FAST JACOBIANS AND HESSIANS BY LEVERAGING SPARSITY

An Illustrated Guide to Automatic Sparse Differentiation

Adrian Hill[1,2], Guillaume Dalle[3] and Alexis Montoison[4]

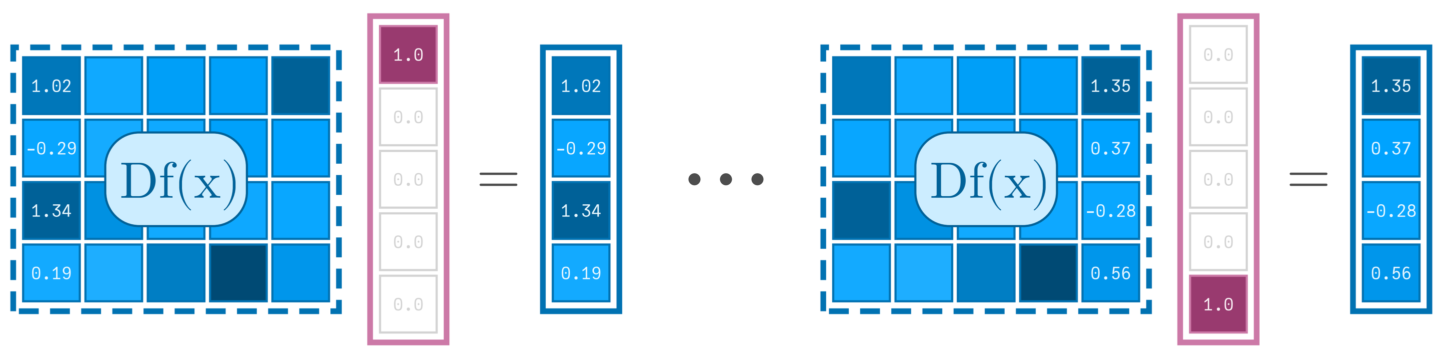
[1]BIFOLD – Berlin Institute for the Foundations of Learning and Data, Berlin, Germany,
[2]Machine Learning Group, Technical University of Berlin, Berlin, Germany,
[3]LVMT, ENPC, Institut Polytechnique de Paris, Univ Gustave Eiffel, Marne-la-Vallée, France,
[4]Argonne National Laboratory, Lemont, USA

BLOG & CODE

Recap: Automatic Differentiation (AD)

The use of AD in deep learning is ubiquitous: Instead of having to compute gradients and Jacobians by hand, AD automatically computes them given PyTorch, JAX or Julia code.

Matrix-free Jacobian operators (dashed) lie at the core of AD. While we illustrate them as matrices to provide intuition, they are best thought of as **black-box functions** with unknown structure.
To turn such Jacobian operators into **Jacobian matrices** (solid), they are evaluated with all standard basis vectors.

