

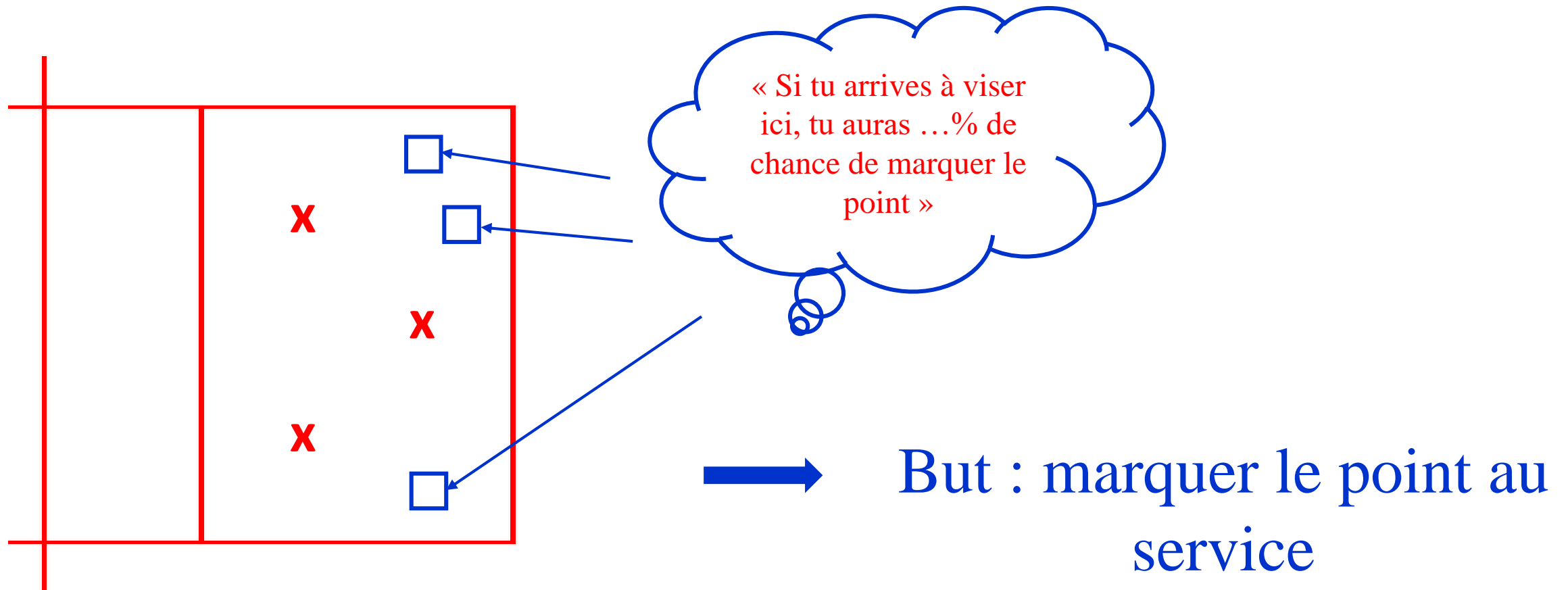


Optimisation d'un service de volley-ball

- Paramètres :
- de position : x, y, z
 - de vitesse : v
 - angulaires : λ, α
 - relatifs au système {ballon} : masse (m), coefficient de frottements (coef)
 - N : nombre de répétitions

OBJECTIF

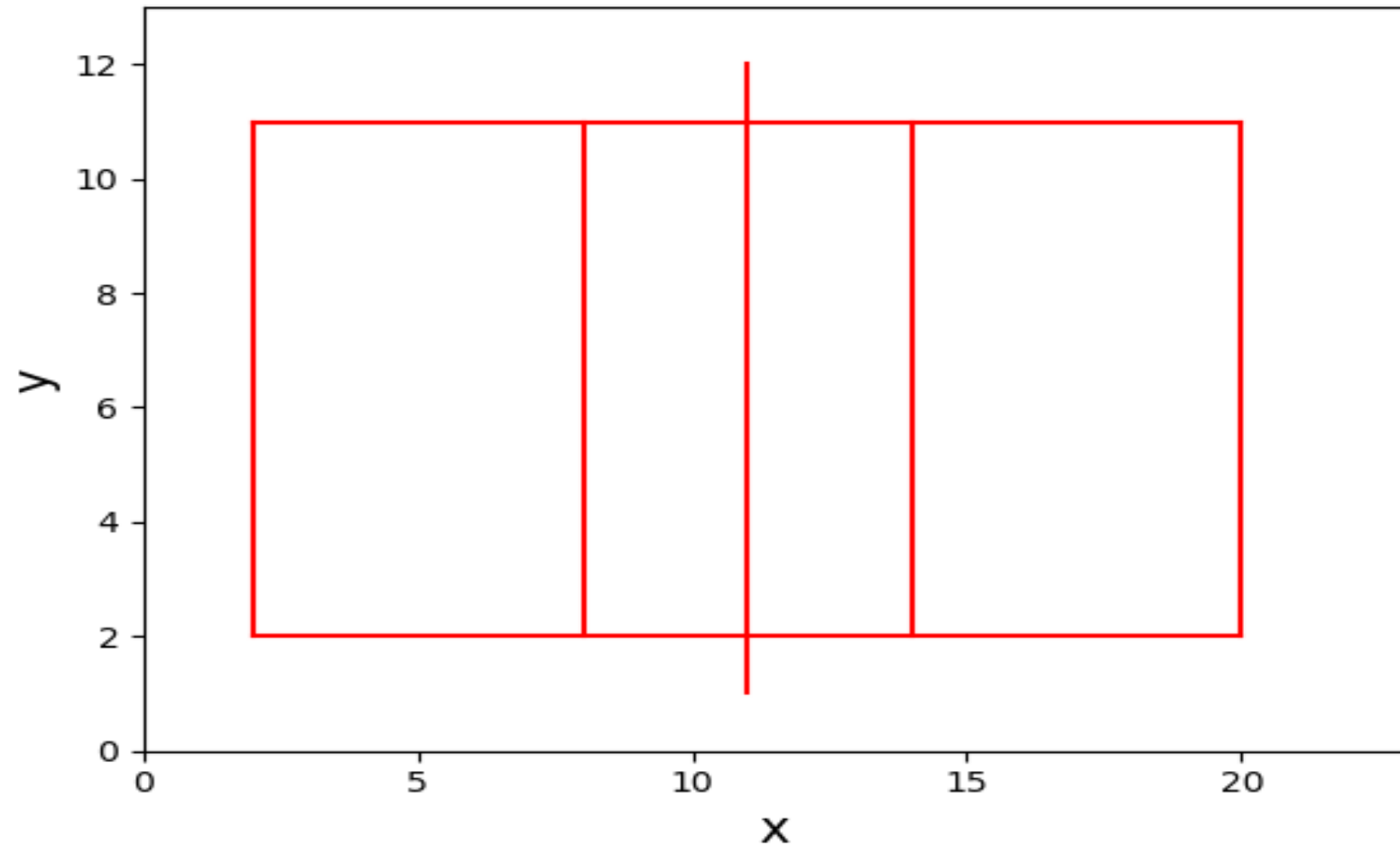
Déterminer les zones du terrain à viser lors du service



PLAN

- Modélisation :
 - du terrain
 - de la trajectoire
- Optimisation du service
- Adaptation de la réception
- Conclusion

Le Terrain ·



Le Terrain :

