UNITE DE SENSOMETRIE ET CHIMIOMETRIE

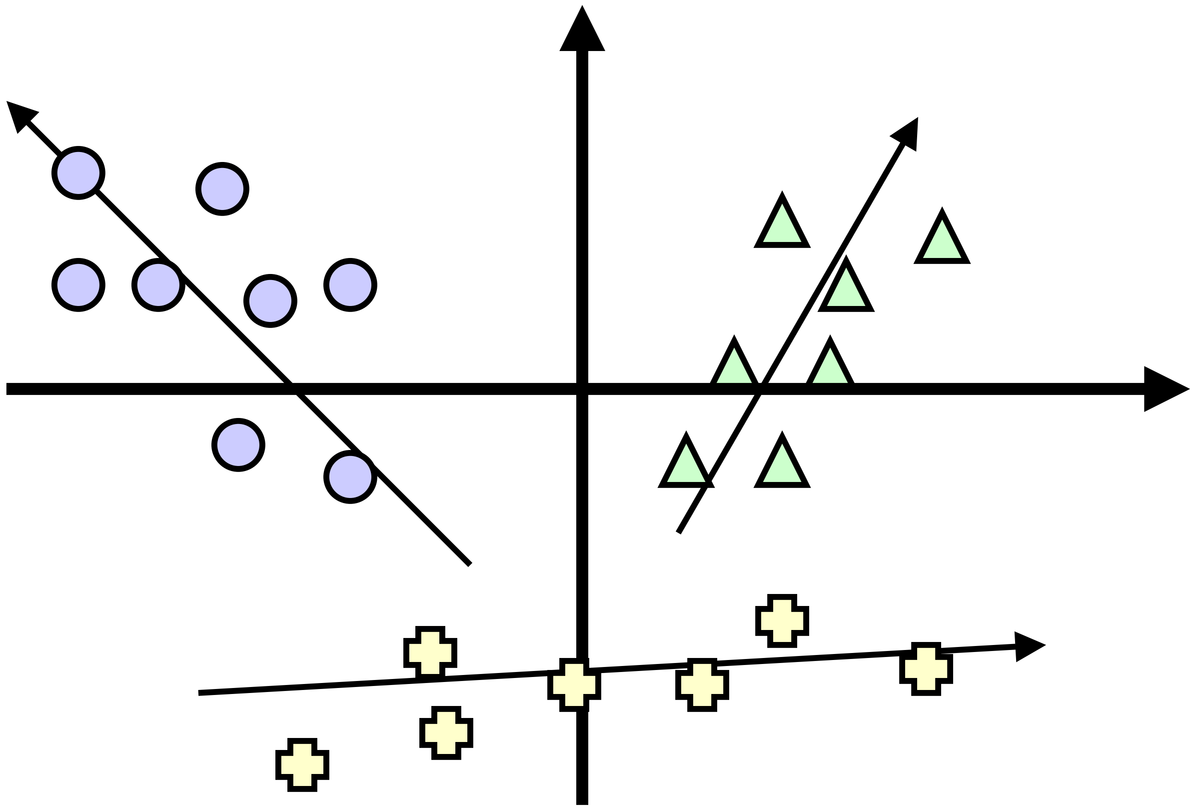
Les fonctions radiales de base

présentation générale

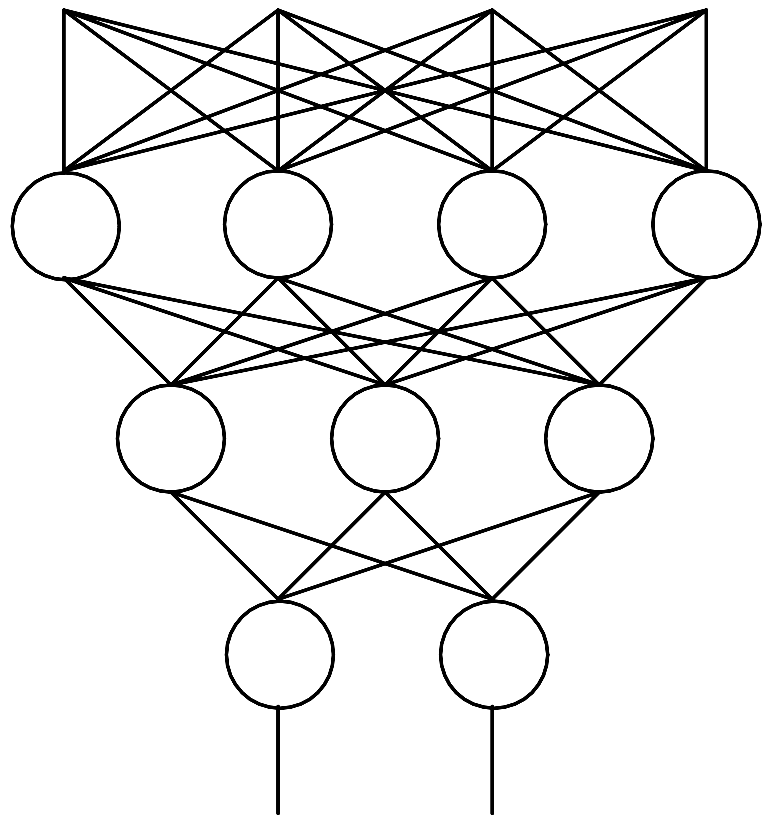
et premiers essais

Dominique Bertrand

Recherche de méthodes prédictives non linéaires



Réseau de neurones
à rétropropagation de l'erreur



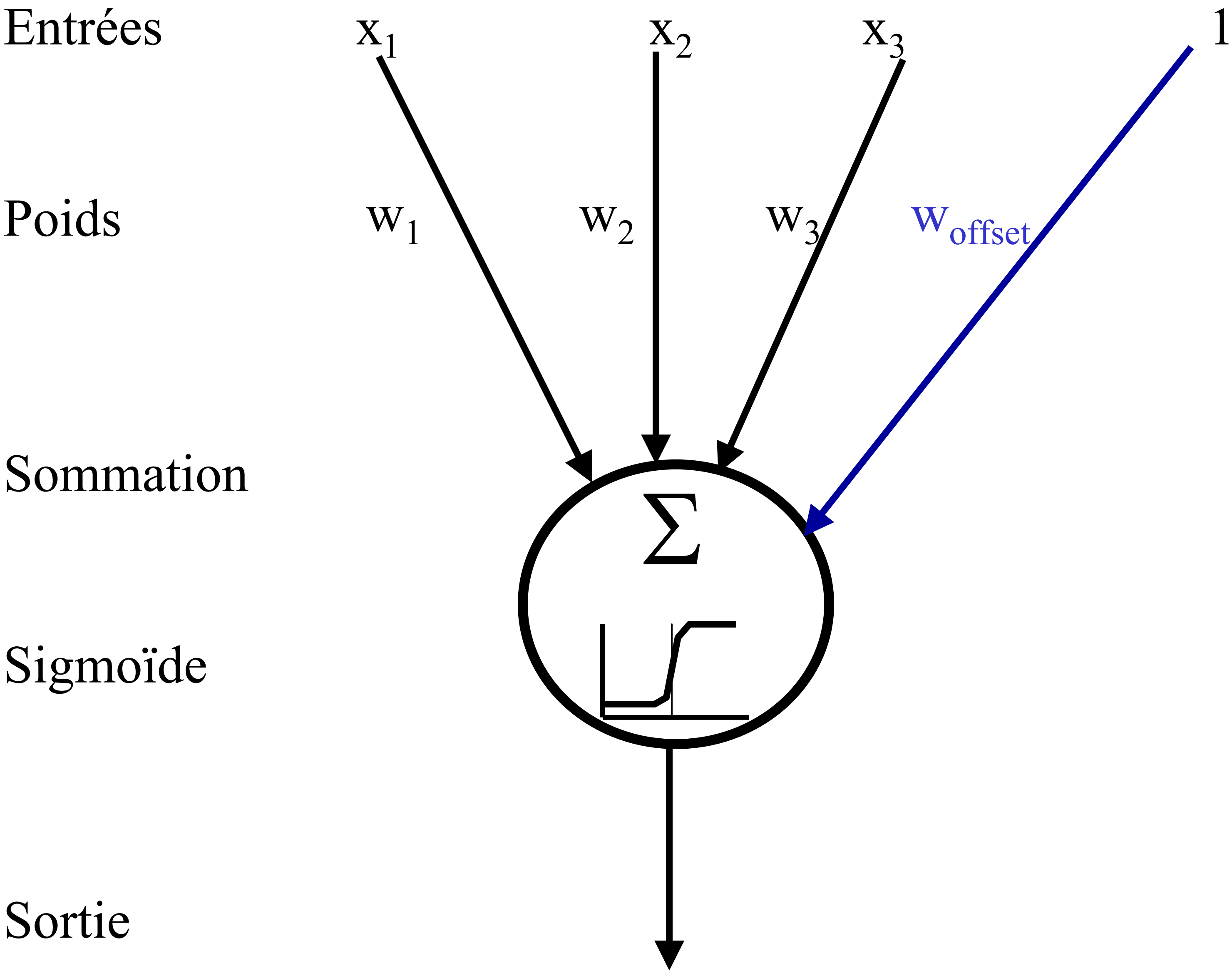
Couche d'entrée

Première couche
active

Deuxième couche
active ("cachée")

Troisième couche
active

Sortie



Neurone élémentaire

Principe fondamental des fonctions radiales de base

Espace initial de dimension *p*

Soit **x** une observation (*p* valeurs)

On utilise des fonctions non linéaires pour créer un nouvel espace dont les dimensions peuvent être supérieures à p.

