# Recent Progress in SONIC

**S**olver and **O**ptimizer for **N**onlinear Problems
based on **I**nterval **C**omputation

Elke Just
University of Wuppertal

June 4, 2012

# Contents

# Basic Facts

- Main contributors:
  Thomas Beelitz, Bruno Lang, Elke Just, Paul Willems
  (Applied Computer Science, University of Wuppertal)
  Peer Ueberholz (Hochschule Niederrhein, University of Applied Sciences)

- initial goal: **efficient** and **robust** design of dynamic systems

# Basic Facts

- based on branch-and-bound
- contains rigorous nonlinear solver and (constrained) optimizer
- written in C++
- "generic" interval code provides **performance** and **portability**
  - supported interval libraries: `C-XSC`, `SUN C++`, `filib++`
- parallelization with **OpenMP** and **MPI** available

# Key features of SONIC

- basic branch-and-bound algorithm
- **hybrid subdivision strategy**

  apply subdivision scheme that seems most promising for current box
- **Constraint Propagation** (CP)

  on finite unions of real intervals
- **Taylor refinement**
  - first and second order
  - optionally integrated into CP

# Key features of SONIC (continued)

- **Interval Newton method**

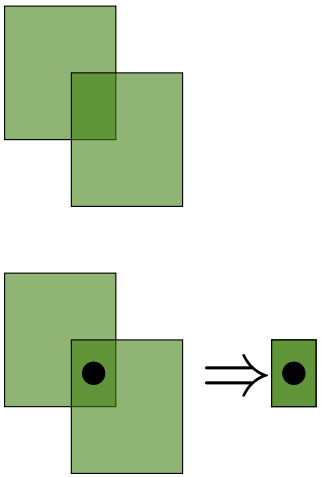
  on a hierarchy of extended systems

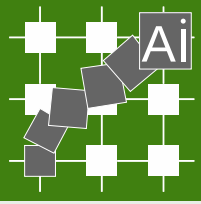
- **Several preconditioners**
  - inverse midpoint preconditioner
  - optimal linear programming preconditioners

- **Verification**
  - Newton
  - Miranda
  - Borsuk
  - Tests based on topological degree

# New features

- solution **boxes can be saved before verification**
  - different methods can be tested without calculating anew
- option for **reducing the number of solution boxes**
  - boxes containing unique tested for intersections (may occur due to epsilon-inflation)
  - intersections tested for uniqueness
  - but: test for intersections is expensive
    $\Rightarrow$ not used in default settings



