Experiments with Range Computations using Extrapolation

P. S. V. Nataraj
And
Shanta Sondur

Systems & Control Engineering,
IIT Bombay.

Overview

- Introduction
- NIE and Uniform Subdivision
- Objective
- Extrapolation Process – Sequence Transformation
- The Richardson Extrapolation Process (REP)
- Other Extrapolation Algorithms
- Error control in Convergence Acceleration Processes
- Brezinski's Theorem
- Proposed Algorithm
- Illustrative Example – Vehicle Clutch System
- Comparison of proposed method with the existing method and with Globsol
- Conclusions

Introduction

- We present a new method 'Extrapolated NIE' to compute the range enclosure of the functions using the NIE with uniform subdivision and Extrapolation Algorithms such as, Richardson Extrapolation Algorithm (REP).

- Aim is to 'Accelerate the Order of Convergence of the NIE' by extrapolation.

NIE and Uniform Subdivision

- Consider the interval vector $\mathrm{x} = (\mathrm{x}_1, \ldots, \mathrm{x}_l)^T$

  With components $\mathrm{x}_j = [\underline{x}_j, \overline{x}_j]$.
- Suppose we uniformly subdivide the interval x using the subdivision factor N, as follows,

$$\mathrm{x}_{i,j} = [\underline{x}_i + (j-1)\,\mathrm{wid}\,\mathrm{x}_i / N,\ \underline{x}_i + j\,\mathrm{wid}\,\mathrm{x}_i / N],\ j = 1,2,\ldots,N$$

$$\mathrm{x}_i = \bigcup_{j=1}^{N} \mathrm{x}_{i,j}$$

$$\mathrm{x} = \bigcup_{j_l=1}^{N} (\mathrm{x}_{1,j_1}, \mathrm{x}_{2,j_2}, \ldots, \mathrm{x}_{l,j_l})$$

NIE continued…

- Let $e_{(N)}(\mathrm{x})$ be the error interval associated with N partitions of the interval vector x expressed as

$$e_{(N)}(\mathrm{x}) = \bigcup_{j_l=1}^{N} e\left(\mathrm{x}_{1,j_1}, \mathrm{x}_{2,j_2}, \ldots, \mathrm{x}_{l,j_l}\right)$$

Define $f_{(N)}(\mathrm{x})$ as

$$f_{(N)}(\mathrm{x}) = \bigcup_{j_l=1}^{N} f\left(\mathrm{x}_{1,j_1}, \mathrm{x}_{2,j_2}, \ldots, \mathrm{x}_{l,j_l}\right) = f^{\,range}(\mathrm{x}) + e_{(N)}(\mathrm{x})$$

The excess width is defined as

$$\mathrm{wid}\,\boldsymbol{e}_{(N)}(\mathrm{x}) \le \frac{\sigma}{\boldsymbol{N}}\,\mathrm{wid}\,\mathrm{x}$$

NIE continued…

- The excess width also can be expressed as

$$\operatorname{wid} e_{(N)}(\mathrm{x}) \le \frac{\sigma}{N} \operatorname{wid} \mathrm{x} + O\left(\operatorname{wid} \mathrm{x}^{2}\right)$$

- The range infimum and range supremum can be expressed as

$$\underline{f_{(N)}(\mathrm{x})} = \underline{f^{range}(\mathrm{x})} + \frac{\sigma}{N} \operatorname{wid} \mathrm{x} + O\left(\mathrm{x}^{2}\right)$$

$$\overline{f_{(N)}(\mathrm{x})} = \overline{f^{range}(\mathrm{x})} + \frac{\sigma}{N} \operatorname{wid} \mathrm{x} + O\left(\mathrm{x}^{2}\right)$$

Order of Convergence of NIE with Uniform Subdivision Process

- NIE with uniform subdivision process has linear order of convergence.

'The excess width goes down linearly with the domain width'.

Objective

Acceleration

of

Linear Convergence Process

of

NIE with Uniform Subdivision.

Extrapolation Process – Sequence Transformation

- Extrapolation Algorithms are popularly used for '*accelerating* ' the convergence of a given sequence.