On fait donc correspondre à **x** un vecteur

$$\varphi(\mathbf{x}) = [\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}),..., \varphi_M(\mathbf{x})]'$$

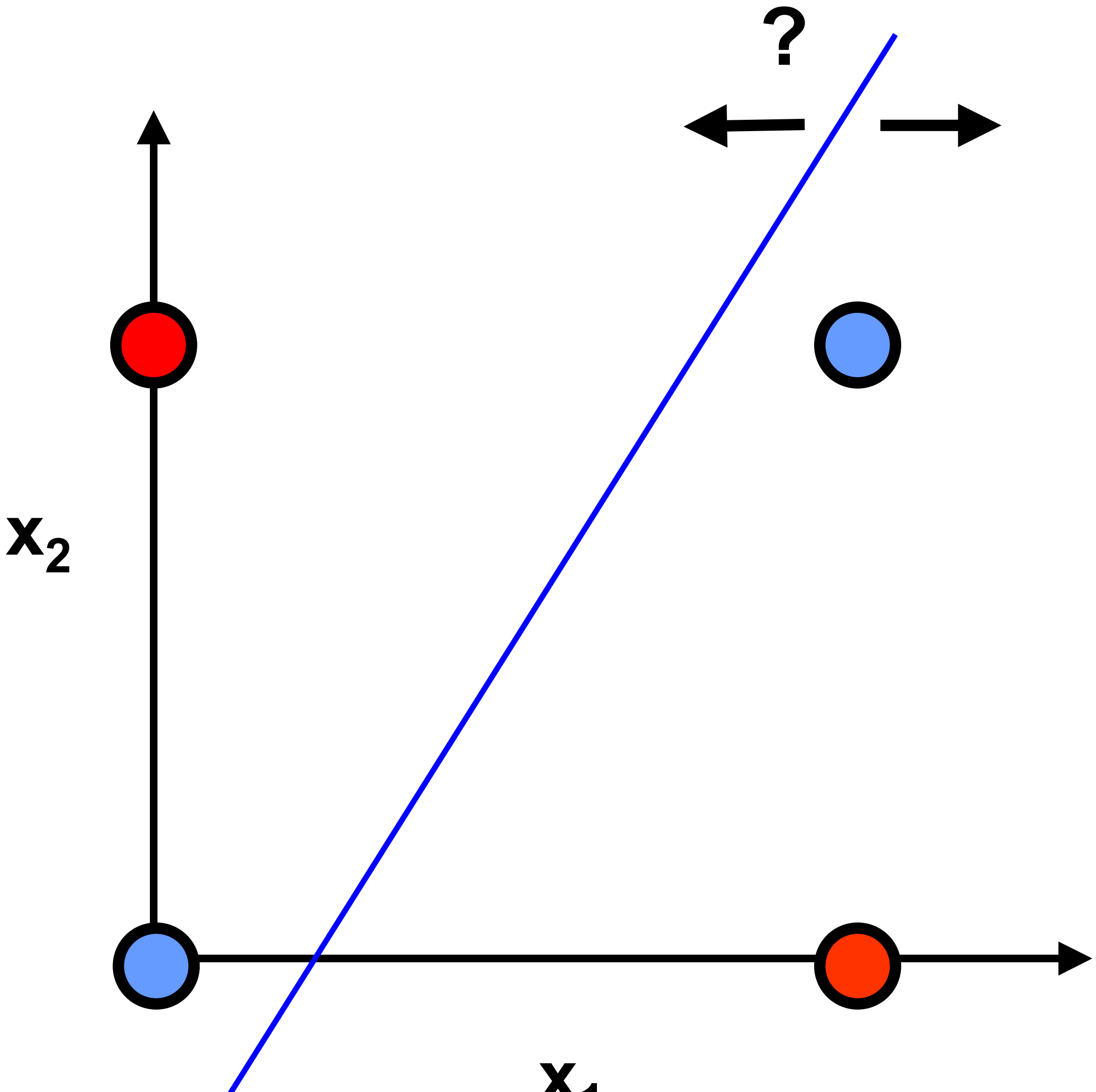
On applique les méthodes classiques (régression, discrimination ...) sur les vecteurs $\varphi(\mathbf{x})$ au lieu de **x**

Les fonctions radiales

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \rightarrow \begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1n} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{n1} & \varphi_{n2} & \cdots & \varphi_{nn} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

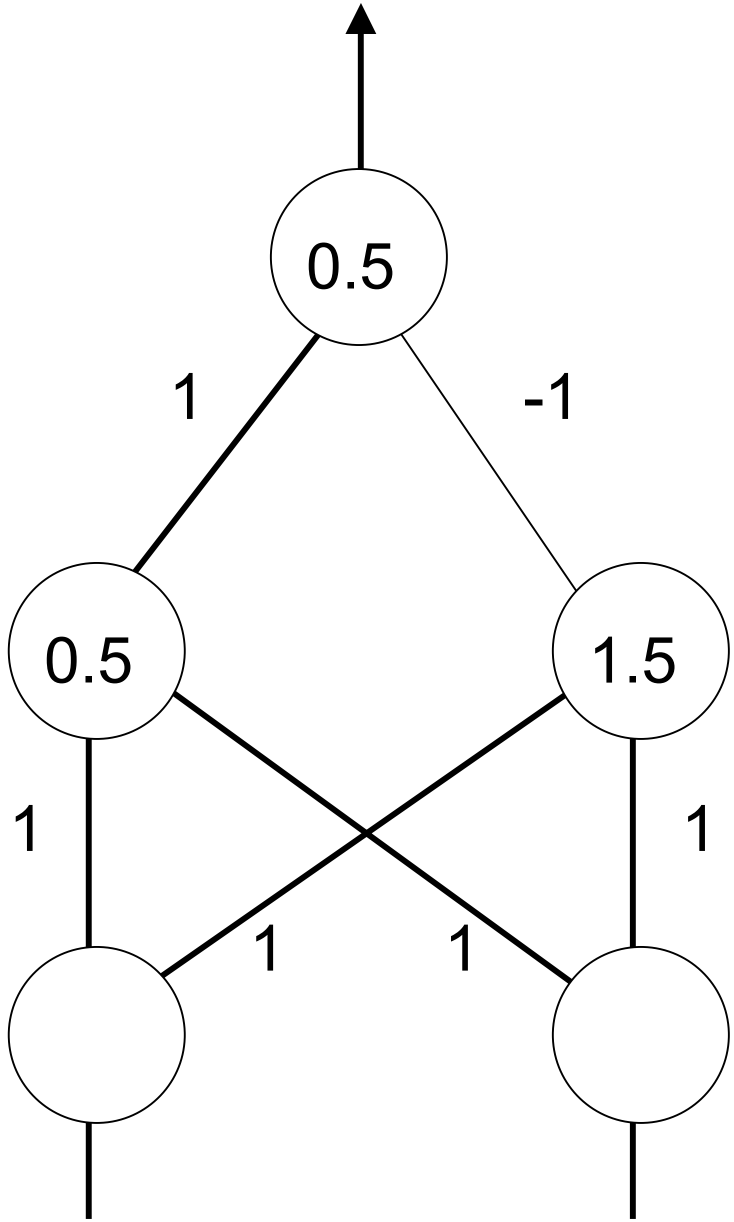
avec:

$$\varphi_{ij} = \varphi\left(\left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|\right)$$



x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Le problème *XOR* ('ou' exclusif)



x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Le problème *XOR* résolu par
les réseaux de neurones classiques

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

→

d_{xa}	d_{xb}
0	2
1	1
1	1
2	0

→

φ_a	φ_b	b
1	0.135	1
0.368	0.368	1
0.368	0.368	1
0.135	1	1

X **y** **d^2** φ

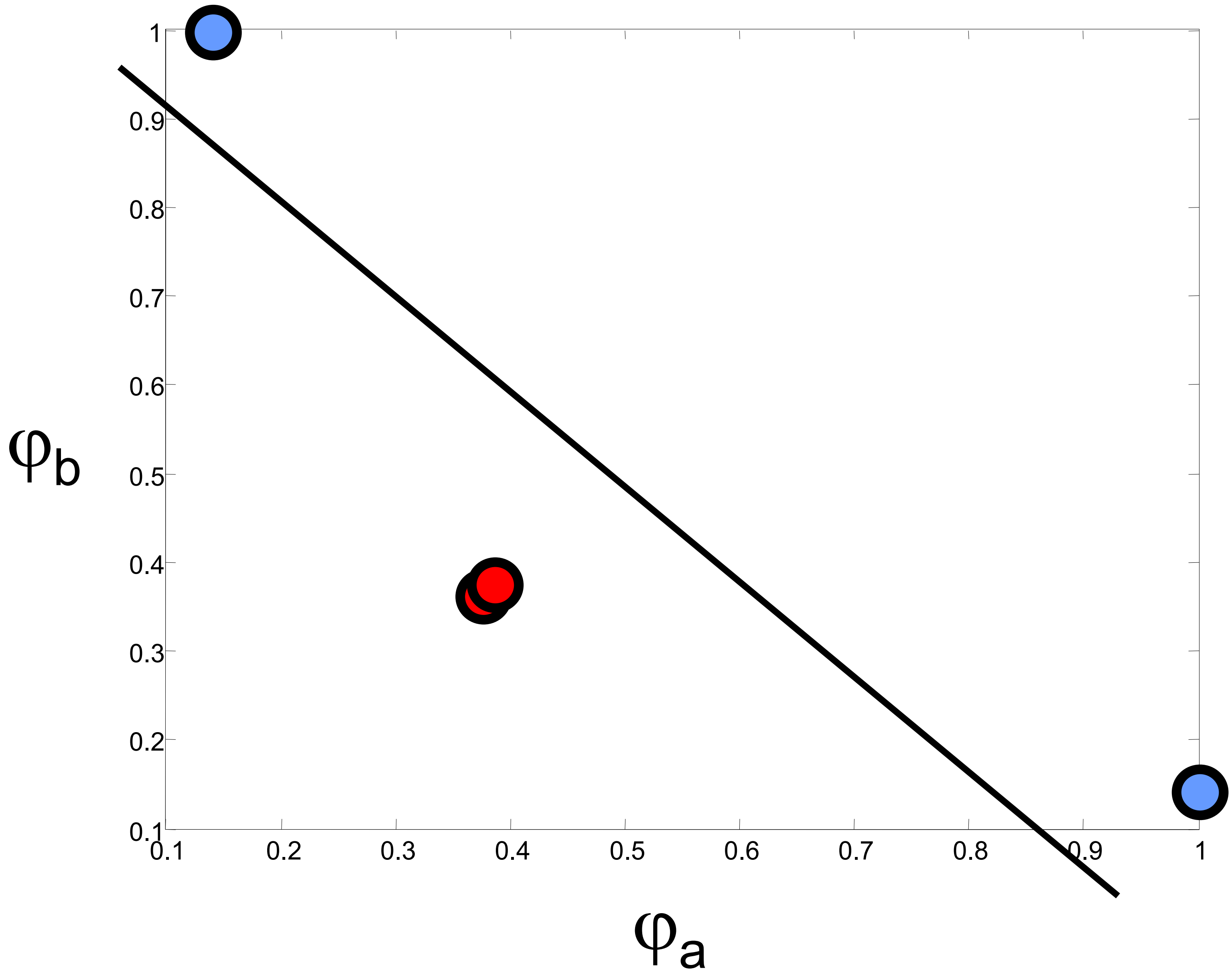
$$\varphi_a = \exp(-\|\boldsymbol{x}-\boldsymbol{a}\|^2)$$
$$\varphi_b = \exp(-\|\boldsymbol{x}-\boldsymbol{b}\|^2)$$

