Formal verification for numerical methods

Ioana Paşca

PhD thesis at INRIA Sophia Antipolis, Marelle Team
Advisor: Yves Bertot

Robots

Robots

- efficiency

Robots

- efficiency
- safety

Inside the robot

lots of wires

Inside the robot

lots of wires

code

- computations
- numerical methods

```
Public Function GetChecksum(ByVal sentence As String) As String
    Dim Character As Char
    Dim Checksum As Integer
    For Each Character In sentence
        Select Case Character
            Case "$"c
                ' Ignore the dollar sign
            Case "*"c
                ' Stop processing before the asterisk
                Exit For
            Case Else
                ' Is this the first value for the checksum?
                If Checksum = 0 Then
                    ' Yes. Set the checksum to the value
                    Checksum = Convert.ToByte(Character)
                Else
                    Checksum = Checksum Xor Convert.ToByte(Character)
                End If
        End Select
    Next
    Return Checksum.ToString("X2")
End Function
```

Ensure correctness of the code

tests

Ensure correctness of the code

tests

proofs

- proof assistants
- formal verification

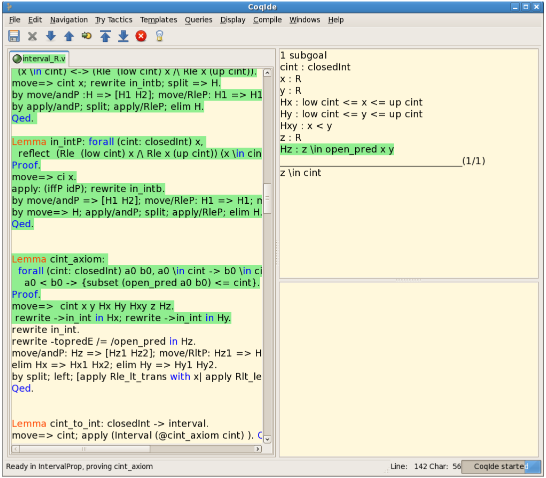
Proof assistants

- define data
- define functions
- prove properties
- trust proofs

Proof assistants

Coq and SSReflect

- define data
- define functions
- prove properties
- trust proofs



Example tasks and methods

compute a valid position for the robot by solving a system of equations

- use Newton's method and its properties
- use interval analysis based methods

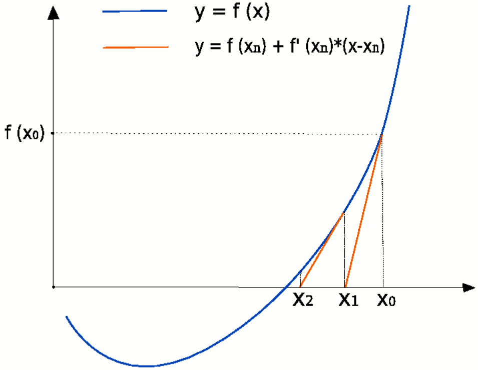
Newton's method

Definition:

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

Properties:

- convergence to the root of function f
- speed of convergence
- local unicity of the root
- local stability



Kantorovitch's theorem in one dimension

Consider an equation $f(x) = 0$, where $f :]a, b[\to \mathbb{R}$, $a, b \in \mathbb{R}$,
$f \in C^{(1)}(]a, b[)$.
Let $x_0 \in]a, b[$ such that $\overline{U_\varepsilon}(x_0) = \{x : |x - x_0| \le \varepsilon\} \subset]a, b[$.
If:

1. $f'(x_0) \neq 0$ and $\left|\dfrac{1}{f'(x_0)}\right| \le A_0$;
2. $\left|\dfrac{f(x_0)}{f'(x_0)}\right| \le B_0 \le \dfrac{\varepsilon}{2}$;
3. $\forall x, y \in]a, b[, |f'(x) - f'(y)| \le C|x - y|$
4. $\mu_0 = 2A_0B_0C \le 1$.

then, the Newton process $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$, $n = 0, 1, 2, \ldots$ converges and $\lim\limits_{n\to\infty} x_n = x^*$ is a solution of the initial equation, so that $|x^* - x_0| \le 2B_0 \le \varepsilon$.

