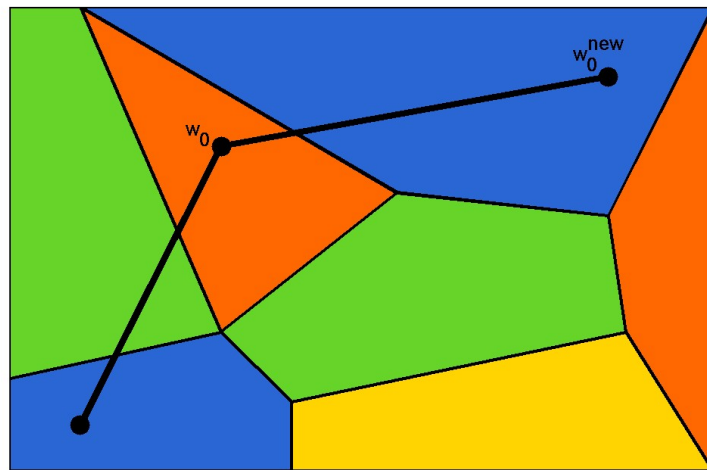# Additional Notes on
qpOASES_e 3.1 (July 2015)

(a plain C, static memory translation of qpOASES 3.1)

Hans Joachim Ferreau et al.[1,2]

ABB Corporate Research, Switzerland

support@qpOASES.org

General Remarks

- This is an almost complete plain C, static memory translation of `qpOASES` 3.1 from February 2015. It might be further maintained in the future.

- This translation aims at providing most of the functionality of `qpOASES` 3.1; see its manual located at <install-dir>/doc/manual.pdf for further information.

- The translation process made necessary several adaptations of the calling syntax of all provided functions. In particular,

    - modified constructor/destructor calls for `QProblem(B)` or `Options` objects,
    - addition of the prefix "`QProblem(B)_`" to all former member functions,
    - addition of a pointer to the corresponding `QProblem(B)` object as first argument to all former member functions,
    - replacement of references by pointers (relevant for the argument "`int* nWSR`" within the `init` and `hotstart` functions),
    - addition of the prefix "`qpOASES_`" to all global constants and utility functions.

  Besides the modified calling syntax, purpose and effect of these functions remain the same.

- `SQProblem` objects are not yet supported.

- Parts of the MATLAB and SIMULINK interfaces have also been transformed to plain C, but no SCILAB, OCTAVE or PYTHON interface is provided.

- The folder <install-dir>/examples contains simple examples to illustrate the usage of the new syntax. Moreover, examples for using the MATLAB and SIMULINK interfaces can be found within <install-dir>/interfaces/matlab and <install-dir>/interfaces/simulink, respectively.

- If you think the translated code is ugly, you are right. It is rather a quick and dirty translation of the existing C++ version than a tidy plain C design.

- If you have further questions concerning `qpOASES_e` 3.1, contact its main author: Hans Joachim Ferreau, support@qpOASES.org.

Known Issues

- All parts of the code need further testing.

- Code has not yet been optimized for speed or object code size. In particular, computation times of `qpOASES_e` 3.1 are up to a factor of two higher(!) than that of `qpOASES` 3.1.

- The use of the keywords `static` and `inline` needs to be reviewed and possibly adapted.