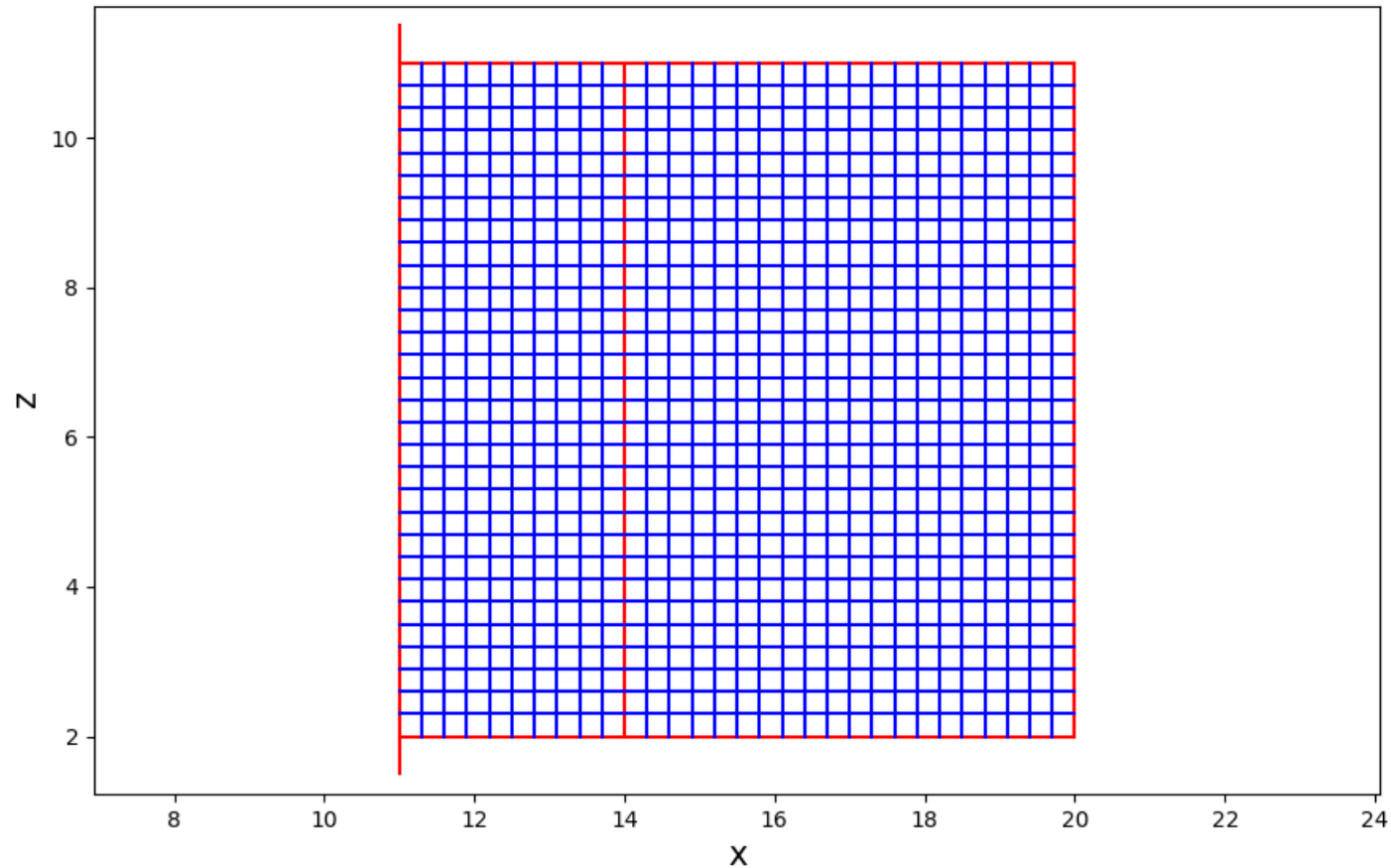
1^{er} choix : 100*100 cases

MODELISATION

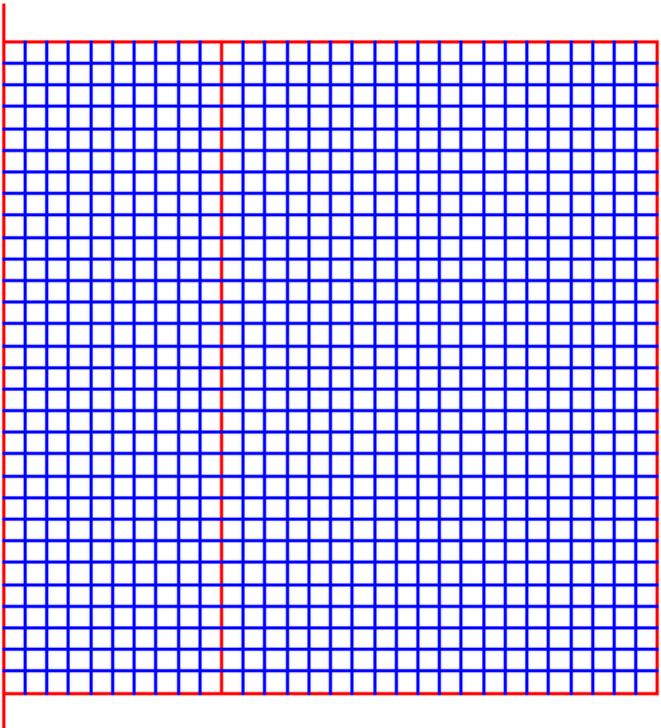
Le Terrain :

~~1^{er} choix : 100*100 cases~~

2nd choix : 30*30 cases



Le Terrain :



```
[ [ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ],  
  [ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ],  
  [ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ],  
  [ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ],
```

....

Programme : "Terrain()"

```
[ [ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ],  
  [ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ] ]
```


La trajectoire :

Frottements négligés :

Equation de la trajectoire dans le plan xOz :

$$z = -\frac{1}{2} \frac{g}{(v_0 \cos(\alpha))^2} x^2 + \tan(\alpha)x + h_0$$

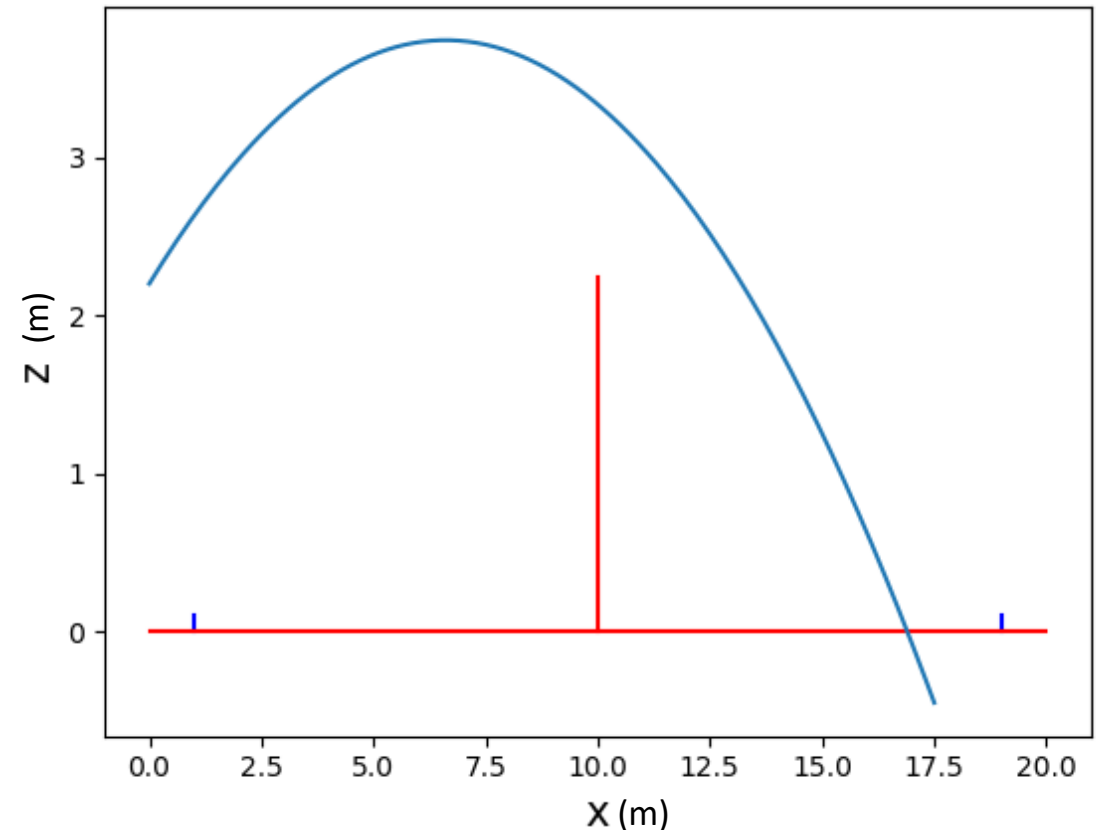
La trajectoire : Choix des conditions initiales : $\alpha = 25^\circ$ $v_0 = 13\text{m} \cdot \text{s}^{-1}$ $h_0 = 2,20\text{m}$