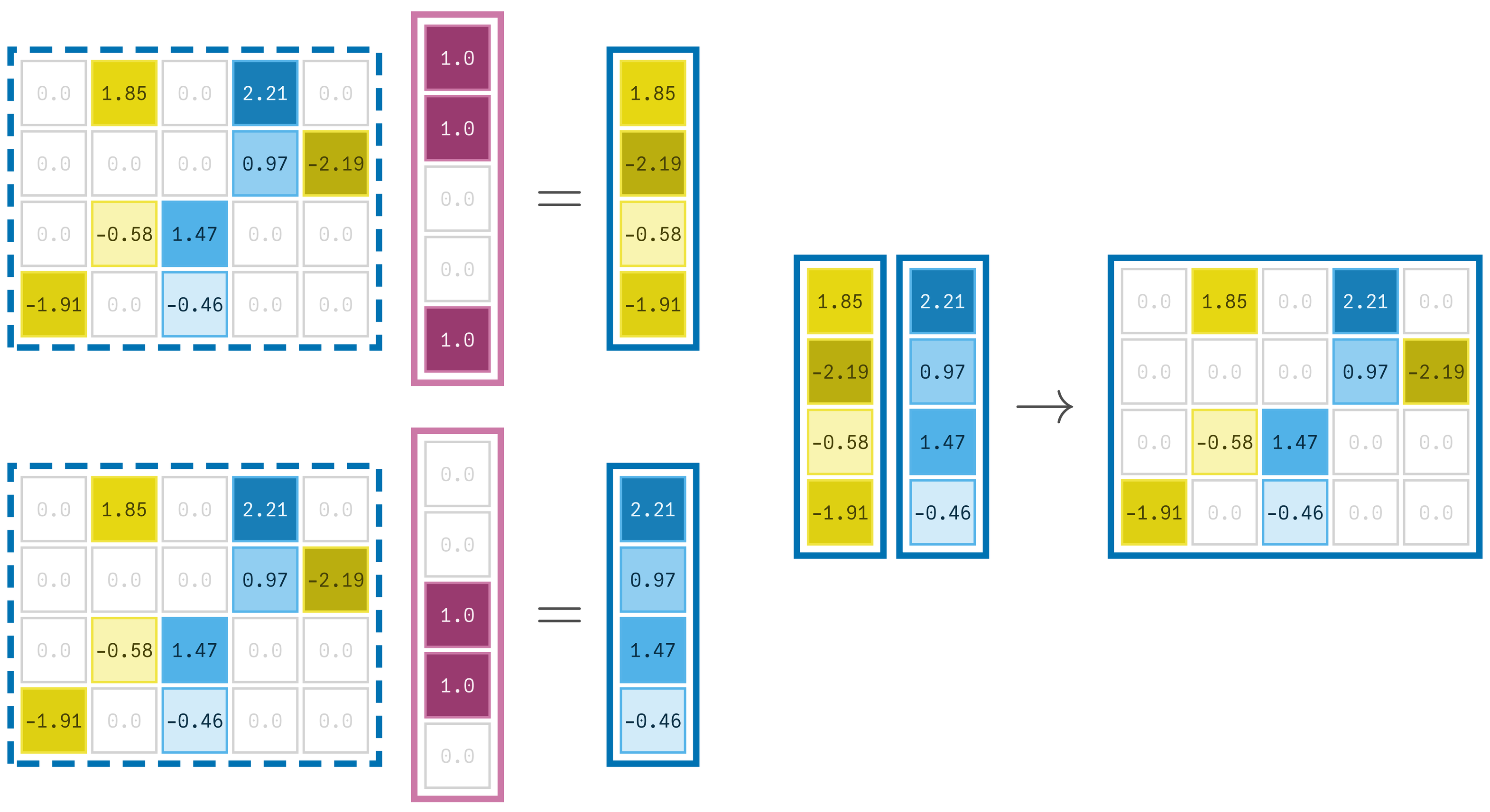


This constructs Jacobian matrices column-by-column[1] or row-by-row[2].

[1] Forward mode, computing as many JVPs as there are inputs (pictured).
[2] Reverse mode, computing as many VJPs as there are outputs.

Automatic Sparse Differentiation (ASD)

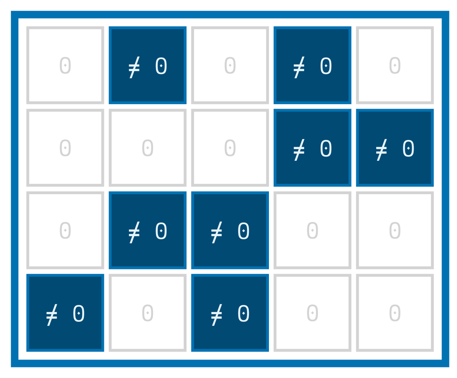
Since Jacobian operators are linear maps, we can **simultaneously compute the values of multiple orthogonal columns** (or rows) and decompress the resulting vectors into the Jacobian matrix [1, 2].



To do this, ASD requires knowledge of the structure of the resulting Jacobian matrix. Since Jacobian operators have unknown structure, two preliminary steps are required.

Step 1: Sparsity Pattern Detection

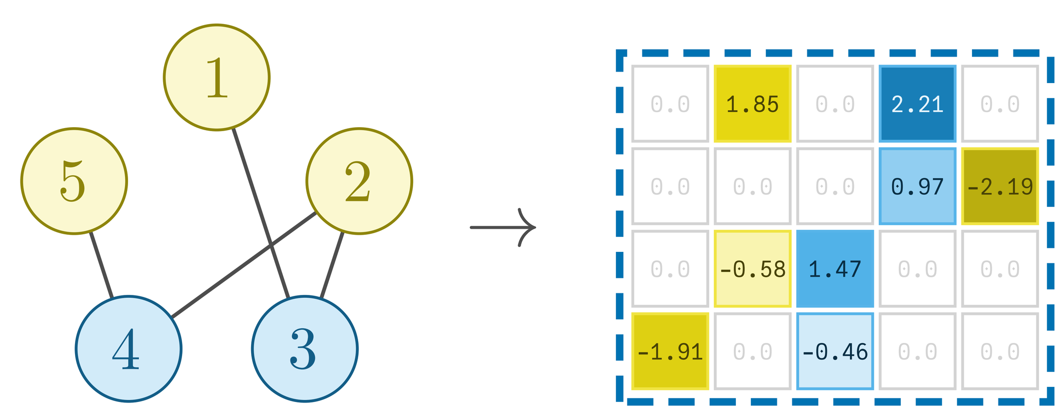
To find orthogonal columns, the pattern of non-zero values in the Jacobian matrix has to be computed. This requires a binary AD system.



