Relaxed Algorithmic Differentiation

Neil Kichler, Uwe Naumann CSE25 Wednesday 5th March, 2025

Algorithmic Differentiation (AD)

Computes tangents (forward-mode) or adjoints (reverse-mode) for

$$F^* : \mathbb{R}^n \to \mathbb{R}^m$$

with machine accuracy **at a point** x.

This talk:

Get derivative information over a compact **range of inputs**.

*implemented as a computer program

Relaxations

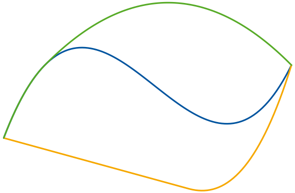
A deterministic way to bound outputs of a function.

Interval Arithmetic:

- Box with lower/upperbound in range

McCormick Relaxations:

- Interval +
- Convex/Concave relaxation



Relaxation of nonconvex, nonconcave function.

Why relax?

- Uncertainty quantification
- Verified computing
- Constraint propagation/satisfaction
- Nonconvex optimization:
  - Subdomain separability [1]
  - Lower bounding in (deterministic) global optimization [2]

Outline

1. Interval Arithmetic
2. McCormick Relaxations
3. AD $\leftrightarrow$ Relaxations
4. Examples
5. Conclusion

Interval Arithmetic

Interval Arithmetic

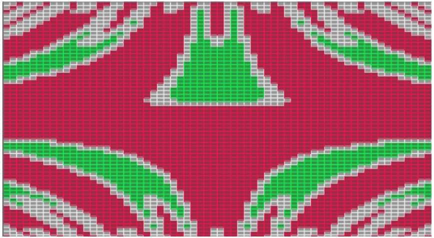
Computes function bounds:

Replace $x \in \mathbb{R}$ with $X \in \mathbb{IR}$:

$$\mathbb{IR} := \{[a, b] \mid a \leq b \wedge a, b \in \overline{\mathbb{R}}\},$$
$$X = [a, b] := \{x \in \mathbb{R} \mid a \leq x \leq b\}.$$

$\hookrightarrow$ interval extension:

$$F(X) \supseteq \{f(x) \mid x \in X\}.$$



Refinement of: $\sin(\cos(xy)y) > 0.5$

Fundamental Theorem of Interval Arithmetic

The interval extension $F : \mathbb{IR}^n \rightarrow \mathbb{IR}$ of $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is guaranteed to enclose the range of f over the inputs in $X = (X_0, \cdots, X_n)$, i.e., $\mathrm{range}(f) \subseteq F(X)$ [3].

Interval Arithmetic

IA has fundamental shortcomings:

- Dependency problem: E.g., $x^2 - 4x$ vs. $(x - 2)^2 - 4$.
  $\hookrightarrow$ partially addressable through symbolic rewriting.
- Wrapping Effect: fix would require different arithmetic.
- Limited by hardware intrinsics accuracy and rounding support.

McCormick Relaxations

McCormick Relaxations

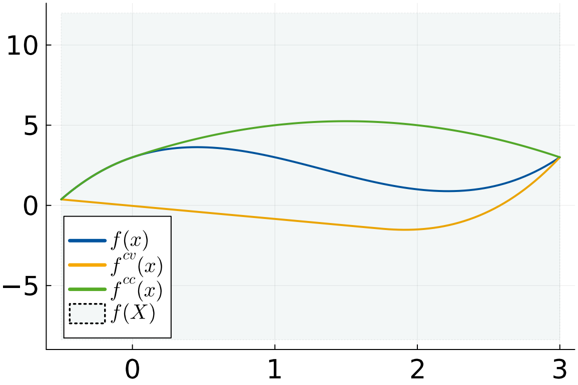
Used in lower bounding of Branch & Bound methods.

Provides:

$f^{cv}(x)$: Convex relaxation at x

$f^{cc}(x)$: Concave relaxation at x

$f(X)$: Interval over X.