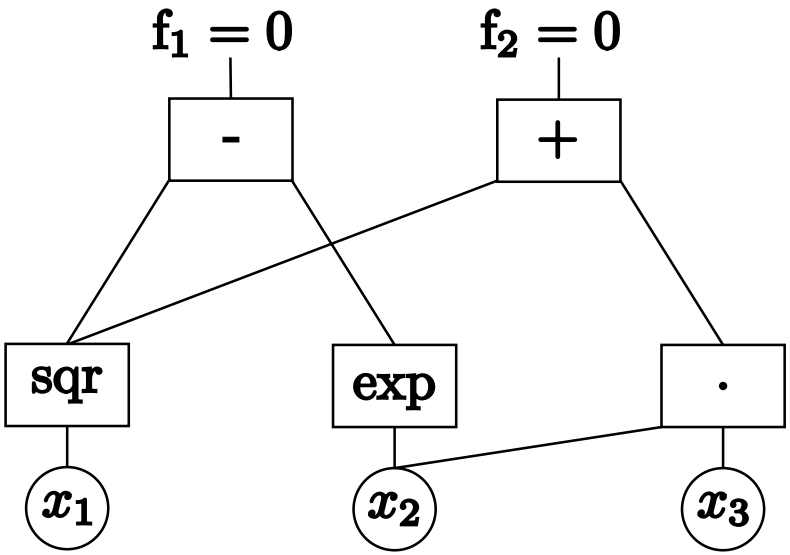



# Extended systems (split levels)

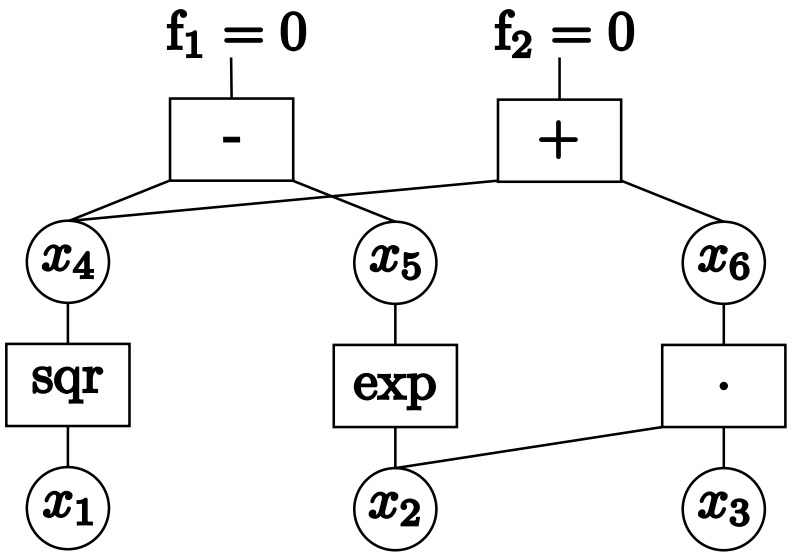
- **decomposition of arithmetic expressions**
- using intermediate variables
- results in another termnet

# Extended systems - example

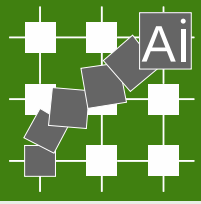
**System:** $f_1(x_1, x_2, x_3) = x_1^2 - e^{x_2}, \quad f_2(x_1, x_2, x_3) = x_1^2 + x_2 \cdot x_3$




(a) original termnet




(b) extended system (fullsplit)

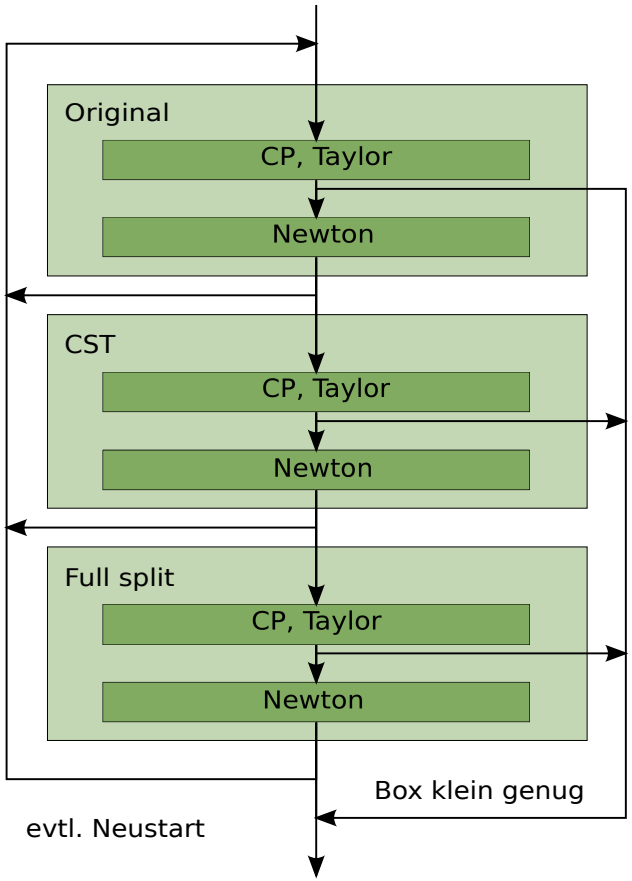



# Extended systems

- can be used for CP and Interval Newton
- default in SONIC:
  - **original**
  - **CST** (common subterms)
  - **fullsplit**
- more extended systems have been discussed in [Wil04]
- but: larger systems are usually also more costly !

