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

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.*

• *Compare the two methods* **SoftThrd** *and* **HardThrd** *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.*