Frottements négligés :

Equation de la trajectoire dans le plan xOz :

$$z = -\frac{1}{2} \frac{g}{(v_0 \cos(\alpha))^2} x^2 + \tan(\alpha)x + h_0$$

Programme : “trajectoire_sans_frot (g,v,a,h0)”



La trajectoire :

Avec frottements :

$$\vec{F} = -kv^2 \hat{v}$$

$$\text{avec } k = \frac{1}{2} C_x \rho S = 0,0095 \text{ kg} \cdot \text{m}^{-1}$$

La trajectoire : Choix des conditions initiales : $\alpha = 25^\circ$ $v_0 = 13\text{m} \cdot \text{s}^{-1}$ $h_0 = 2,20\text{m}$

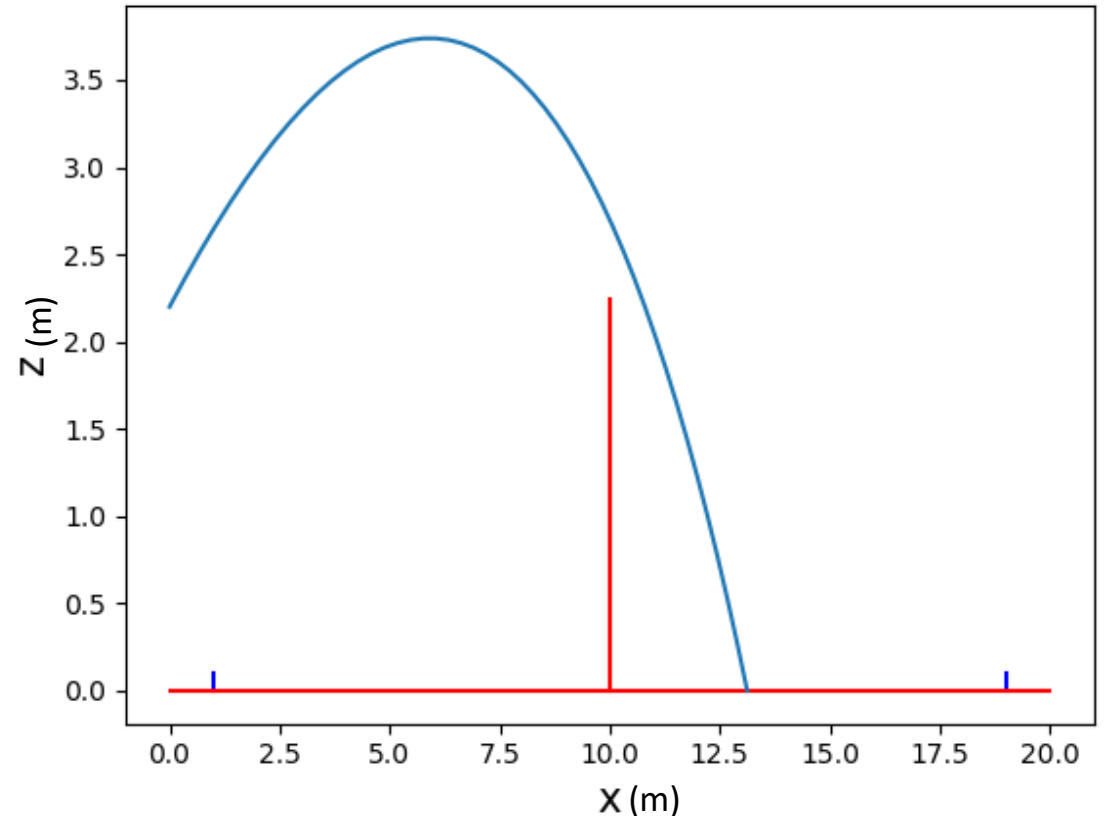
Avec frottements :

$$\vec{F} = -k v^2 \hat{v}$$

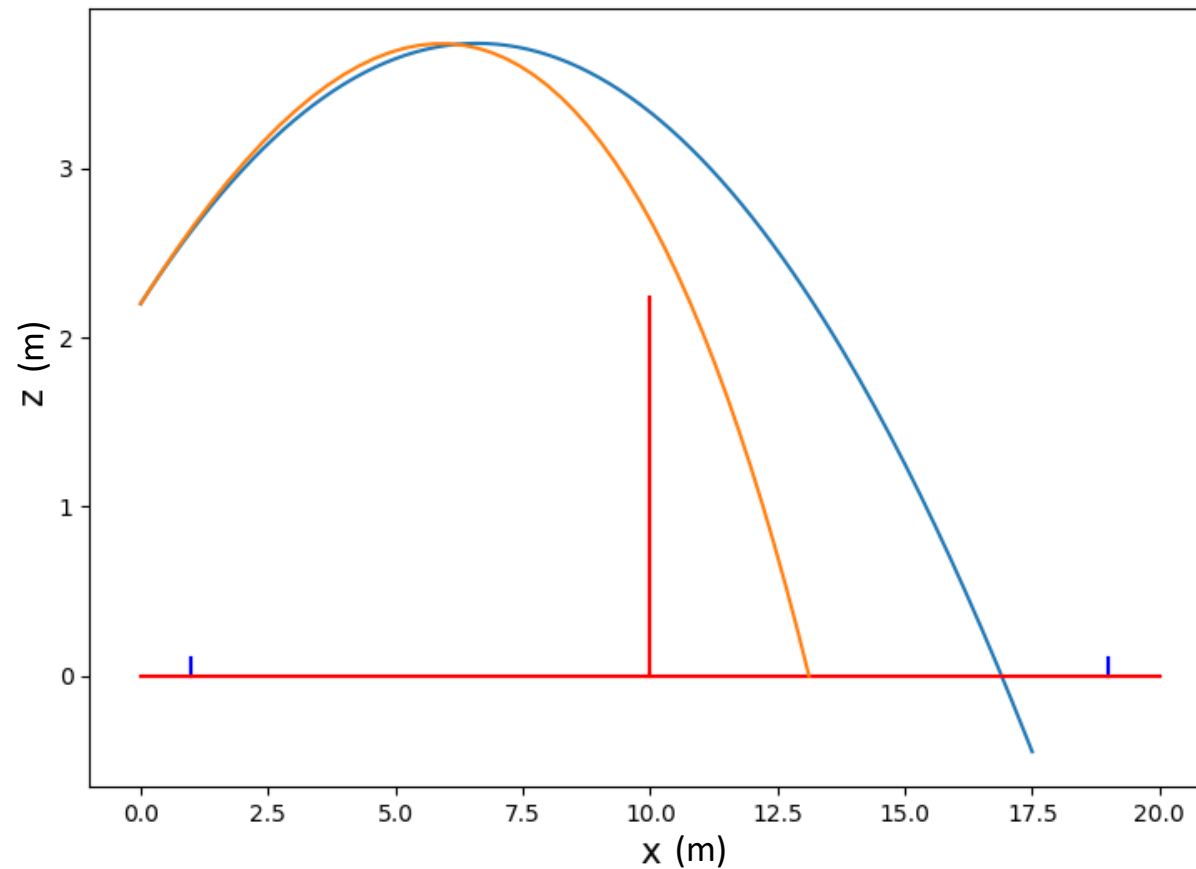
$$\text{avec } k = \frac{1}{2} C_x \rho S = 0,0095 \text{ kg} \cdot \text{m}^{-1}$$

Programme :

“traj_avec_frot(t0,tf,x0,y0,z0,v0x,v0y,v0z,N,coef,m)”