# Old heuristic

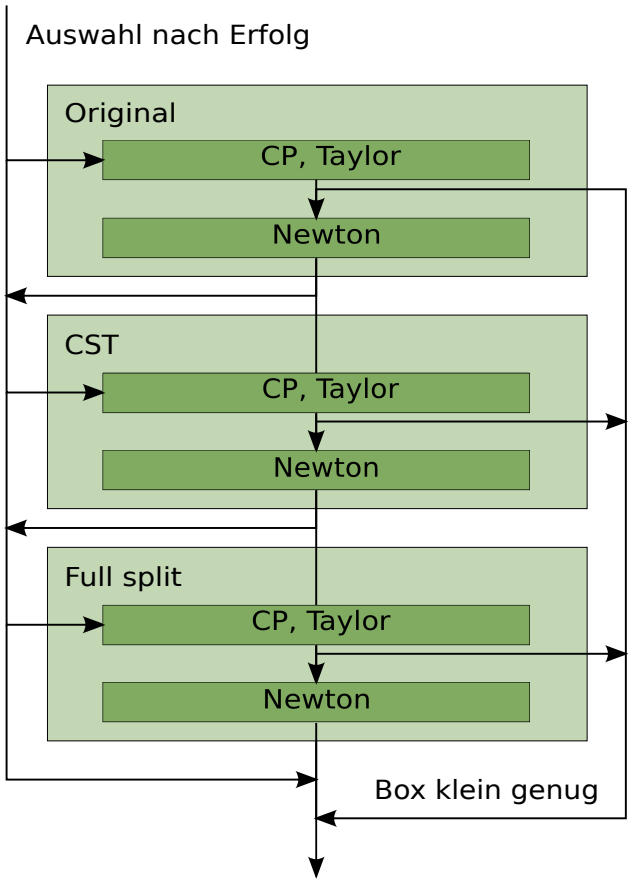
- uses original, CST, fullsplit in this order
- **restart** with original when box was contracted by a certain percentage
- **stop** when
  - box is small enough or
  - a certain number of tests has been applied



# New heuristic

- uses also original, CST, fullsplit
- **no restarts**
- systems are used depending on their **success**
  - success= time * ratio of contraction
  - all systems are used in the beginning
  - success is averaged over previous runs
  - stored in box for each level

More details can be found in [JL12].