$\exp(0)=1$
$\exp(-1)=0.368$
$\exp(-2)=0.135$

$$\boldsymbol{a} = [0\ 0]^{\mathrm{T}} \quad \boldsymbol{b} = [1\ 1]^{\mathrm{T}}$$

Le problème *XOR* résolu par
les fonctions radiales de base

Le problème *XOR* résolu par les fonctions radiales de base



y
0
1
1
0

y

φ_a	φ_b	b
1	0.135	1
0.368	0.368	1
0.368	0.368	1
0.135	1	1

φ

y =φ w

On cherche w qui résout ce système par exemple:

w= φ^+ y

φ^+ : inverse généralisé

ou régression OLS

Le problème *XOR* résolu par les fonctions radiales de base

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

→

d_{xa}	d_{xb}
0	2
1	1
1	1
2	0

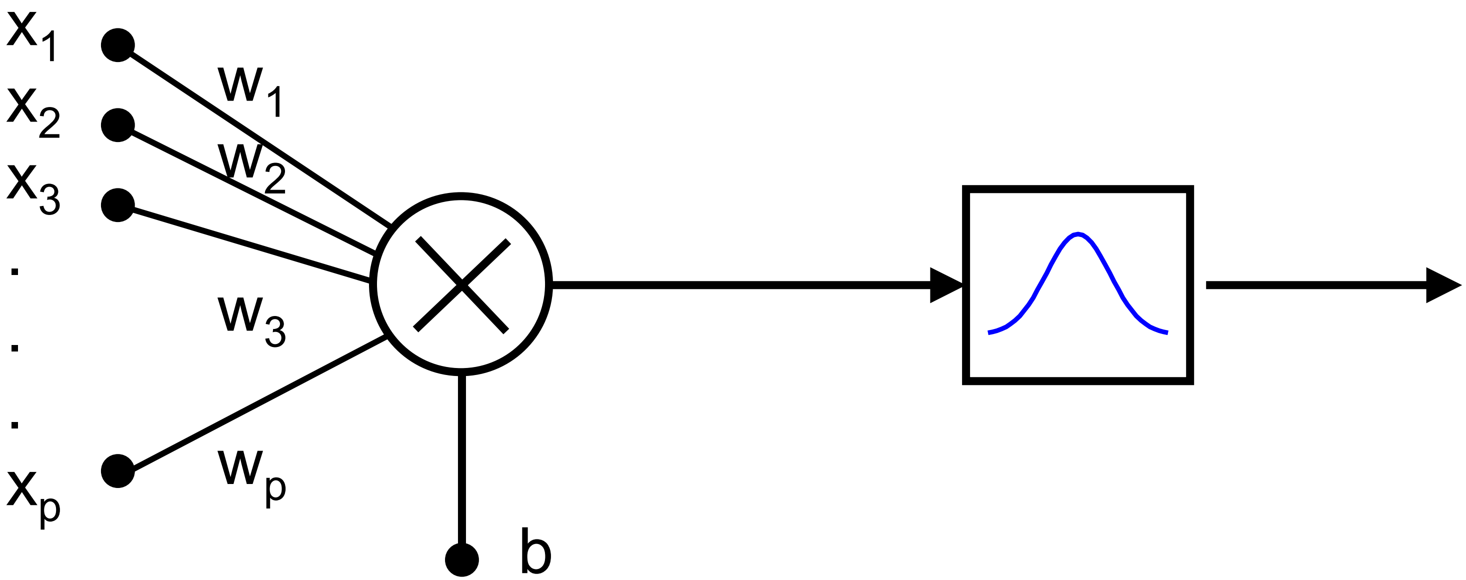
→

φ_a	φ_b	b
1	0.135	1
0.368	0.368	1
0.368	0.368	1
0.135	1	1

```
1 >> x
x =
      0        2
      1        1
      1        1
      2        0
2 >> x1=exp(-x)
x1 =
     1.0000    0.1353
     0.3679    0.3679
     0.3679    0.3679
     0.1353    1.0000
```

```
3 >> x1=[x1 ones(4,1)]
x1 =
    1.0000    0.1353    1.0000
    0.3679    0.3679    1.0000
    0.3679    0.3679    1.0000
    0.1353    1.0000    1.0000
4 >> w=inv((x1'*x1))*x1'*y
w =
   -2.5027
   -2.5027
    2.8413
```

Exemple de calculs sous Matlab



Entrée:
distance entre
x et **w**
Multiplication par b

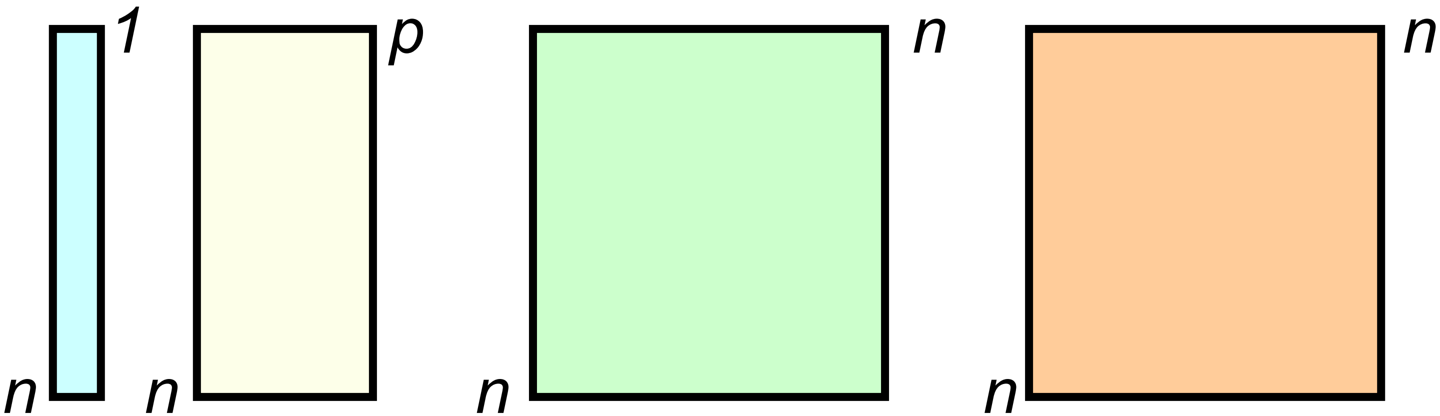
Activation par
une fonction

Présentation des fonctions radiales de base dans un style "réseau de neurones"

Premiers essais

Principe:
Utilisation d'un système très simple de fonctions radiales de base

Étape d'étalonnage :



$\mathbf{y}$	$\mathbf{X}$	$\mathbf{D^2}$	$\varphi = [\exp(-b * d^2)]$
Données à prédire	Données prédictive	Distance euclidienne	Fonctions radiales

On établit le modèle

$y = \varphi\ \mathbf{w}$ par $\mathbf{w} = \varphi^{-1}\mathbf{y}$ (simple inversion de φ)

Premiers essais (suite 1)

Principe:
Utilisation d'un système très simple de fonctions radiales de base

Étape d'étalonnage :

$$\varphi = [\exp(-b * d^2)]$$

Fonctions radiales

On établit le modèle

$y = \varphi\ \mathbf{w}$ par $\mathbf{w} = \varphi^{-1}\mathbf{y}$ (simple inversion de φ)

Le seul paramètre à ajuster est b (spread).

Premiers essais (suite 2)

<u>Principe</u>:
Utilisation d'un système très simple de fonctions radiales de base
Les collections sont divisées aléatoirement
en un jeu d'étalonnage et
un jeu de validation comprenant chacun
environ la moitié des observations.

Tous les modèles rbf (*radial basis function*) sont testés en faisant varier les valeurs du paramètre *b* par pas constants.
Dans chaque essai, on teste 50 valeurs de *b*.

Les modèles sont établis sur le jeu d'étalonnage,
et testés sur les jeux de validation.

