



Algorithme de Clustering/Fitting pour le spectromètre à muons d'Alice

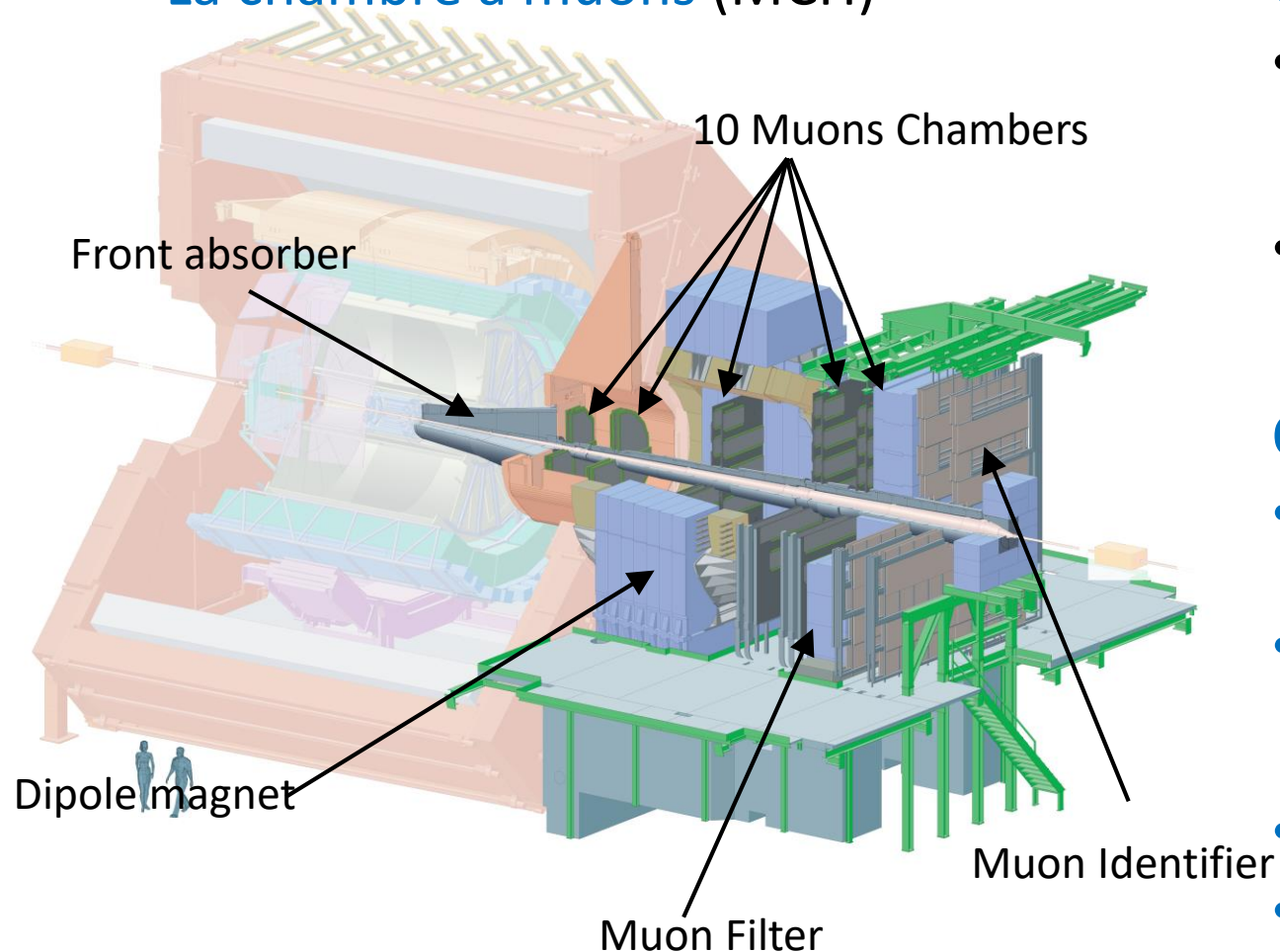
Journées du Laboratoire, 19-20 mai 2022

G. Grasseau, L. Aphecetche, P. Pillot



Motivations

La chambre à muons (MCH)



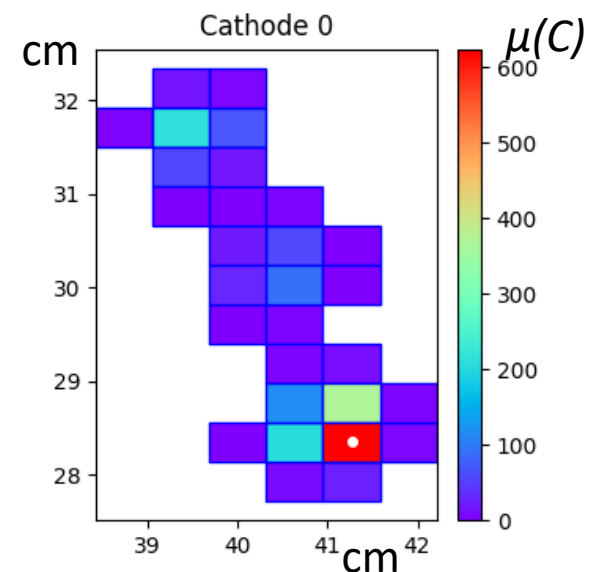
Contexte de l'algorithme

- Localisation du passage de particules chargées dans des différentes chambres à muons (MCH)
- Positions transmises au *trackFinder* pour la reconstruction de traces et leurs sélections

Objectifs

- Retrouver le modèle/l'algorithme (développement initial complexe)
- Construire un nouvel algorithme et une implémentation plus simple à comprendre et à maintenir
- Algorithme / implémentation efficace
- Documentation

