4-1982

An Interval Arithmetic Newton Method for Solving Systems of Nonlinear Equations

Ronald I. Greenberg
Washington University in St. Louis, Rgreen@luc.edu

Eldon R. Hansen
Lockheed Missiles & Space Company

AN INTERVAL ARITHMETIC NEWTON METHOD FOR SOLVING SYSTEMS OF NONLINEAR EQUATIONS

WE INTRODUCE AN INTERVAL NEWTON METHOD FOR BOUNDING SOLUTIONS OF SYSTEMS OF NONLINEAR EQUATIONS.

IT ENTAILS THREE SUB-ALGORITHMS:

- A GAUSS-SEIDEL TYPE STEP
- A REAL (NON-INTERVAL) NEWTON ITERATION
- SOLUTION OF THE LINEARIZED EQUATIONS BY ELIMINATION

WE EXPLAIN WHY EACH SUB-ALGORITHM IS DESIRABLE AND HOW THEY FIT TOGETHER TO PROVIDE SOLUTIONS IN AS LITTLE AS 1/3 TO 1/4 THE TIME REQUIRED BY KRAWCZYK'S METHOD.

I. INTRODUCTION TO INTERVAL ARITHMETIC
II. INTRODUCTION TO INTERVAL NEWTON METHODS
III. COMPOSITION OF THE ALGORITHM
IV. EXPERIMENTAL RESULTS

I. INTERVAL ARITHMETIC

A. MOTIVATION FOR INTERVAL ARITHMETIC

B. THE FOUR ARITHMETIC OPERATIONS

C. EXTENDED INTERVAL ARITHMETIC

D. GENERAL FUNCTIONS AND DIRECTED ROUNDING

E. COMPUTER IMPLEMENTATIONS

A. MOTIVATION FOR INTERVAL ARITHMETIC

** AUTOMATIC ERROR ANALYSIS!!

PRIMARY SOURCES OF ERROR ARE:

- ROUNDING
- LIMITED ACCURACY OF INPUT DATA
- LOSS OF LEADING SIGNIFICANT DIGITS IN SUBTRACTION OF NEARLY EQUAL VALUES (CANCELLATION)

FOR EXAMPLE, ON A FIVE DECIMAL DIGIT MACHINE WITH X=.99999 AND Y=.99998, SUPPOSE WE COMPUTE Z=X-Y. BY HAND WE GET

$$\begin{array}{r} .99999 \\ -.99998 \\ \hline .00001 \end{array} \quad \text{OR } .1 \times 10^{-4}.$$

SINCE WE KNOW NOTHING ABOUT X AND Y AFTER THE FIFTH DIGIT, WE CAN SAY ONLY THAT $0<Z<.2 \times 10^{-4}$. BUT THE COMPUTER WILL GIVE A FIVE DIGIT ANSWER, E.G. Z=.10000E-4. (THE ANSWER MAY BE SLIGHTLY DIFFERENT DUE TO BASE CONVERSIONS.) THE COMPUTER GIVES NO INDICATION OF THE LIMITED ACCURACY OF ITS FIVE DIGIT ANSWER.

B. THE FOUR ARITHMETIC OPERATIONS

$A=[A_1,A_r]$ $\qquad\qquad$ $B=[B_1,B_r]$

$$A+B = [A_1+B_1,A_r+B_r] = \{a+b \mid a\in A, b\in B\}$$

$$A-B = [A_1-B_r,A_r-B_1] = \{a-b \mid a\in A, b\in B\}$$

$$A\cdot B = [\min\{A_1\cdot B_1,A_1\cdot B_r,A_r\cdot B_1,A_r\cdot B_r\},\ \max\{A_1\cdot B_1,A_1\cdot B_r,A_r\cdot B_1,A_r\cdot B_r\}]$$

A/B similar

IN MULTIPLICATION AND DIVISION THE NUMBER OF PRODUCTS OR QUOTIENTS CALCULATED CAN BE REDUCED BY TESTING THE SIGNS OF THE INTERVAL ENDPOINTS.