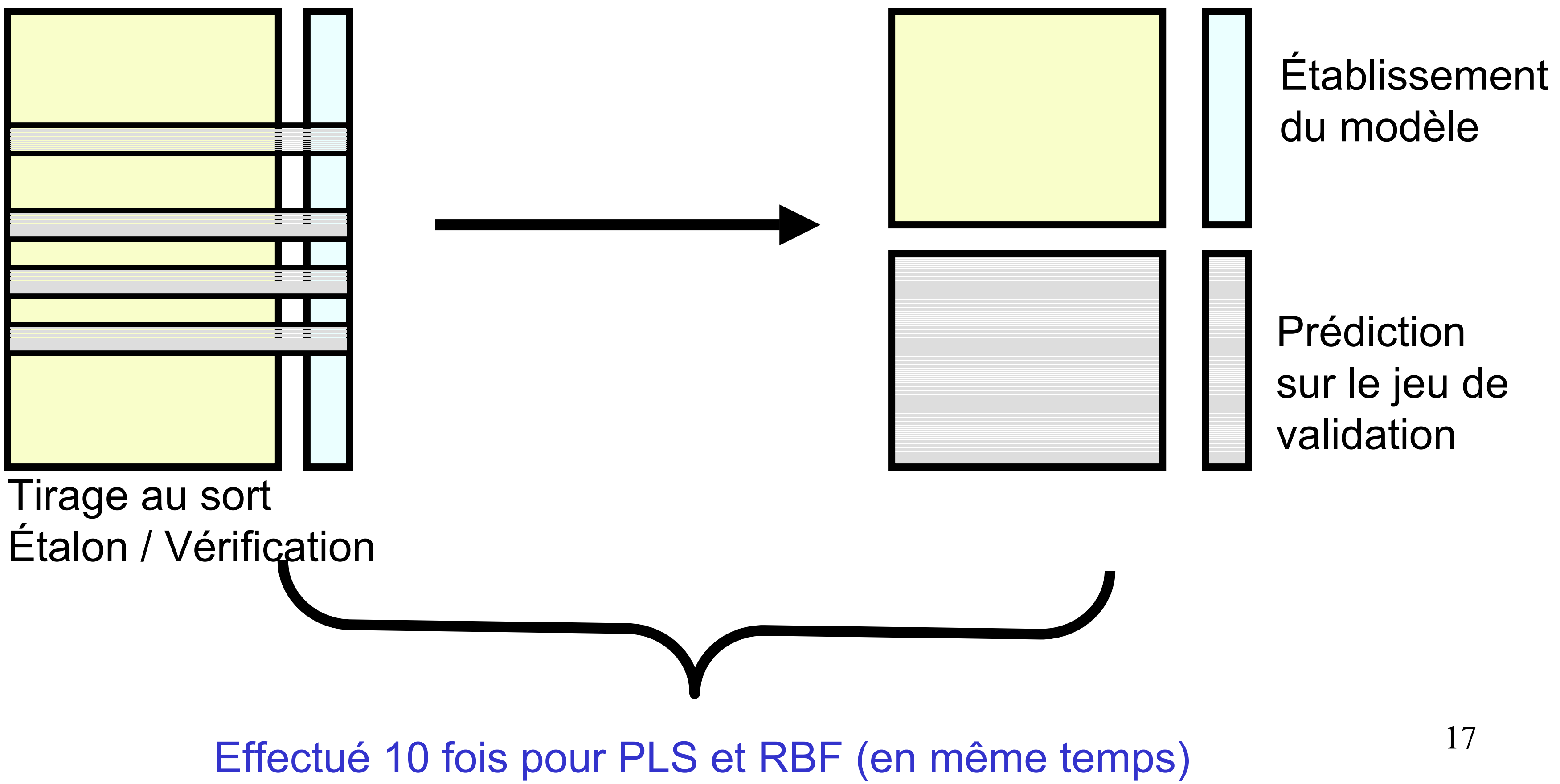
Les résultats sont comparés avec ceux obtenus par
la régression *partial least square*, dans les même conditions en testant les modèles comprenant de 1 à 20 dimensions.

Premiers essais (suite 3)

<u>Principe</u>:

Pour chaque collection, on effectue 10 fois le tirage au sort des jeux d'étalonnage et de vérification.

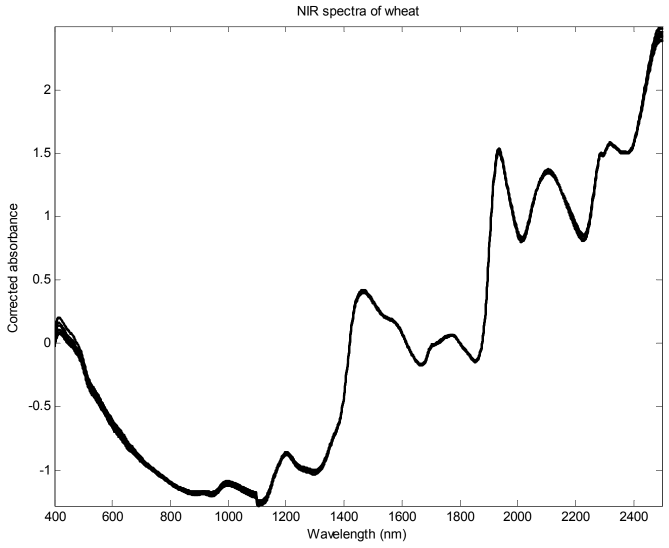
La qualité des résultats de chaque essai est estimée par l'erreur résiduelle *(root mean square error)* dans le jeu de validation.