# Heuristics in comparison

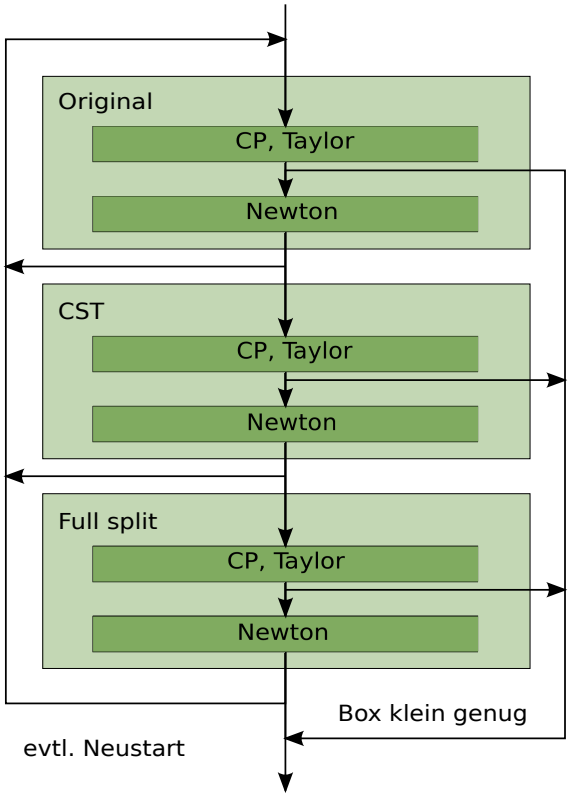



Figure: old heuristic

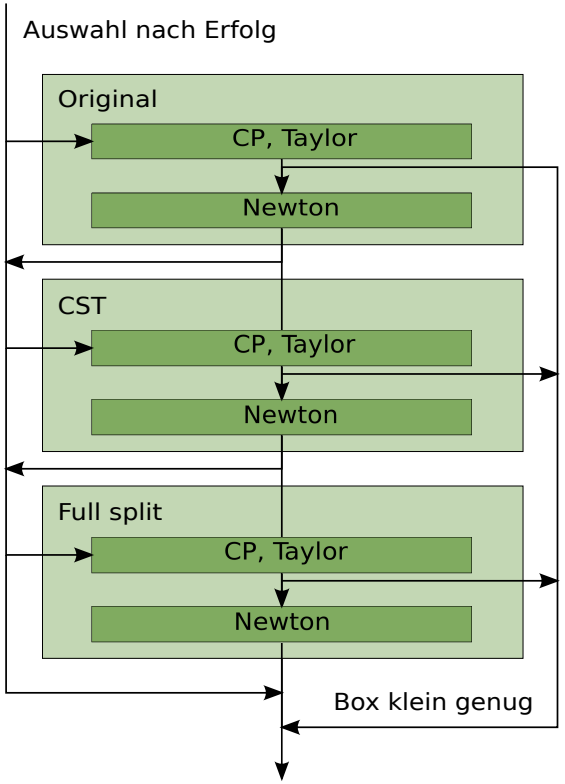
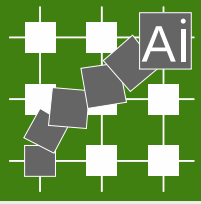


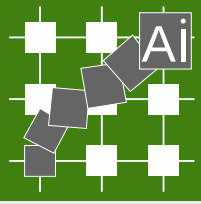

Figure: new heuristic




# Heuristics in comparison

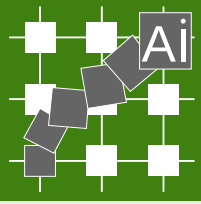
| box number | | time (in s) | |
|---|---|---|---|
| old | new | old | new |
| 51851 | 55421 | 79.09 | 29.28 |
| 1385 | 1405 | 105.57 | 31.73 |
| 18569 | 31751 | 132.70 | 133.87 |
| 49195 | 49201 | 185.85 | 88.16 |
| 27949 | 44497 | 195.02 | 112.15 |
| 11523 | 17375 | 446.28 | 79.67 |
| 68185 | 68699 | 1894.40 | 1408.86 |
| 2959459 | 2976919 | 32902.30 | 13886.70 |

Arithmetic mean of ratios of boxes: 1.17 (14 test problems)
Arithmetic mean of ratios of times: 0.53




# More extended systems

- which systems are constructed and used can be set in the code
- also “handmade” extended systems are possible
- idea: choosing systems to be used in heuristics by number of variables
  - tests were not encouraging
  - one reason: not all variables of larger systems are used in the systems
    (of course term structure is important too)




# Optimal preconditioners

CWLP preconditoner: width-optimal contraction preconditioner by Kearfott ([Kea96])

- can be computed rowwise
- optimal linear programming problems
- really expensive for large extended systems

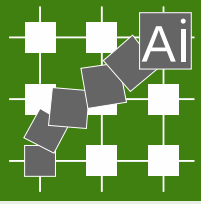
Ansatz: do not use for all rows/variables in the extended systems

# Choosing variables - before

CWLP preconditoner is expensive
$\Rightarrow$ not used for all variables in the extended systems