Données d'entrées / pré-cluster

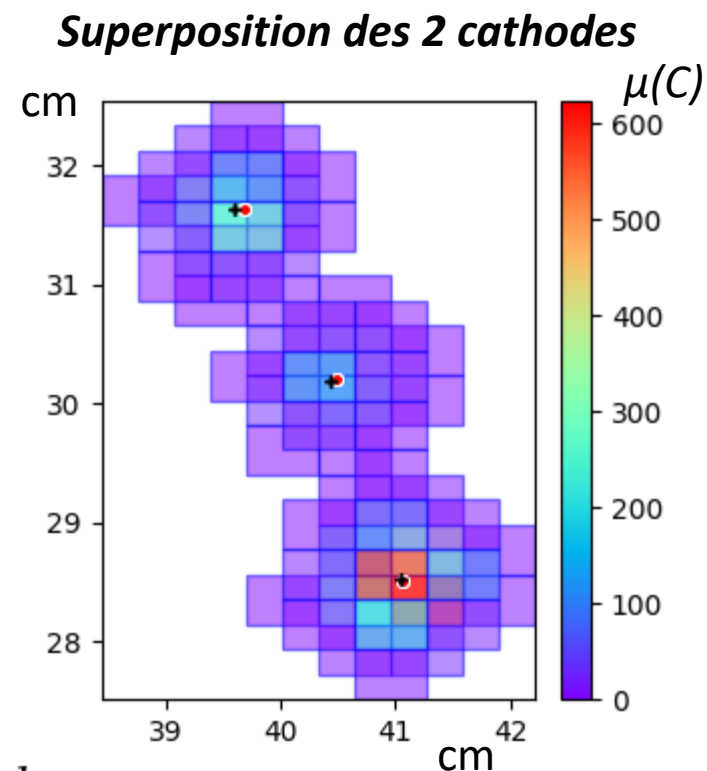
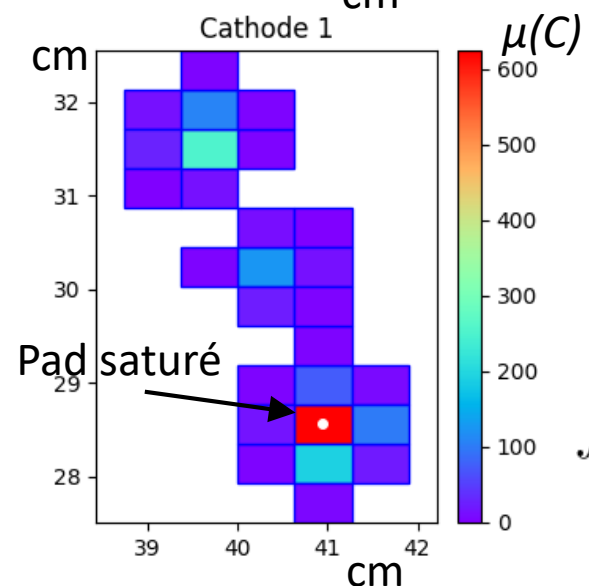


- Objets à traiter : des **pré-clusters**, ensemble de *pads*, sur 2 plans de mesures (2 cathodes)

- **Distribution de la charge** sur les *pads* : $q_a \rho_K(x)$, où $\rho_K(x)$ est une fonction de Mathieson (~gaussienne, intégrale 1), et q_a la charge totale déposée.

- **Calcul de la charge** d'un pad

$$\int_{pad} (2D) = 4 q_a \int_{x_1}^{x_2} \rho(x - x_c) dx \int_{y_1}^{y_2} \rho(y - y_c) dy$$



Simulation d'év
pré-cluster = 0/66, chld=2
formé de 3 clusters

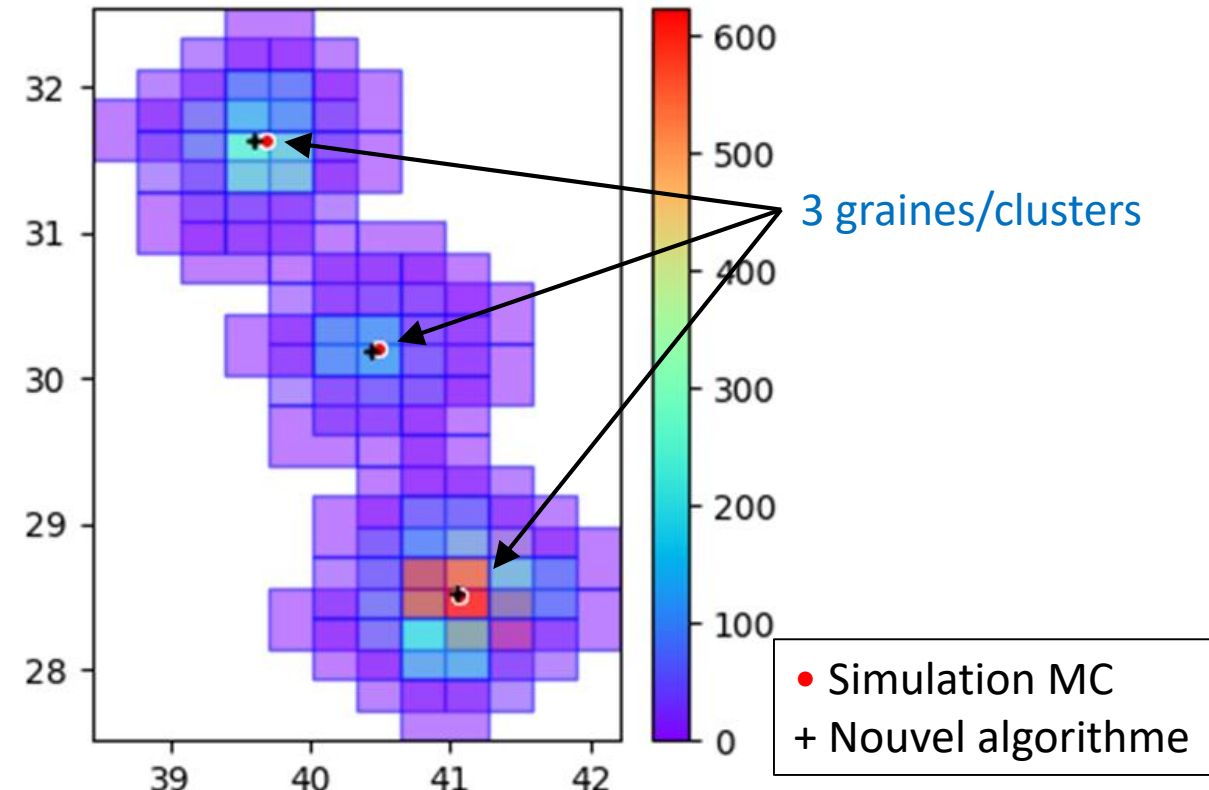
Deux étapes : le *clustering* et le *fitting*

Clustering

- déterminer le nombre d'agrégats / de **graines** (nombre de Mathieson) issues du passage de particules dans la chambre