- Let (S_n) be a sequence of (real or complex) numbers converging to **S**. Extrapolation method *transforms* the sequence (S_n) into another sequence (T_n) which converges *to the same limit **S**, faster than* (S_n).

Richardson Extrapolation Algorithm (REP)

- Let K∈N, ρ ≥ 2, and $\{S_n\}$ be the sequence to be accelerated. The REP can be given as

$$T_0^{(j)} = S_j,\ j = 0,1,\ldots,\ K$$

$$T_k^{(j)} = T_{k-1}^{(j)} + \frac{\left(T_{k-1}^{(j)} - T_{k-1}^{(j-1)}\right)}{\left(\rho^k - 1\right)}, \begin{cases} k = 1,2,\ldots,K, \\ j = k,\ldots,K. \end{cases}$$

Romberg Table by REP

i	$k=0$	$k=1$	$k=2\ldots$
0	$A_0^{(0)}$		
		$A_1^{(0)} = \dfrac{2\,A_0^{(1)} - A_0^{(0)}}{(2-1)}$	
1	$A_0^{(1)}$		$A_2^{(0)} = \dfrac{2^2\,A_1^{(1)} - A_1^{(0)}}{(2^2-1)}$
		$A_1^{(1)} = \dfrac{2\,A_0^{(2)} - A_0^{(1)}}{(2-1)}$	
2	$A_0^{(2)}$		$A_2^{(1)} = \dfrac{2^2\,A_1^{(2)} - A_1^{(1)}}{(2^2-1)}$
		$A_1^{(2)} = \dfrac{2\,A_0^{(3)} - A_0^{(2)}}{(2-1)}$	
3	$A_0^{(3)}$		
	$O(h)$	$O(h^2)$	$O(h^3)\ldots$

Other Extrapolation Algorithms

1. Aitkin's Δ^2 iterated process
2. Rational Extrapolation
3. The ε - algorithm
4. The E – algorithm
5. The G - transformation
6. Levin's transform
7. Overholt's process

Error control in Convergence Acceleration Processes

- From user's point of view it <u>is not sufficient</u> to know that, for a given sequence (S_n), the extrapolated sequence (T_n) will converge faster.

- One should have an estimate of the error $(T_n - S)$ or, still better, to know a sequence of intervals containing the unknown limit S of the sequence (S_n).

- Brezinski's method can be used for estimating the error on the transformed sequences (T_n) obtained through Extrapolation process.

Brezinski's Interval

- Let (T_n) and (V_n) be the transformed sequences of (S_n) converging to the same limit S, and let $V_n(b)$ and $J_n(b)$ be defined as

$$V_n(b) = V_n - b(V_n - T_n), \; n \in \mathrm{N}, \mathrm{b} \in \Re,$$

$$J_n(b) = [\min(V_n(b), V_n(-b)), \max(V_n(b), V_n(-b))].$$

Brezinski's Theorem

If the sequence (T_n) converges to S faster than the original sequence (S_n) with limit S and if the sequence (V_n) converges to S faster than the (T_n), *then for all* $b \neq 0$, $\exists N: \forall n \geq N$, *the true value* S *will be contained in an interval* $J_n(b)$.

Moreover, the sequence $V_n(\pm b)$ will converge faster than the (S_n).

Proposed Algorithm

1. Generate the two separate sequences of Lower and Upper bounds on the range infium and range supremum, respectively.

2. Apply REP and generate Romberg Tables for range infimum and supremum.

3. Construct Brezinski's Tables of intervals approximating the range infimum (BTII) and supremum (BTIS).

4. Construct the final table of intervals approximating the range from BTII and BTIS.

Procedure to construct the sequences

Do j = 1, 2,…,K

1. Subdivide the box x uniformly into $N_j = 2^j$ subboxes. Let,

$$\mathrm{x} = \bigcup_{j_l=1}^{N} \left(\mathrm{x}_{1,j_1}, \mathrm{x}_{2,j_2}, \ldots, \mathrm{x}_{l,j_l}\right)$$

