HAL open science

# GPU-based Implementations of MM Algorithms. Application to Spectroscopy Signal Restoration

Mouna Gharbi, Emilie Chouzenoux, Jean-Christophe Pesquet, Laurent Duval

**To cite this version:**

Mouna Gharbi, Emilie Chouzenoux, Jean-Christophe Pesquet, Laurent Duval. GPU-based Implementations of MM Algorithms. Application to Spectroscopy Signal Restoration. EUSIPCO 2021 - 29th European Signal Processing Conference, Aug 2021, Dublin / Virtual, Ireland. hal-03275427v2

**HAL Id: hal-03275427**

**https://inria.hal.science/hal-03275427v2**

Submitted on 26 Jul 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# GPU-based Implementations of MM Algorithms. Application to Spectroscopy Signal Restoration

Mouna Gharbi(1), Emilie Chouzenoux(1), Jean-Christophe Pesquet(1) and Laurent Duval(2)
(1) *Université Paris-Saclay, Inria, CentraleSupélec, CVN, France.* firstname.name@centralesupelec.fr
(2) *IFP Energies nouvelles, Rueil-Malmaison, France.* laurent.duval@ifpen.fr

***Abstract*—Restoration of analytical chemistry data from degraded physical acquisitions is an important task for chemists to obtain accurate component analysis and sound interpretation. The high-dimensional nature of these signals and the large amount of data to be processed call for fast and efficient reconstruction methods. Existing works have primarily relied on optimization algorithms to solve a penalized formulation. Although very powerful, such methods can be computationally heavy, and hyperparameter tuning can be a tedious task for non-experts. Another family of approaches explored recently consists in adopting deep learning to perform the signal recovery task in a supervised fashion. Although fast, thanks to their formulations amenable to GPU implementations, these methods usually need large annotated databases and are not explainable. In this work, we propose to combine the best of both worlds, by proposing unfolded Majorization-Minimization (MM) algorithms with the aim to reach fast and accurate methods for sparse spectroscopy signal restoration. Two state-of-the-art iterative MM algorithms are unfolded onto deep network architectures. This allows both the deployment of GPU-friendly tools for accelerated implementation, as well as the introduction of a supervised learning strategy for tuning automatically the regularization parameter. The effectiveness of our approach is demonstrated on the restoration of a large dataset of realistic mass spectrometry data.**

***Index Terms*—Majorization-Minimization, subspace acceleration, unfolding, regularization parameter, GPU, mass spectrometry.**

## I. Introduction

Several applications for analyzing chemical sample properties, such as mass spectrometry (MS) [1] or nuclear magnetic resonance (NMR) spectroscopy [2], require the restoration of source signals $\overline{x} \in \mathbb{R}^n$ from acquired measurements $y$, related to the sought signal through

$$y = H\overline{x} + e. \tag{1}$$

Hereabove, $y \in \mathbb{R}^m$ is the acquired signal, $H \in \mathbb{R}^{m \times n}$ is the linear operator modeling the acquisition/degradation process (e.g., blur, discrete Laplace transform, etc.) and $e \in \mathbb{R}^m$ is an additive noise corrupting the data. The final goal for restoring $\overline{x}$ is, for chemists, to produce meaningful spectra to analyze and extract relevant information: for instance relative abundance of compounds, chemical formulae, etc. In such an applicative context, the inverse problem in (1) must be solved a large number of times (at least, thousands) as each system corresponds to a given acquisition setting (e.g., relaxation time in NMR) and/or chemical compound. It is thus fundamental to have access to fast resolution tools. Furthermore, Problem (1) is itself of large dimension and is often ill-posed due to the presence of noise and ill-conditioning of $H$. Thus, basic inversion or filtering procedures might fail, and more sophisticated recovery methods should be sought for. A first family of techniques amounts to minimizing a penalized loss function, combining data fidelity term and regularization term that incorporates prior knowledge (e.g., sparsity, positivity) on $\overline{x}$. This approach was adopted, for instance in [3], [4] in the context of NMR relaxometry, in [5] in DOSY NMR, and in [6], [7] for MS data processing.