C. EXTENDED INTERVAL ARITHMETIC

$A = [A_1, A_r]$ $\quad B = [B_1, B_r]$ $\quad 0 \notin B$

$A/B =$

$[A_r/B_1, +\infty]$ IF $A_r < 0$ AND $B_r = 0$

$[-\infty, A_r/B_r] \cup [A_r/B_1, +\infty]$ IF $A_r < 0$, $B_1 < 0$, AND $B_r > 0$

$[-\infty, A_r/B_r]$ IF $A_r < 0$ AND $B_1 = 0$

$[-\infty, A_1/B_1]$ IF $A_1 > 0$ AND $B_r = 0$

$[-\infty, A_1/B_1] \cup [A_1/B_r, +\infty]$ IF $A_1 > 0$, $B_1 < 0$, and $B_r > 0$

$[A_1/B_r, +\infty]$ IF $A_1 > 0$ AND $B_1 = 0$

$[-\infty, +\infty]$ IF $A_1 \leq 0$ AND $A_r \geq 0$

D. GENERAL FUNCTIONS AND DIRECTED ROUNDING

USING EXACT INTERVAL ARITHMETIC TO EVALUATE ANY FUNCTION f INVOLVING THE FOUR BASIC OPERATIONS, WE CAN BE SURE THAT $f(X) \supseteq \{f(x) \mid X_l < x < X_r\}$ WHERE $X = [X_l, X_r]$. IN ACTUAL COMPUTATIONS WITH A FINITE ACCURACY MACHINE, WE MUST PERFORM DIRECTED ROUNDING TO INSURE THAT THE RESULT OF OUR CALCULATION IS AN INTERVAL CONTAINING THE DESIRED RESULT. AS EACH ARITHMETIC OPERATION IS PERFORMED, WE ROUND THE LEFT ENDPOINT OF THE RESULT DOWN AND THE RIGHT ENDPOINT UP.

THOUGH WE ARE SURE TO GET RESULTS BRACKETING THE CORRECT ANSWER, OUR RESULT MAY BE PESSIMISTIC. IT IS A CHALLENGE IN INTERVAL ANALYSIS TO FIND ALGORITHMS WHICH DO NOT LEAD TO OVERLY PESSIMISTIC NUMERICAL RESULTS. OFTEN SIMPLE REARRANGEMENT OF AN ARITHMETIC EXPRESSION CAN MAKE A SUBSTANTIAL IMPROVEMENT.

FOR EXAMPLE, $(X+[0,1])/X$ SHOULD BE WRITTEN AS $1 + [0,1]/X$.

LETTING X = 1,3 WE CAN CALCULATE

$$\frac{X+[0,1]}{X} = \frac{[1,4]}{[1,3]} = [1/3,4] \qquad \text{AND} \qquad 1 + \frac{[0,1]}{X} = [1,1] + [0,1] = [1,2].$$

E. COMPUTER IMPLEMENTATIONS

1) KLUGE

2) PRECOMPILER

3) INTEL 8087 OR 432 MICROPROCESSORS

4) ADDITIONAL VARIABLE TYPE "XOTHER" ADDED TO THE MNF AND M77 FORTRAN COMPILERS DEVELOPED AT THE UNIVERSITY OF MINNESOTA COMPUTING CENTER.

(Notes for this slide at end.)

II. INTRODUCTION TO INTERVAL NEWTON METHODS

— CONSIDER A CONTINUOUSLY DIFFERENTIABLE FUNCTION $f:R^n \to R^n$ WITH JACOBIAN J. AND LET y BE A ZERO OF f.

BY TAYLOR'S THEOREM,

$$f(x) + J(\xi)(y-x) = f(y) = 0.$$

— IF X IS A BOX (AN INTERVAL VECTOR) CONTAINING x AND y, THEN THE POINTS INDICATED BY THE NOTATION ξ ARE CONTAINED IN X. THUS y IS CONTAINED IN THE SET Z OF POINTS SATISFYING $f(x) + J(X)(z-x) = 0$.