2. Compute the range enclosure $f_{(N_j)}(\mathrm{x})$

$$f_{(N_j)}(\mathrm{x}) = \left[\underline{f_{(N_j)}(\mathrm{x})}, \overline{f_{(N_j)}(\mathrm{x})}\right] = \bigcup_{j_l=1}^{N} f\left(\mathrm{x}_{1,j_1}, \mathrm{x}_{2,j_2}, \ldots, \mathrm{x}_{l,j_l}\right)$$

3. Set

$$A_0^{(j)} \leftarrow \underline{f(N_j)(\mathrm{x})}, \quad B_0^{(j)} \leftarrow \overline{f(N_j)(\mathrm{x})}$$

End do

Romberg Table for Infimum

$k=0$	$k=1$	$k=2\ldots..$
$A_0^{(0)} = inf\,(F_2(X))$		
	$A_1^{(0)} = A_0^{(1)} + \dfrac{A_0^{(1)} - A_0^{(0)}}{(2^{k+1}-1)}$	
$A_0^{(1)} = inf\,(F_4(X))$		$A_2^{(0)} = A_1^{(1)} + \dfrac{A_1^{(1)} - A_1^{(0)}}{(2^{k+1}-1)}$
	$A_1^{(1)} = A_0^{(2)} + \dfrac{A_0^{(2)} - A_0^{(1)}}{(2^{k+1}-1)}$	
$A_0^{(2)} = inf\,(F_8(X))$		$A_2^{(1)} = A_1^{(2)} + \dfrac{A_1^{(2)} - A_1^{(1)}}{(2^{k+1}-1)}$
	$A_1^{(2)} = A_0^{(3)} + \dfrac{A_0^{(3)} - A_0^{(2)}}{(2^{k+1}-1)}$	
$A_0^{(3)} = inf\,(F_{16}(X))$		$A_2^{(2)} = A_1^{(3)} + \dfrac{A_1^{(3)} - A_1^{(2)}}{(2^{k+1}-1)}$
	$A_1^{(3)} = A_0^{(4)} + \dfrac{A_0^{(4)} - A_0^{(3)}}{(2^{k+1}-1)}$	
$A_0^{(4)} = inf\,(F_{32}(X))$		

Romberg Table for Supremum

$k=0$	$k=1$	$k=2\ldots..$

$$A_0^{(0)} = sup\,(F_2(X))$$

$$A_1^{(0)} = A_0^{(1)} + \frac{A_0^{(1)} - A_0^{(0)}}{(2^{k+1}-1)}$$

$$A_0^{(1)} = sup\,(F_4(X))$$

$$A_2^{(0)} = A_1^{(1)} + \frac{A_1^{(1)} - A_1^{(0)}}{(2^{k+1}-1)}$$

$$A_1^{(1)} = A_0^{(2)} + \frac{A_0^{(2)} - A_0^{(1)}}{(2^{k+1}-1)}$$

$$A_0^{(2)} = sup\,(F_8(X))$$

$$A_2^{(1)} = A_1^{(2)} + \frac{A_1^{(2)} - A_1^{(1)}}{(2^{k+1}-1)}$$

$$A_1^{(2)} = A_0^{(3)} + \frac{A_0^{(3)} - A_0^{(2)}}{(2^{k+1}-1)}$$

$$A_0^{(3)} = sup\,(F_{16}(X))$$

$$A_2^{(2)} = A_1^{(3)} + \frac{A_1^{(3)} - A_1^{(2)}}{(2^{k+1}-1)}$$

$$A_1^{(3)} = A_0^{(4)} + \frac{A_0^{(4)} - A_0^{(3)}}{(2^{k+1}-1)}$$

$$A_0^{(4)} = sup\,(F_{32}(X))$$

Procedure to construct the Brezenski's Interval

1. Let $S_n = A_k^{(i)}$, $T_n = A_{k+1}^{(i)}$, and $V_n = A_{k+2}^{(i)}$, $k=1,2,\ldots,K$ and $i = k,\ldots,K$, and $b = 1$.