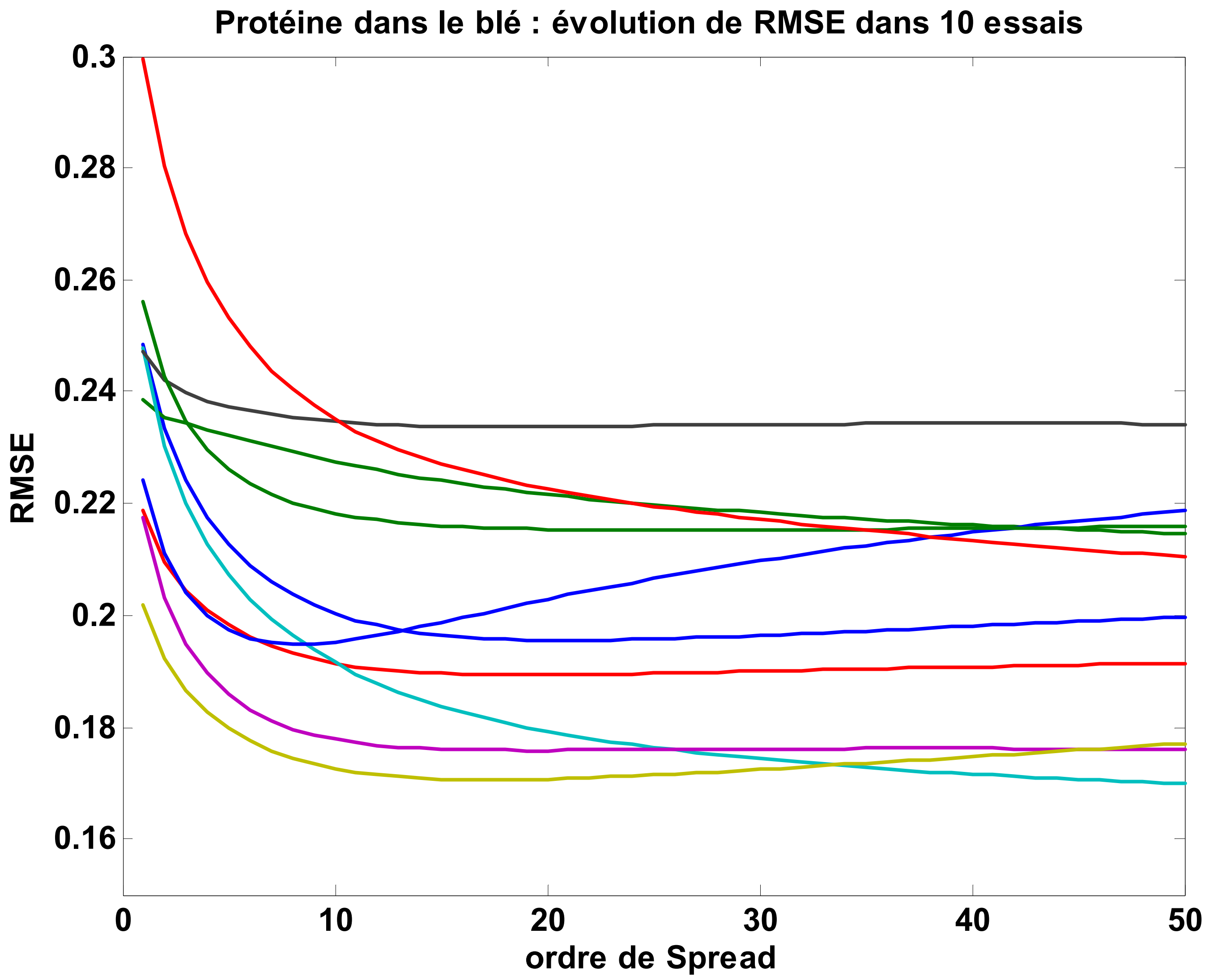


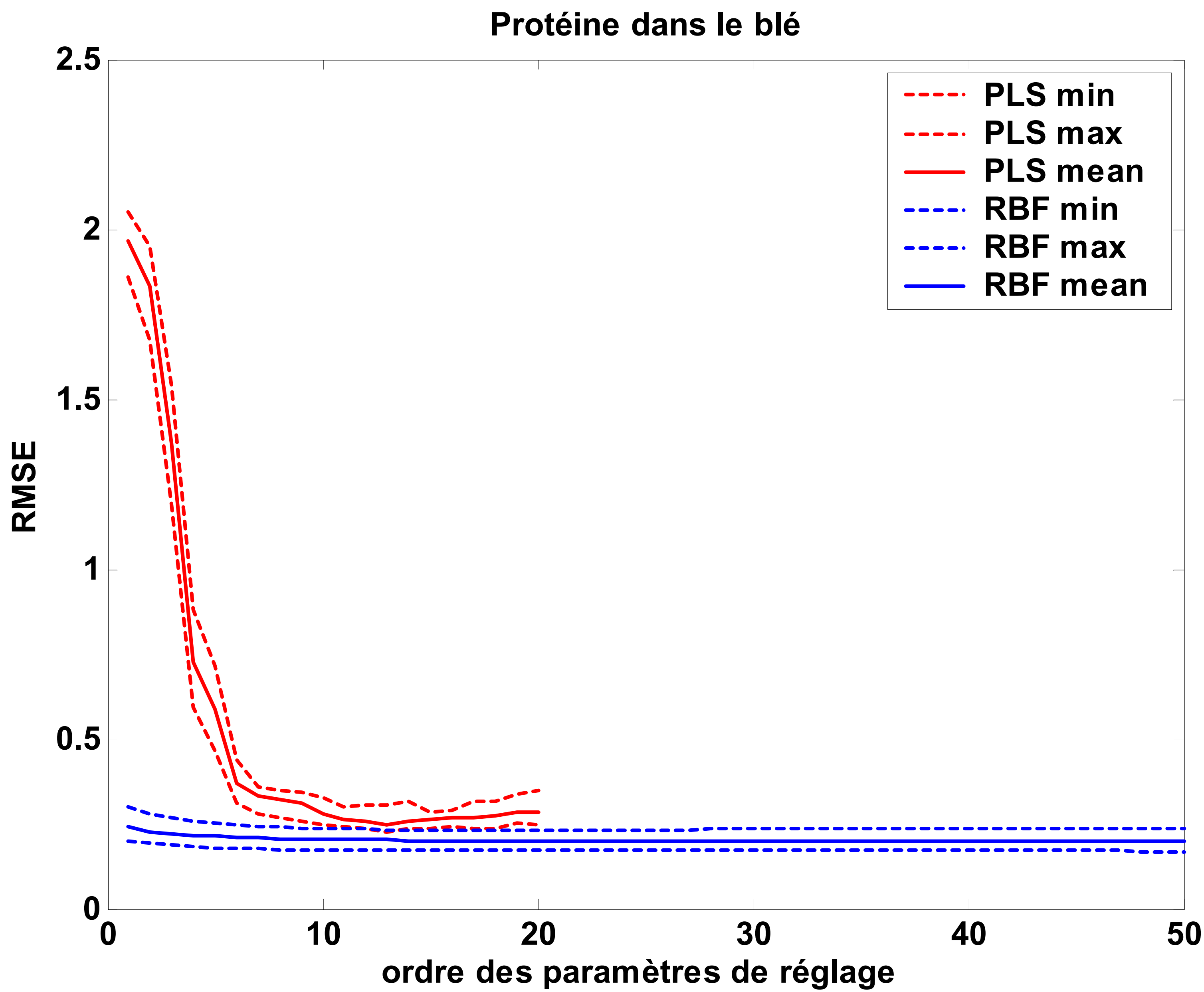
Protéines dans le blé

140 spectres proche infrarouge
Divisés en 70 dans jeu d'étalonnage
70 en jeu de validation

"Spread factor": $2*10^{-5}$ à 0.0010



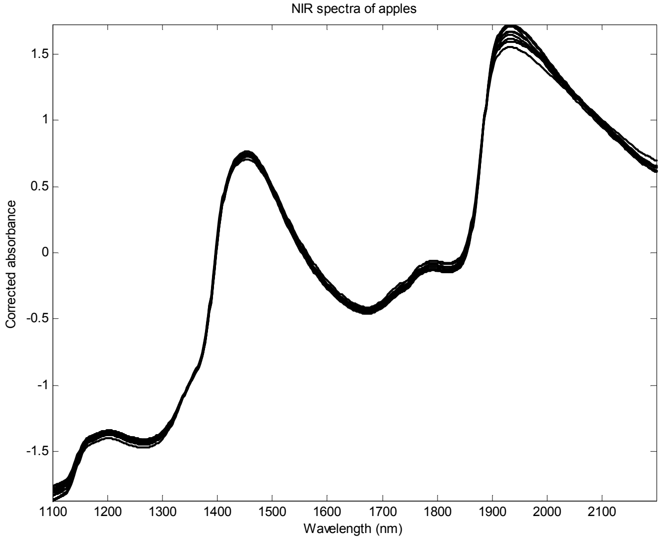

Protéine dans le blé : évolution de RMSE dans 10 essais
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0.16
RMSE
0
10
20
30
40
50
ordre de Spread

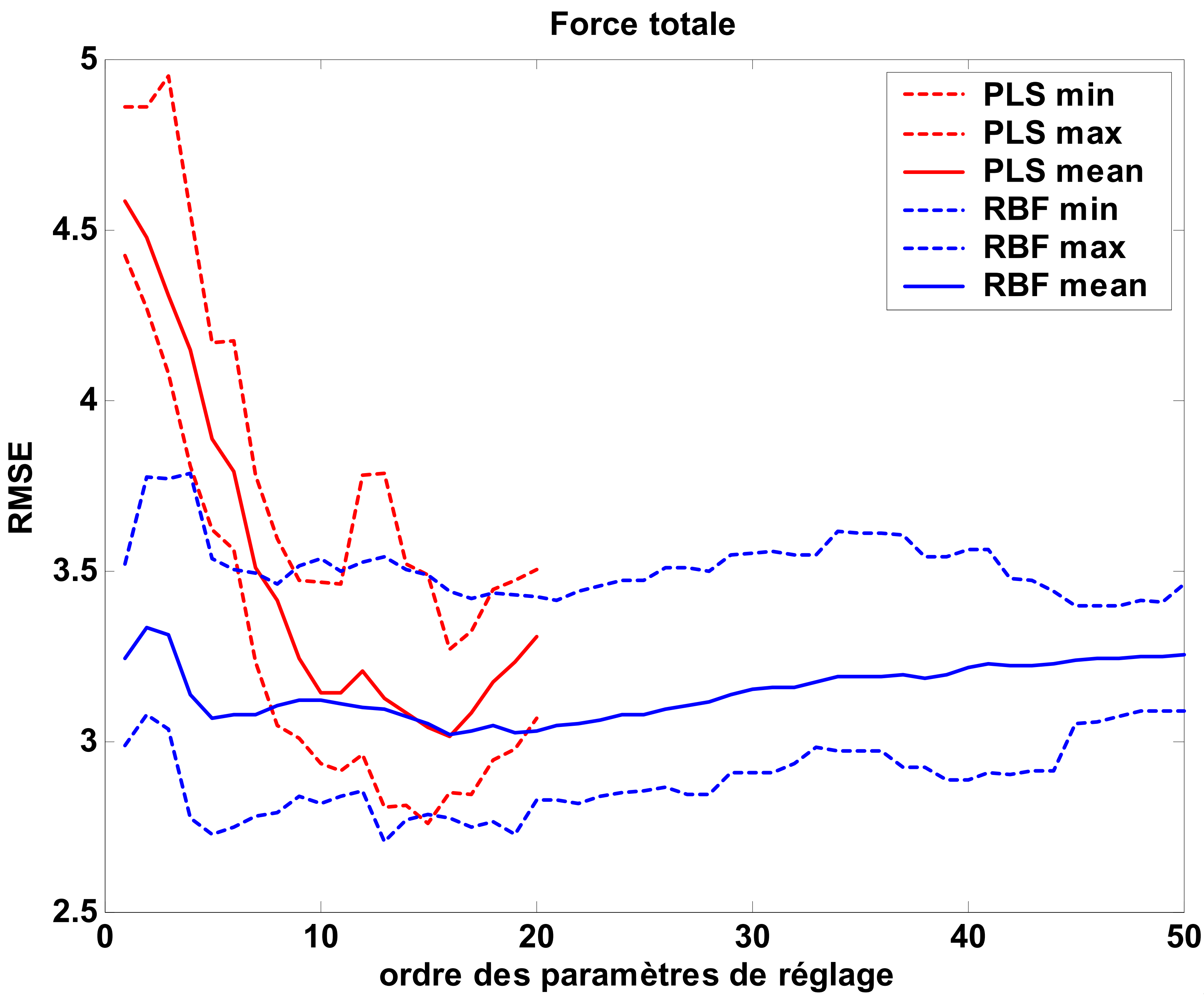

Protéine dans le blé
RMSE
ordre des paramètres de réglage
PLS min
PLS max
PLS mean
RBF min
RBF max
RBF mean
2.5
2
1.5
1
0.5
0
0
10
20
30
40
50

Force totale pommes

337 spectres proche infrarouge
Divisés en 187 dans jeu d'étalonnage
150 en jeu de validation