McCormick relaxation of $f(x) = (x-1)^3 - x^2 + 4$

McCormick Relaxations from an AD perspective

Both conceptually start with a computational graph. Then:

AD:

- diff. rules of basic ops
- chain rule
- faster derivatives through symbolic AD (e.g., matmul)

McCormick relaxation:

- relaxations of basic ops
- composition rule
- tighter relaxations of special composite functions possible (e.g., $x \log(x)$)

Relaxations more broadly

Many ways to relax:

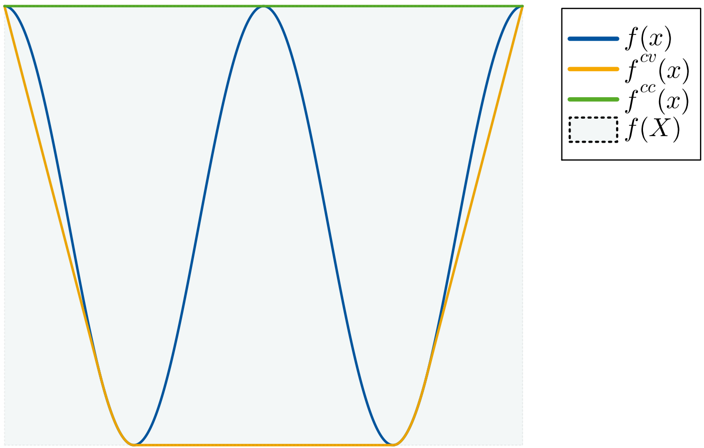
Primitive relaxations via, e.g., αBB using Hessian information.

Aside: Finding the optimal relaxation is again an optimization problem
$\hookrightarrow$ Can be solved by SDP/SOCP.

But: Tradeoff between computation time of relaxation and number of executed branches in Branch & Bound methods.

$\hookrightarrow$ McCormick relaxations form a good compromise

McCormick Elementary Relaxation: Cosine



McCormick Composition Rule

Let $X \subseteq \mathbb{R}^n, Z \subseteq \mathbb{R}$ be nonempty convex sets. For a composite function $g = F \circ f$, where $f : X \to Z$ and $F : Z \to \mathbb{R}$, with known convex relaxations $f^{cv} : X \to \mathbb{R}$, $F^{cv} : Z \to \mathbb{R}$, and concave relaxations $f^{cc} : X \to \mathbb{R}$, $F^{cc} : Z \to \mathbb{R}$, the convex and concave relaxations of g can be computed by

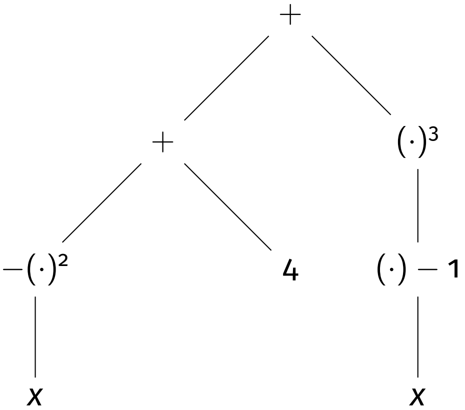
$$g^{cv}(\boldsymbol{x}) = F^{cv}(\mathtt{mid}(f^{cv}(\boldsymbol{x}), f^{cc}(\boldsymbol{x}), z_{\min})), \tag{1}$$

and

$$g^{cc}(\boldsymbol{x}) = F^{cc}(\mathtt{mid}(f^{cv}(\boldsymbol{x}), f^{cc}(\boldsymbol{x}), z_{\max})), \tag{2}$$

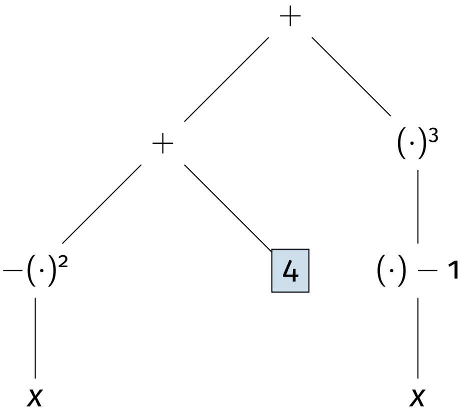
respectively.