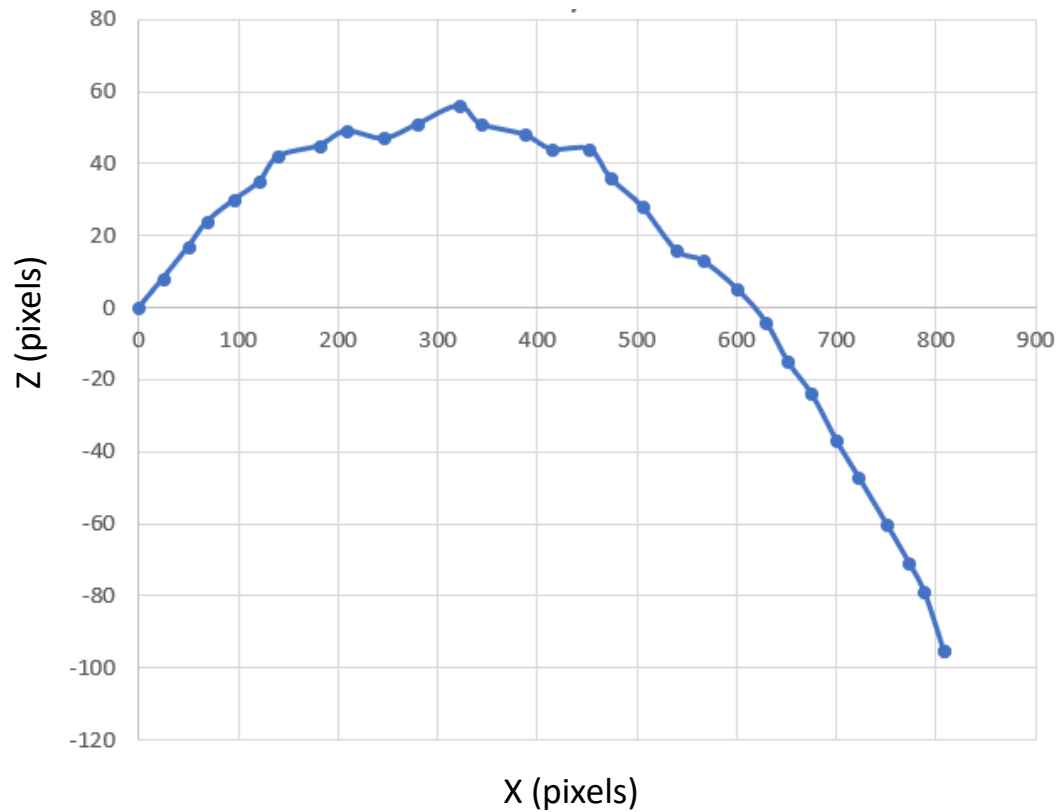


La trajectoire : Choix des conditions initiales : $\alpha = 25^\circ$ $v_0 = 13\text{m} \cdot \text{s}^{-1}$ $h_0 = 2,20\text{m}$



— Sans frottement
— Avec frottements

La trajectoire : Expérimentalement



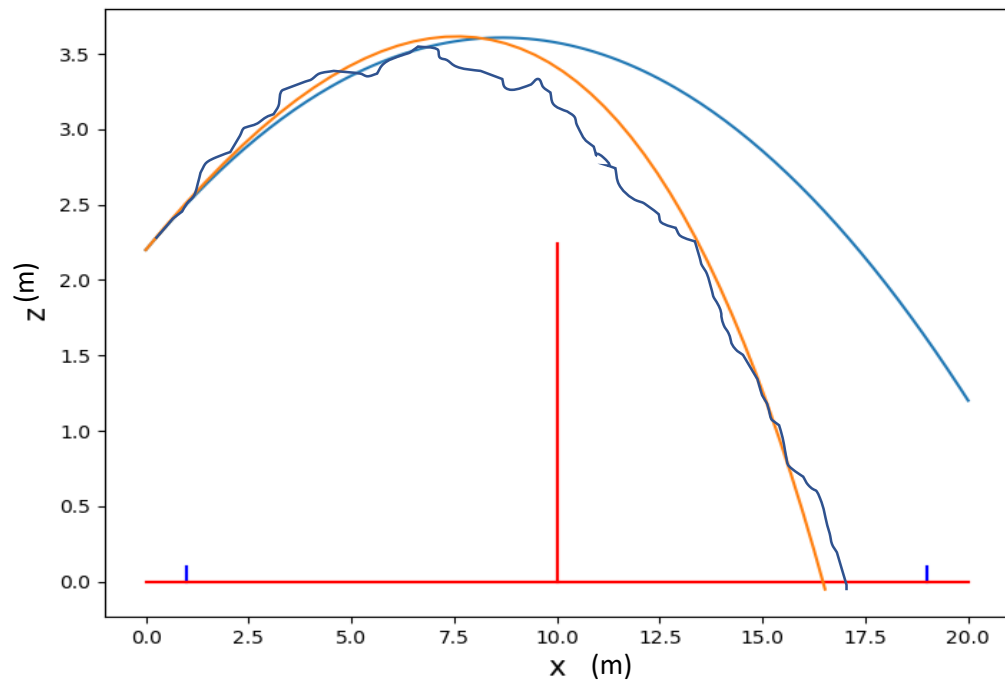
(Kinovea)

$$\alpha_{exp} = 18^\circ$$

$$v_{exp} = 17m \cdot s^{-1}$$

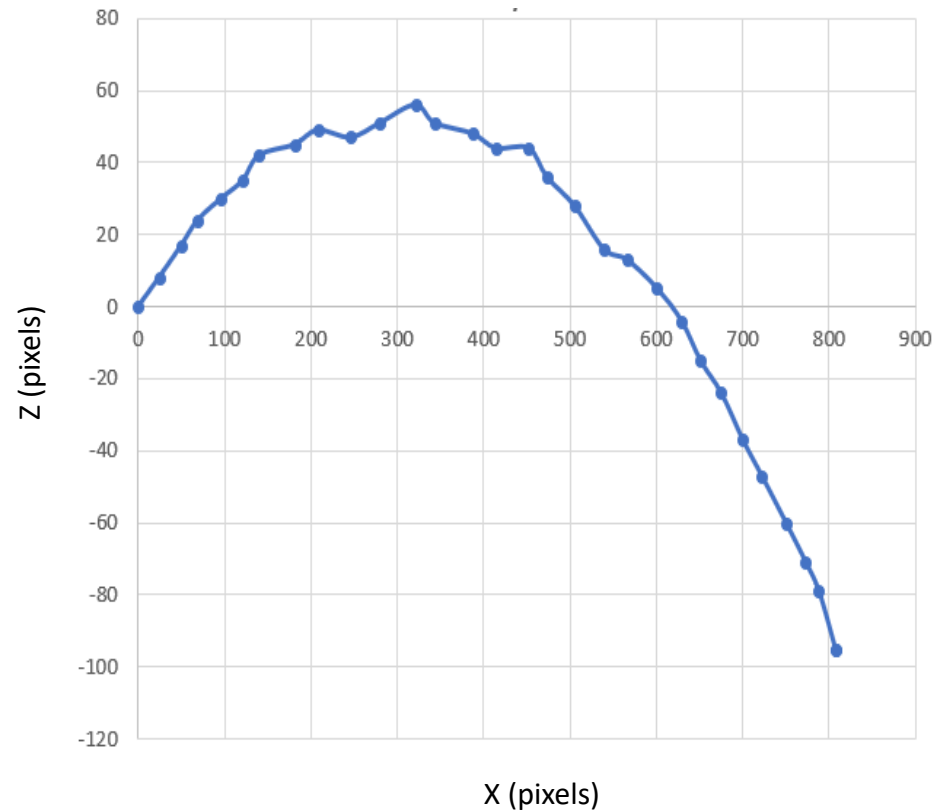
$$h_{exp} = 2,20m$$

La trajectoire : Choix des conditions initiales : $\alpha = 18^\circ$ $v_0 = 17\text{m} \cdot \text{s}^{-1}$ $h_0 = 2,20\text{m}$