Fitting

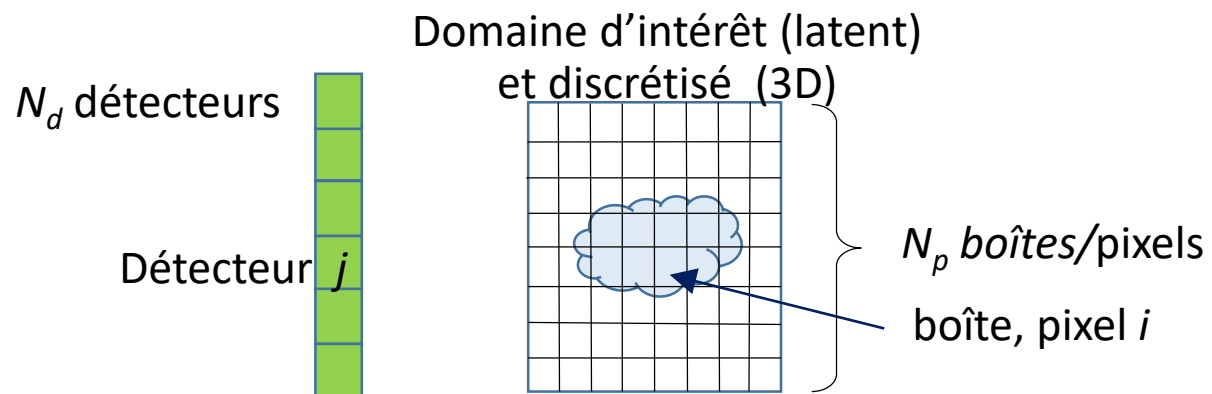
- connaissant le nombre de graines
- ... et la position approchée de chaque graine
- Ajustement de la position des graines et de leur poids (MSE, algorithme de Levenberg-Marquardt)



Par la suite on ne s'occupera que du clustering, étape la plus délicate

Algorithme MLEM TEP

Schéma TEP simplifié



Avec

- i décrit les boîtes/pixels, j les détecteurs
- $n_j = \#$ de γ (paires de γ) vus pas le détecteur j
- $\lambda = (\lambda_1, \dots, \lambda_{N_p})$, activité des pixels (champ discrétisé)
- $\lambda_{ij} = c_{ij} \lambda_i$, activité du pixel i vue par le détecteur j
- c_{ij} angle solide (géométrie)

Expectation-Maximization (EM ou MLEM)

- Maximisation de la log-vraisemblance en fct de $\vartheta = (\lambda_1, \dots, \lambda_{N_p})$
- Spécifique à une loi de proba : **loi de Poisson**
- Forme itérative (étape k)

$$\lambda_i^{k+1} = \lambda_i^k \times \sum_j \frac{c_{ij}}{\sum_{j'} c_{ij'}} \cdot \frac{n_j}{\sum_{i'} c_{i'j} \lambda_{i'}^k}$$

Mesure de n_j

Poids relatif de l'angle solide c_{ij} vis-à-vis de l'angle solide total du pixel i (géométrie)

Prédiction de n_j^k

Algorithme EM MCH

Correspondance avec MLEM TEP

n_j -> charge mesurée q_j du pad j

λ_i -> intensité du pixel i (charge p_i)

N_d -> nombre de pads dans le pré-cluster

N_p -> nombre de pixels dans le domaine (latent)

c_{ij} , influence du pixel i sur la charge du pad j

- charge du pad j si une particule (charge=1.0) passe au centre du pixel i (x_i, y_i)
- géométrique : dépend de la distance entre le pad j et le pixel i et des dimensions du pad

$$c_{ij} = \int_{pad_j} (2D | \text{particule en } i) = 4 \int_{x_1}^{x_2} \rho(x - x_i) dx \int_{y_1}^{y_2} \rho(y - y_i) dy$$

Mesure de q_j

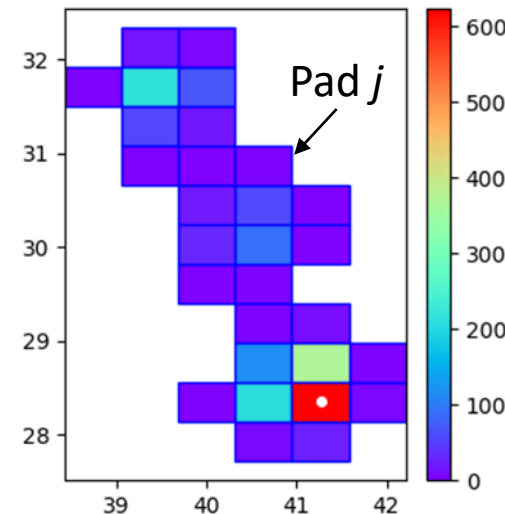
Expectation-Maximization MCH

$$p_i^{k+1} = p_i^k \times \sum_j \frac{c_{ij}}{\sum_{j'} c_{ij'}} \frac{q_j}{\sum_{i'} c_{i'j} p_{i'}^k}$$

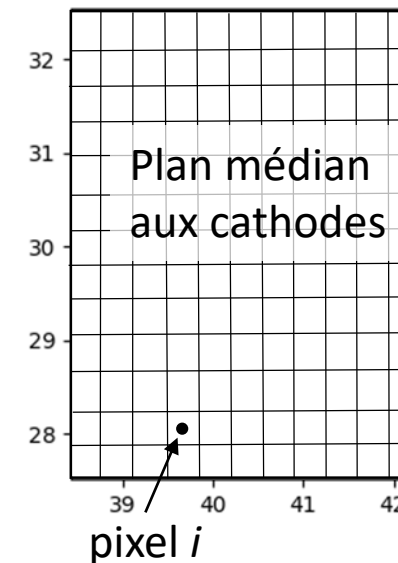
Poids relatif de c_{ij} vis-à-vis de $\int_{\text{pré-cluster}} (2D | \text{particule en } i)$

Prédiction de q_j^k

Cathode 0 (observation)



Pixels /boites



Problème

- q_j pas discret, mathématiquement l'algorithme (Poisson) n'est pas adapté,
- Décision de prendre un autre algorithme EM avec un mélange de gaussiennes
- -> résultats : pas assez sensible (perte ~ 3-5 %)

Généralisation : recherche de maxima locaux (1)

