

ADiMat: Automatic Differentiation for MATLAB and Octave

Evaluate derivatives automatically

ADiMat differentiates m-file functions

- augmented m-files are generated
 - in forward mode: for Jacobians
 - in reverse mode: for efficient gradients
- Using OO techniques we obtain
 - vector mode: multiple directional derivatives
 - forward-over-reverse mode: for efficient Hessians

ADiMat support for Octave

ADiMat is designed to be 100% Octave compatible

- generated code is compatible
- all runtime m-files are compatible
- ADiMat has OCT-files for its C++ runtime functions
 - stacks using the RIOS library and
 - (experimental) derivative classes
- classes are written in @folder-syntax

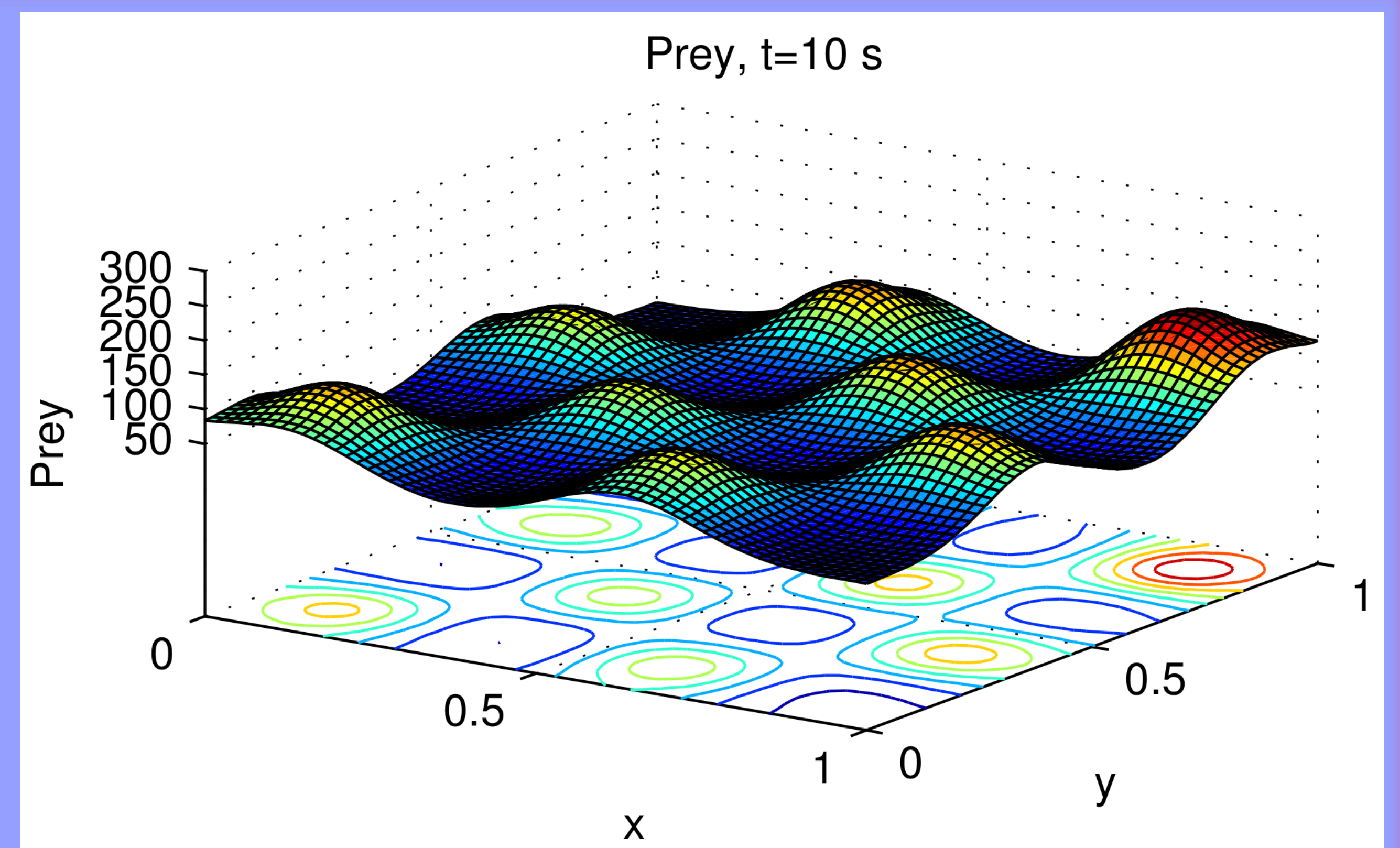
Use case: DASSL/DASPK + forward mode

The DAE solvers DASSL and DASPK available in Octave can make use of derivatives computed with ADiMat.