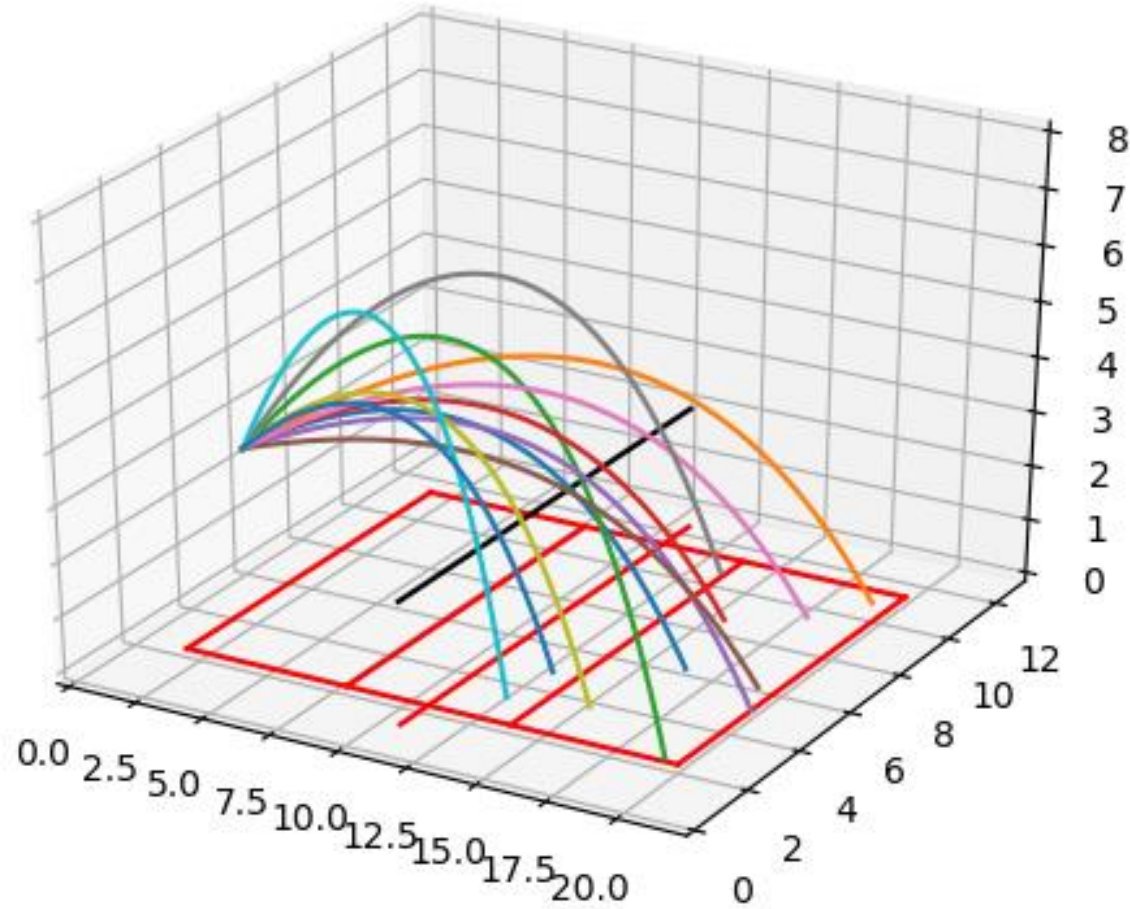


— Sans frottement

— Avec frottements



Modèle adopté

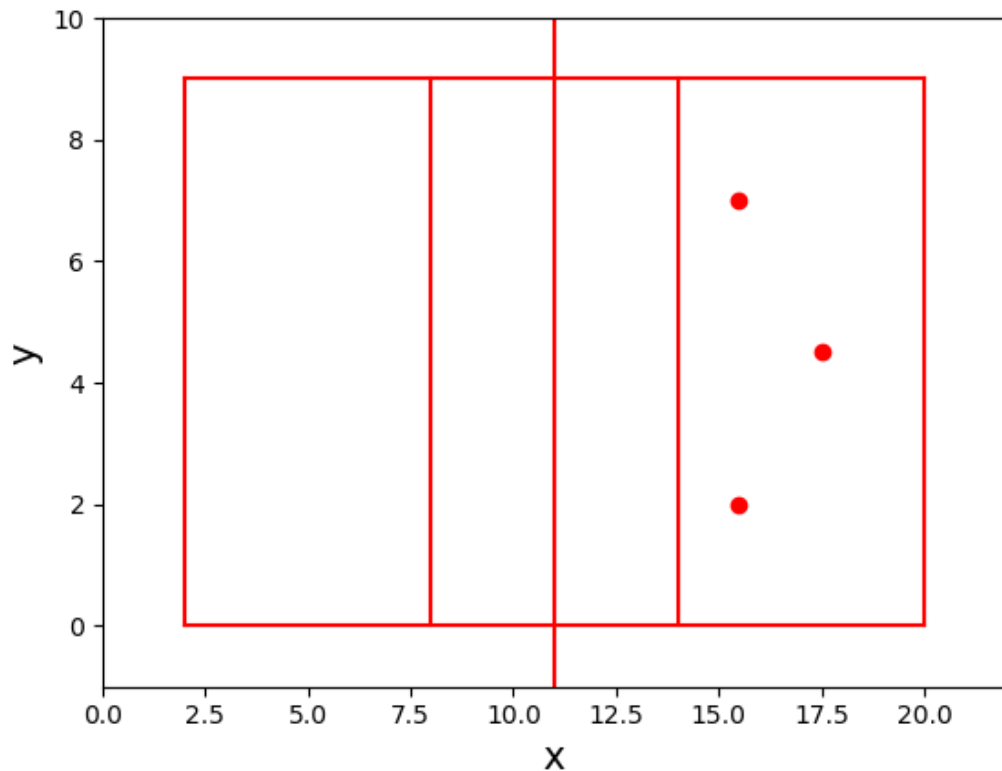


Choix de condition initiales
diverses mais position du serveur
fixe : $x_0 = 0, y_0 = 6, z_0 = 2.20$

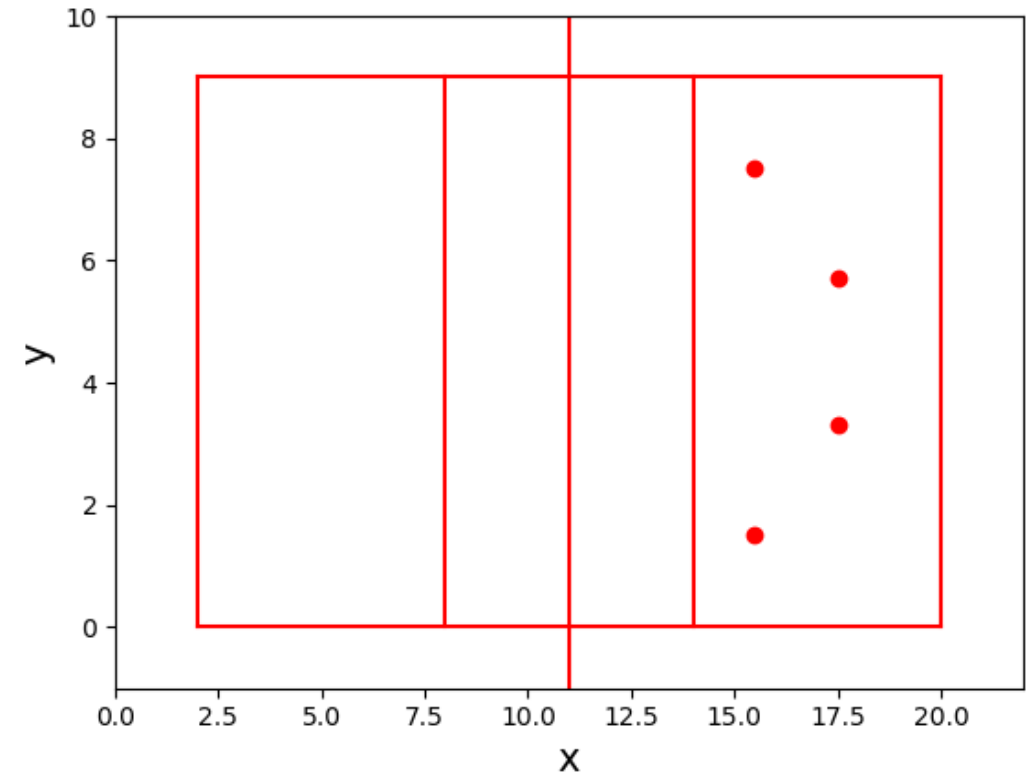
→ mise en place d'un processus
aléatoire

