MC++ (version 1.0)

MC++ provides a collection of C++ classes for bounding factorable functions, including interval bounds, convex/concave relaxations, Taylor model estimators, and spectral bounds. The implementation of MC++ relies on operator/function overloading and templates. Our main emphasis in developing MC++ has been to make the computation of bounds as simple and natural as possible, similar to computing function values in real arithmetics. In particular, we find MC++ most useful for the fast prototyping and testing of new algorithms and ideas, for instance in such areas as global and robust optimisation.

McCormick Relaxation Arithmetic

McCormick’s technique [McCormick, 1976] provides a means for computing pairs of convex/concave relaxations of a multivariate function on interval domains provided that this function is factorable and that the intrinsic univariate functions in its factored form have known convex/concave envelopes or, at least, relaxations.

The class mc::McCormick in MC++ provides an implementation of the McCormick relaxation technique and its recent extensions; see [McCormick, 1976; Scott et al., 2011; Tsoukalas & Mitsos, 2012; Wechsung & Barton, 2013]. mc::McCormick also has the capability to propagate subgradients for these relaxations, which are guaranteed to exist in the interior of the domain of definition of any convex/concave function. This propagation is similar in essence to the forward mode of automatic differentiation; see [Mitsos et al., 2009].
Examples of McCormick relaxations constructed with mc::McCormick are shown on the left plot of Fig B for the factorable function $f(x) = \cos(x^2) \sin(x^{-3})$ for $x \in \left[ \frac{\pi}{6}, \frac{\pi}{3} \right]$. Also shown on the right plot of Fig B are the affine relaxations constructed from a subgradient at $\frac{\pi}{4}$ of the McCormick relaxations of $f$ on $\left[ \frac{\pi}{6}, \frac{\pi}{3} \right]$.

Taylor Model Arithmetic

A $q^{th}$-order Taylor model of a multivariate function $f : \mathbb{R}^n \to \mathbb{R}$ that is (at least) $(q + 1)$-times continuously differentiable on the domain $D$, consists of the $(q + 1)$-order multivariate Taylor polynomial $\mathcal{P}$, expanded around a point $\hat{x} \in D$, plus a remainder term $\mathcal{R}$, so that (Fig C):

$$f(x) \in \mathcal{P}(x - \hat{x}) \oplus \mathcal{R}, \quad \forall x \in D.$$ 

The polynomial part $\mathcal{P}$ is propagated symbolically and accounts for functional dependencies. The remainder term $\mathcal{R}$, on the other hand, is traditionally computed using interval analysis [Neumaier, 2002; Makino & Berz, 2003]; see figure opposite. More generally, convex/concave bounds for the remainder term can be propagated using McCormick relaxation arithmetic, leading to so-called McCormick-Taylor models [Sahloedin & Chachuat, 2011]. Ellipsoidal enclosure too can be computed for the remainder term of vector-valued functions [Houska et al., 2013]. It can be established that the remainder term has convergence order (no less than) $(q + 1)$ with respect to the diameter of the domain set $D$ under sufficient differentiability conditions [Bompadre et al., 2012].
The classes `mc::TModel` and `mc::TVar` provide an implementation of Taylor model arithmetic for factorable functions, together with a few methods for computing bounds on the Taylor model range (multivariate polynomial part). We note that computing exact bounds for multivariate polynomials is a hard problem in general. Instead, a number of computationally tractable, but typically conservative, the following bounding approaches are implemented in `mc::TModel` / `mc::TVar`:

- Bounding every monomial term independently and adding these bounds (naive approach);
- Bounding the first- and diagonal second-order terms exactly and adding bounds for the second-order off-diagonal and higher-order terms computed independently [Lin & Stadtherr, 2007];
- Bounding the terms up to order 2 based on an eigenvalue decomposition of the corresponding Hessian matrix and adding bounds for the higher-order terms computed independently;
- Rewriting the multivariate polynomial in Bernstein form, thereby providing bounds as the minimum/maximum among all Bernstein coefficients [Lin & Rokne, 1995; 1996].

Examples of Taylor and McCormick-Taylor models (blue lines) constructed with `mc::TModel` / `mc::TVar` are shown on the left and right plots of Fig D, respectively, for the factorable function $f(x) = x \exp(-x^2)$ (red line) for $x \in [-0.5, 1]$. Also shown on these plots are the bounds (green lines), either interval or convex/concave bounds, computed from the Taylor models.

**Spectral Bound Arithmetic**

Given a factorable, multivariate function $f : \mathbb{R}^n \to \mathbb{R}$, that is twice-continuously differentiable on a box $X := [x^l, x^u]$, Mönnigmann's technique [Mönnigmann, 2008; 2011] provides a means for computing spectral bounds for its Hessian matrix $H_f(x)$ at any point $x \in X$—without actually computing $H_f$. Applications of this technique are for determining whether a function is convex or concave on a particular domain, as well as for constructing convex/concave relaxations for complete search approaches in global optimisation e.g. as an alternative to McCormick relaxation arithmetic. Alternative techniques for determining spectral bounds include the interval variant of Gershgorin's circle criterion [Adjiman et al., 1998] as well as Hertz & Rohn's method [Hertz, 1992], which rely on an interval enclosure of the set of all possible Hessian matrices $[H_f] \supseteq \{H_f(x) \mid x \in X\}$. 
The class mc::Specbnd in MC++ provides an implementation of eigenvalue arithmetic for factorable functions, along with the ability for computing these bounds directly from the interval extension of $H_f$ using either Gershgorin's circle criterion or Hertz & Rohn's method. As established by [Darup et al., 2012], the spectral bound arithmetic may or may not produce tighter spectral bounds than these latter techniques, depending on the factorable function $f$ at hand and its variable range $X$.

Sample results obtained for the factorable function $f(x, y) = 1 + x - \sin(2x + 3y) - \cos(3x - 5y)$, for $x \in [-0.5, 0.5]^2$ are shown on Fig E. This function (red line) and corresponding interval bounds (blue line) are shown on the left plot, while the minimal and maximal eigenvalues (red and green lines) as well as the computed spectral bounds (blue line) of the Hessian matrix $H_f$ are shown on the right plot.

**Getting and Using MC++**

MC++ is released as open source code under the Eclipse Public License (EPL). We are planning to make the source code available through the COIN-OR repository (the code is currently under review). In the meantime, a link to the pre-release version of MC++ is: [MC++_1.0_distrib_24-07-2013.tgz](file:///C:/temp/MC++_1.0_distrib_24-07-2013.tgz)

MC++ uses a number of third-party libraries. The following ones are mandatory to use all of its features:

- Library FADBAD++ (version 2.1) for automatic differentiation (AD) using forward, backward and Taylor expansion methods
- Libraries BLAS and LAPACK (version 4.3) for linear algebra computations

Optional libraries include:

- Libraries PROFIL and/or FILIB++ for verified interval arithmetic
- CppUnit for unit testing