McCormick Relaxations: By Example



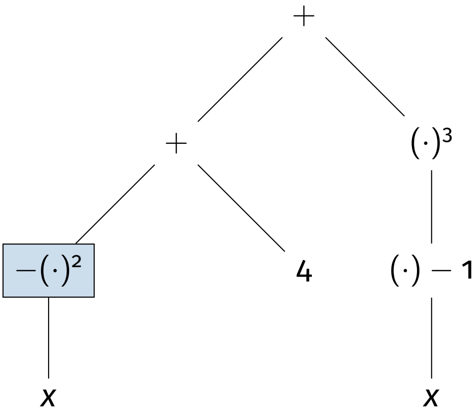
Example: $f(x) = (x-1)^3 - x^2 + 4$ with $x \in [0, 3] \subset \mathbb{R}$.

McCormick Relaxations: Constant



$$f_1(x) = 4 \Rightarrow f_1^{cv}(x) = f_1^{cc}(x) = 4$$

McCormick Relaxations: Concave



$f_2(x) = -x^2 \Rightarrow f_2^{cv}(x) = -3x, \quad f_2^{cc}(x) = -x^2$ (for $x \in [0, 3]$)

McCormick Relaxations: Concave

$f_2(x) = -x^2$ over $x \in [0, 3]$

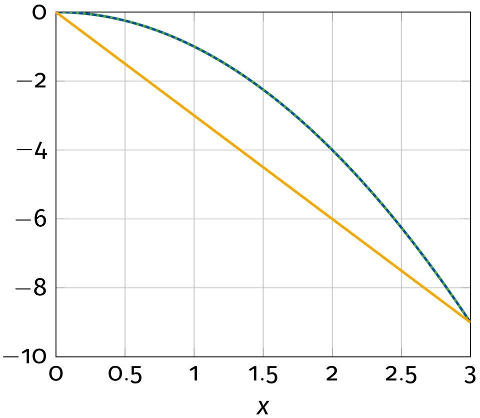
As f_2 is concave $\rightarrow$ compute chord:

$$x^L = 0, \quad x^U = 3$$

$$f_2^{cv}(x)$$

$$= -(x^U)^2 + \frac{-(x^U)^2 - (-(x^L)^2)}{x^U - x^L}(x - x^U)$$
$$= -3x$$

$$f_2^{cc}(x) = -x^2$$



McCormick Relaxations: Add

In general:

$$f_3(x) = f_1(x) + f_2(x)$$

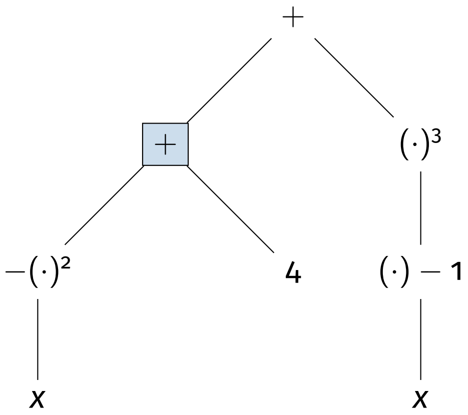
$$f_3^{cv}(x) = f_1^{cv}(x) + f_2^{cv}(x)$$

$$f_3^{cc}(x) = f_1^{cc}(x) + f_2^{cc}(x)$$

So:

$$f_3^{cv}(x) = -3x + 4$$

$$f_3^{cc}(x) = -x^2 + 4$$



McCormick Relaxations: Subtract

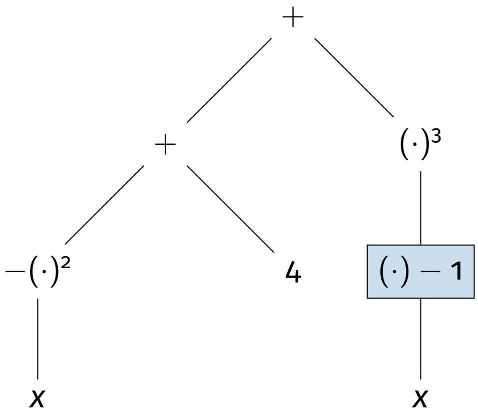
As

$$f_4(x) = x - 1$$

is affine:

$$f_4^{cv}(x) = x - 1$$

$$f_4^{cc}(x) = x - 1$$

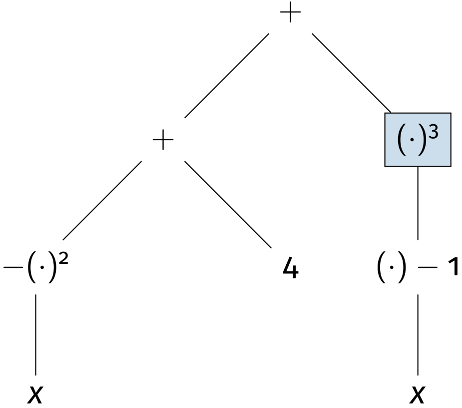


McCormick Relaxations: Cubed

Let

$$f_5(x) = x^3,$$

for $x \in [-1, 2]$.



McCormick Composition

$$g^{cv}(x) = F^{cv}(\text{mid}\{f^{cv}(x), f^{cc}(x), z^{min}\})$$
$$g^{cc}(x) = F^{cc}(\text{mid}\{f^{cv}(x), f^{cc}(x), z^{max}\})$$

See Desmos.

McCormick Relaxations: Result

Back to

$$f(x) = (x-1)^3 + (-x^2+4)$$

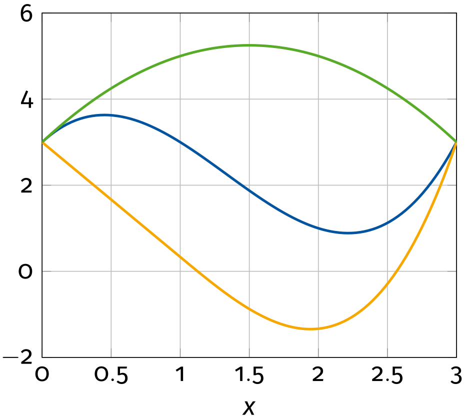
for $x \in [0,3]$.

Relaxations:

$$f^{cv}(x) = f_5^{cv}(x) + f_3^{cv}(x)$$

$$f^{cc}(x) = f_5^{cc}(x) + f_3^{cc}(x)$$

See Desmos.



AD $\leftrightarrow$ Relaxations

Tangent AD $\leftrightarrow$ McCormick relaxation

`mccormick<tangent<T>>`	`tangent<mccormick<T>>`
Tangent of McCormick relaxation	McCormick relaxation of Tangent
↓	↓
Linearized relaxations	Nonlocal derivative information

Adjoint AD $\leftrightarrow$ McCormick relaxation

`mccormick<adjoint<T>>`	`adjoint<mccormick<T>>`
Adjoint of McCormick relaxation [4]	McCormick relaxation of Adjoint
↓	↓
Linearized relaxations	Non-local derivative information

Tightness of non-local derivative bounds

`ad<relaxation<T>>` provides enclosures of derivatives that contain all possible values of the derivative over the specified domain.

Tightness?

How many domain splits are required to approach true bound?

- Interval: converges linearly
- McCormick: converges quadratically

What can we do with relaxed AD?

Example: Subdomain Separability [1]

Function f is partially separable, if:

$$f(x) = \sum_{i=1}^{p} f^{[i]}(x^{[i]}).$$

Optimization can simplify from:

$$\min_{x} f(x),$$

to:

$$\sum_{j=1}^{p} \min_{x^{[j]}} f^{[j]}(x^{[j]}).$$

Example: Subdomain Separability [1]

But, consider:

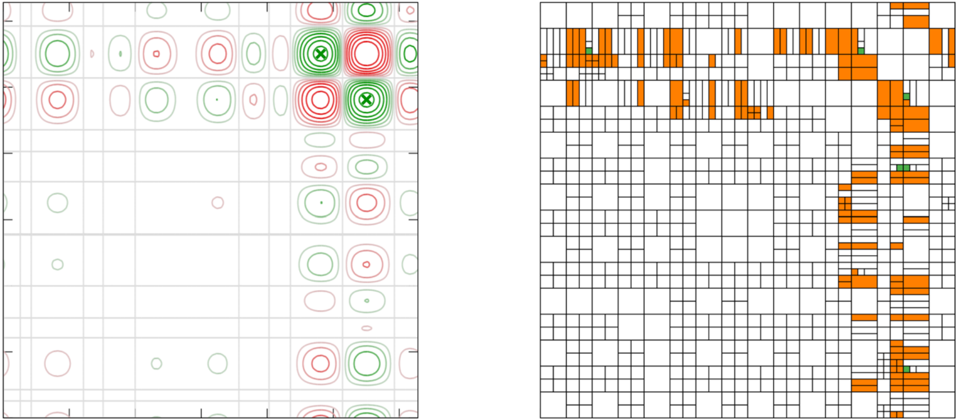
$$f(x) = -\exp(-\frac{1}{2}\sum_{i=1}^{n} x_i^2)$$

Not partially seperable in the typical sense.
$\hookrightarrow$ structural separability correctly identifies decomposibility of f.

How?

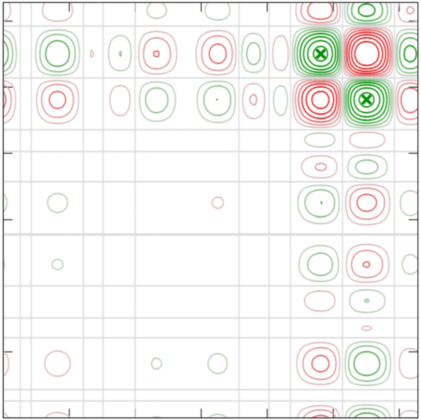
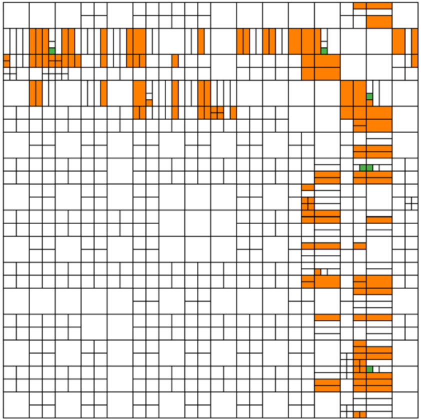
Monotonicity test of parts of computational graph using relaxed AD.

Example: Subdomain Separability [1]



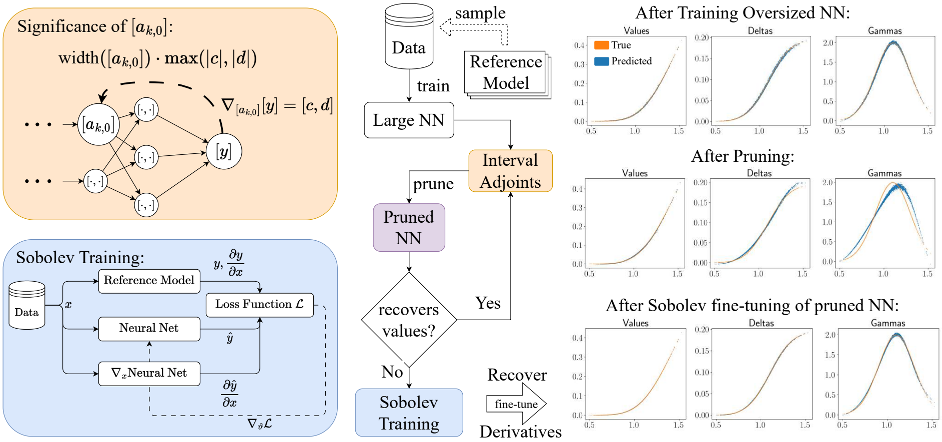
B&B optimization of Shubert function with cutoff and monotonicity tests.

Example: Subdomain Separability [1]

More than 40 times fewer B&B nodes generated by otherwise same algorithm.