Reconsidération de l'algorithme EM
MCH en généralisant le problème

Généralisation

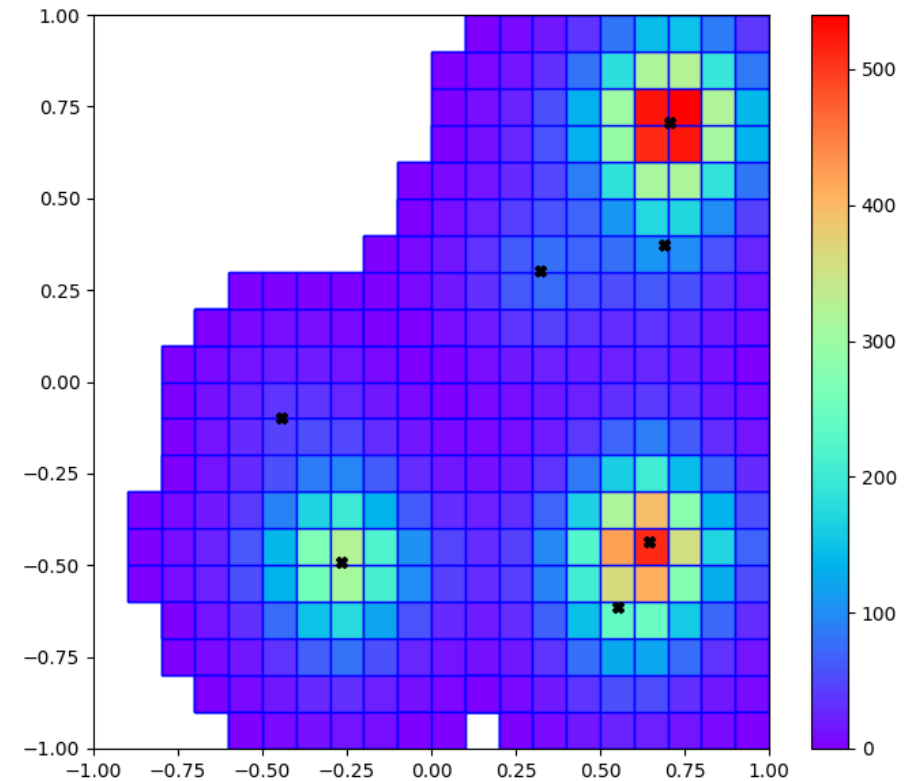
- Trouver les maxima locaux d'une mesure/observation constituée d'un mélange de fonctions (gaussiennes, ...)

$$\text{mesure}(\mathbf{x}) = w_i * f(\mathbf{x}_i),$$

avec i nombre de fct (graines)

Exemple

- Mélange de 7 graines / 7 fonctions
- Séparation $\min(w_i) / \max(w_i) \sim 5\%$
- Algorithmes classiques pas assez sensibles



Simulation d'un mélange de 7 graines
tirées aléatoirement (w_i, x_i) avec $f=\rho$

Généralisation : recherche de maxima locaux (2)

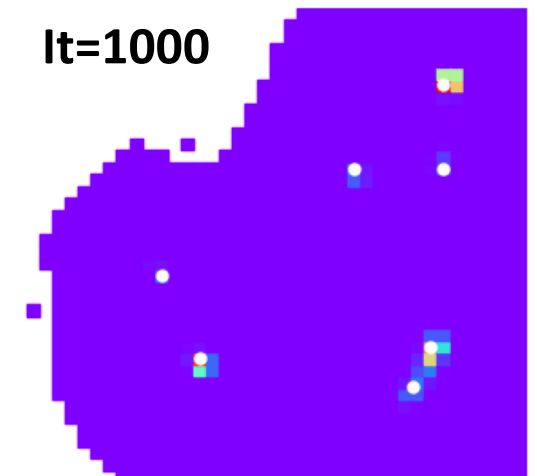
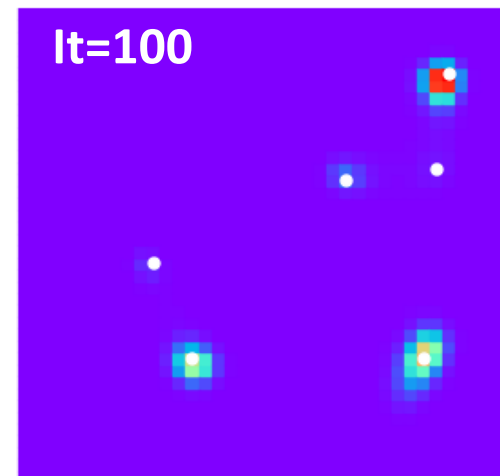
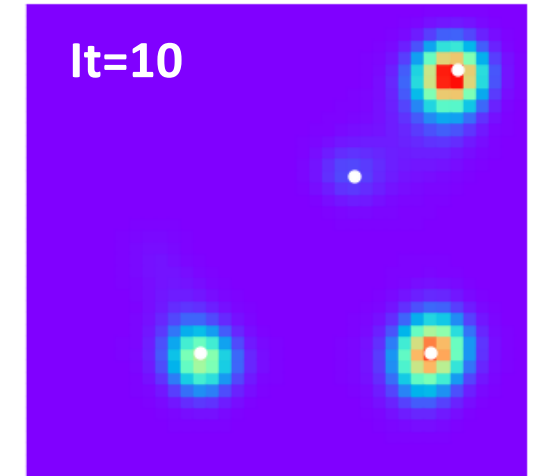
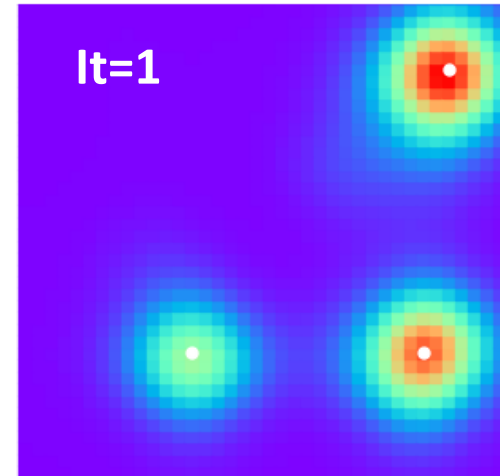
Application de EM MCH

- Domaine latent = plan d'observation
- Boîtes p_i de résolution $\frac{1}{2}$ vis-à-vis des données/observations
- Conditions initiales $p_i^0 = 1.0$
- Convergence $|p_i^{k+1} - p_i^k| < \varepsilon$
- Puis algorithme classique de détection de max
- Critère de sélection du # graines

