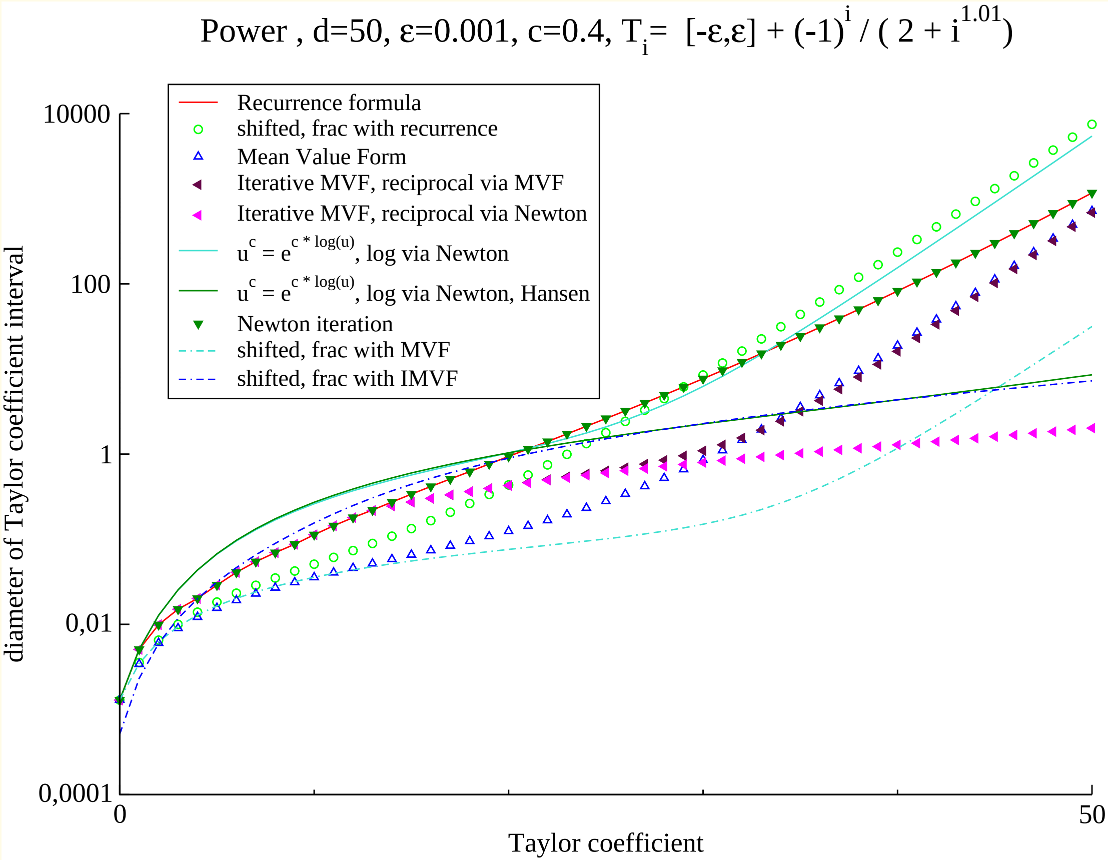
Logarithm , d=50, ε=0.001, T_i= [-ε,ε] + (–1)^i / √1 + i
Recurrence
Mean Value Form
Integralform with reciprocal via Newton (19)
Newton iteration (14)
Lower Bound
1e+06
10000
100
1
0.01
0
5
10
15
20
25
30
35
40
45
50
diameter of Taylor coefficient interval
Taylor coefficient


Exponential , d=100, ε=0.001, T_i= [-ε,ε] + $(-1)^i$ / ($2+i^{1.01}$)
Recurrence
Mean Value Form
Interative Mean Value Form
Newton iteration (0), init via recurrence, log via recurrence
Newton iteration (0), init via recurrence, log via Integral
Lower bound
diameter of Taylor coefficient interval
Taylor coefficient
0
0.02
0.04
0.06
0.08
0.1
0
10
20
30
40
50
60
70
80
90
100


Exponential , d=100, ε=0.001, T_i= [-ε,ε] + (-1)^i / √(1 + i)
50
40
30
20
10
0
0
100
Recurrence
Mean Value Form
Interative Mean Value Form
Newton iteration (0), init via recurrence, log via recurrence
Newton iteration (0), init via recurrence, log via Integral
Lower bound
diameter of Taylor coefficient interval
Taylor coefficient



Power , d=50, ε=0.001, c=0.4, T_i= [-ε,ε] + (-1)^i / (2 + i^1.01)
Recurrence formula
multiplied
Mean Value Form
Iterative MVF, reciprocal via MVF
Iterative MVF, reciprocal via Newton
u^c = e^(c * log(u)), log via Newton
u^c = e^(c * log(u)), log via Newton, Hansen
Newton iteration
shifted, frac via MVF
shifted, frac via IMVF
10000
100
1
0,01
0
50
diameter of Taylor coefficient interval
Taylor coefficient


Power , d=50, ε=0.001, c=0.4, T_i= [-ε,ε] + (-1)^i / (2 + i^1.01)
Recurrence formula
shifted, frac with recurrence
Mean Value Form
Iterative MVF, reciprocal via MVF
Iterative MVF, reciprocal via Newton
u^c = e^(c * log(u)), log via Newton
u^c = e^(c * log(u)), log via Newton, Hansen
Newton iteration
shifted, frac with MVF
shifted, frac with IMVF
10000
100
1
0,01
0,0001
0
50
diameter of Taylor coefficient interval
Taylor coefficient

Concluding Remarks and Questions

- Conversion: multivariate $\longrightarrow$ univariate family efficient.
- Conversion: multivariate $\longrightarrow$ single univariate dubious.
- Conflict between fast and thight interval evaluation ????.
- Interval result can/must be tightened by postiterations.

Concluding Remarks and Questions

- Conversion: multivariate $\longrightarrow$ univariate family efficient.
- Conversion: multivariate $\longrightarrow$ single univariate dubious.
- Conflict between fast and thight interval evaluation ????.
- Interval result can/must be tightened by postiterations.

Concluding Remarks and Questions

- Conversion: multivariate $\longrightarrow$ univariate family efficient.
- Conversion: multivariate $\longrightarrow$ single univariate dubious.
- Conflict between fast and thight interval evaluation ????.
- Interval result can/must be tightened by postiterations.

Concluding Remarks and Questions

- Conversion: multivariate $\longrightarrow$ univariate family efficient.
- Conversion: multivariate $\longrightarrow$ single univariate dubious.
- Conflict between fast and thight interval evaluation ????.
- Interval result can/must be tightened by postiterations.