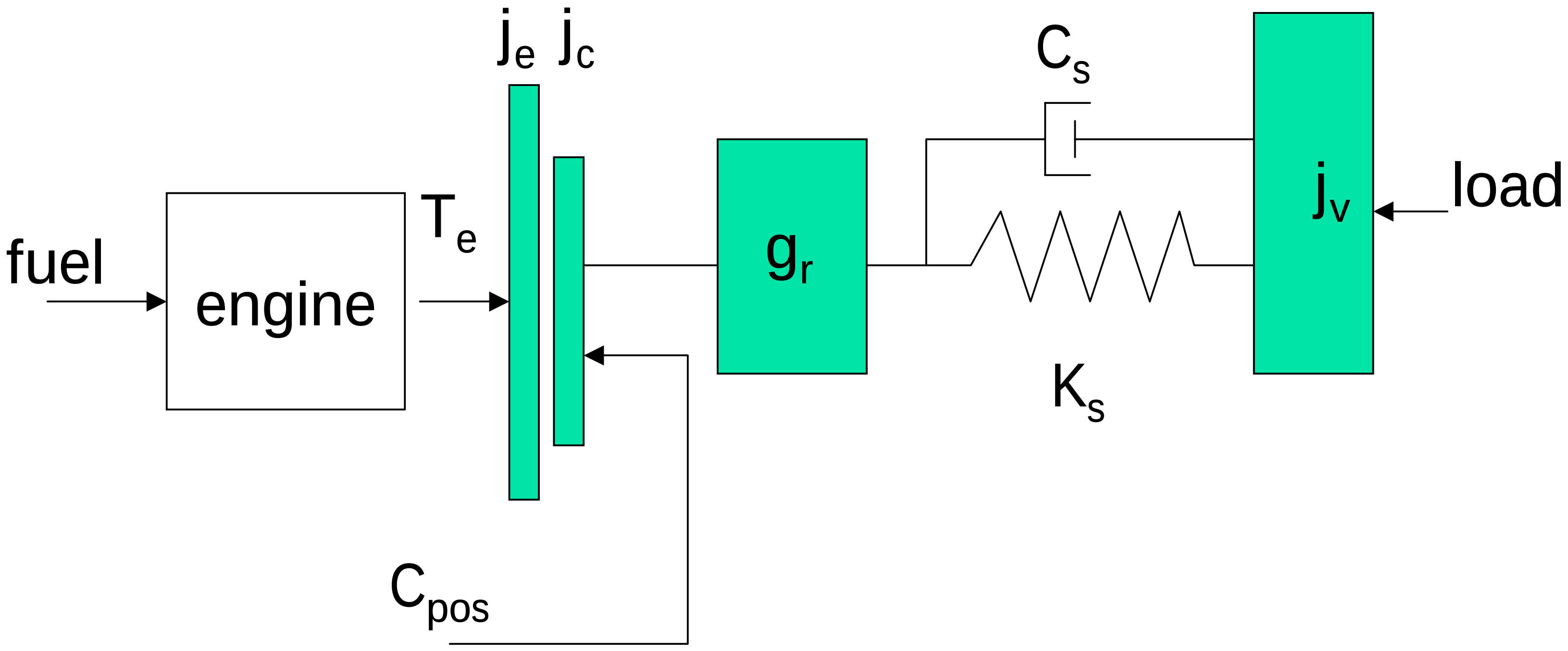
2. Construct Brezenski's Interval $J_n(b)$ for the entries in the Romberg Table for Infimum and Supremum.

3. Call the resulting table as Brezenski's Table of Intervals for Infimum (BTII) and Brezenski's Table of Intervals for Supremum (BTIS).

4. Construct the range enclosure by

   $F(X) = [\inf(BTII), \sup(BTIS)]$.