"Spread factor": 0.01 à 0.5000

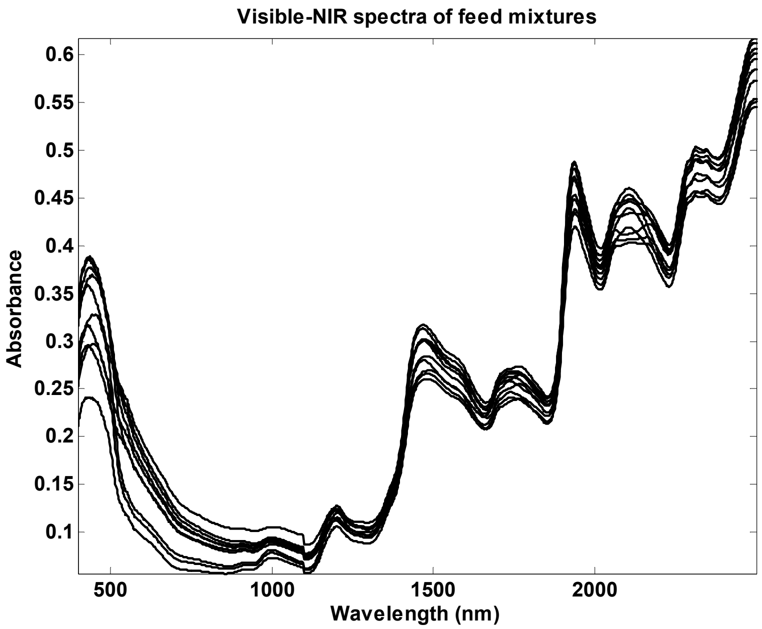


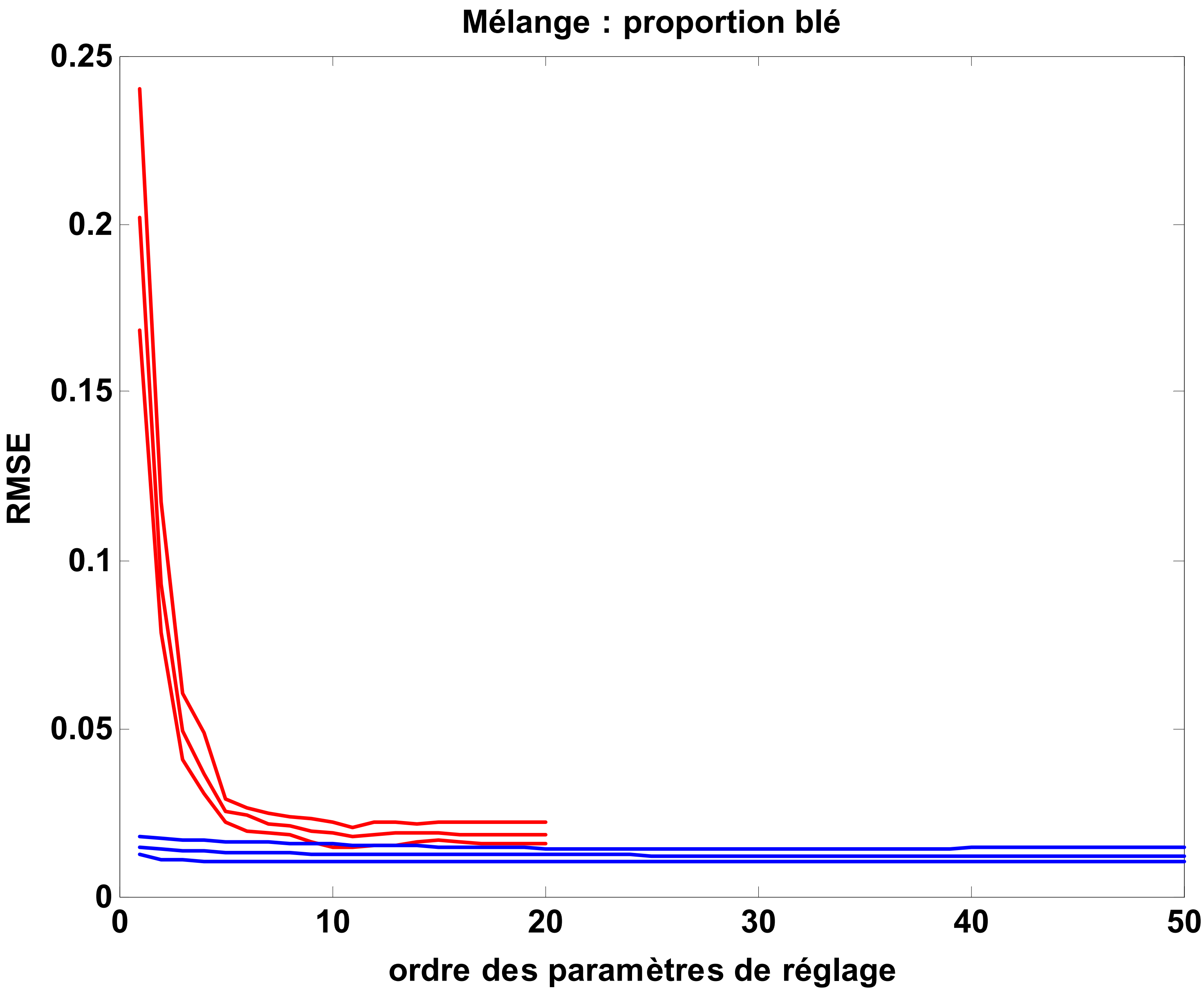

Force totale
PLS min
PLS max
PLS mean
RBF min
RBF max
RBF mean
RMSE
ordre des paramètres de réglage
5
4.5
4
3.5
3
2.5
0
10
20
30
40
50