Majorization-Minimization (MM) approaches have shown their efficiency in solving large scale optimization problems arising in signal and image restoration [8]. In particular, the MM quadratic approach, also called half-quadratic (HQ) [9], was deployed with success for chemical signal restoration [10], [11]. The convergence rate of the aforementioned algorithm has later on been greatly improved thanks to a careful subspace strategy [12], leading to the MM Memory Gradient (3MG) algorithm [13], [14].

A second strategy to tackle this inverse problem lies in adopting a deep learning-based framework so as to learn, from annotated database of signals, an optimal restoration architecture. This was adopted for instance in [15], where dense convolutional neural networks (CNNs) were trained to remove artifacts from high-dimensional NMR signals. Similarly, in [16], a residual convolutional neural network, including CNNs and fully connected layers, is built to perform compressive sensing spectroscopy.

On the one hand, deep learning methods can make great use of recent GPU-based computing architecture yielding a limited processing time. However, the obtained architectures may often be difficult to interpret, and may suffer from instability [17] [18]. On the other hand, optimization-based methods rely on sound mathematical concepts, that makes them more reliable and interpretable than deep learning strategies. But they may require tedious settings, for example of regularization parameters [19]. Moreover, their parallelization on recent GPU-friendly languages (e.g., Pytorch, TensorFlow) remains scarce in the literature. One can only mention the parallel CPU-based MM approaches from [20], [21], although

This work has been supported by the ITN-ETN project TraDE-OPT funded by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 861137 and by the European Research Council Starting Grant MAJORIS ERC-2019-STG-850925.

requiring to make significant changes in the algorithm scheme itself. A recent promising avenue combines the two aforementioned techniques, by relying on the so-called 'unfolding' principle. Deep unfolding allows the design of novel neural network architectures based on iterative model-based methods. Model parameters are 'untied' across the deep network layers, simulating the consecutive iterations and offering training possibility for a better performance [22]. Deep unfolding was used in [23] to learn operators and penalty function shape parameters. Shrinkage functions and biases are learnt to perform optimal image denoising in [24]. Hyperparameters such as stepsize and regularization weight can also be learnt [25].

The aim of this work is threefold. First, we show how to cast HQ and 3MG iterations as layers of a specific type of feedforward neural-network architecture. This yields a simple way to reach fast implementation of those methods, by using modern GPU tools. Second, we address the learning of the regularization parameter involved in the methods, so as to reach optimal signal restoration results on a large dataset. Third, we illustrate the performance of the resulting methods in the context of MS signal restoration.

The paper is organized as follows. First, in Section II the optimization problem is introduced. HQ and 3MG algorithms are detailed. Section III presents our main contribution, that is the unfolding of HQ and 3MG strategies under a feedforward network architecture. We describe our strategy for optimizing the regularization parameter, in a supervised fashion. Section IV summarizes experimental tests, results and analysis for all proposed methods. Section V concludes the work and draws perspectives.

## II. MM-BASED SIGNAL RESTORATION APPROACH

### A. Problem statement

Let us define $\hat{x} \in \mathbb{R}^n$, an estimate of $\overline{x}$ obtained by solving the minimization problem:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \Big(F(x) = \frac{1}{2}\|Hx - y\|^2 + \lambda\Psi(x)\Big). \tag{2}$$

Hereabove, $\Psi : \mathbb{R}^n \mapsto \mathbb{R}$ is a regularization term defined as follows

$$(\forall x \in \mathbb{R}^N) \quad \Psi(x) = \sum_{s=1}^{S} \psi([Lx]_s), \tag{3}$$

where $L \in \mathbb{R}^{S\times n}$, $\psi : \mathbb{R} \mapsto \mathbb{R}$ is a differentiable potential function and $\lambda > 0$ is a regularization hyperparameter. The goal is to minimize the objective function $F$. This can be achieved by adopting MM-based algorithms described in the next section.