Adaptation au positionnement de la réception adverse :

3 récepteurs



4 récepteurs

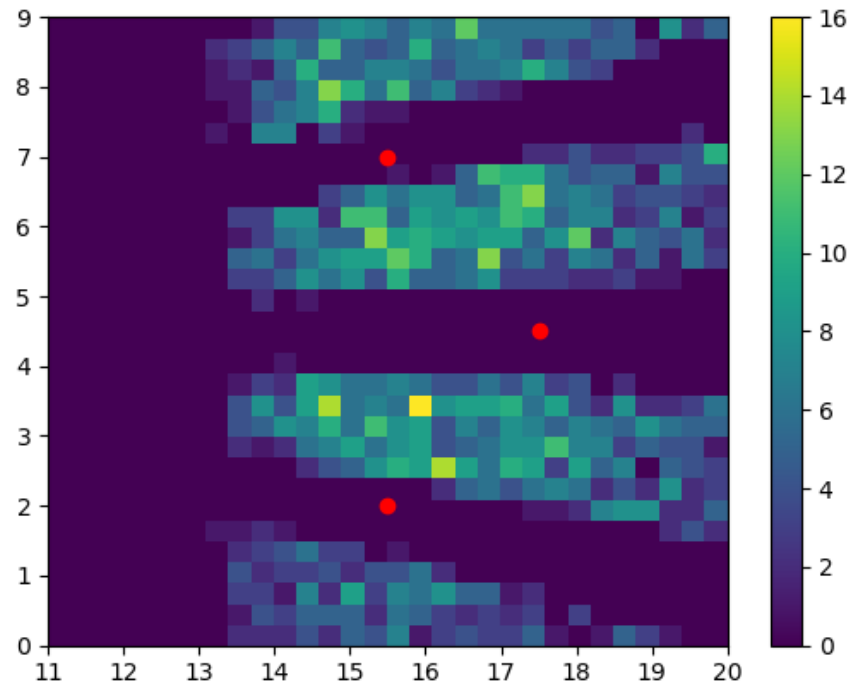


1^{ère} idée : mettre en évidence des « zones à viser » en évitant les joueurs.



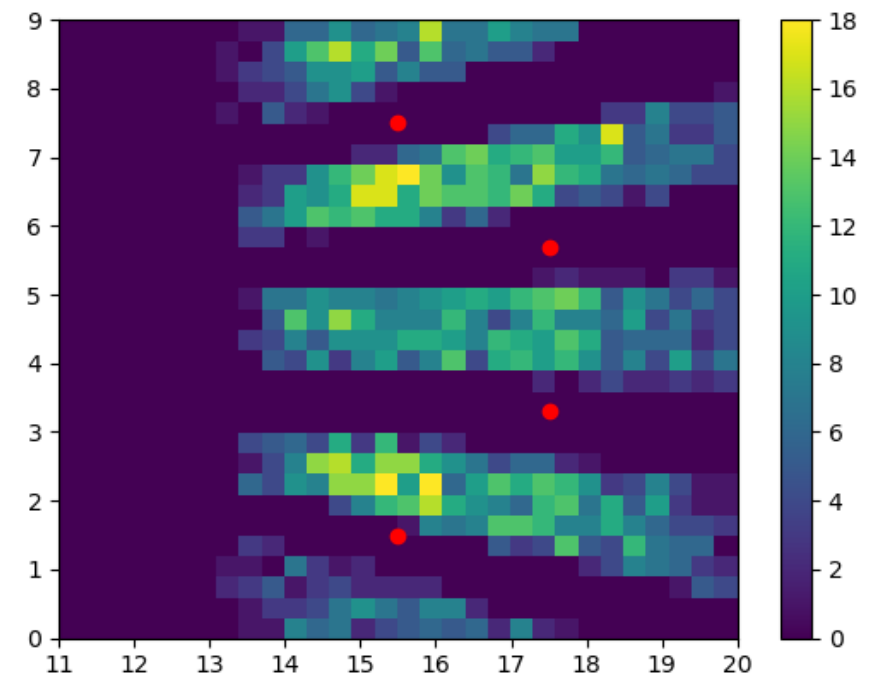
Programme : “remplissage_en_fonction_de_recep_3j(paramètres)”

3 réceptionneurs



1000 lancers

4 réceptionneurs



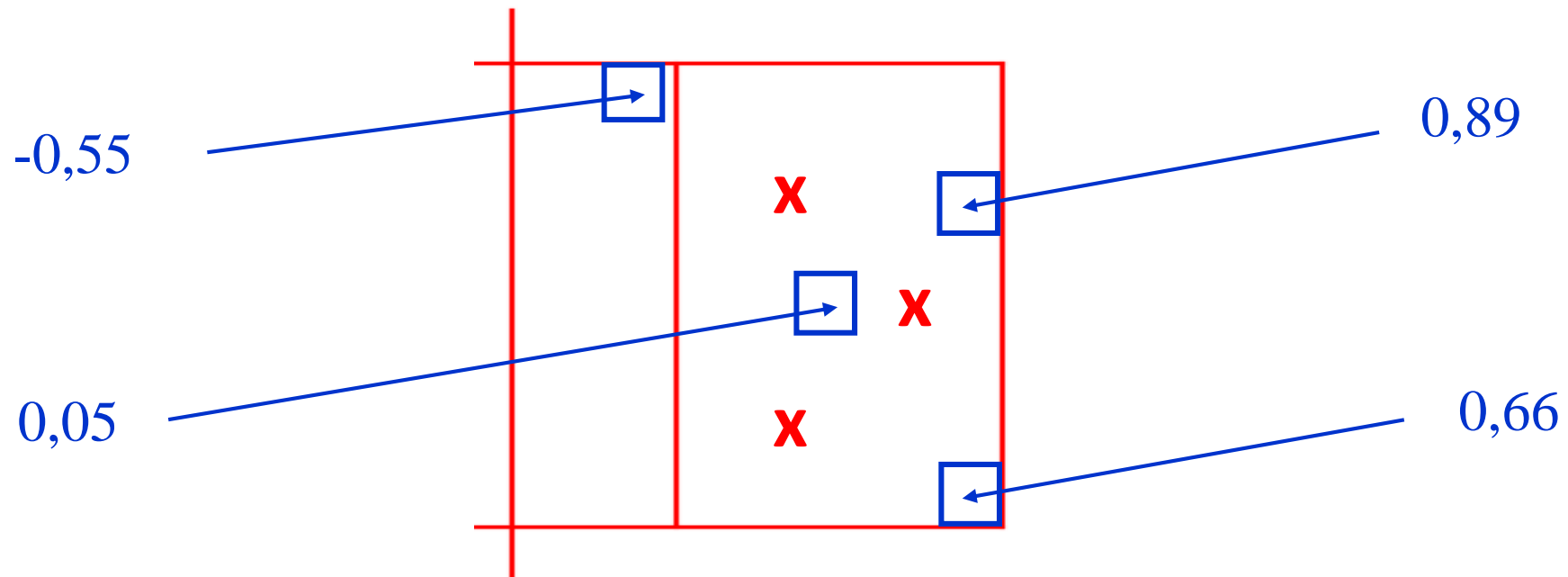
1000 lancers

2^{ème} idée et objectif final : indiquer au serveur les zones dont la probabilité de gain est maximale.