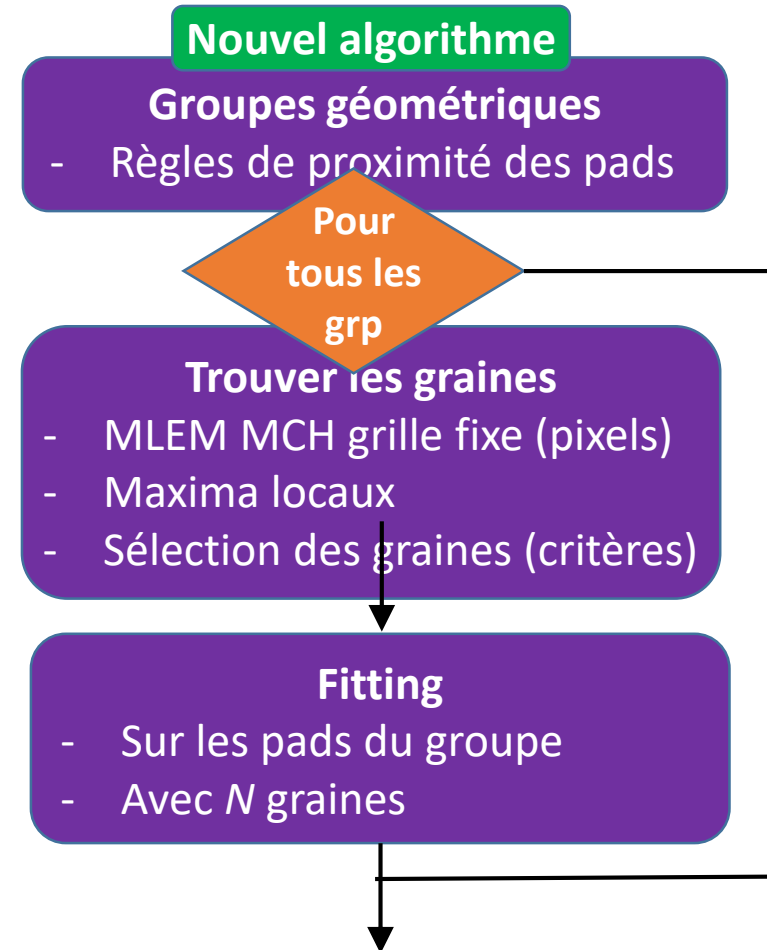
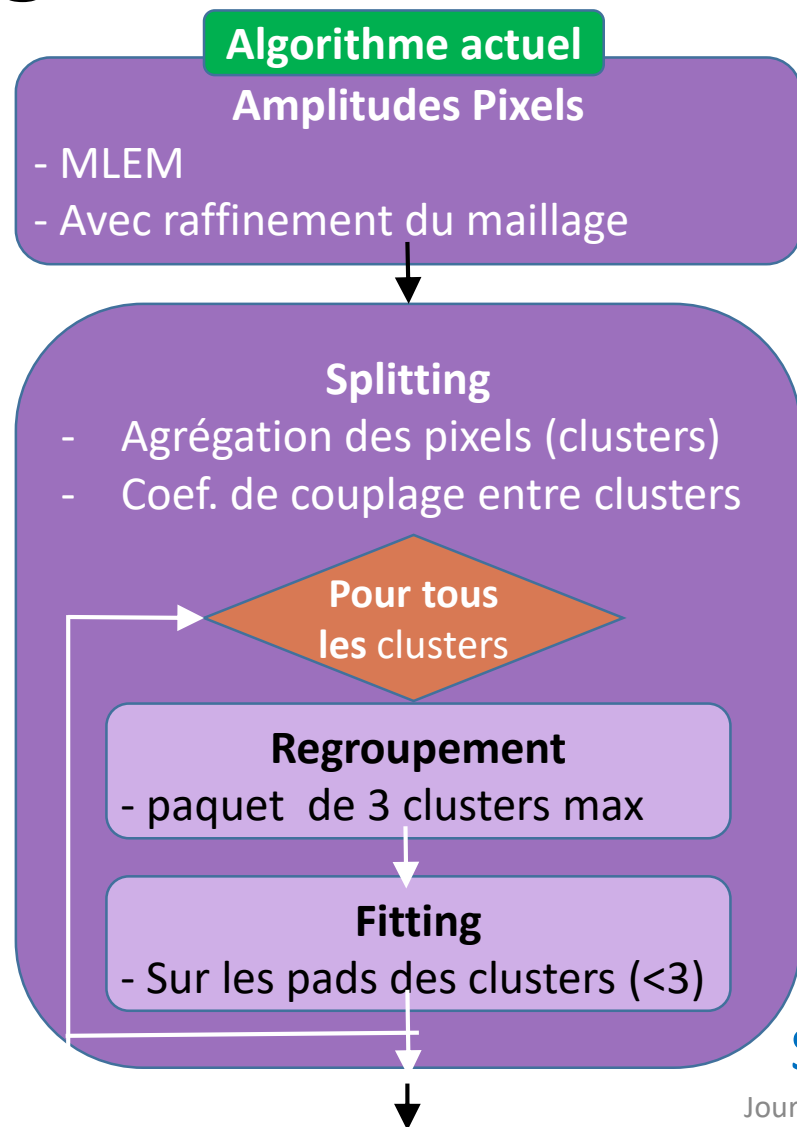
$$\sqrt{\chi^2/n_{dof}}$$

- L'algorithme EM MCH est capable de trouver le # de maxima locaux

Algorithme semble être adapté à une large classe d'applications (fonctions f)

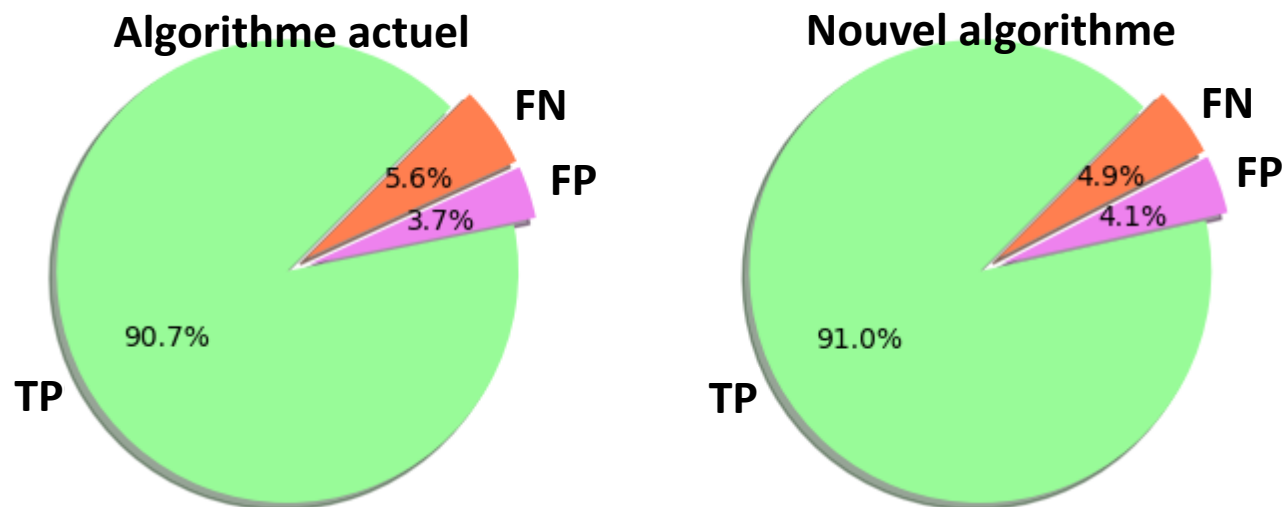


Algorithmes différents



Simplification du découpage en sous agrégats (splitting)