### B. Half-quadratic algorithm

The main principle of MM algorithms lies in replacing the objective function to minimize, by a sequence of majorizing functions $Q(\cdot, x_k)$ approximating $F$ at each current estimate $x_k$ with $k \in \mathbb{N}$ and chosen so as to satisfy the following conditions:

$$(\forall x \in \mathbb{R}^n) \quad F(x) \leq Q(x, x_k), \quad F(x_k) = Q(x_k, x_k). \tag{4}$$

HQ approach [9], [26] assumes that $\psi$ is a differentiable even function on $\mathbb{R}$, increasing on $[0, +\infty)$, and such that $\psi(\sqrt{\cdot})$ is concave on $[0, +\infty)$. Then, one can set for every iterate $x_k \in \mathbb{R}^n$ the quadratic majorizing tangent function as:

$$\begin{aligned}(\forall x \in \mathbb{R}^n) \quad Q(x, x_k) &= F(x_k) + \nabla F(x_k)^\top (x - x_k) \\ &\quad + \frac{1}{2}(x - x_k)^\top A(x_k)(x - x_k),\end{aligned} \tag{5}$$

where $A(x_k) \in \mathbb{R}^{n\times n}$ is symmetric definite positive, defined as

$$(\forall x \in \mathbb{R}^n) \quad A(x) = H^\top H + \lambda L^\top \operatorname{Diag}\{(\omega([Lx]_s))_{1\leq s\leq S}\}L,$$

where, for $u \in \mathbb{R}$, $\omega(u) = \dot{\psi}(u)/u$ (extended by continuity in $u = 0$), and $\dot{\psi}$ is the derivative of $\psi$. The HQ algorithm reads:

$$(\forall k \in \mathbb{N}) \quad x_{k+1} = x_k - A(x_k)^{-1}\nabla F(x_k). \tag{6}$$

### C. Majorize Minimize-Memory Gradient (3MG) algorithm

The 3MG algorithm [14] reduces the complexity of the inversion required in (6) by limiting the search for each iterate to a vector subspace. To do so, one defines the memory gradient matrix $D_k \in \mathbb{R}^{n\times 2}$, gathering the search directions

$$D_k = \left[-\nabla F(x_k) \quad x_k - x_{k-1}\right], \tag{7}$$

with the convention $x_{-1} = x_0$. The 3MG algorithm reads:

$$(\forall k \in \mathbb{N}) \quad x_{k+1} = x_k - D_k(D_k^\top A(x_k)D_k)^\dagger D_k^\top \nabla F(x_k), \tag{8}$$

where $\dagger$ denotes the pseudo-inverse operation. The computational cost necessary for one iteration is considerably reduced when compared to the HQ algorithm in (6). Note that the 3MG scheme is highly related to the nonlinear conjugate gradient algorithm [27], to the momentum-based accelerated gradient schemes from [28] and to trust-region approaches [29]. The main advantage of 3MG over those methods is that the MM framework allows to have a simple closed form update in (8), avoiding any tedious search of stepsize and momentum / conjugation weight. The convergence properties of 3MG and its practical relevance on various examples have been investigated in [13], [14], [30].

## III. UNFOLDED MM ALGORITHMS

### A. Unfolded architecture

Both HQ and 3MG iterations can be reexpressed as

$$(\forall k \in \mathbb{N}) \quad x_{k+1} = x_k - W_k\nabla F(x_k), \tag{9}$$

where, for HQ algorithm (6),

$$W_k = A(x_k)^{-1}, \tag{10}$$

while, for 3MG algorithm (8),

$$W_k = D_k(D_k^\top A(x_k)D_k)^\dagger D_k^\top. \tag{11}$$

Moreover, from the expression of $F$ in (2)-(3), we have

$$(\forall k \in \mathbb{N}) \quad \nabla F(x_k) = H^\top(Hx_k - y) + \lambda L^\top \Omega_k L x_k \quad (12)$$

with

$$(\forall k \in \mathbb{N}) \quad \Omega_k = \mathrm{Diag}\{(\omega([Lx_k]_s))_{1 \leq s \leq S}\}. \quad (13)$$