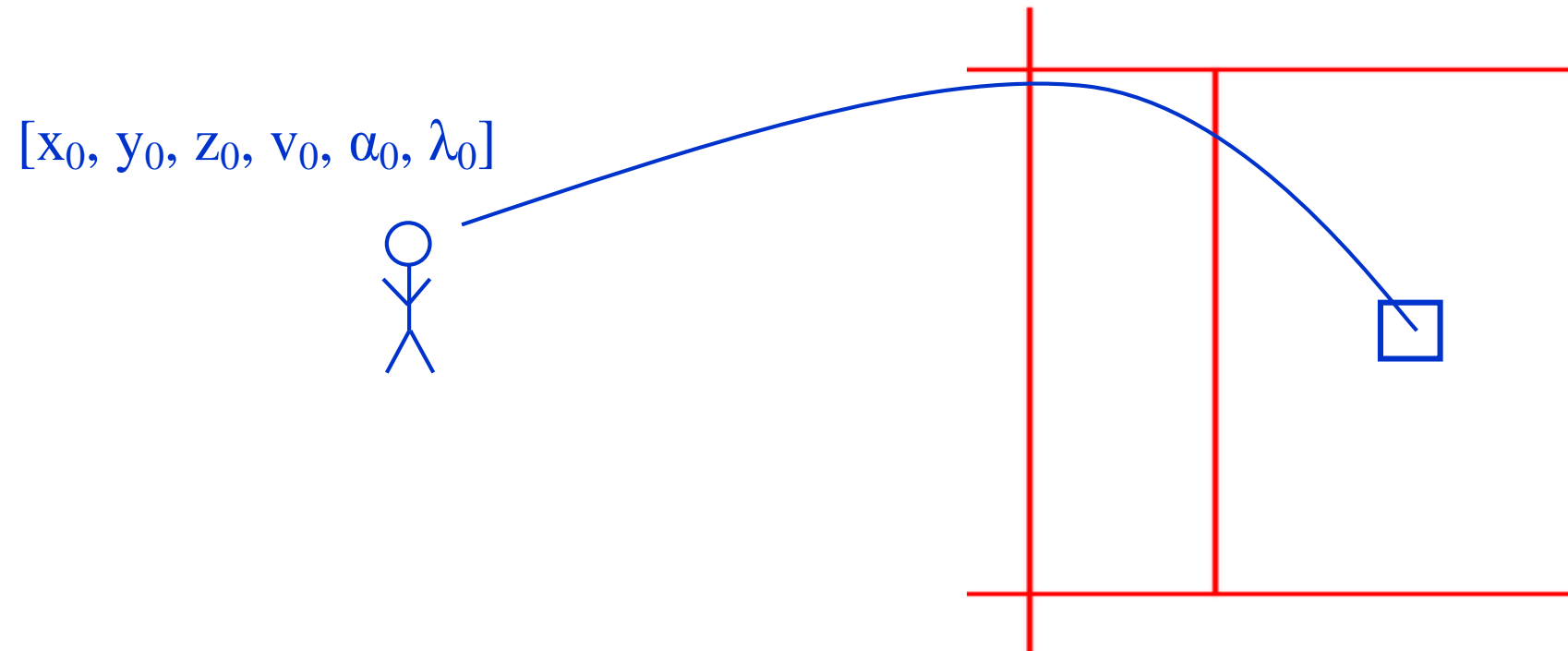
2^{ème} idée et objectif final : indiquer au serveur les zones dont la probabilité de gain est maximale.



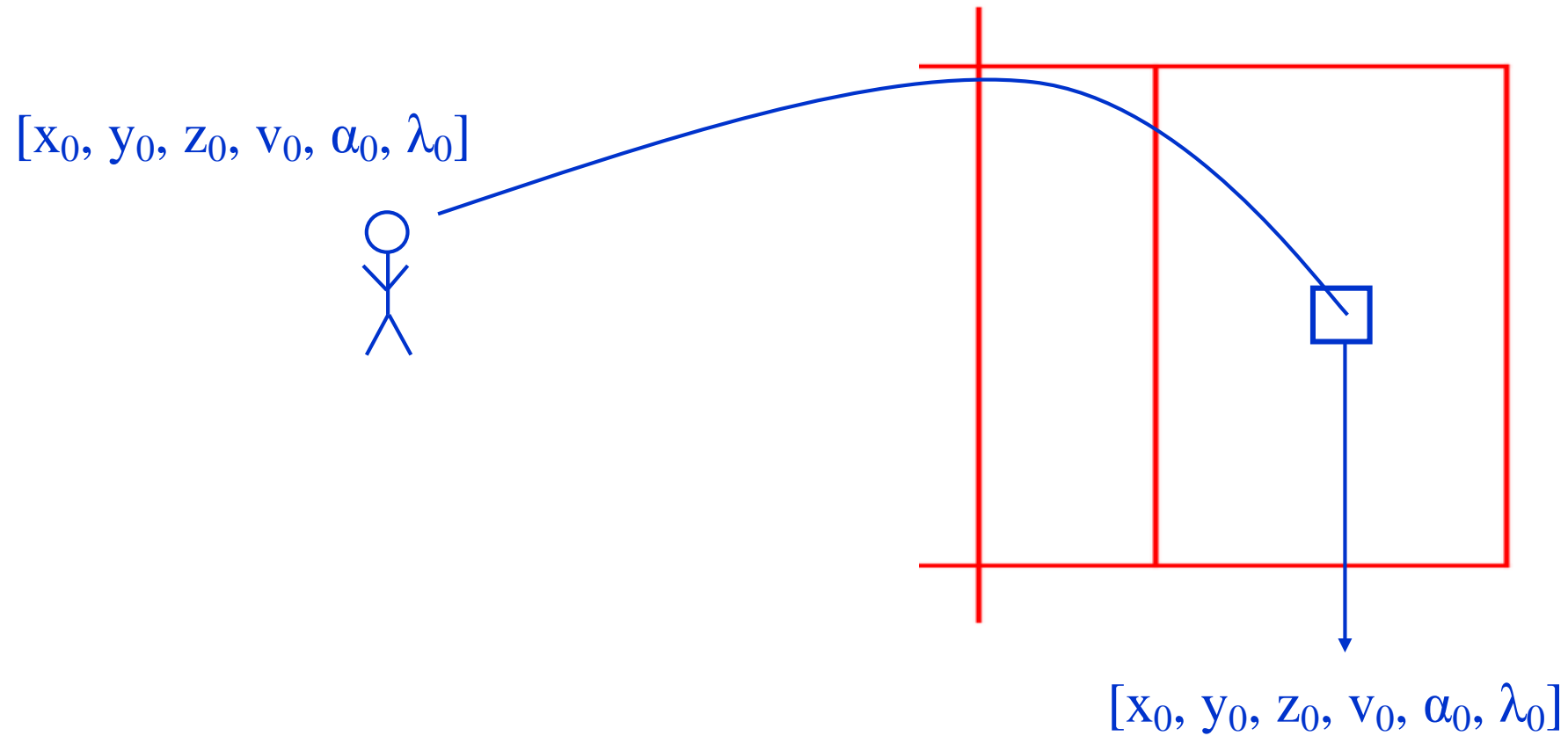
Remplir la matrice Terrain avec les espérances de gain associées à chaque case.