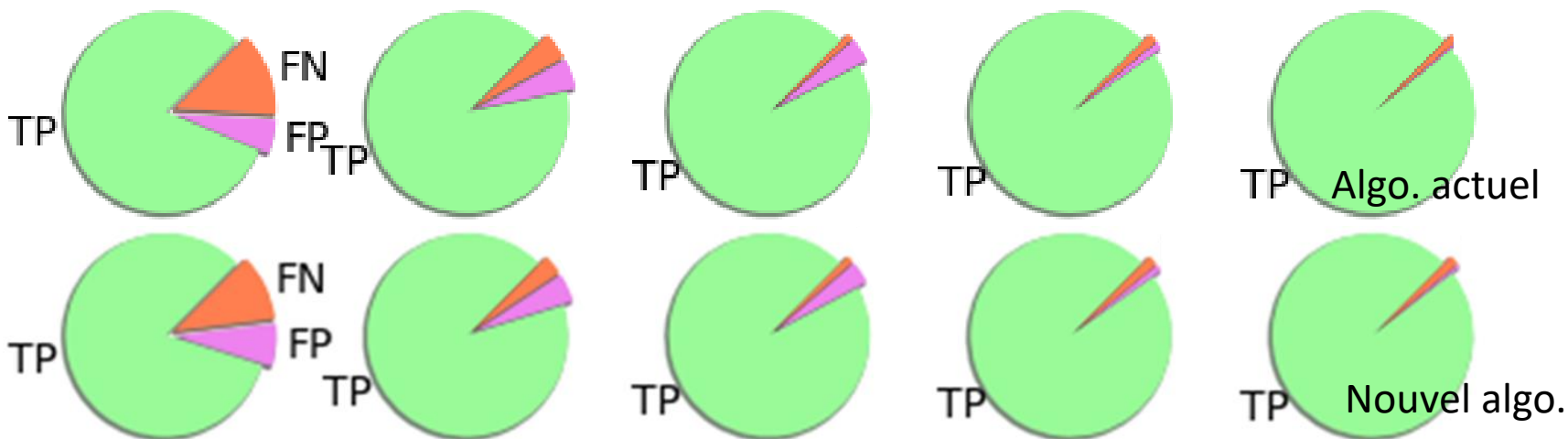
Qualité - Données de simulation



Simulation de muons jusqu'au TeV

- # evts = 50, ~ 11 k pré-clusters
- Inclus les evts sur lesquels on n'a pas de détails (particules qui bouclent, ...). Augmente les FP
- Histogrammes des résidus $d(X_{mc}, X_{algo})$ similaires

Détails par station



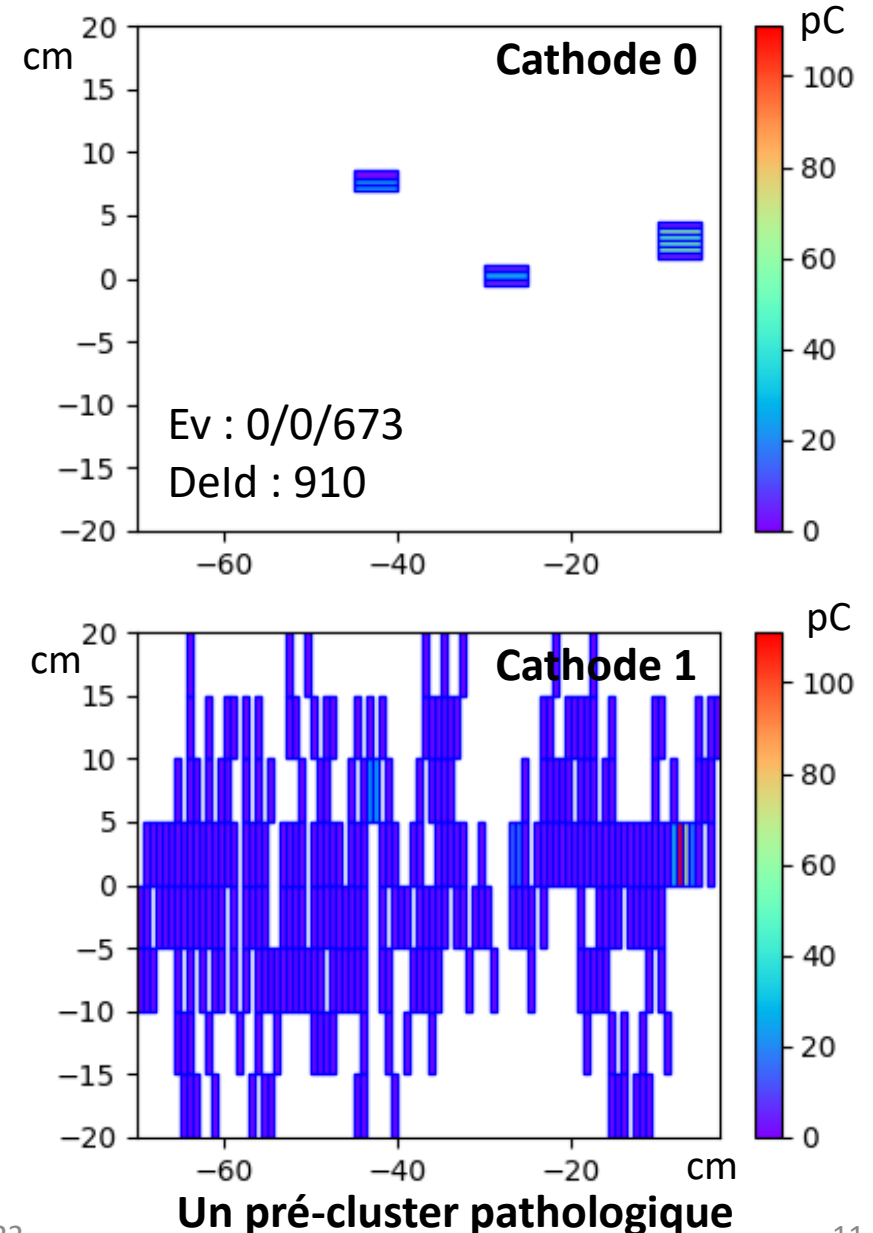
Les 2 algorithmes sont similaires en qualité