The key idea of our unfolding procedure is to rewrite one iteration $k \in \{0, \ldots, K-1\}$, $K > 0$, of (9) through compositions of non-linear and linear operations, mimicking one layer of a neural network [31]. To do so, let us introduce three useful non-linear operators $R_1$, $R_2$ and $R_3$, and matrix/vector terms $V_1$, $V_2$, $V_3$ and $b_1$, $b_2$, $b_3$, respectively. One layer $\mathcal{L}_k$ of our architecture, i.e., one iteration $k$ of the MM algorithm, is decomposed into three branches. In the first branch, we use $V_1 = H^\top H$ , $b_1 = -H^\top y$ and $R_1 : \mathbb{R}^n \to \mathbb{R}^n$ reduces to the identity function. For the second branch, we use $V_2 = L$, $b_2 = 0$, and $R_2 : \mathbb{R}^S \to \mathbb{R}^S$ given by

$$(\forall u \in \mathbb{R}^S) \quad R_2 : u \to (\dot{\psi}(u_s))_{1 \leq s \leq S}. \quad (14)$$

The last branch concatenates the outputs of $R_1$ and $R_2$ and composes the result with the linear operator[1] $V_3 = [I_n \quad \lambda L^\top]$ with null shift $b_3$. Here, $R_3 : \mathbb{R}^n \to \mathbb{R}^n$ reduces to the identity function like $R_1$. For every $k \in \{0, ..., K-1\}$, the update (9) then reads

$$x_{k+1} = x_k - W_k \left( R_3 \left( V_3 \begin{bmatrix} R_1(V_1 x_k + b_1) \\ R_2(V_2 x_k + b_2 \end{bmatrix} + b_3 \right) \right). \quad (15)$$

As described at the bottom of Fig. 1, the latter update equation can be interpreted as adding a skip connection in a feed-forward architecture. At a given layer $\mathcal{L}_k$ of the network, $V_1$, $V_2$, $V_3$ and $W_k$ can be interpreted as linear weight operators, $b_1$, $b_2$, $b_3$ as bias terms, while operators $R_1$, $R_2$ and $R_3$ can be seen as specific activation functions. It is worth noting that the linear weight operator $W_k$ actually depends on each layer's entry $x_k$. This model leads to a deep unfolding of $K$ iterations of HQ and 3MG algorithms as shown at the top of Fig. 1. In the remaining, we will interchangeably use layer and iteration to describe one step of the algorithms.

Such deep unfolding allows straightforward parallel implementation of the algorithms since such networks are, by nature, suitable for GPU-friendly frameworks such as Pytorch or TensorFlow. Moreover, the unfolding procedure offers the possibility of learning certain terms such as the regularization parameter instead of manually choosing their values, as we will show in the next section.

### B. Learning regularization parameter

We propose to adopt a supervised learning strategy in order to determine from a training set, an optimal setting of the regularization parameter for the 3MG (8) method, starting from an initial given regularization parameter. To this aim, 3MG schemes are unfolded over $K$ iterations, as described in the previous section. Moreover, the regularization parameter $\lambda$ is now untied across the network, leading to a sequence of parameters $(\lambda_k)_{0 \leq k \leq K-1}$ to be learnt instead of a single one.

[1] $I_n$ denotes the identity matrix of $\mathbb{R}^n$

For every $k \in \{0, ..., K-1\}$, layer $k$ of the network is built by combining a hidden structure $\mathcal{L}_k^{(\lambda)}$ computing the regularization weight $\lambda_k$ and an explicit structure mirroring the original 3MG algorithm and generating the iterate $x_k$ using the learnt hyperparameter $\lambda_k$ at layer $k$. The resulting architecture is depicted in Fig. 2.