Example: Sobolev pruning [5]



Relaxed AD implementation

- Directly possible for operator-overloading based tools [†].
- Source-to-Source needs understanding of McCormick relaxations:
  $\hookrightarrow$ Could be implemented as another pass, similar in style to AD.

[†] modulo details of OO-tool, like passive values etc.

Existing/Upcoming Tools

CPU:

- `dco/c++` with boost-interval/MC++ (or any other operator-overloading based AD-tool)
- `MC++`
- McCormick.jl

GPU:

- `CuInterval`
- `CuMcCormick`
- `CuTangent`

Conclusion

Takeaway:

- Potential to extend local derivative information to a whole region.
- Relaxations of AD are underexplored.
- Both mccormick<ad<T>> and ad<mccormick<T>> useful concepts.
- Development for GPUs ongoing.

Slides online:

References

[1] Jens Deussen and Uwe Naumann. “Subdomain separability in global optimization”. In: *Journal of Global Optimization* 86.3 (2023), pp. 573–588.

[2] Dominik Bongartz et al. “Deterministic global optimization of steam cycles using the IAPWS-IF97 model”. In: *Optimization & Engineering* 21 (2020), pp. 1095–1131.

[3] Ramon E. Moore et al. *Introduction to Interval Analysis*. Society for Industrial and Applied Mathematics, 2009.

[4] Markus Beckers et al. “Adjoint Mode Computation of Subgradients for McCormick Relaxations”. In: *Recent Advances in Algorithmic Differentiation*. 2012, pp. 103–113. ISBN: 978-3-642-30023-3.

[5] Neil Kichler et al. “Towards Sobolev Pruning”. In: *Proceedings of the Platform for Advanced Scientific Computing Conference*. PASC ’24. Zurich, Switzerland: Association for Computing Machinery, 2024. ISBN: 9798400706394.

Backup Slides

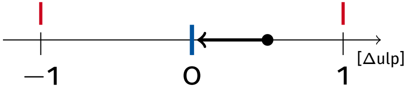
More on Interval Arithmetic Rounding

We use the CUDA intrinsic functions which use round-to-nearest-even.

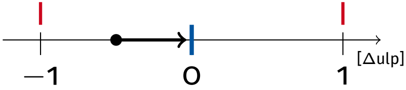
- if halway between two floating point numbers $\rightarrow$ pick even.
- else $\rightarrow$ pick nearest.
- Error of 1 ulp in intrinsic results in 2.5 ulp max error for interval lower & upper bound, i.e. for lower bound: $|\inf(F_{\text{analytic}}(X)) - \inf(F(X))| \leq 2.5$.
- 8 Scenarios:
  - halfway (odd above/below),
  - closer to even/odd (left/right),
  - exact (even/odd).

Interval Arithmetic Rounding

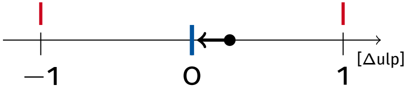
Scenarios given 1 ulp error (round-to-nearest-even):



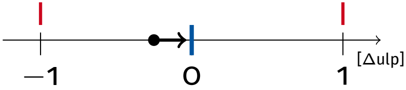
halfway (odd above)



halfway (odd below)



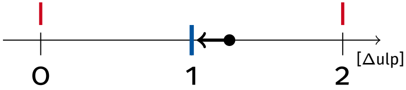
closer to even (left)



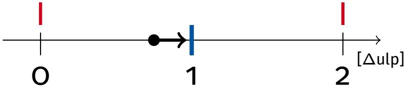
closer to even (right)

Interval Arithmetic Rounding

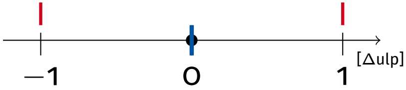
Scenarios given 1 ulp error (round-to-nearest-even):



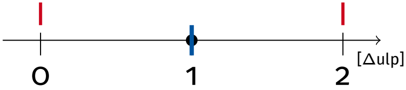
closer to odd (left)



closer to odd (right)



exact (even)



exact (odd)