Idea of the proof

- for every element of the Newton's sequence show that hypotheses 1 - 4 are verified, with different constants;

- infer that Newton's sequence is a Cauchy sequence and, by the completeness of $\mathbb{R}$, a convergent sequence;

- prove that the limit of the sequence is a root of the given function.

Working with real numbers

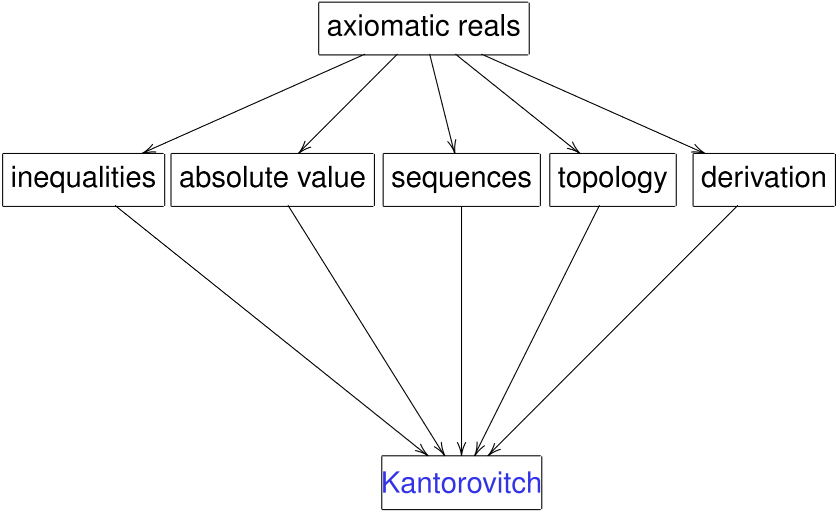
use the Coq standard library on real numbers

- axiomatic definition: archimedean, total order field, satisfying the least upper bound principle
- classical real analysis
- proofs close to “pen and paper” mathematics

other approaches

- constructive definition
- intuitionistic logic
- non-standard analysis

Formal proof in one dimension



Some COQ code:

```
Theorem kanto_exist:
  ∃ xs:R, Un_cv Xn xs ∧ c_disc X0 (2*B0) xs ∧ f xs = 0.
```

Kantorovitch's theorem in several dimensions

Let $f(x) = 0$ be a system with p equations and p variables, with $f \in C^{(2)}(\omega)$ and $\overline{U_\varepsilon}(x_0) = \{\|x - x_0\| \leq \varepsilon\} \subset \omega$.
If:

1. the Jacobian matrix $W(x) = [\frac{\partial f_i}{\partial x_j}]$ for $x = x_0$ has an inverse $\Gamma_0 = W^{-1}$ with $\|\Gamma_0\| \leq A_0$;
2. $\|\Gamma_0 f(x_0)\| \leq B_0 \leq \frac{\varepsilon}{2}$;
3. $\sum_{k=1}^{p} |\frac{\partial^2 f_i(x)}{\partial x_j \partial x_k}| \leq C$ for $i, j = 1, 2, ..., p$ and $x \in \overline{U_\varepsilon}(x_0)$;
4. $2pA_0B_0C \leq 1$.

Then, Newton's process:
$x_{n+1} = x_n - W^{-1}(x_n)f(x_n)$ converges to $x^* = \lim_{n\to\infty} x_n$.