In order to enforce positivity of $\lambda_k$ for each layer $k \in \{0, \ldots, K-1\}$, we define:

$$\lambda_k = \mathcal{L}_k^{(\lambda)} = \mathrm{Softplus}(z_k), \quad (16)$$

where $(\forall z \in \mathbb{R})$ $\mathrm{Softplus}(z) = \ln(1 + \exp(z))$ is the smooth approximation of the ReLU activation function and $(z_k)_{0 \leq k \leq K-1}$ are scalar parameters of the network to be learnt during the training phase. In our proposal, the training will be performed so as to minimize the mean square error loss between the original signal $\overline{x}$ and its estimation $\hat{x}$, over a training dataset.

## IV. Restoration of mass spectrometry signals

We now evaluate the proposed algorithms when applied to the restoration of blurry and noisy mass spectrometry sparse signals.

### A. Settings and dataset

*a) Dataset:* We build synthetic datasets of MS signals. Groundtruth signals $\overline{x}$ are simulated with the `averagine` model [32], so as to mimick typical mass spectra of a compound containing 10 proteins with charge state $+1$, on the mass range $[1100, 1200]$ ppm. These signals are of size 2000, containing randomly distributed positive valued peaks, with different intensities. (Fig. 4 (top left)). Corrupted signals $y$ are then constructed, following model (1) (Fig. 4 (top right)). $H$ models the convolution, with circulant padding, with a standard normal kernel. The additive noise $e$ is zero-mean white Gaussian with standard deviation 2. In Sec. IV-B, 1000 signals are used to compare CPU versus GPU implementations. In Sec. IV-C, we build a larger dataset, split into 1000, 200 and 200 pairs $(\overline{x}, y)$ for training, validation and test, respectively.

*b) Evaluation of the results:* The performance of the methods, and the training of the unfolded architecture, relies on the mean squared error (MSE) defined as $\frac{1}{n}\|\overline{x} - \hat{x}\|^2$, averaged over the dataset considered in each experiment.

*c) Penalization function:* Experiments were conducted using the convex sparsity promoting penalization function

$$\psi(t) = \delta(|t| - \delta \log(|t|/\delta + 1)), \quad (17)$$

with $\delta = 2$.

*d) Initialization:* All the tests are performed by using the Wiener-like initialization:

$$x_0 = (H^\top H + \sigma I_n)^{-1} H^\top y \quad (18)$$

with $\sigma = 0.5$.

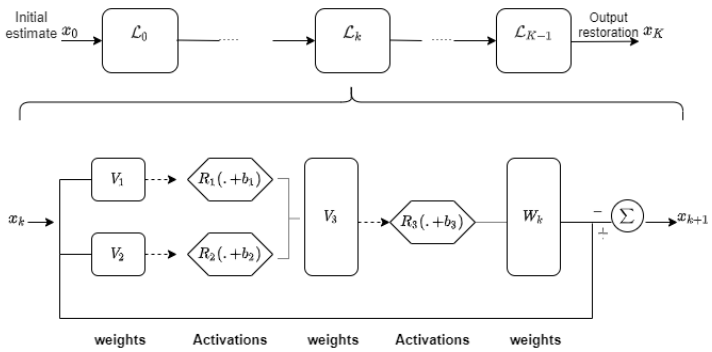


Fig. 1. Unfolded MM algorithm architecture: Unfolded layer $\mathcal{L}_k$ corresponds to iteration $k$ of (9). $V_1$, $V_2$ and $V_3$ are weight operators. $b_1$, $b_2$, $b_3$ are biases, and $R_1$, $R_2$ and $R_3$ specific activation functions. $W_k$ is the last weight operator corresponding to (10) for HQ algorithm and (11) for 3MG algorithm.

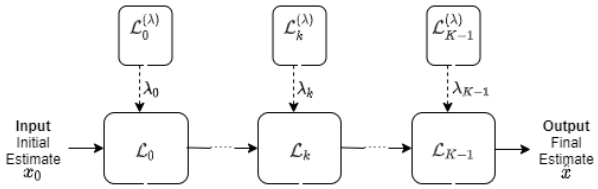


Fig. 2. Architecture for learning regularization weight.