— INTERVAL NEWTON METHODS FIND A BOX $N(x,X)$ CONTAINING Z. THE PROCESS IS INTERATED. GIVEN A BOX $X^{(0)}$, NEW ITERATES ARE OBTAINED AS

$$X^{(k+1)} = X^{(k)} \cap N(x^{(k)}, X^{(k)}).$$

— IT IS BEST FIRST TO MULTIPLY BY AN APPROXIMATE INVERSE OF J^c, WHERE J^c DENOTES THE CENTER OF J(X).

LETTING B DENOTE OUR APPROXIMATE INVERSE, THE MULTIPLICATION YIELDS

$$M(z-x) = b \quad \text{WHERE } M=BJ(X) \text{ AND } b=-Bf(x).$$

$$M(z-x)=b \qquad (M=BJ(X) \text{ AND } b=Bf(x))$$

INTERVAL NEWTON METHODS DIFFER IN HOW THIS EQUATION IS SOLVED.

- KRAWCZYK METHOD: DOES NOT ATTEMPT TO OBTAIN Z, BUT ONLY A BOUND FOR Z.

- HANSEN-SENGUPTA METHOD: SAME APPLIES, BUT CONVERGENCE SUPERIOR TO KRAWCZYK METHOD.

- ELIMINATION METHOD: SOLVE THE EQUATION BY GAUSSIAN ELIMINATION. SEEKS Z RATHER THAN A BOUND FOR Z, BUT THIS METHOD CANNOT ALWAYS BE EMPLOYED SINCE M MAY CONTAIN A SINGULAR MATRIX, AND MORE IMPORTANTLY, INTERVALS TEND TO GROW DURING THE ELIMINATION PROCESS, SO THAT POOR RESULTS MAY BE OBTAINED.

- REAL ITERATION IS EMPLOYED TO IMPROVE THE VALUE OF x BEFORE ELIMINATION IS ATTEMPTED.

- INNER ITERATION REDUCES THE NUMBER OF TIMES J(X), B, AND BJ(X) MUST BE COMPUTED.

III. COMPOSITION OF THE ALGORITHM

A. INITIAL HANSEN-SENGUPTA STEP

B. REAL ITERATION

C. ELIMINATION ITERATION

D. ADDITIONAL HANSEN-SENGUPTA ITERATION

E. SPLITTING THE BOX

A. INITIAL HANSEN-SENGUPTA STEP

— AFTER CALCULATING J AND B, AND PERFORMING THE MULTIPLICATION BY B, WE FIRST PERFORM ONE HANSEN-SENGUPTA STEP. WE WISH TO SOLVE THE i-TH EQUATION OF $M(z-x)=b$ FOR THE i-TH VARIABLE. THAT IS

$$Y_i = x_i + R_i / M_{ii}$$

WHERE

$$R_i = b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} M_{ij}(X_j - x_j).$$

HERE WE HAVE REPLACED z_j BY X_j FOR ALL $j \neq i$. THE INTERVAL Y_i CONTAINS EVERY SOLUTION z_i IN X_i.

— WE PERFORM THE CALCULATIONS FIRST FOR THOSE i SUCH THAT $0 \notin M_{ii}$, AND IN EACH CASE, WE REPLACE X_i BY

$$X_i' = X_i \cap Y_i.$$

— WE THEN PROCEED TO DEAL WITH THOSE COMPONENTS FOR WHICH $0 \in M_{ii}$ IN A SIMILAR MANNER. HERE WE MAY OBTAIN A "GAP" WHICH MAY BE USED LATER TO SPLIT THE BOX.

B. REAL ITERATION

- WE ATTEMPT TO FIND AN IMPROVED APPROXIMATION x FOR A ZERO OF f IN X (IF ONE EXISTS)

  BY STARTING AT $x^{(0)}=m(X)$ AND COMPUTING

$$x^{(k+1)} = x^{(k)} - Bf(x^{(k)}),$$

C. ELIMINATION ITERATION

- IF WE HAVE FOUND AN x SUCH THAT $||f(x)||$ IS SUFFICIENTLY SMALL, WE ATTEMPT TO PERFORM AN LU DECOMPOSITION OF M.

- IF SUCCESSFUL, WE CALCULATE $b=-Bf(x)$ AND PERFORM FORWARD AND BACK SUBSTITUTION TO SOLVE $M(z-x)=B$. THEN WE INTERSECT OUR SOLUTION WITH X AND CALCULATE A NEW x AT THE CENTER OF THE NEW X.