Kantorovitch's theorem in several dimensions

Let $f(x) = 0$ be a system with p equations and p variables, with $f \in C^{(2)}(\omega)$ and $\overline{U_\varepsilon}(x_0) = \{\|x - x_0\| \leq \varepsilon\} \subset \omega$.
If:

1. the Jacobian matrix $W(x) = [\frac{\partial f_i}{\partial x_j}]$ for $x = x_0$ has an inverse $\Gamma_0 = W^{-1}$ with $\|\Gamma_0\| \leq A_0$;
2. $\|\Gamma_0 f(x_0)\| \leq B_0 \leq \frac{\varepsilon}{2}$;
3. $\sum_{k=1}^{p} |\frac{\partial^2 f_i(x)}{\partial x_j \partial x_k}| \leq C$ for $i, j = 1, 2, ..., p$ and $x \in \overline{U_\varepsilon}(x_0)$;
4. $2pA_0B_0C \leq 1$.

Then, Newton's process:
$x_{n+1} = x_n - W^{-1}(x_n)f(x_n)$ converges to
$x^* = \lim_{n\to\infty} x_n$.

Kantorovitch's theorem in several dimensions

Let $f(x) = 0$ be a system with p equations and p variables, with $f \in C^{(2)}(\omega)$ and $\overline{U_\varepsilon}(x_0) = \{\|x - x_0\| \leq \varepsilon\} \subset \omega$.
If:

1. the Jacobian matrix $W(x) = [\frac{\partial f_i}{\partial x_j}]$ for $x = x_0$ has an inverse $\Gamma_0 = W^{-1}$ with $\|\Gamma_0\| \leq A_0$;
2. $\|\Gamma_0 f(x_0)\| \leq B_0 \leq \frac{\varepsilon}{2}$;
3. $\sum_{k=1}^{p} |\frac{\partial^2 f_i(x)}{\partial x_j \partial x_k}| \leq C$ for $i, j = 1, 2, ..., p$ and $x \in \overline{U_\varepsilon}(x_0)$;
4. $2pA_0 B_0 C \leq 1$.

Then, Newton's process: $x_{n+1} = x_n - W^{-1}(x_n) f(x_n)$ converges to $x^* = \lim_{n\to\infty} x_n$.

- real vectors
- real matrices
- sequences of vectors
- functions of several variables
- partial derivatives

Kantorovitch's theorem in several dimensions

Let $f(x) = 0$ be a system with p equations and p variables, with $f \in C^{(2)}(\omega)$ and $\overline{U_\varepsilon}(x_0) = \{\|x - x_0\| \le \varepsilon\} \subset \omega$.
If:

1. the Jacobian matrix $W(x) = [\frac{\partial f_i}{\partial x_j}]$ for $x = x_0$ has an inverse $\Gamma_0 = W^{-1}$ with $\|\Gamma_0\| \le A_0$;
2. $\|\Gamma_0 f(x_0)\| \le B_0 \le \frac{\varepsilon}{2}$;
3. $\sum_{k=1}^{p} |\frac{\partial^2 f_i(x)}{\partial x_j \partial x_k}| \le C$ for $i, j = 1, 2, ..., p$ and $x \in \overline{U_\varepsilon}(x_0)$;
4. $2pA_0B_0C \le 1$.

Then, Newton's process:
$x_{n+1} = x_n - W^{-1}(x_n)f(x_n)$ converges to $x^* = \lim_{n\to\infty} x_n$.

- real vectors
- real matrices
- sequences of vectors
- functions of several variables
- partial derivatives

Setting up the formalization

We need to talk about

- real numbers
- matrices

We use

- COQ standard library Reals
- SSREFLECT library matrix

Mix SSREFLECT and standard COQ !

Mix SSREFLECT and COQ

in SSREFLECT

- hierarchy of algebraic structures
- abstract matrices, but operations when elements are from a ring

in COQ's Reals library

- real numbers defined by axioms
- ring structure

Dealing with real matrices

- use existing results for SSREFLECT library
- define new concepts: e.g. the norm of a real matrix, matrix sequences and series
- prove properties