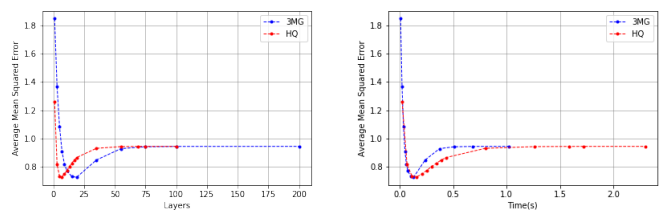


Fig. 3. Evolution of averaged MSE over layers (left) and time (right), for HQ and 3MG.

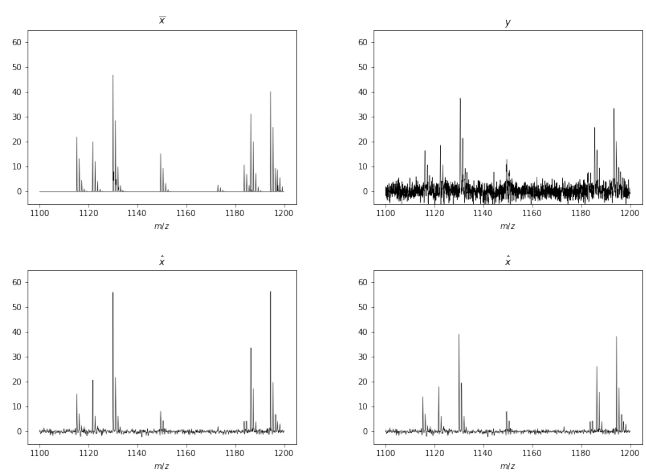


Fig. 4. Example of groundtruth signal $\overline{x}$ (top left), degraded signal $y$ (top right), reconstructed signal using 3MG with fixed $\lambda$, MSE=1.1213 (bottom left) and 3MG with learnt $(\lambda_k)_{0\leq k\leq K-1}$, MSE=0.7303 (bottom right).

*e) Programming framework:* We compare GPU-based implementations, using Pytorch, with classical CPU-based implementation. The codes are run onto an Intel(r) Xeon(R) w-2135 cpu @ 3.70GHz, with 6 CPU active cores, equipped with a Quadro RTX 5000. We make use of Python 3.8.3 and Pytorch 1.7.1.

### *B. CPU versus GPU-based implementations*

In our first set of experiments, HQ and 3MG CPU and GPU implementations are compared. The regularization parameter is manually set to $\lambda = 2$, which gives the best regularization in terms of average MSE on the dataset. An example of restoration is depicted in Fig. 4 (bottom left). As expected, the execution time increases linearly with iterations (i.e., layers), for both GPU and CPU implementations. More precisely, the ratio (*Execution time/Number of iterations*) respectively for HQ GPU, HQ CPU, 3MG GPU and 3MG CPU implementations are $0.023$, $0.2473$, $0.0068$ and $0.1612$. The number of iterations necessary to reach convergence of 3MG is slightly higher than that for HQ, as shown on Fig. 3 (left). However, the latter requires more time per iteration (see Fig. 3 (right)). For comparison, 3MG GPU requires 0.6812 s. to reach a stable value of MSE while HQ GPU requires 1.5854 s. to converge to the same MSE value (here, 0.9448). Execution time per iteration increases on a logarithmic scale with signal length $n$ as shown on Fig. 5. It is worth noting that GPU implementations always show much faster performance than their CPU equivalent, for both HQ and 3MG, while reproducing exactly the same iterations and thus result quality. This illustrates the clear advantage of our unfolding procedure.

### *C. Learning the regularization parameter*

We now proceed with learning the regularization parameter by training the unfolded 3MG. The latter was retained as it led to the lowest convergence time in the previous experiments. We initialize the network weights to be learnt to $z_k \equiv 2$. We set $K = 75$ layers, as it appeared sufficient to reach convergence as can be observed in Fig. 3. ADAM optimizer is used to minimize the MSE on the training set, with a learning rate $10^{-4}$, decreasing by $0.9$ factor after each epoch.