Heuristics

- original variables are always considered
- for further variables: use stepsize depending on number of variables
  ($\Rightarrow$ some variables are never used)
- option available for choosing first variable randomly




# Choosing variables - first findings

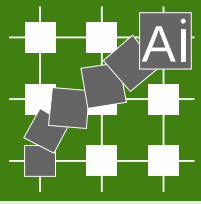
Tests showed

- using all variables (and all preconditioners)
  - can get really expensive in runtime (esp. for large systems)
  - can lead to significantly smaller number of boxes needed
  - some numbers:

    | problem | time factor | box factor | |
    |---|---|---|---|
    | `Trigonometric` | 2.60 | 0.24 | (compared with |
    | `Brent7` | 94.4 | 0.42 | |

    deterministic hierarchy of extended systems)

- contraction in variables differs widely




# Choosing variables - first findings

- **first variable** chosen **random** or **traversed**
  - seems to have no significant influence on runtime
- considering **less variables** in deeper bisection levels

  (first variable = max( 1, n - (bisection level mod n)))
  - time factor 0.6 to 2.0
  - time changes maybe only due to new heuristics

# Choosing variables - upcoming studies

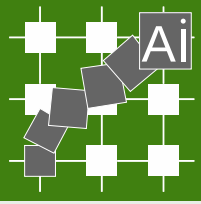
- stepsize depending on recursion depth
- choose variables to consider with regard to Jacobian

# Choosing variables - vision

Can the success in contracting a variable in (preconditioned) Newton be predicted by the structure of the system?

We could then

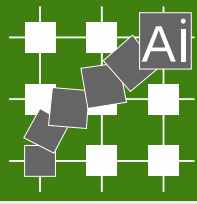
- choose the "right" variables
- save huge amounts of time
- construct extended systems designed to contain many promising variables




# Choosing variables - implementational details

Already possible: **marking** variables to be used in Newton

- all used in CP, less in Newton
- marked when systems are constructed
- can be done according to term structure




# Reusing preconditioner information

Second approach: reuse preconditioner rows to save computation time

- maybe not best information - but cheap
- tests show: can save us time
  (tested without using new heuristic, stepsize)
- but: much data has to be stored and forwarded
- all preconditioner rows calculated only once in the beginning of branch-and-bound
  $\Rightarrow$ also expensive in time for numerous variables

# Reusing preconditioner information - test results

| | box number | | time (in s) | |
|---|---|---|---|---|
| problem | using all variables | saved precond. | using all variables | saved precond. |
| Brent7 | 23767 | 43671 | 2990 | 231 |
| Eco9 | 25949 | 27087 | 9011 | 227 |
| Trigexp1 | 417 | 983 | 10902 | 1377 |
| Trigonometric | 7431 | 8861 | 374 | 298 |
| DesignProblem9 | 10963 | 21051 | 3071 | 412 |

(all problems shown here can be found at [COP])

# Reusing preconditioner information - comparisons

In comparison to calculation of **all preconditioners for all variables**:

- less expensive in time
- more boxes needed

In comparison to **new hierarchy with stepsize** for variables:

- more expensive in time
- usually better with respect to number of boxes

# Further plans and ideas

- using Taylor models for better enclosures
- general acceleration of SONIC for more automatic proofs of spherical designs
- GUI
- calculation of optimal preconditioner using sparse structure of extended systems
- further research, which structures may “predict” contraction success in Newton

# Contact

For further questions or suggestions write to

**just@math.uni-wuppertal.de**

COPRIN-Homepage.

Elke Just and Bruno Lang.
Success-guided selection of expanded systems for result-verifying nonlinear solvers.
*Reliable Computing*, 16:73–83, 2012.

R. Baker Kearfott.
*Rigorous Global Search: Continuous Problems*.
Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, 1996.

Paul Willems.
Symbolisch-numerische Techniken zur verifizierten Lösung nichtlinearer Gleichungssysteme.
Diploma thesis, RWTH Aachen University, 2004.