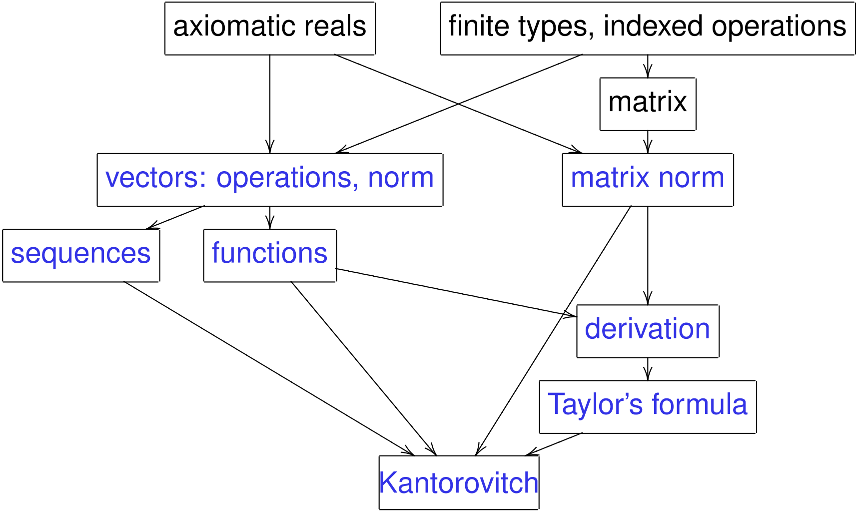
$$\|A\| < 1 \Rightarrow \exists S, \sum_{n=1}^{\infty} A^n = S$$

$$\|A\| < 1 \Rightarrow \det(I_p - A) \neq 0$$

- instantiate to a given norm: e.g. the maximum norm

```
Definition norm_m (A: 'M_(m, n)): R :=
  \big[Rmax/R0]_i \sum_j (Rabs (A i j)).
```

Formal proof in several dimensions



Some Coq code:

```
Theorem kantoroRp_exist: ∃ xs: vec R p,
  conv Xn xs ∧ norm (dif_v xs X0) ≤ 2*b0 ∧ f xs = vect0.
```

What about computations?

What about computations?

- in Coq axiomatic real numbers are appropriate for proofs, but not for computations

- use libraries for exact real arithmetic

Exact real arithmetic with co-inductive streams

Representation

- compute a real number in $[-1, 1]$ with arbitrary precision
- real numbers represented as streams of signed digits in base β
  e.g. $\frac{1}{3} = 0.333\ldots = [\![3::3::3\ldots]\!]_{10} = [\![4::-7::4::-7\ldots]\!]_{10}$
  $$[\![s]\!]_\beta = [\![d_1::d_2::d_3::\ldots]\!]_\beta = \sum_{i=1}^{\infty} \frac{d_i}{\beta^i};\ -\beta < d_i < \beta$$
- notice $[\![d_1::\overline{s}]\!]_\beta = \frac{d_1 + [\![\overline{s}]\!]_\beta}{\beta}$

Exact real arithmetic with co-inductive streams

Representation

- compute a real number in $[-1, 1]$ with arbitrary precision
- real numbers represented as streams of signed digits in base β

  e.g. $\frac{1}{3} = 0.333\ldots = [\![3::3::3\ldots]\!]_{10} = [\![4::-7::4::-7\ldots]\!]_{10}$

  $$[\![s]\!]_\beta = [\![d_1::d_2::d_3::\ldots]\!]_\beta = \sum_{i=1}^{\infty} \frac{d_i}{\beta^i};\ -\beta < d_i < \beta$$

- notice $[\![d_1::\overline{s}]\!]_\beta = \frac{d_1 + [\![\overline{s}]\!]_\beta}{\beta}$
- redundant representation $\rightarrow$ useful for designing algorithms

  e.g. $[\![0::3::\ldots]\!]_{10} + [\![0::6::\ldots]\!]_{10} =?$

  $[\![0::3::3::\ldots]\!]_{10} + [\![0::6::5::\ldots]\!]_{10} = [\![1::-1::\ldots]\!]_{10}$