Final average MSE value on the test set, for unfolded 3MG

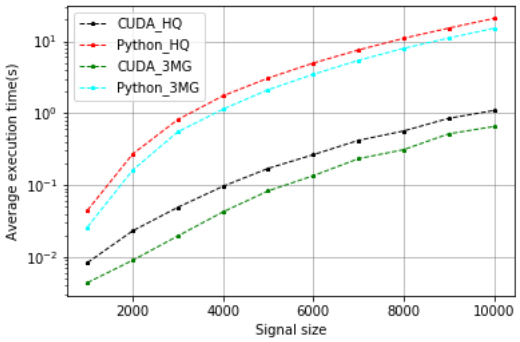


Fig. 5. Averaged execution time w.r.t. signal size $n$, for one iteration.

algorithm is equal to $0.7312$, while the use of 75 iterations of 3MG method with fixed $\lambda$ returns $0.9730$ . We conclude that learning the regularization parameter leads to a gain in terms of output signal quality, as can also be seen in Fig. 4 (bottom right).

## V. Conclusion

In this work, we have explored efficient approaches for fast resolution of linear inverse problems arising in chemical signal restoration. We have proposed a GPU-friendly reformulations of HQ and 3MG algorithms, by unfolding their iterations as deep network architectures. We have also shown how to further benefit from unfolding to learn the regularization parameter and thus improve restoration quality.

## References


[1] L. Heaney, D. Jones, and T. Suzuki, "Mass spectrometry in medicine: a technology for the future?" *Future Science OA*, vol. 3, 06 2017.

[2] R. R. Ernst, G. Bodenhausen, and A. Wokaun, *Principles of Nuclear Magnetic Resonance in One and Two Dimensions*. 2nd ed., ser. International Series of Monographs on Chemistry. Oxford, NY: Oxford University Press, 19, 1987.

[3] E. Chouzenoux, S. Moussaoui, J. Idier, and F. Mariette, "Efficient maximum entropy reconstruction of nuclear magnetic resonance T1-T2 spectra," Dec. 2010.

[4] E. D. Lade, J. Skilling, J. Staunton, S. Sibisi, and R. G. Brereton, "Maximum entropy method in nuclear magnetic resonance spectroscopy," *Journal of Magnetic Resonance (1969)*, vol. 62, no. 3, pp. 437–452, 1985.

[5] A. Cherni, E. Chouzenoux, and M.-A. Delsuc, "PALMA, an improved algorithm for DOSY signal processing," *Analyst*, vol. 142, no. 5, pp. 772–779, 2016.

[6] A. Cherni, E. Chouzenoux, L. Duval, and J. C. Pesquet, "Spoq $\ell_p$-over-$\ell_q$ regularization for sparse signal recovery applied to mass spectrometry," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6070–6084, 2020.

[7] A. Mohammad-Djafari, J.-F. Giovannelli, G. Demoment, and J. Idier, "Regularization, maximum entropy and probabilistic methods in mass spectrometry data processing problems," *International Journal of Mass Spectrometry*, vol. 215, no. 1, pp. 175–193, 2002, detectors and the Measurement of Mass Spectra.

[8] Y. Sun, P. Babu, and D. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, pp. 794–816, Feb 2017.

[9] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 298–311, 1997.

[10] V. Mazet, D. Brie, and J. Idier, "Baseline spectrum estimation using half-quadratic minimization," in *Proceedings of the 12th European Signal Processing Conference (EUSIPCO 2004)*, 2004, pp. 305–308.

[11] J. Liu, J. Liu, J. Sun, X. Huang, G. Li, and B. Liu, "Goldindec: A novel algorithm for Raman spectrum baseline correction," *Applied Spectroscopy*, vol. 69, no. 7, 2015.

[12] A. Wald and T. Schuster, "Sequential subspace optimization for nonlinear inverse problems," *Journal of Inverse and Ill-posed Problems*, vol. 25, no. 1, pp. 99–117, 2017.

[13] E. Chouzenoux, J. Idier, and S. Moussaoui, "A majorize–minimize strategy for subspace optimization applied to image restoration," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1517–1528, 2011.