In the well-known predator-prey index-1 DAE model the Jacobian for DASPK can be computed efficiently in forward mode using compression and a custom seed matrix.

Fig: The plot from the DASPK paper [Brown, Hindmarsh, and Petzold 1999] recreated in Octave using ADiMat to compute the derivatives

J. Willkomm, C. H. Bischof, and H. M. Bücker. The New User Interface of ADiMat and How to Use it with DAE Solvers in Matlab and Octave. *Twelfth European Workshop on Automatic Differentiation*, Humboldt-Universität zu Berlin. December 9, 2011
 Peter N. Brown, Alan C. Hindmarsh, and Linda R. Petzold. Using Krylov Methods in the Solution of Large-Scale Differential Algebraic Systems. *SIAM Journal on Scientific Computing*, Volume 15(6), 1994



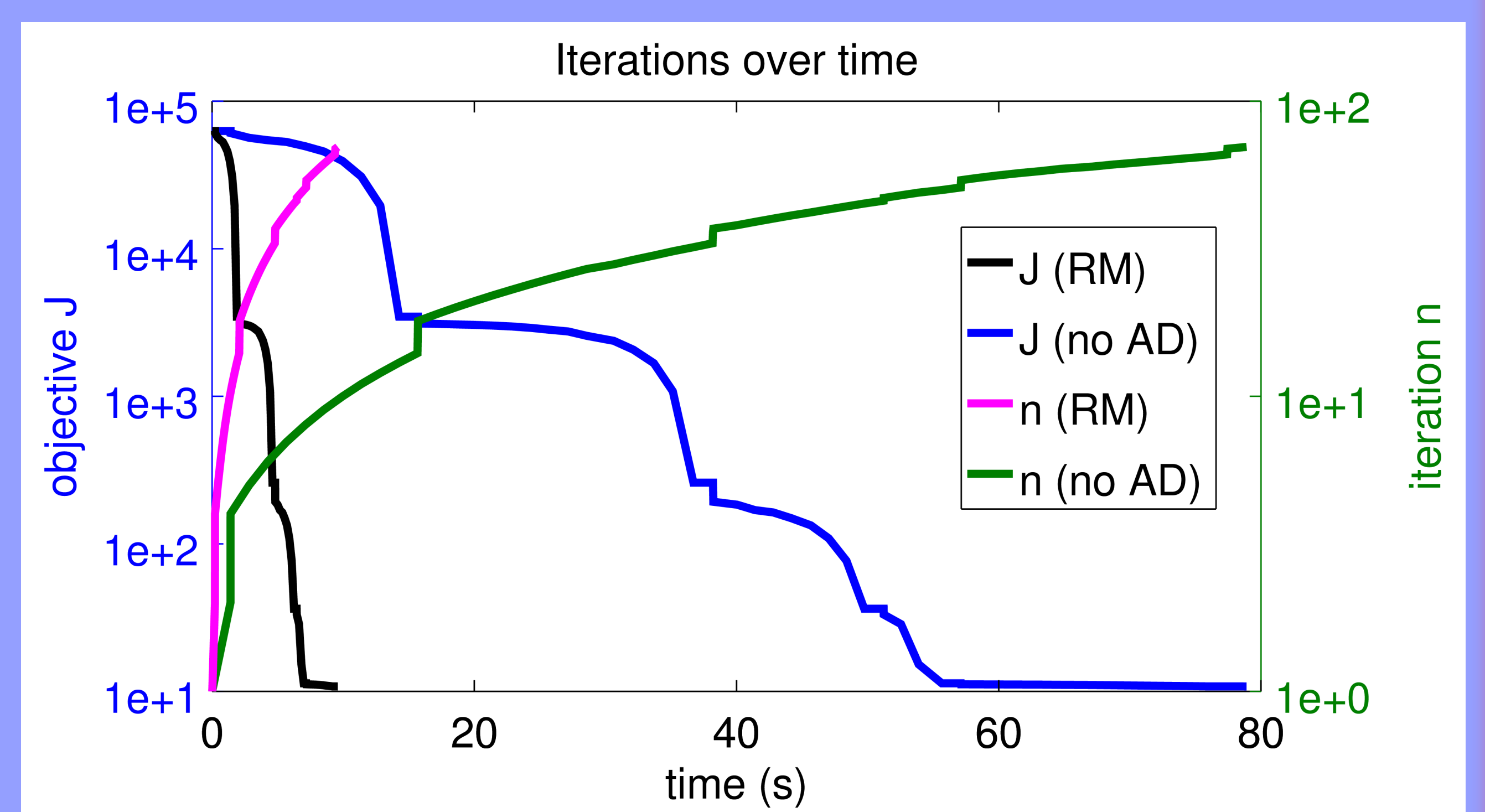
Use case: FMINUNC + reverse mode

The FMINUNC solver contained in the optim package is an ideal candidate to employ the reverse mode of ADiMat: The gradient of the scalar objective function can be computed with $O(1)$ runtime overhead.