Illustrative Example
Vehicle Clutch System



Vehicle Driveline Model

Parameters of Vehicle

Parameter	Symbols	Range of Values
Clutch inertia	j_c	0.09 kg m^2
Engine inertia	j_e	3.07 kg m^2
Vehicle inertia	j_v	[1400, 11000] kg m^2
Axle shaft spring rate	k_s	[58000, 115000] N m/rad
Axle and tyre damping	C_c	377 N m/rad/s
Engine viscous friction	C_e	18.5 N m/rad/s
Overall gear ratio	g_r	27.0
$\frac{\Delta \text{ Clutch torque}}{\Delta C_{pos}}$	K_c	[100, 800]N m/mm

Transfer function

- Three uncertain physical parameters are:
  - vehicle inertia j_v,
  - axel shaft spring rate, k_s, and
  - the ratio between change in clutch torque and change in clutch position, K_c.

The transfer function between the clutch position C_{pos} and the transmission input speed S_i is given as:

$$G(s) = \frac{65.61K_c\left(j_v s^2 + 377s + K_s\right)}{3.07s\left(65.61j_v s^2 + 377(65.61 + j_v)s + K_s(65.61 + j_v)\right)}$$

Magnitude Function

$$f(x)=10\log_{10}\left\{\frac{\left(65.61\left(x_1-x_2\omega^2\right)\right)^2+\left(24734.97\omega\right)^2}{\left(-1157.39\omega\right)^2+\left(3.07\left(x_1-\frac{\omega^2}{\left(\frac{1}{65.61}+\frac{1}{x_2}\right)}\right)\right)^2}+20\log_{10}\left(\frac{x_3}{\left(65.61+x_2\right)\omega}\right)\right\}$$

With parameter uncertainties as follows:

$$x_1\in[5800,11500],\quad x_2\in[1400,11000],\quad x_3\in[100,800]$$

for frequency ω=50.

Range computed using NIE and Uniform subdivision

True_range = [64.17591070389160, 92.11586932433393]

N	Range computed using NIE and Uniform subdivision
64	[63.30270186117932, 92.43277538797648]
128	[63.72755957264531, 92.17531914664293]
256	[63.94859696170153, 92.14554381891074]
512	[64.06144077281130, 92.13069401039323]
1024	[64.11846865754421, 92.12327853239701]
2048	[64.14713741612914, 92.11957314529018]
4096	[64.16151093091627, 92.11772103912668]
8192	[64.16870752718803, 92.11679513281965]

Romberg Table for Infimum

	K=0	*K=1*	*K=2*	*K=3*
1.	63.30270186117932			
2.	63.72755957264531	64.15241728411131		
3.	63.94859696170154	64.16963435075778	64.17537337297326	
4.	64.06144077281131	64.17428458392108	64.17583466164219	64.17590056002345
5.	64.11846865754423	64.17549654227715	64.17590052839584	64.17590993793208
6.	64.14713741612916	64.17580617471408	64.17590938552640	64.17591065083076
7.	64.16151093091628	64.17588444570340	64.17591053603319	64.17591070039130
8.	64.16870752718805	64.17590412345982	64.17591068271196	64.17591070366606

	K=4	*K=5*	*K=6*	*K=7*
5.	64.17591056312600			
6.	64.17591069835733	64.17591070271963		
7.	64.17591070369534	64.17591070386753	64.17591070388575	
8.	64.17591070388438	64.17591070389048	64.17591070389084	64.17591070389088

BTII for Infimum

i	*K=1*	*K=2*	*K=3*
3.	[63.9485, 64.3907]		
4.	[64.0614, 64.2872]	[64.1742, 64.1774]	
5.	[64.1184, 64.2326]	[64.1754, 64.1764]	[64.1759, 64.1760]
6.	[64.1471, 64.2045]	[64.1758, 64.1761]	[64.1759, 64.1760]
7.	[64.1615, 64.1903]	[64.1758, 64.1760]	[64.1759, 64.1760]
8.	[64.1687, 64.1832]	[64.1759, 64.1760]	[64.1759, 64.1760]

	K=4	*K=5*	*K=6*
6.	[64.1759, 64.1760]		
7.	[64.1759, 64.1760]	[64.1759, 64.1760]	
8.	[64.1759, 64.1760]	[64.1759, 64.1760]	[64.1759, 64.1760]