  $[\![0::3::3::\ldots]\!]_{10} + [\![0::6::7::\ldots]\!]_{10} = [\![1::0::\ldots]\!]_{10}$

Exact real arithmetic with co-inductive streams

Implementation

$$[\![s]\!]_\beta = [\![d_1 :: d_2 :: d_3 :: \ldots]\!]_\beta = [\![d_1 :: \overline{s}]\!]_\beta; \; -\beta < d_i < \beta$$

- in COQ: co-inductive definitions and co-recursive functions

```
CoInductive Stream (A: Type): Type:=
  | Cons: A → Stream A → Stream A.
Notation "x :: s" := Cons x s.

CoFixpoint Sopp (s: Stream digit): Stream digit:=
  match s with | d₁ :: s̄ ⇒ (−d₁) :: Sopp s̄ end.
```

Exact real arithmetic with co-inductive streams

Certification

$$[\![d_1 :: \overline{s}]\!]_\beta = \frac{d_1 + [\![\overline{s}]\!]_\beta}{\beta}$$

- link the exact reals with axiomatic reals

```
Variable β : ℕ.
CoInductive represents: Stream digit → R → Prop :=
  | rep: ∀ s r k, −β < k < β → −1 ≤ r ≤ 1 →
  represents s r → represents (k :: s) (k+r)/β .
Notation " s ≃ r " := represents s r.
```

- certify implementations via this relation

```
Theorem Sopp_correct: ∀ s r, s ≃ r → (Sopp s) ≃ (−r).
```

Implementation of Newton's method

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

Newton on streams

$Sx_0 := s_0$

$Sx_{n+1} := Sx_n \ominus g(Sx_n)$

Newton on axiomatic reals

$Rx_0 := r_0$

$Rx_{n+1} := Rx_n - \frac{f(Rx_n)}{f'(Rx_n)}$

Theorem `Snewt_correct:` *(∗ ... ∗)* $s_0 \simeq r_0 \to g(x) \simeq f(x)/f'(x) \to$
`(Sxn g` s_0 `n)` $\simeq$ `(Rxn f f'` r_0 `n).`

- we can express properties on elements of Newton's sequence

Implementation of Newton's method

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

Newton on streams

$$Sx_0 := s_0$$
$$Sx_{n+1} := Sx_n \ominus g(Sx_n)$$

Newton on axiomatic reals

$$Rx_0 := r_0$$
$$Rx_{n+1} := Rx_n - \frac{f(Rx_n)}{f'(Rx_n)}$$

Theorem Snewt_correct: $(* \ \ldots \ *)$ $s_0 \simeq r_0 \rightarrow g(x) \simeq f(x)/f'(x) \rightarrow$
$(\mathrm{Sxn}\ g\ s_0\ n) \simeq (\mathrm{Rxn}\ f\ f'\ r_0\ n)$.

- we can express properties on elements of Newton's sequence
- but, we cannot reason about the root of the function
- we want to compute the root in arbitrary precision

Newton for streams

Goal define a co-recursive algorithm to compute the root x^* of the function f

- produce the first digit
- use a guarded co-recursive call

Newton for streams

Goal define a co-recursive algorithm to compute the root x^* of the function f

- produce the first digit
- use a guarded co-recursive call

Idea

- start with f and x_0
- speed of convergence $\Rightarrow$ n s.t. $x_n = \frac{d_1 + \overline{x_n}}{\beta} \Rightarrow x^* = \frac{d_1 + \overline{x^*}}{\beta}$

Newton for streams

Goal define a co-recursive algorithm to compute the root x^* of the function f

- produce the first digit
- use a guarded co-recursive call

Idea

- start with f and x_0
- speed of convergence $\Rightarrow$ n s.t. $x_n = \frac{d_1 + \overline{x_n}}{\beta} \Rightarrow x^* = \frac{d_1 + \overline{x^*}}{\beta}$
- $f(x^*) = 0 \Rightarrow f(\frac{d_1 + \overline{x^*}}{\beta}) = 0$
- define $f_1(x) := f(\frac{d_1 + x}{\beta}) \Rightarrow f_1(\overline{x^*}) = 0$
- repeat process to get the first digit of $\overline{x^*}$; start with f_1 and $\overline{x_n}$