Mirroring the multitude of approaches to AD, many viable approaches to pattern detection exist [3–5].

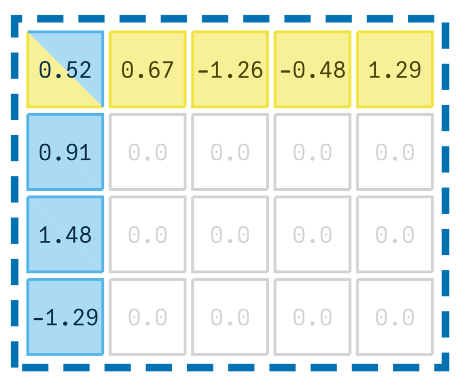
Step 2: Coloring

Graph coloring algorithms are applied to the sparsity pattern to group together orthogonal columns/rows [2].



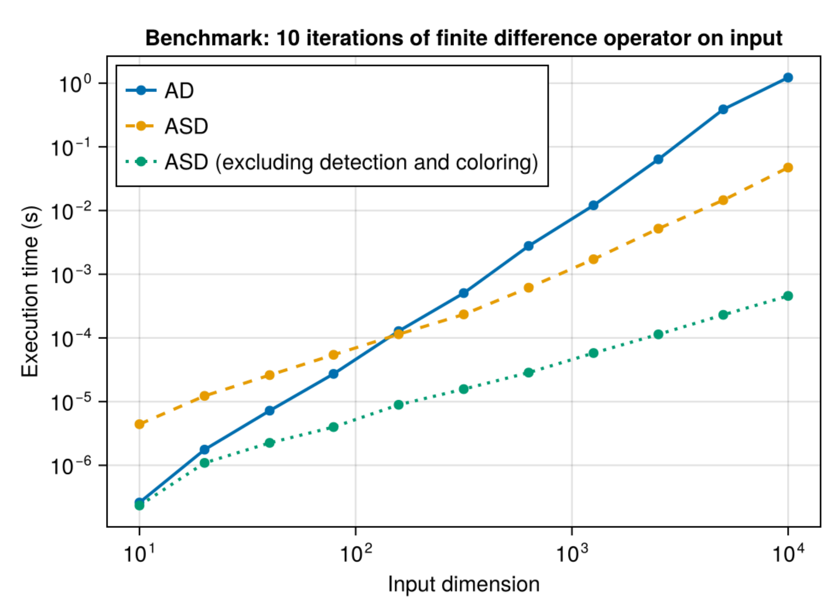
Bicoloring

ASD can be accelerated even further by coloring both rows and columns and combining forward and reverse modes [6, 7].



Benchmarks

ASD can drastically outperform AD. The performance depends on the sparsity of the Jacobian matrix: the cost of sparsity pattern detection and coloring has to be amortized by having to compute fewer matrix-vector products.



References

[1] A. Griewank and A. Walther, Evaluating derivatives: principles and techniques of algorithmic differentiation, 2nd ed. Philadelphia, PA: Society for Industrial, Applied Mathematics, 2008. [Online]. Available: https://epubs.siam.org/doi/book/10.1137/1.9780898717761

[2] A. H. Gebremedhin, F. Manne, and A. Pothen, "What Color Is Your Jacobian? Graph Coloring for Computing Derivatives," SIAM Review, vol. 47, no. 4, pp. 629–705, Jan. 2005, doi: 10/cmwds4.

[3] L. C. W. Dixon, Z. Maany, and M. Mohseninia, "Automatic differentiation of large sparse systems," Journal of Economic Dynamics and Control, vol. 14, no. 2, pp. 299–311, May 1990, doi: 10.1016/0165-1889(90)90023-A.

[4] C. H. Bischof, P. M. Khademi, A. Buaricha, and C. Alan, "Efficient computation of gradients and Jacobians by dynamic exploitation of sparsity in automatic differentiation," Optimization Methods and Software, vol. 7, no. 1, pp. 1–39, Jan. 1996, doi: 10.1080/10556789608805642.

[5] A. Walther, "Computing sparse Hessians with automatic differentiation," ACM Transactions on Mathematical Software, vol. 34, no. 1, pp. 1–15, Jan. 2008, doi: 10.1145/1322436.1322439.

[6] A. K. M. S. Hossain and T. Steihaug, "Computing a sparse Jacobian matrix by rows and columns," Optimization Methods and Software, vol. 10, no. 1, pp. 33–48, Jan. 1998, doi: 10.1080/10556789808805700.

[7] T. F. Coleman and A. Verma, "The Efficient Computation of Sparse Jacobian Matrices Using Automatic Differentiation," SIAM Journal on Scientific Computing, vol. 19, no. 4, pp. 1210–1233, Jan. 1998, doi: 10.1137/S1064827595295349.