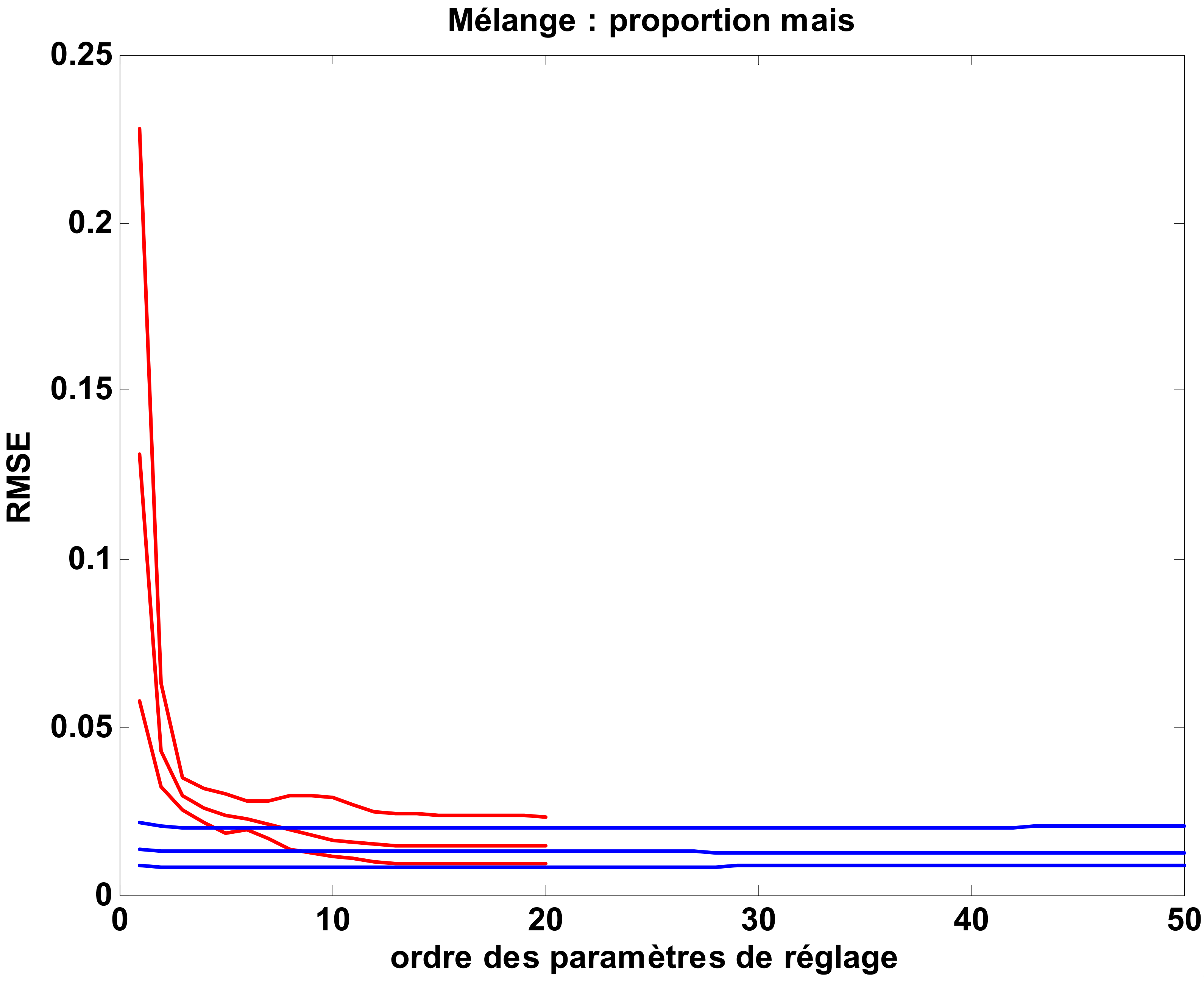
Proportion mélanges

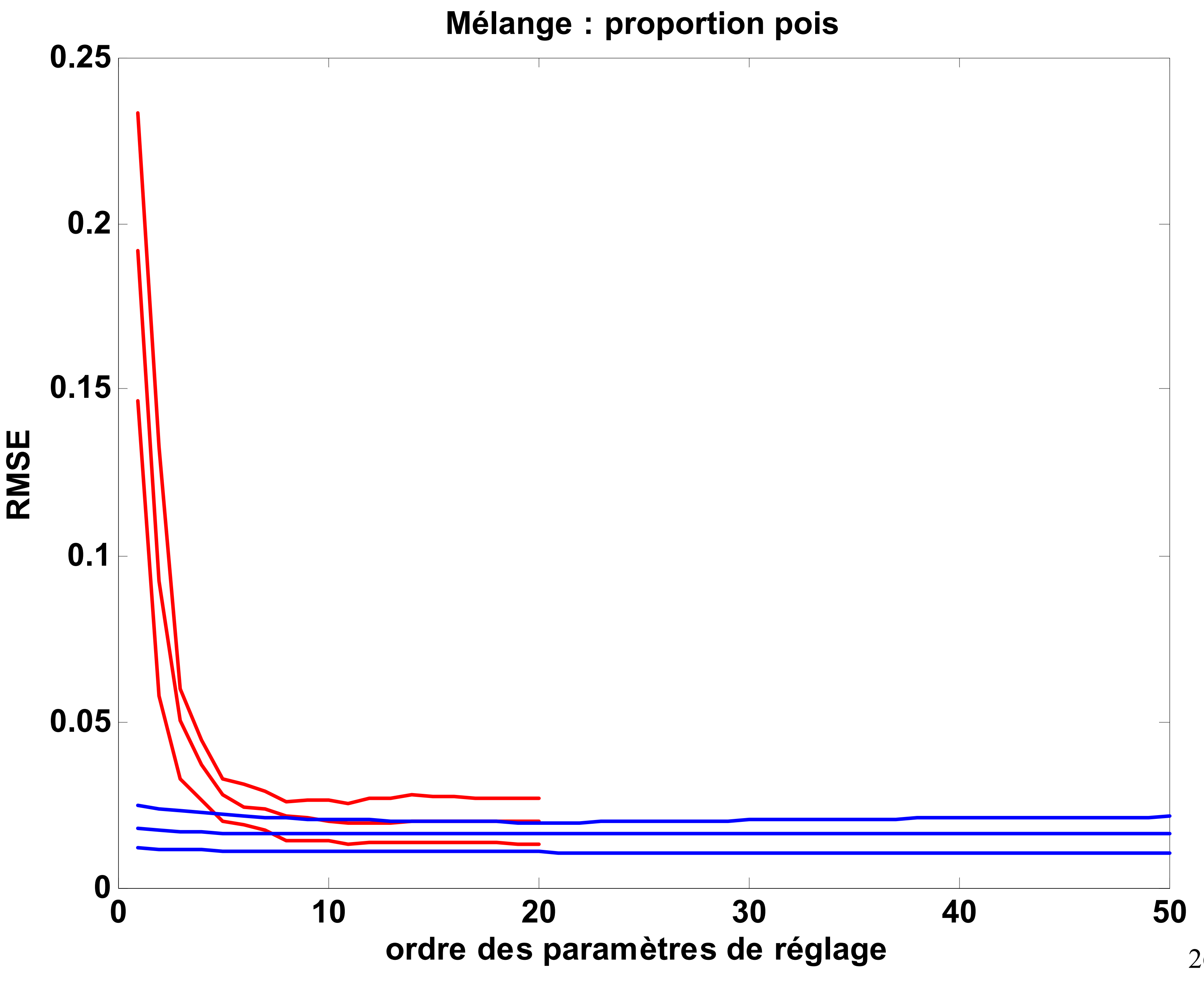
63 spectres proche infrarouge
Divisés en 33 dans jeu d'étalonnage
30 en jeu de validation

*"Spread factor": 2*10^{-5} à 0.0010*

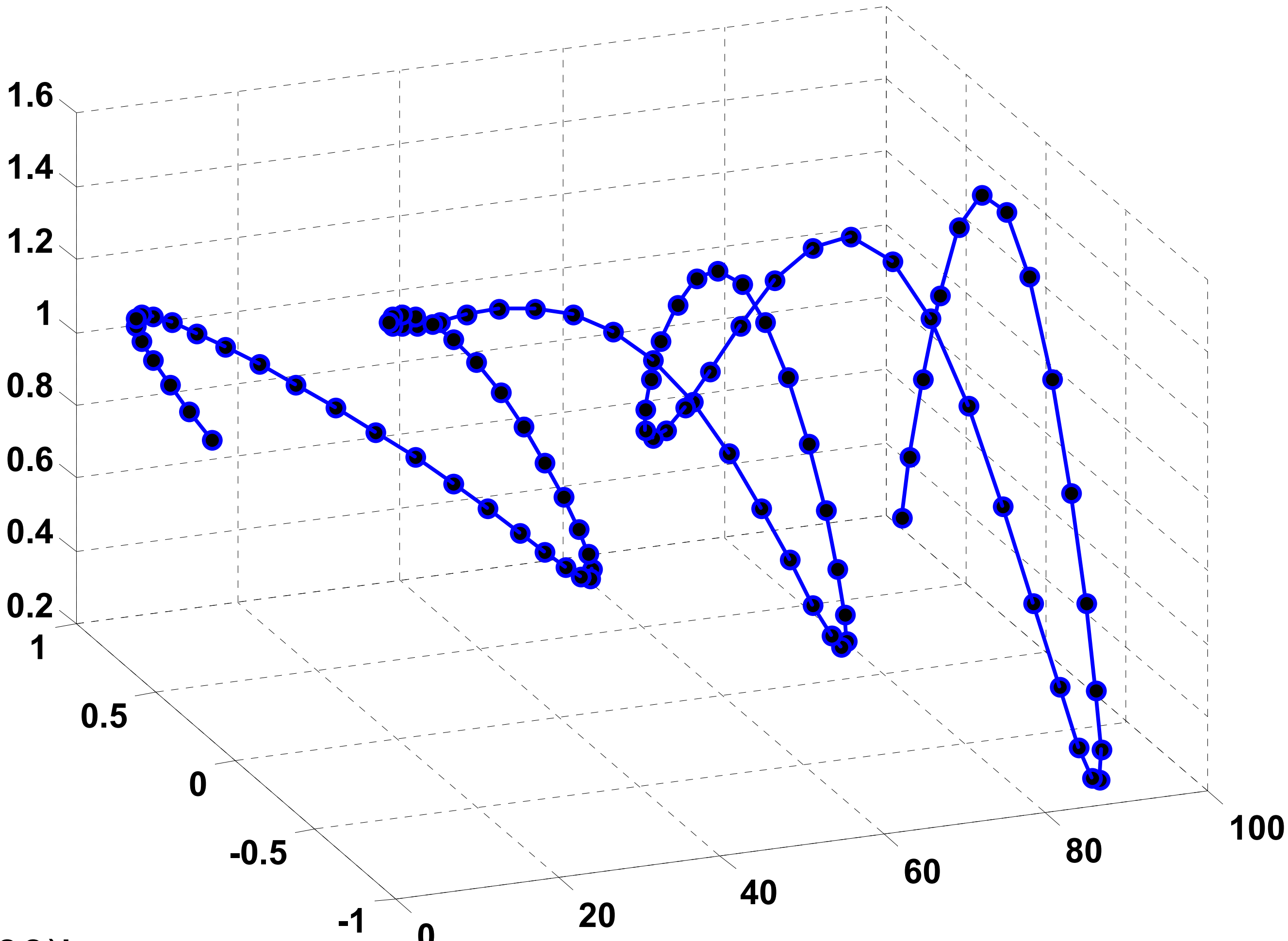



Mélange : proportion blé
RMSE
ordre des paramètres de réglage
0.25
0.2
0.15
0.1
0.05
0
0
10
20
30
40
50


Mélange : proportion mais
0.25
0.2
0.15
0.1
0.05
0
RMSE
0
10
20
30
40
50
ordre des paramètres de réglage


Mélange : proportion pois
RMSE
0.25
0.2
0.15
0.1
0.05
0
0
10
20
30
40
50
ordre des paramètres de réglage