TP : True Positive
 FP : False Positive
 FN : False Negative

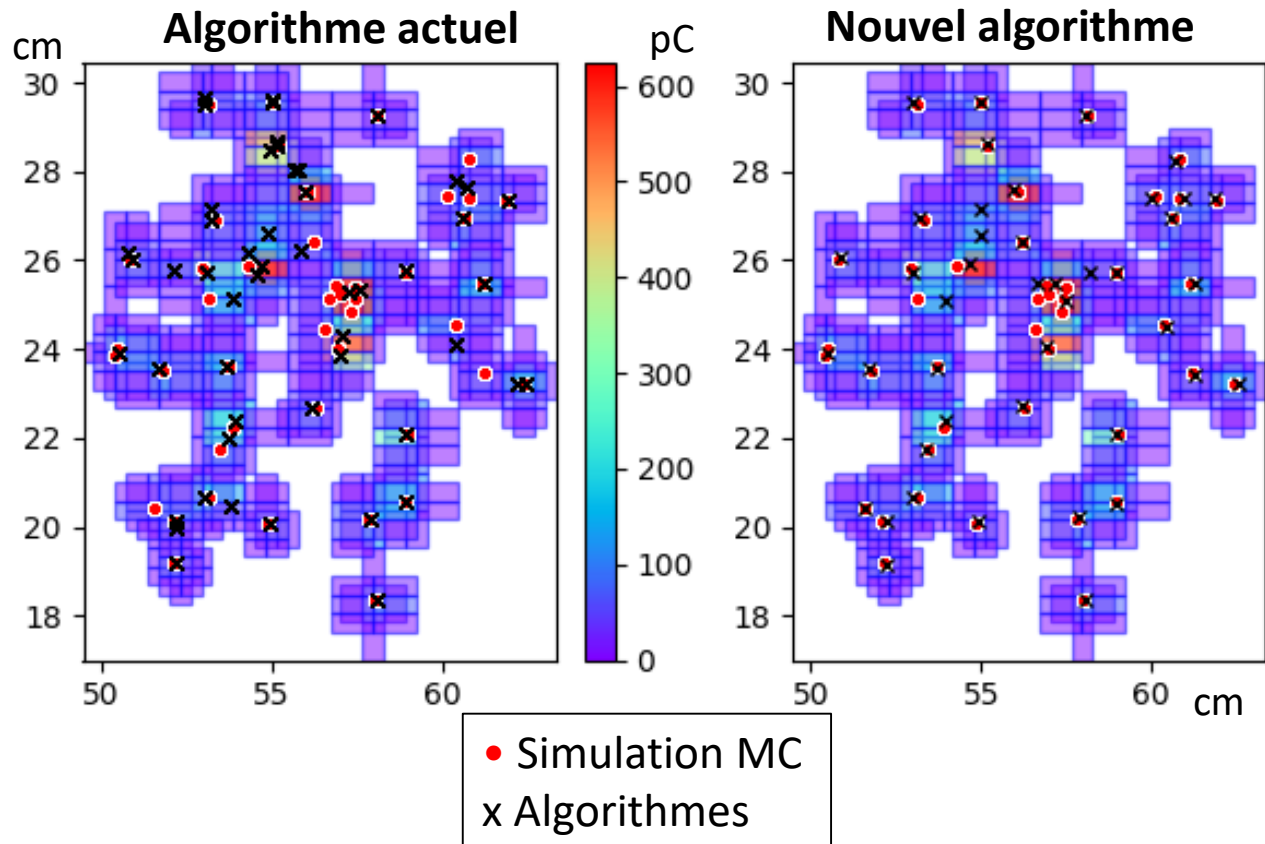
Evènements réels (Run2)

Complexification

- Bruit important pour les faibles charges
- Parfois, 1 seul plan de cathode, pads en panne
- Difficile de comparer les 2 algorithmes (pas d'automatisation)



Conclusion



Nouvel algorithme en général plus performant sur les « gros » pré-clusters

Implémentation

- Version 0 : « sale » et lente dans le *framework* d'Alice (O2) pour démarrer les tests
- Version 1 : réorganisation du code & optimisation, quasiment prête