Newton for streams

Goal define a co-recursive algorithm to compute the root x^* of the function f

- produce the first digit
- use a guarded co-recursive call

Idea

- start with f and x_0
- speed of convergence $\Rightarrow n$ s.t. $x_n = \frac{d_1 + \overline{x_n}}{\beta} \Rightarrow x^* = \frac{d_1 + \overline{x^*}}{\beta}$
- $f(x^*) = 0 \Rightarrow f(\frac{d_1 + \overline{x^*}}{\beta}) = 0$
- define $f_1(x) := f(\frac{d_1 + x}{\beta}) \Rightarrow f_1(\overline{x^*}) = 0$
- repeat process to get the first digit of $\overline{x^*}$; start with f_1 and $\overline{x_n}$
- $g = \frac{f}{f'} \Rightarrow g_1(x) := \frac{f_1(x)}{f_1'(x)} = \frac{f(\frac{d_1+x}{\beta})}{\frac{1}{\beta} f'(\frac{d_1+x}{\beta})} = \beta \times g(\frac{d_1+x}{\beta})$

Newton for streams

Idea

- to produce a first digit of x^* determine $x_n = \frac{d_1 + \overline{x_n}}{\beta}$ s.t. $x^* = \frac{d_1 + \overline{x^*}}{\beta}$
- do a co-recursive call with function $g_1(x) = \beta \times g(\frac{d_1 + x}{\beta})$ and $\overline{x_n}$

Algorithm

```
CoFixpoint exact_newton g s0 n:=
  match (make_digit (Sxn g s0 n)) with
    | d1 :: s̄n ⇒ d1 :: exact_newton (fun s ⇒ (β ⊙ g(d1 :: s))) s̄n n
  end.
```

Newton for streams

Idea

- to produce a first digit of x^* determine $x_n = \frac{d_1 + \overline{x_n}}{\beta}$ s.t. $x^* = \frac{d_1 + \overline{x^*}}{\beta}$
- do a co-recursive call with function $g_1(x) = \beta \times g(\frac{d_1 + x}{\beta})$ and $\overline{x_n}$

Algorithm

```
CoFixpoint exact_newton g s₀ n:=
  match (make_digit (Sxn g s₀ n)) with
    | d₁ :: s̄ₙ ⇒ d₁ :: exact_newton (fun s ⇒ (β ⊙ g(d₁ :: s))) s̄ₙ n
  end.

Theorem exact_newton_correct: (* ... *)
  (exact_newton g s₀ n) ≃ x*.
```

- ensure the same hypotheses for $\overline{x_n}$ and g_1 as for x_0 and g

Computation with rounding

- computations on machines are inexact
- modify Newton's method the method to include rounding

$$x_0 \qquad\qquad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$t_0 = x_0 \qquad\qquad t_{n+1} = rnd_{n+1}\left(t_n - \frac{f(t_n)}{f'(t_n)}\right)$$

Certified rounding

prove that $t_n \to x^*$

use local stability: $\forall x_0' \in U_{x_0}, x_n(x_0') \to x^*$

- $x_n(x_0)$: $x_0, x_1, x_2, x_3, \ldots \to x^*$
- $x_n(x_1)$: $x_1, x_2, x_3 \ldots \to x^*$
- $x_n(\widetilde{x}_1)$: $\widetilde{x}_1, \widetilde{x}_2, \widetilde{x}_3 \ldots \to x^*$
- $x_n(\widetilde{x}_2)$: $\widetilde{x}_2, \widetilde{x}_3 \ldots \to x^*$
- $x_n(\widetilde{\widetilde{x}}_2)$: $\widetilde{\widetilde{x}}_2, \widetilde{\widetilde{x}}_3 \ldots \to x^*$
- $\ldots$