[14] E. Chouzenoux, A. Jezierska, J.-C. Pesquet, and H. Talbot, "A Majorize-Minimize subspace approach for l2-l0 image regularization," *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 563–591, 2013.

[15] X. Qu, Y. Huang, H. Lu, T. Qiu, D. Guo, T. Agback, V. Orekhov, and Z. Chen, "Accelerated nuclear magnetic resonance spectroscopy with deep learning," *Angewandte Chemie (International ed. in English)*, vol. 59, no. 26, p. 10297—10300, June 2020.

[16] C. Kim, D. Park, and H.-N. Lee, "Compressive sensing spectroscopy using a residual convolutional neural network," *Sensors*, vol. 20, p. 594, 01 2020.

[17] V. Antun, F. Renna, C. Poon, B. Adcock, and A. Hansen, "On instabilities of deep learning in image reconstruction and the potential costs of AI," *Proceedings of the National Academy of Sciences*, vol. 117, p. 201907377, 05 2020.

[18] P. L. Combettes and J.-C. Pesquet, "Lipschitz Certificates for Layered Network Structures Driven by Averaged Activation Operators," *SIAM Journal on Mathematics of Data Science (to appear)*, 2020.

[19] T. Edwards and S. Stoll, "Optimal tikhonov regularization for deer spectroscopy," *Journal of Magnetic Resonance*, vol. 288, 02 2018.

[20] S. Cadoni, E. Chouzenoux, J.-C. Pesquet, and C. Chaux, "A block parallel majorize-minimize memory gradient algorithm," in *Proceedings of the IEEE International Conference on Image Processing (ICIP 2016)*, 09 2016, pp. 3194–3198.

[21] M. Chalvidal and E. Chouzenoux, "Block distributed 3mg algorithm and its application to 3d image restoration," in *Proceedings of the 27th IEEE International Conference on Image Processing (ICIP 2020)*, 2020.

[22] J. Hershey, J. Le Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," Tech. Rep., 2014, https://arxiv.org/abs/1409.2574.

[23] D. Ito, S. Takabe, and T. Wadayama, "Trainable ISTA for sparse signal recovery," *IEEE Transactions on Signal Processing*, vol. 67, no. 12, pp. 3113–3125, 2019.

[24] J. Sun and Z. Xu, "Color image denoising via discriminatively learned iterative shrinkage," *IEEE Transactions on Image Processing*, vol. 24, pp. 4148–59, 09 2015.

[25] C. Bertocchi, E. Chouzenoux, M. Corbineau, J. Pesquet, and M. Prato, "Deep Unfolding of a Proximal Interior Point Method for Image Restoration," *Inverse Problems*, vol. 36, no. 3, p. 034005, Feb. 2020.

[26] M. Nikolova and M. K. Ng, "Analysis of half-quadratic minimization methods for signal and image recovery," *SIAM journal on Scientific Computing*, vol. 27, pp. 937–966, 2005.

[27] W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient method," *Pacific Journal of Optimization*, vol. 2, pp. 35–58, 2006.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Tech. Rep., 2014, https://arxiv.org/abs/1412.6980.

[29] P.-A. Absil and K. A. Gallivan, "Accelerated line-search and trust-region methods," *SIAM J. Numer. Anal.*, vol. 47, no. 2, p. 997–1018, Feb. 2009.

[30] E. Chouzenoux and J. Pesquet, "Convergence rate analysis of the majorize-minimize subspace algorithm," *IEEE Signal Processing Letters*, vol. 23, no. 9, pp. 1284–1288, Sep. 2016.

[31] P. L. Combettes and J.-C. Pesquet, "Deep Neural Network Structures Solving Variational Inequalities," *Set-Valued and Variational Analysis (to appear)*, 2020.

[32] M. W. Senko, S. C. Beu, and F. W. McLaffertycor, "Determination of monoisotopic masses and ion populations for large biomolecules from resolved isotopic distributions," *Journal of the American Society for Mass Spectrometry*, vol. 6, no. 4, pp. 229–233, 1995.