Compare this to the $O(n)$ overhead required for forward differentiation and finite differences: The larger the number of inputs n , the greater the advantage of the reverse mode

In this simple test setup FMINUNC solves $\text{norm}(\expm(X)-D)$ for X , where D is a random matrix of order 24 (hence $n=576$).

Fig: Iteration history of the FMINUNC solver using reverse mode derivatives from ADiMat (RM) and no user provided derivatives (no-AD).



RIOS: Efficient I/O for large stacks

RIOS is an open source library for efficiently writing large stack files and reading them backwards, using buffered and asynchronous I/O.

In the reverse mode, push and pop operations in the augmented code can generate very large traces of values which can easily exceed the main memory.

RIOS is used by ADiMat to write such large streams to disk or other secondary storage. The asynchronous operation mitigates the storage latency as the file operations are interleaved with the main computations.

J. Willkomm, C. H. Bischof, and H. M. Bücker. RIOS: Efficient I/O in Reverse Direction. *Software: Practices and Experiences*, 45(10):1399-1427, 2015.

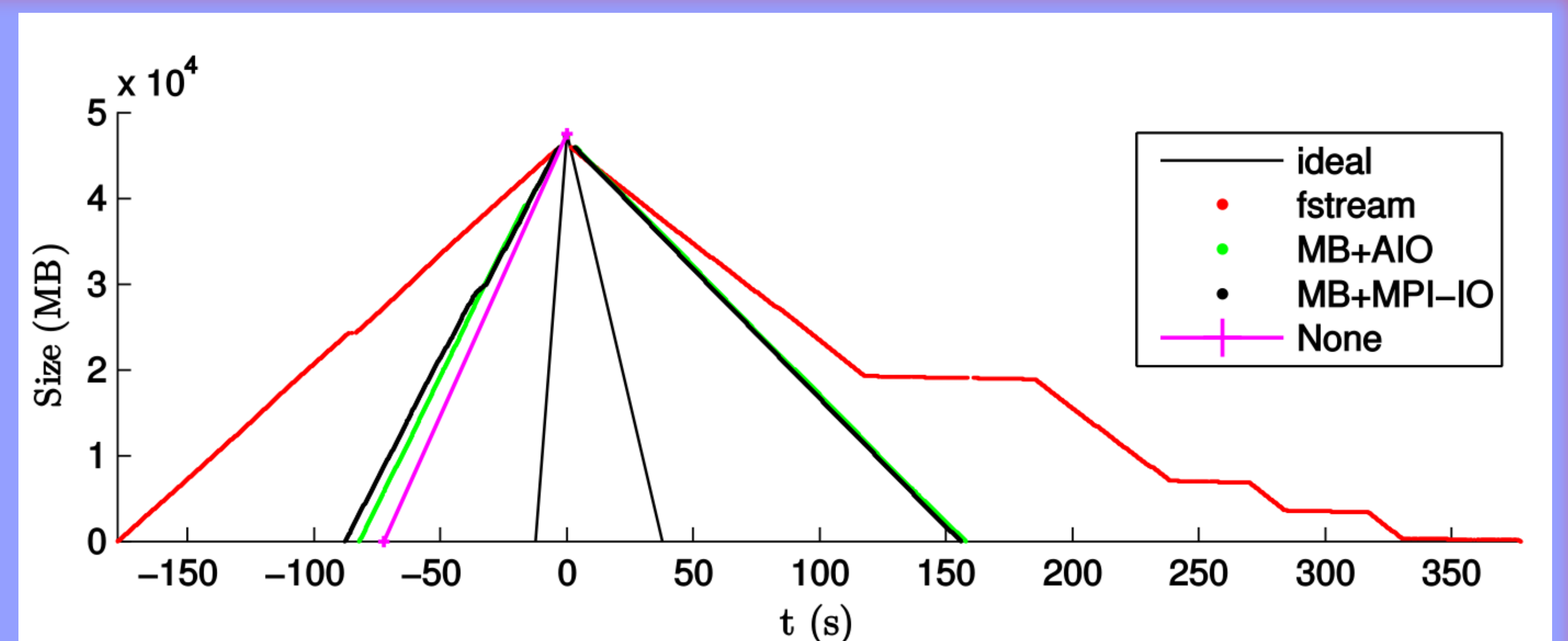


Fig: The stack history of a program differentiated by ADiMat in reverse mode using various I/O streams: C++ std::fstream, RIOS using AIO, and RIOS using MPICH MPI-IO.