- EACH TIME THE WIDTH OF X (THE WIDTH OF ITS WIDEST COMPONENT) DECREASES SIGNIFICANTLY (TO .9 OF THE PREVIOUS WIDTH, SAY), WE REPEAT THE ELIMINATION STEP, STARTING AT THE CALCULATION OF $b=-Bf(x)$ WITH THE NEW x.

D. ADDITIONAL HANSEN-SENGUPTA ITERATION

- IF WE ARE UNABLE TO COMPLETE AN ELIMINATION STEP, WE PERFORM HANSEN-SENGUPTA ITERATION INSTEAD.

- SINCE WE HAVE ALREADY DONE OUR BEST TO FIND A WIDE GAP IN THE BOX X, WE PERFORM THE CALCULATIONS

$$Y_i = X_i + R_i / M_{ii} \quad \text{WHERE} \quad R_i = b_i - \sum_{\substack{j=1 \\ j \neq 1}}^{n} M_{ij}(X_j - x_j)$$

$$X_i' = X_i \cap Y_i$$

ONLY FOR THOSE i SUCH THAT $0 \notin M_{ii}$.

- EACH TIME X IMPROVES SIGNIFICANTLY, WE CALCULATE A NEW x AT ITS CENTER AND REPEAT THE HANSEN-SENGUPTA STEP.

E. SPLITTING THE BOX

- IF WE HAVE FOUND A GAP USING THE INITIAL HANSEN-SENGUPTA STEP DESCRIBED EARLIER, WE MAKE USE OF THAT GAP TO SPLIT THE BOX (IF IT STILL DOES SO)

- IF WE HAVE FOUND NO GAP AND WE HAVE NOT MANAGED TO IMPROVE THE BOX WIDTH SIGNIFICANTLY AT ANY STAGE, WE SPLIT THE BOX AT THE CENTER OF ITS WIDEST COMPONENT.

- WHEN THE BOX IS SPLIT, ONE PART BECOMES THE CURRENT BOX, AND THE OTHER IS PLACED ON A STACK FOR LATER PROCESSING.

- WHETHER OR NOT WE SPLIT THE BOX, WE NOW RETURN TO THE BEGINNING OF THE OUTER STEP, THE CALCULATION OF THE JACOBIAN MATRIX.

F. SUMMARY OF ALGORITHM

1. CALCULATE J(X).
2. CALCULATE B, THE APPROXIMATE INVERSE OF J^C.
3. CALCULATE M=BJ(X).
4. SET x EQUAL TO THE CENTER OF X.
5. PERFORM A HANSEN-SENGUPTA STEP FOR THOSE i SUCH THAT $0 \notin M_{ii}$.
6. PERFORM A HANSEN-SENGUPTA STEP FOR THOSE i SUCH THAT $0 \in M_{ii}$, AND SAVE THE LARGEST GAP.
7. PERFORM REAL ITERATION TO IMPROVE x, STARTING AT THE CENTER OF X.
8. IF ||f(x)|| IS NOT SUFFICIENTLY SMALL,SKIP TO STEP 14.
9. PERFORM THE LU DECOMPOSITION OF M IF POSSIBLE; OTHERWISE SKIP TO STEP 14.
10. CALCULATE Bf(x) AND SOLVE FOR Z BY FORWARD AND BACK SUBSTITUTION.
11. REPLACE X BY $X \cap Z$.
12. IF THE WIDTH OF X IMPROVED SIGNIFICANTLY, SET x EQUAL TO THE CENTER OF X, AND RETURN TO STEP 10.
13. SKIP TO STEP 16.
14. PERFORM A HANSEN-SENGUPTA STEP FOR THOSE COMPONENTS SUCH THAT $0 \notin M_{ii}$.
15. IF X IMPROVED SIGNIFICANTLY, SET x EQUAL TO THE CENTER OF X, AND RETURN TO STEP 14.
16. IF WE HAVE A GAP SAVED FROM STEP 6, USE IT. IF THE BOX SPLITS, PUT ONE PART ON THE STACK, AND KEEP THE OTHER PART AS THE NEW X. RETURN TO STEP 1.
17. IF THE BOX DID NOT IMPROVE SIGNIFICANTLY IN STEPS 5 AND 6, IN STEP 11, OR IN STEP 14, THAN SPLIT AT THE CENTER OF THE WIDEST COMPONENT. SAVE ONE HALF ON THE STACK, AND USE THE OTHER HALF AS THE NEW X.
18. RETURN TO STEP 1.