Overestimation and Ratio for BTII

Overestimation

i	*k=1*	*k=2*	*k=3*	*k=4*	*k=5*
	max (true)	max (true)	max (true)	max (true)	max (true)
3.	4.4e-1(2.3e-1)				
4.	2.3e-1(1.1e-1)	3.1e-3(1.6e-3)			
5.	1.1e-1(5.7e-2)	8.1e-4(4.1e-4)	1.9e-5(1.0e-5)		
6.	5.7e-1(2.9e-2)	2.1e-4(1.0e-4)	2.5e-6(1.3e-6)	9.5e-08(5.3e-08)	
7.	2.9e-1(1.4e-2)	5.2e-5(2.6e-5)	3.3e-7(1.7e-7)	6.6e-09(3.5e-09)	3.4e-10(1.9e-10)
8.	1.4e-2(7.2e-3)	1.3e-5(6.6e-6)	4.2e-8(2.1e-8)	4.4e-10(2.3e-10)	1.2e-11(7.3e-12)

Overestimation Ratio

i	*k=1*	*k=2*	*k=3*	*k=4*	*k=5*
4.	1.9858				
5.	1.9928	3.9263			
6.	1.9964	3.9622	7.7183		
7.	1.9982	3.9808	7.8540	15.158	
8.	1.9991	3.9903	7.9254	15.515	26.787
	$O(2^1=2)$	$O(2^2=4)$	$O(2^3=8)$	$O(2^4=16)$	$O(2^5=32)$

Romberg Table for Supremum

	K=0	K=1	K=2	K=3
1.	92.43277538797648			
2.	92.17531914664292	91.91786290530936		
3.	92.14554381891072	92.11576849117853	92.18173701980159	
4.	92.13069401039321	92.11584420187570	92.11586943877477	92.10645978434236
5.	92.12327853239701	92.11586305440081	92.11586933857585	92.11586932426171
6.	92.11957314529018	92.11586775818334	92.11586932611085	92.11586932433013
7.	92.11772103912668	92.11586893296318	92.11586932455647	92.11586932433441
8.	92.11679513281965	92.11586922651262	92.11586932436244	92.11586932433471

	K=4	K=5	K=6	K=7
5.	92.11649662692301			
6.	92.11586932433470	92.11584908876733		
7.	92.11586932433470	92.11586932433470	92.11586964553419	
8.	92.11586932433474	92.11586932433474	92.11586932433474	92.11586932180560

BTIS for Supremum

	K=1	*K=2*	*K=3*
3.	[92.0859, 92.1456]		
4.	[92.1009, 92.1307]	[92.1158, 92.1159]	
5.	[92.1084, 92.1233]	[92.1158, 92.1159]	[92.1158, 92.1159]
6.	[92.1121, 92.1196]	[92.1158, 92.1159]	[92.1158, 92.1159]
7.	[92.1140, 92.1178]	[92.1158, 92.1159]	[92.1158, 92.1159]
8.	[92.1149, 92.1168]	[92.1158, 92.1159]	[92.1158, 92.1159]

	K=4
6.	[92.1158, 92.1159]
7.	[92.1158, 92.1159]
8.	[92.1158, 92.1159]

Overestimation and Ratio for BTIS

Overestimation

i	*k=1*	*k=2*	*k=3*	*k=4*
	max (true)	max (true)	max (true)	max (true)
3.	6.0e-02(3.0e-02)			
4.	3.0e-02(1.5e-02)	5.0e-05(2.5e-05)		
5.	1.5e-02(7.4e-03)	1.3e-05(6.3e-06)	2.9e-08(1.4e-08)	
6.	7.4e-03(3.7e-03)	3.1e-06(1.6e-06)	3.6e-09(1.8e-09)	(9.3e-12)(5.4e-12)
7.	3.7e-03(1.9e-03)	7.8e-07(3.9e-07)	4.4e-10(2.2e-10)	(*)(1.1e-12)
8.	1.9e-03(9.3e-04)	2.0e-07(9.8e-08)	5.6e-11(2.9e-11)	(*)(8.8e-13)

Overestimation Ratio

i	*k=1*	*k=2*	*k=3*	*k=4*
4.	2.0017			
5.	2.0008	4.0250		
6.	2.0004	4.0125	8.0148	
7.	2.0002	4.0062	7.9838	-
8.	2.0001	4.0031	7.7998	-
	$O(2^1=2)$	$O(2^2=4)$	$O(2^3=8)$	

(*) – denotes the zero within machine precision.