Certified rounding

prove that $t_n \to x^*$

use local stability: $\forall x_0' \in U_{x_0}, x_n(x_0') \to x^*$

- $x_n(x_0)$: $x_0, x_1, x_2, x_3, \ldots \to x^*$ $\qquad x_0 = t_0$
- $x_n(x_1)$: $x_1, x_2, x_3 \ldots \to x^*$
- $x_n(\widetilde{x}_1)$: $\widetilde{x}_1, \widetilde{x}_2, \widetilde{x}_3 \ldots \to x^*$ $\qquad \widetilde{x}_1 = t_1$
- $x_n(\widetilde{x}_2)$: $\widetilde{x}_2, \widetilde{x}_3 \ldots \to x^*$
- $x_n(\widetilde{\widetilde{x}}_2)$: $\widetilde{\widetilde{x}}_2, \widetilde{\widetilde{x}}_3 \ldots \to x^*$ $\qquad \widetilde{\widetilde{x}}_2 = t_2$
- $\ldots$

Newton with rounding

Let $f :]a, b[\to \mathbb{R}$ and x_0 satisfying the conditions in Kantorovitch's theorem and let $rnd : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$. If

1. $\forall n \forall x, x \in]a, b[\Rightarrow rnd_n(x) \in]a, b[$
2. $\frac{1}{2} \leq \mu_0 < 1$
3. $[x_0 - 3B_0, x_0 + 3B_0] \subset]a, b[$
4. $\forall n \forall x, |x - rnd_n(x)| \leq \frac{1}{3^n} R_0$, where $R_0 = \frac{1-\mu_0^2}{8\mu_0} B_0$

then, for the perturbed Newton's sequence

$$t_0 = x_0 \quad \text{and} \quad t_{n+1} = rnd_{n+1}(t_n - f(t_n)/f'(t_n))$$

a. the sequence $\{t_n\}_{n\in\mathbb{N}}$ converges and $\lim_{n\to\infty} t_n = x^*$ where x^* is the root of the function f given by Kantorovitch's theorem

b. $\forall n, |x^* - t_n| \leq \frac{1}{2^{n-1}} B_0$

Formalization of a numerical method

1. formalize the necessary mathematical theories
2. prove the theorems
3. handle computations
4. handle optimizations

Another formal study

Interval arithmetic

- tool used to handle inaccuracies in computations.

$$-\pi * \sqrt{2} \approx -3.14 * 1.41 = -4.4274$$

$$[-3.15, -3.14] * [1.41, 1.42] = [-4.473, -4.4274]$$

Another formal study

Interval arithmetic

- tool used to handle inaccuracies in computations.

$$-\pi * \sqrt{2} \approx -3.14 * 1.41 = -4.4274$$

$$[-3.15, -3.14] * [1.41, 1.42] = [-4.473, -4.4274]$$

- solve systems of linear interval equations

$$\begin{cases} [1,2]x_1 & + & [2,4]x_2 & = [-1,1] \\ [2,4]x_1 & + & [1,2]x_2 & = [1,2] \end{cases}$$

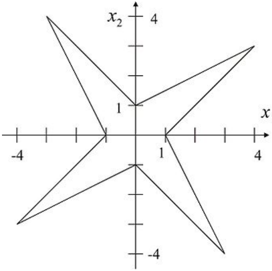
Solving systems of linear interval equations

Two steps:

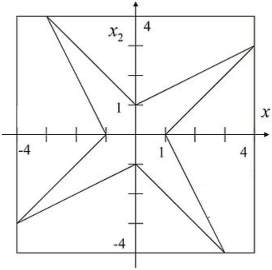
1. checking regularity of the associated interval matrix

2. computing bounds of the solution set

exact solution



bounds for the solution set



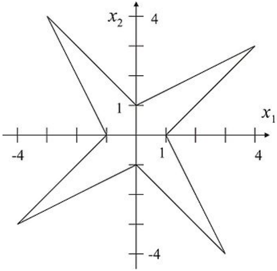
Solving systems of linear interval equations