IF IN STEPS 5 AND 6, IN STEP 11, OR IN STEP 14, THE BOX IS NARROWED TO WITHIN THE ACCEPTABLE TOLERANCE OR IS FOUND NOT TO CONTAIN A SOLUTION, THE MOST RECENTLY STACKED BOX BECOMES THE NEW X, AND WE RETURN TO STEP 1. WHEN WE ENCOUNTER AN EMPTY STACK, WE ARE DONE.

IV. EXPERIMENTAL RESULTS

- SEVERAL PROBLEMS WERE SOLVED USING VARIOUS VERSIONS OF THE ALGORITHM DESCRIBED ABOVE AND, FOR COMPARISON, VARIOUS VERSIONS OF THE KRAWCZYK METHOD.

- IN THE FINAL VERSION OF OUR ALGORITHM AS DESCRIBED ABOVE, IT IS POSSIBLE TO VARY THE FACTOR REPRESENTING SIGNIFICANT IMPROVEMENT OF THE BOX WIDTH. THIS ALLOWS THE USER TO APPLY SOME CONTROL OF HOW MUCH INNER ITERATION IS PERFORMED, ACCORDING TO HOW DIFFICULT IT IS TO CALCULATE THE JACOBIAN, THE INVERSE OF ITS CENTER, AND THE NECESSARY INTERVAL ARITHMETIC MATRIX PRODUCT.

- WE SHOW REPRESENTATIVE RESULTS FOR THE BROYDEN BANDED FUNCTION

$$f_i(x) = x_i(2+5x_i^2) + 1 - \sum_{j\in J_i} x_j(1+x_j) \qquad (i=1,\ldots,n)$$

WHERE $J_i = \{j \mid j=i,\ \max(1,i-5)\leq j\leq \min(n,i+1)\}$.

THE INITIAL BOX WAS $[-1,1]$ IN EACH COMPONENT AND CONTAINS ONE SOLUTION. IT WAS REQUIRED THAT THE SOLUTION BOX HAVE A WIDTH LESS THAN 10^{-8}.

THE FOLLOWING VALUES OF INTEREST ARE TABULATED FOR THE SAMPLE RUNS:

S = THE SIGNIFICANT BOX WIDTH IMPROVEMENT FACTOR

N_O = THE NO. OF OUTER STEPS = THE NO. OF JACOBIAN EVALUATIONS = THE NO. OF MATRIX INVERSIONS = THE NO. OF INTERVAL ARITHMETIC MATRIX-MATRIX PRODUCTS.

N_F = THE NO. OF INTERVAL ARITHMETIC FUNCTION EVALUATIONS = THE NO. OF INTERVAL ARITHMETIC MATRIX-VECTOR PRODUCTS = N_K OR $N_E + N_{HS1}$

N_K = THE NO. OF KRAWCZYK STEPS

N_R = THE NO. OF REAL ITERATIONS = THE NO. OF REAL FUNCTION EVALUATIONS = THE NO. OF REAL MATRIX-VECTOR PRODUCTS

N_{LU}= THE NO. OF INTERVAL ARITHMETIC LU DECOMPOSITION ATTEMPTS

N_E = THE NO. OF ELIMINATION STEPS

N_{HS1}= THE NO. OF HANSEN-SENGUPTA STEPS FOR THOSE i WITH $0 \notin M_{ii}$

N_{HS2}= THE NO. OF HANSEN-SENGUPTA STEPS FOR THOSE i WITH $0 \in M_{ii}$

THE TIME IS MINUTES:SECONDS. THE EXPERIMENTS WERE RUN ON THE SLOW HP 9845.