Range Enclosures: [inf(BTII), sup(BTIS)]

i	*k=0*	*k=1*
1.	[63.30270186117932, 92.43277538797648]	
2.	[63.72755957264531, 92.17531914664293]	
3.	[63.94859696170153, 92.14554381891074]	[63.94859696170151, 92.14554381891074]
4.	[64.06144077281130, 92.13069401039323]	[64.06144077281130, 92.13069401039323]
5.	[64.11846865754421, 92.12327853239701]	[64.11846865754421, 92.12327853239701]
6.	[64.14713741612914, 92.11957314529018]	[64.14713741612914, 92.11957314529018]
7.	[64.16151093091627, 92.11772103912668]	[64.16151093091627, 92.11772103912668]
8.	[64.16870752718803, 92.11679513281965]	[64.16870752718803, 92.11679513281965]

i	*k=2*	*k=3*
4.	[64.17428458392106, 92.11589467567383]	
5.	[64.17549654227713, 92.11587562275091]	[64.17590052839580, 92.11586933857588]
6.	[64.17580617471405, 92.11587089403837]	[64.17590938552633, 92.11586932611090]
7.	[64.17588444570338, 92.11586971614976]	[64.17591053603312, 92.11586932455650]
8.	[64.17590412345980, 92.11586942221226]	[64.17591068271190, 92.11586932436248]

i	*k=4*
6.	[64.17591065083067, 92.11586932433933]
7.	[64.17591070039119, 92.11586932433507]
8.	[64.17591070366599, 92.11586932433481]

True Range Overestimation and Ratio

Overestimation

i	*k=1*	*k=2*	*k=3*	*k=4*	*k=5*
1.	1.2				
2.	5.1e-01				
3.	2.6e-01	2.6e-01			
4.	1.3e-01	1.3e-01	1.7e-03		
5.	6.5e-02	6.5e-02	4.2e-04	1.0e-05	
6.	3.2e-02	3.2e-02	1.1e-04	1.3e-06	5.3e-08
7.	1.6e-02	1.6e-02	2.7e-05	1.7e-07	3.5e-09
8.	8.1e-03	8.1e-03	6.7e-06	2.1e-08	2.3e-10

Overestimation Ratio

i	*k=1*	*k=2*	*k=3*	*k=4*	*k=5*
1.	2.3437				
2.	1.9760				
3.	1.9876	1.9876			
4.	1.9937	1.9937	3.9278		
5.	1.9968	1.9968	3.9629	7.7187	
6.	1.9984	1.9984	3.9812	7.8542	15.155
7.	1.9992	1.9992	3.9905	7.9253	15.460
	$O(2^1=2)$	**$O(2^2=4)$**	**$O(2^3=8)$**	**$O(2^4=16)$**	**$O(2^5=32)$**

Comparison of Existing method with the proposed method

- Number of subdivisions required and number of boxes generated to compute the Bode magnitude envelope with accuracy $\varepsilon = 10^{-10}$.

	Existing Method	Proposed method
Subdivision factors	2^{40}	2^{13}
Number of Boxes in the final list (with cutoff test)	4974	4872
Time (sec.)	24.6	4.9

Comparison of the proposed method with Globsol

- Time (sec.) required to compute the Bode magnitude envelope with accuracy $\varepsilon = 10^{-10}$.

Proposed method	Globsol
4.9 sec.	19.18 sec.

Conclusions

- We could Accelerate the order of Convergence of NIE by Extrapolation.
- The width of the Brezinski's intervals is a good measure of having attained the specified accuracy
- High accuracy can be achieved with comparatively less computational effort !
- In the vehicle clutch example:
  - Number of subdivisions required to compute the Range enclosure with the proposed method is significantly reduced.
  - The proposed method is faster compared with **Globsol**.