Two steps:

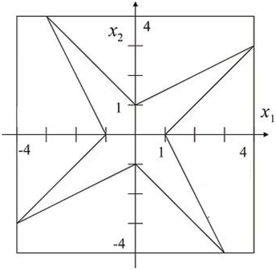
1. checking regularity of the associated interval matrix

2. computing bounds of the solution set

exact solution



bounds for the solution set



Regular interval matrices

$$A = \begin{pmatrix} [2,4] & [-1,1] \\ [-1,1] & [2,4] \end{pmatrix}$$

A is regular iff $\forall \tilde{A} \in A, \det \tilde{A} \neq 0$

$$\tilde{A} = \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{pmatrix}, \quad \begin{matrix} \tilde{A}_{11} \in [2,4], & \tilde{A}_{12} \in [-1,1] \\ \tilde{A}_{21} \in [-1,1], & \tilde{A}_{22} \in [2,4] \end{matrix}$$

$\det \tilde{A} \neq 0$

Criteria for regularity of interval matrices

Criterion

A is regular if and only if $\forall \tilde{x} \in \mathbb{R}^n, 0 \in A\tilde{x} \Rightarrow \tilde{x} = 0$.

Criterion

A is regular if and only if $\forall \tilde{x} \in \mathbb{R}^n, |A_c\tilde{x}| \leq \Delta_A|\tilde{x}| \Rightarrow \tilde{x} = 0$.

Criterion (using positive definiteness)

If the matrix $(A_c^T A_c - \|\Delta_A^T \Delta_A\| I)$ is positive definite for some consistent matrix norm $\|\cdot\|$, then A is regular.

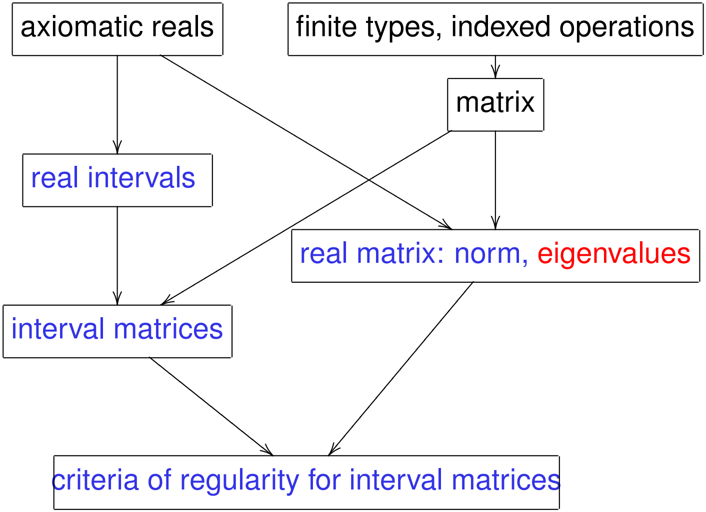
Criterion (using the midpoint inverse)

If the following inequality holds $\rho(|I - RA_c| + |R|\Delta_A) < 1$ for an arbitrary matrix R, then A is regular.

Criterion (using eigenvalues)

If the inequality $\lambda_{max}(\Delta_A^T \Delta_A) < \lambda_{min}(A_c^T A_c)$ holds, then A is regular.

Organization of the formal proof



Work for eigenvalues

spectral radius: $\rho(A) = \max\{|\lambda(A)|\}$

Theorem (Perron Frobenius)

If $A \in \mathbb{R}^{n \times n}$ is nonnegative then the spectral radius $\rho(A)$ is an eigenvalue of A, and there is a real, nonnegative vector $x \neq 0$ with $Ax = \rho(A)x$.

Conclusion

Contributions:

- formalization of mathematical concepts
- two formal studies:
    - Newton's method
    - regularity of interval matrices

Perspectives:

- for interval analysis: study computation
- study for floating point numbers