S REPRESENTS THE SIGNIFICANT BOX IMPROVEMENT FACTOR
N_0 REPRESENTS THE NO. OF OUTER STEPS
TIME IS MINUTES:SECONDS

n=3, KRAWCZYK METHOD WITH INNER ITERATION

S	N_0	TIME
.6	57	8:34
.7	46	7:30
.8	44	9:52
.9	36	11:48

n=3, HANSEN SENGUPTA METHOD WITHOUT INNER ITERATION

S	N_0	TIME
.8	21	2:37

n=3, NEW METHOD

S	N_0	TIME
.6	13	2:31
.8	13	2:29
.9	12	2:28
.99	12	2:28

n=5, KRAWCZYK METHOD WITH INNER ITERATION

S	N_0	TIME
.9	194	80:14

n=5, HANSEN SENGUPTA METHOD WITHOUT INNER ITERATION

S	N_0	TIME
.9	80	30:03

n=5, NEW METHOD

S	N_0	TIME
.9	46	23:17

n=3, KRAWCZYK METHOD WITH INNER ITERATION

S	N_0	N_F	N_K	TIME
.6	57	80	80	8:34
.7	46	74	74	7:30
.8	44	114	114	9:52
.9	36	153	153	11:48

n=3, HANSEN-SENGUPTA METHOD WITHOUT INNER ITERATION

S	N_0	N_F	N_{HS1}	N_{HS2}	TIME
.8	21	21	21	16	2:37

n=3, NEW METHOD

S	N_0	N_F	N_R	N_{LU}	N_E	N_{HS1}	N_{HS2}	TIME
.6	13	27	18	2	3	24	13	2:31
.8	13	26	17	2	3	23	12	2:29
.9	12	27	17	2	3	24	12	2:28
.99	12	27	17	2	3	24	12	2:28

n=5, KRAWCZYK METHOD WITH INNER ITERATION

S	N_0	N_F	N_K	TIME
.9	194	234	234	80:14

n=5, HANSEN-SENGUPTA METHOD WITHOUT INNER ITERATION

S	N_0	N_F	N_{HS1}	N_{HS2}	TIME
.9	80	80	80	45	30:03

n=5, NEW METHOD

S	N_0	N_F	N_R	N_{LU}	N_E	N_{HS1}	N_{HS2}	TIME
.9	46	88	47	2	6	82	38	23:17

E. COMPUTER IMPLEMENTATIONS

1) KLUGE - A SUBROUTINE CALL FOR EVERY ARITHMETIC OPERATION. LABORIOUS PROGRAMMING AND SLOW EXECUTION.

2) PRECOMPILER - THERE DOES EXIST A PRECOMPILER WHICH TRANSLATES FORTRAN TO INCLUDE THE REQUISITE CALLS TO AN EXTERNAL LIBRARY OF INTERVAL SUBROUTINES. EXECUTION SPEED IS 19 TO 192 TIMES SLOWER THAN THE SINGLE PRECISION VERSION OF THE PROGRAM.

3) INTEL 8087 OR 432 MICROPROCESSORS - INTEL HAS DEVELOPED CHIPS TO PERFORM DIRECTED ROUNDING WITH THE INTENTION OF IMPLEMENTING INTERVAL ARITHMETIC, BUT A COMPILER DOES NOT YET EXIST TO EXPLOIT THIS NEW HARDWARE.

4) ADDITIONAL VARIABLE TYPE "XOTHER" ADDED TO THE MNF AND M77 FORTRAN COMPILERS DEVELOPED AT THE UNIVERSITY OF MINNESOTA COMPUTING CENTER. WHENEVER THE COMPILER ENCOUNTERS OPERATIONS INVOLVING VARIABLES OF TYPE XOTHER, APPROPRIATE ROUTINES ARE USED TO PERFORM THE DESIRED OPERATIONS. EASY TO USE. 4 TO 6 TIMES SLOWER THAN ORDINARY SINGLE PRECISION IMPLEMENTATIONS FOR ALGORITHMS DOMINATED BY ARITHMETIC OPERATIONS. PROGRAMS WITH MANY FUNCTION CALLS MAY BE 12 TIMES SLOWER.