PRACTICAL WORK # 2 - SPARSITY, APPLICATIONS TO INVERSE PROBLEMS

We strongly advise the use of either **Matlab** or **Python** for these practical works.
Participants that opt for Python will find the following modules helpful :

- ipython
- scipy/numpy
- matplotlib
- scikit-learn
- pyfits.

Most of them can be set up with easily using standard porting tools (apt-get, macport ... etc).

All the necessary material is available at http://jbobin.cosmostat.org/master-2-mva

Estimation of a multiresolution mask

- *Add a Gaussian noise (sigma=50) to the image simu_sky.fits*

- *Implement a routine that takes the starlet transform of an image and computes its MAD (Median Absolute Deviation) at the finest scale (i.e. first wavelet scale). Estimate the noise level of the noisy version of simu_sky.fits in the wavelet domain. Comment the results.*

- *Implement a routine that estimates automatically the noise level, and subsequently the signal detection level at each wavelet scale using a k-sigma strategy.*

- *Apply the method to derive a multiresolution mask for the image simu_sky.fits.*

Denoising with sparsity constraint in the starlet transform

- *Implement a soft and hard thresholding denoising method, which we will call* **SoftThrd** *and* **HardThrd** *respectively.*

- *Using the noisy image derived from simu_sky.fits, investigate by eyes the reconstructed image and the residual. Comment the results.*

- *Implement the iterative multiresolution mask denoising algorithm, which we will call* **MRDenoise***, and apply it to the noisy image derived from simu_sky.fits.*

- *Compare the three methods* **SoftThrd***,* **HardThrd** *and* **MRDenoise** *on the noisy version of simu_sky.fits. Comment the results.*

For that purpose, you can compute a (normalized) mean-square error :

$$\epsilon_{\text{MSE}} = \frac{\|x^\star - \hat{x}\|_{\ell_2}}{\|x^\star\|_{\ell_2}}$$

where $x^\star$ stands for the original image and $\hat{x}$ for the estimated signal.

Sparsity-based deblurring

- *Write down a version of the Forward-Backward Splitting algorithm that tackles 2D deconvolution problems.*

- *Implement the proposed algorithm.*

- *Convolve simu_sky.fits by its point-spread function (PSF - simu_psf.fits), add some Gaussian noise, and apply the deconvolution method. Comment the results. For that purpose you can investigate by eyes the error $x^\star - \hat{x}$ in the pixel domain as well as the wavelet domain.*

Please note that the PSF simu_psf.fits is defined in the pixel domain not in the Fourier domain.

Applications of proximal algorithms to inpainting

- *Implement Forward-Backward Splitting algorithm so as to minimize the following problem :*

$$\min_x \quad \lambda \|x\mathbf{\Phi}^T\|_{\ell_1} + \frac{1}{2}\|b - \mathcal{M} \odot x\|_2^2$$

where b denotes the observed data, $\mathbf{\Phi}^T$ the forward starlet transform, $\mathcal{M}$ is a binary mask and x is the image to be recovered. The operator $\odot$ is the Hadamard product.

- *Apply a random binary mask with a fixed number of non-zero entries p and apply the above algorithm to estimate an inpainted image. Investigate the performances of the proposed algorithm as a function of p.*