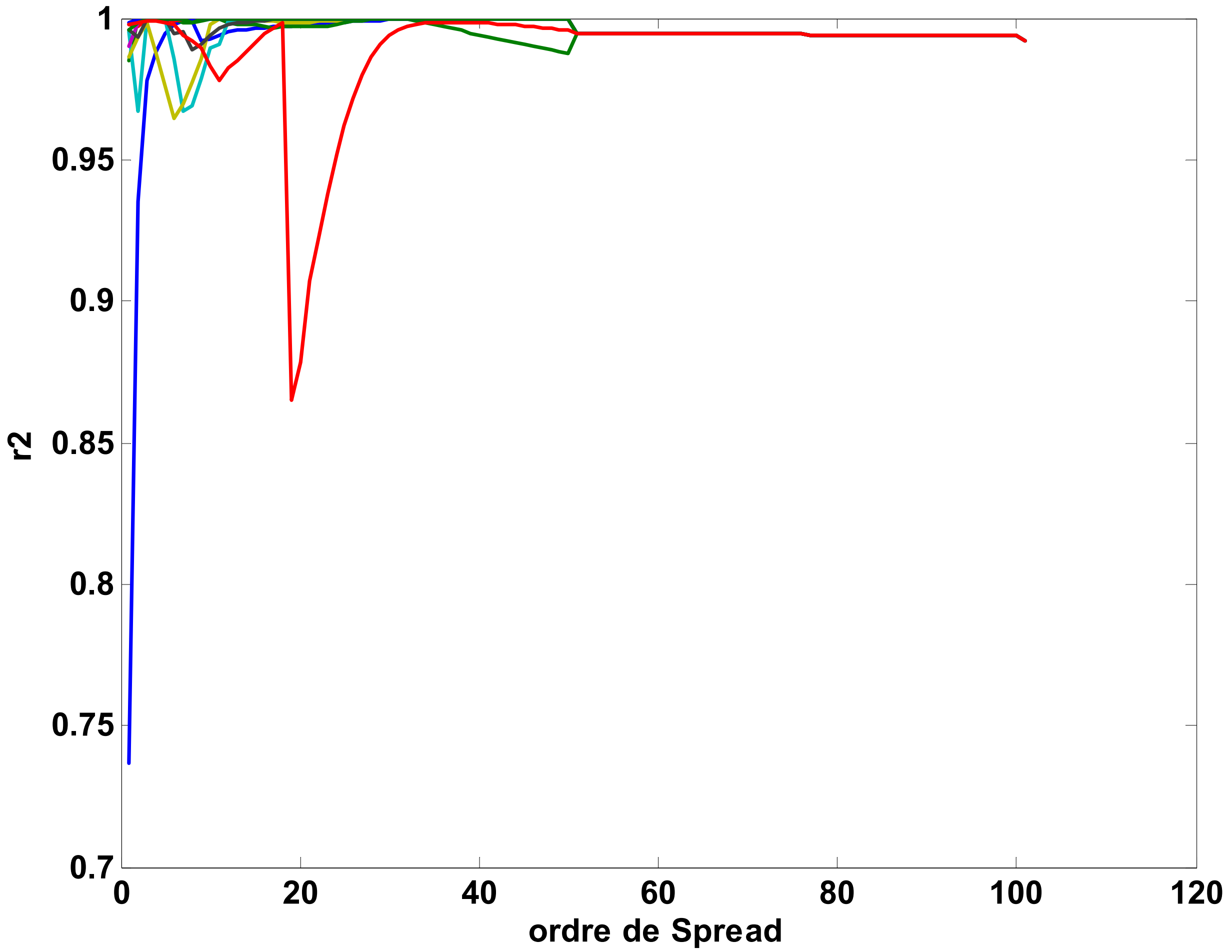
SIMULATION



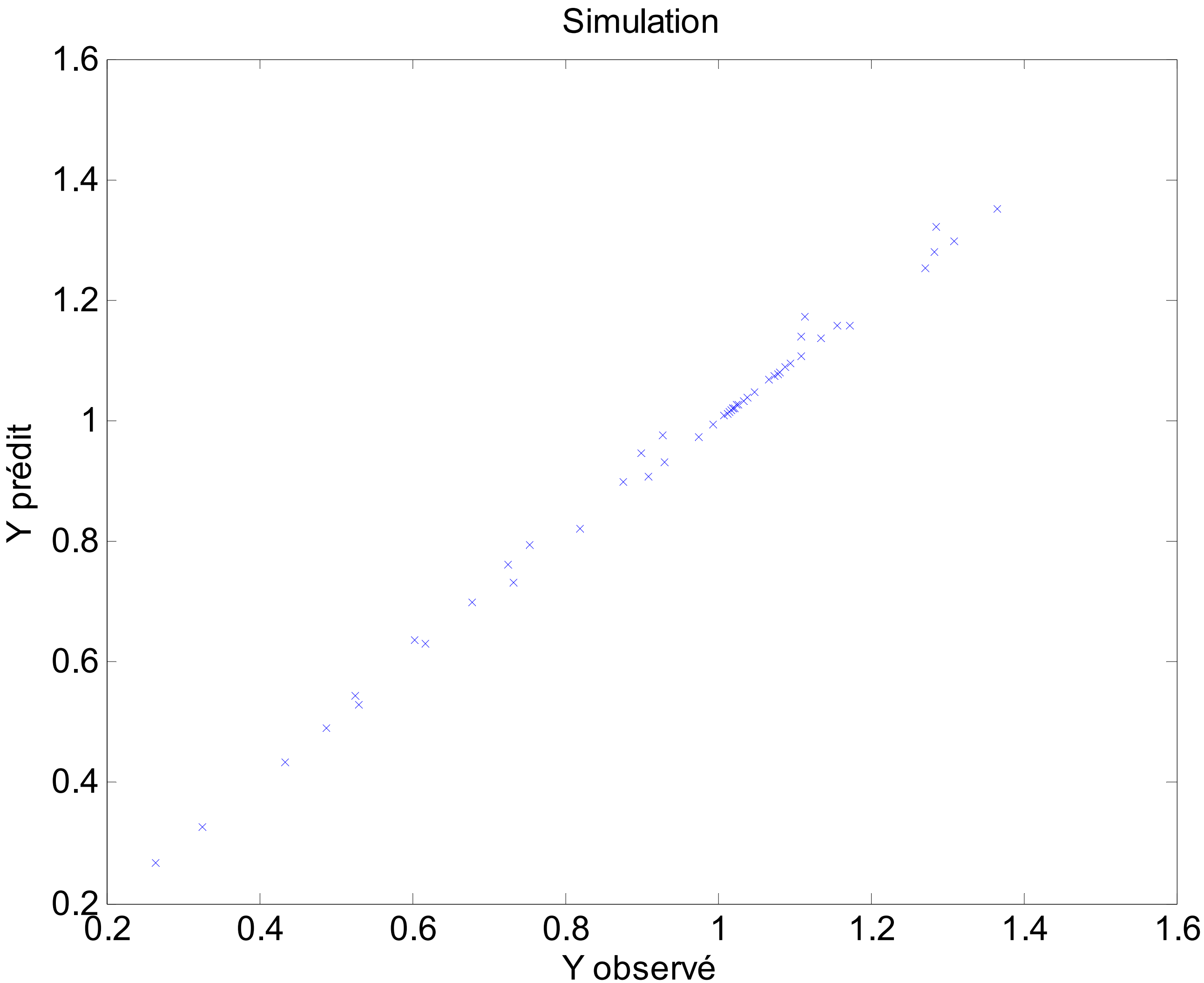
x(:,1)=(1:100)'

x(:,2)=(sin((1:100)/5))';

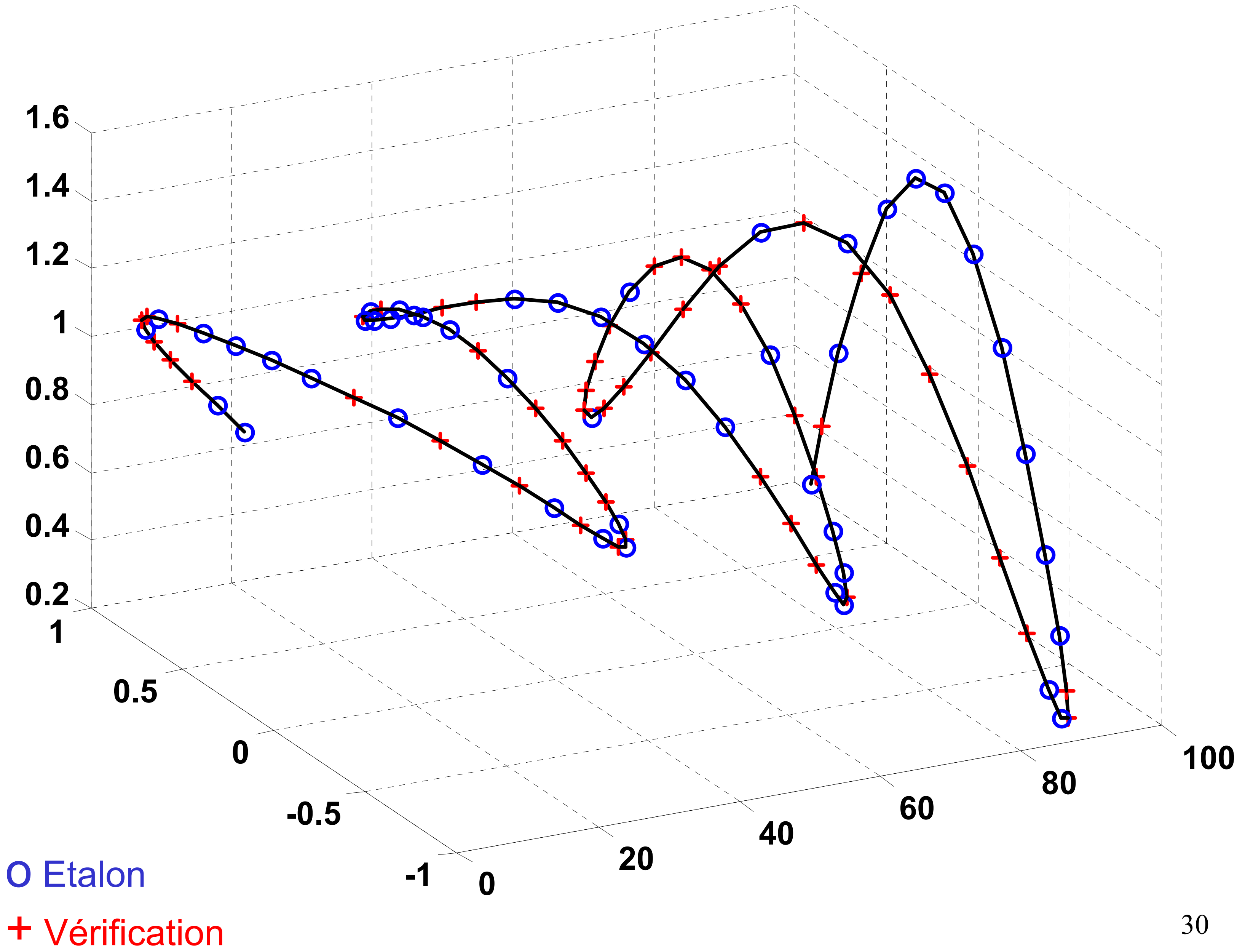
y=cos((x(:,1).*(x(:,2)*10))/500)+sin(x(:,1)/200);



10 essais avec validation
spread=0.001 à 0.05


Simulation
Y prédit
Y observé
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6

Simulation


1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
1
0.5
0
-0.5
-1
0
20
40
60
80
100
O Etalon
+ Vérification

Conclusion:

Les fonctions radiales de base semblent être efficaces.

Beaucoup de sujets à étudier:

- La métrique
- Le choix des prototypes
- L'optimisation des paramètres