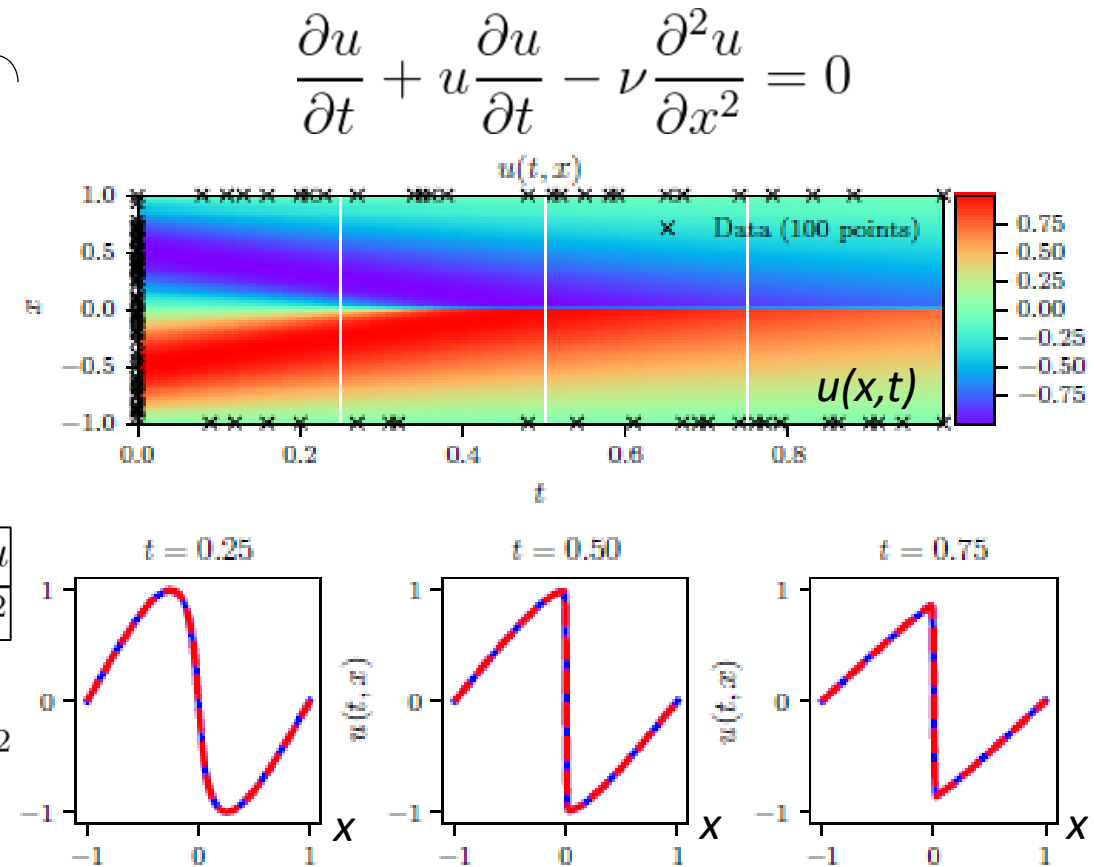
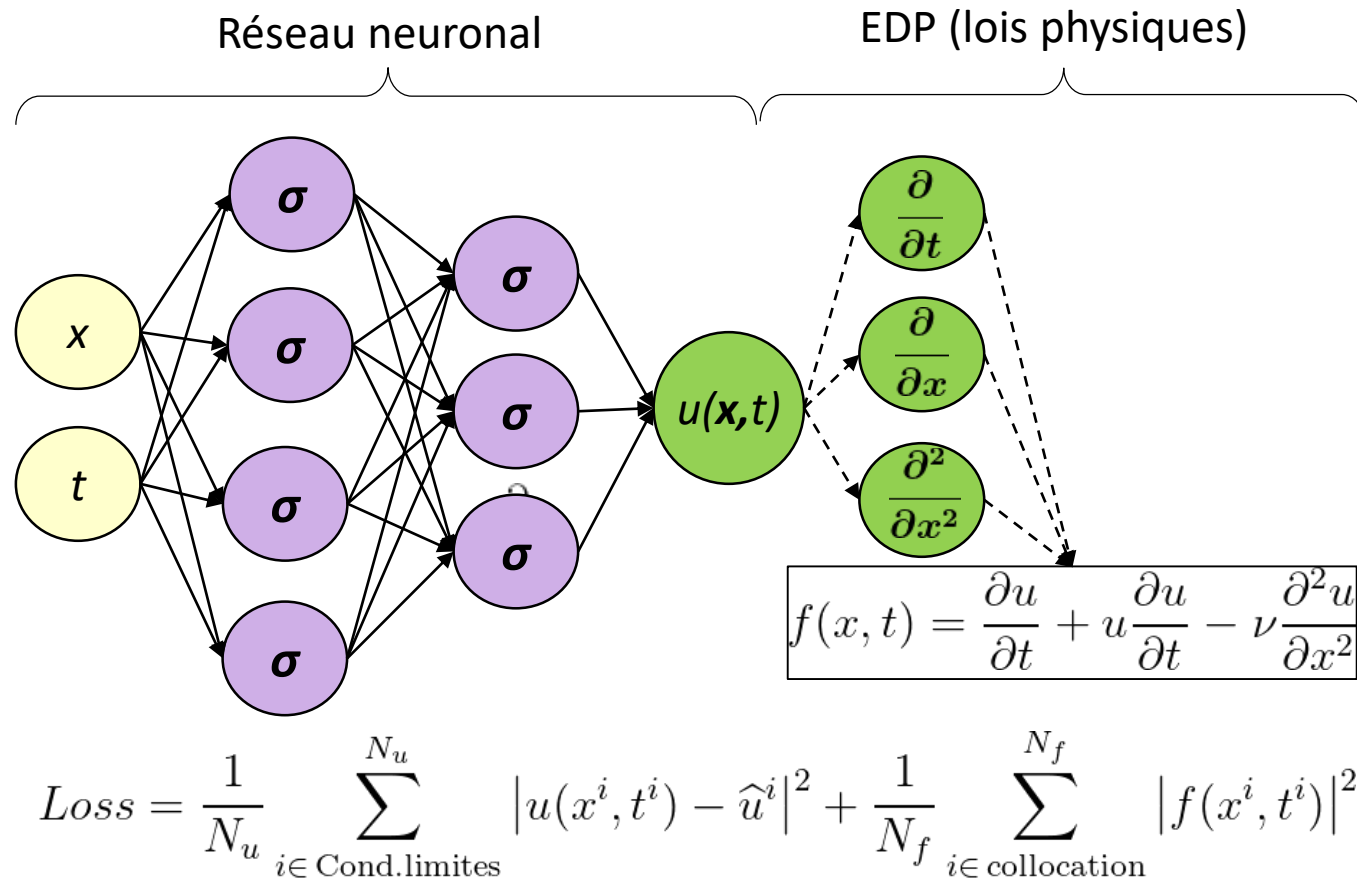
A faire

- Tests Run3
 - changement de l'électronique
 - Données/acquisitions récentes, simulations
- Ajustement des paramètres, puis optimisation
- Documentation

Conséquences

- Utilisation du MLEM MCH généralisé sur d'autres expériences (dépôts d'énergie avec une distribution connue)

Physics-Informed Neural Networks (PINN)



M. Raissi, et al - *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations* - Journal of Computational Physics 378 (2019)

Merci !
Pôle MNDL

Modélisation Numérique et Développement Logiciel

- Xemis 2 : primitives physiques (Xavier)
- EPOS : ajouts de fonctionnalités + « refactorisation » (Damien)
- Juno : Deep Learning (PINN)
- Expertise en physique et architecture logicielle (Laurent)

Backup

Previous point – The 5th February

EM algorithme

Maximisation de la vraisemblance (complète)

- \mathbf{X} données observées
- Modélisées par une loi de proba de paramètres ϑ
- \mathbf{Z} données latentes
- Likelihood, pas de solution

$$L(\boldsymbol{\theta}; \mathbf{X}) = p(\mathbf{X} | \boldsymbol{\theta}) = \int p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}) d\mathbf{Z} = \int p(\mathbf{X} | \mathbf{Z}, \boldsymbol{\theta}) p(\mathbf{Z} | \boldsymbol{\theta}) d\mathbf{Z}$$

Connaissant $\boldsymbol{\theta}^{(t)}$,

- étape E - calcul de l'espérance Q sur \mathbf{Z}

$$Q(\boldsymbol{\theta} | \boldsymbol{\theta}^{(t)}) = E_{\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta}^{(t)}} [\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})]$$

- étape M – maximisation de Q

$$\boldsymbol{\theta}^{(t+1)} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} | \boldsymbol{\theta}^{(t)})$$

- On montre qu'améliorer Q , améliore $L(\mathbf{X}, \boldsymbol{\theta})$