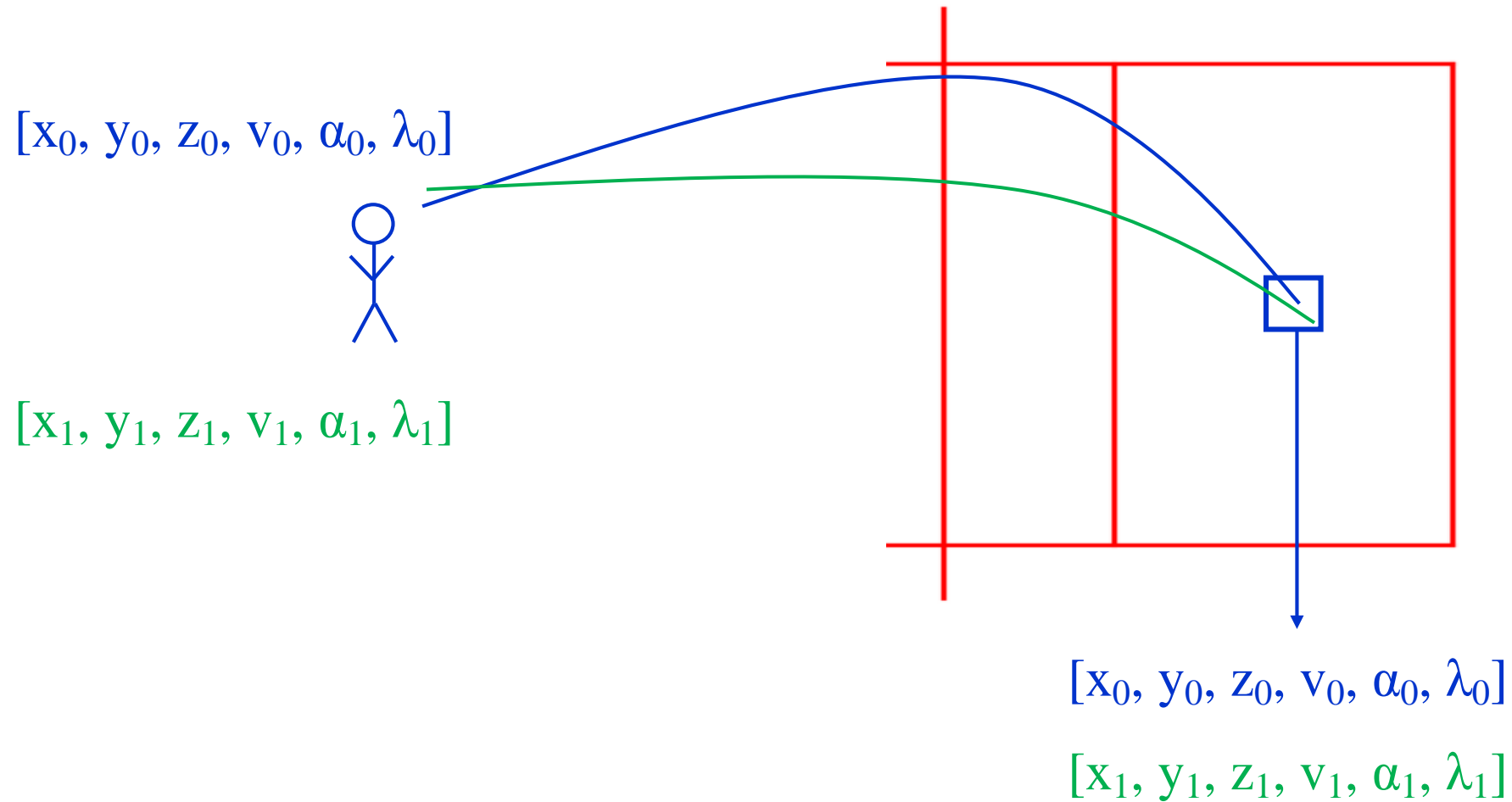
Etape 1 : faire des lancers aléatoires



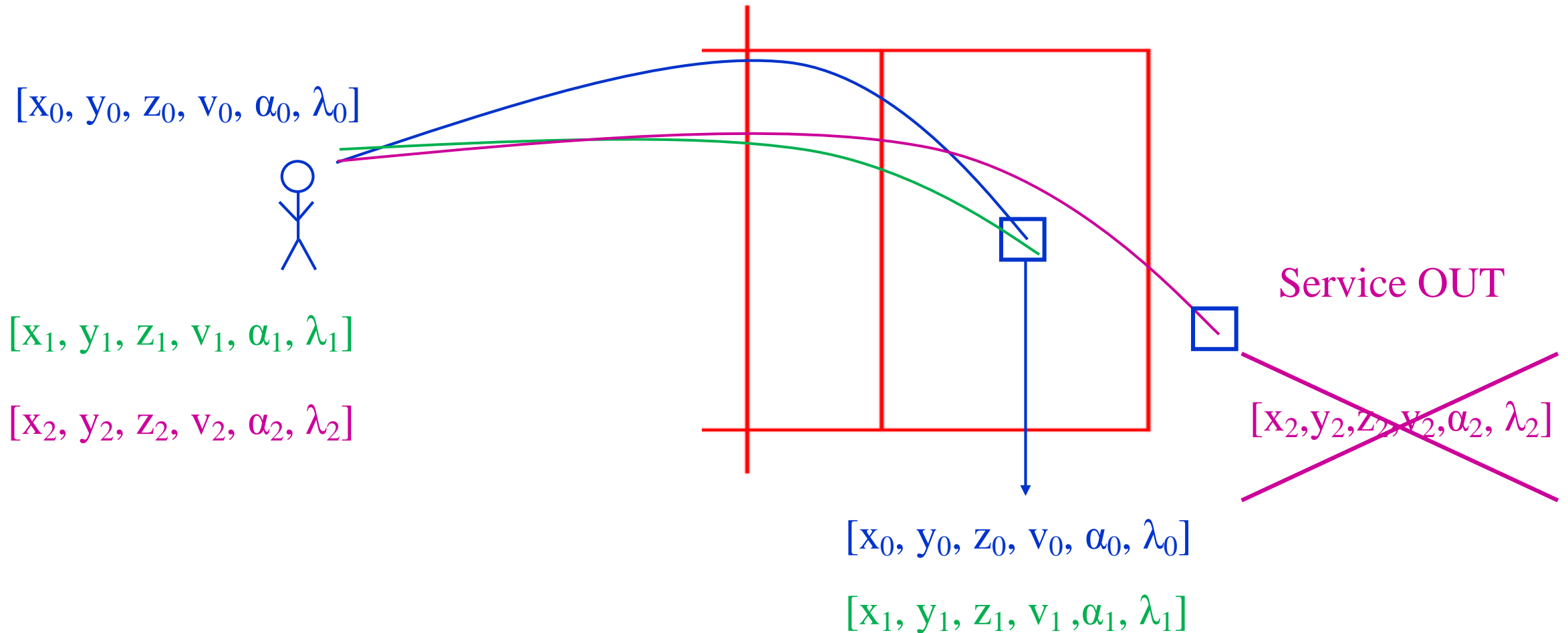
Etape 1 : faire des lancers aléatoires



Étape 1 : faire des lancers aléatoires



Étape 1 : faire des lancers aléatoires

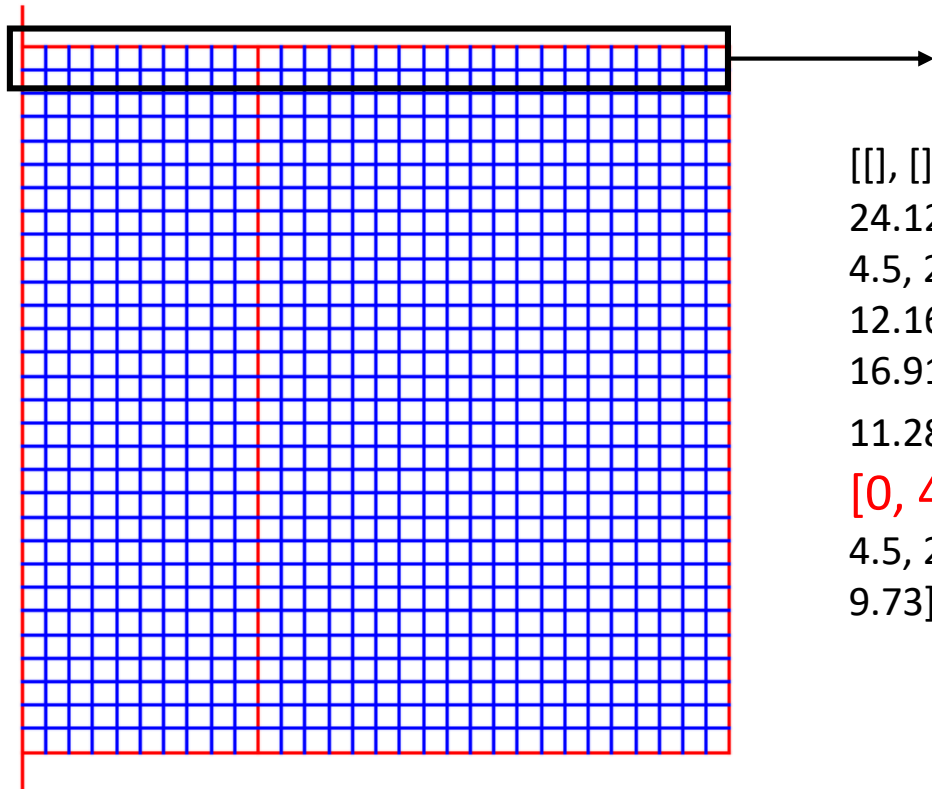


Etape 1 : faire des lancers aléatoires

Exemple de 1000 lancers :

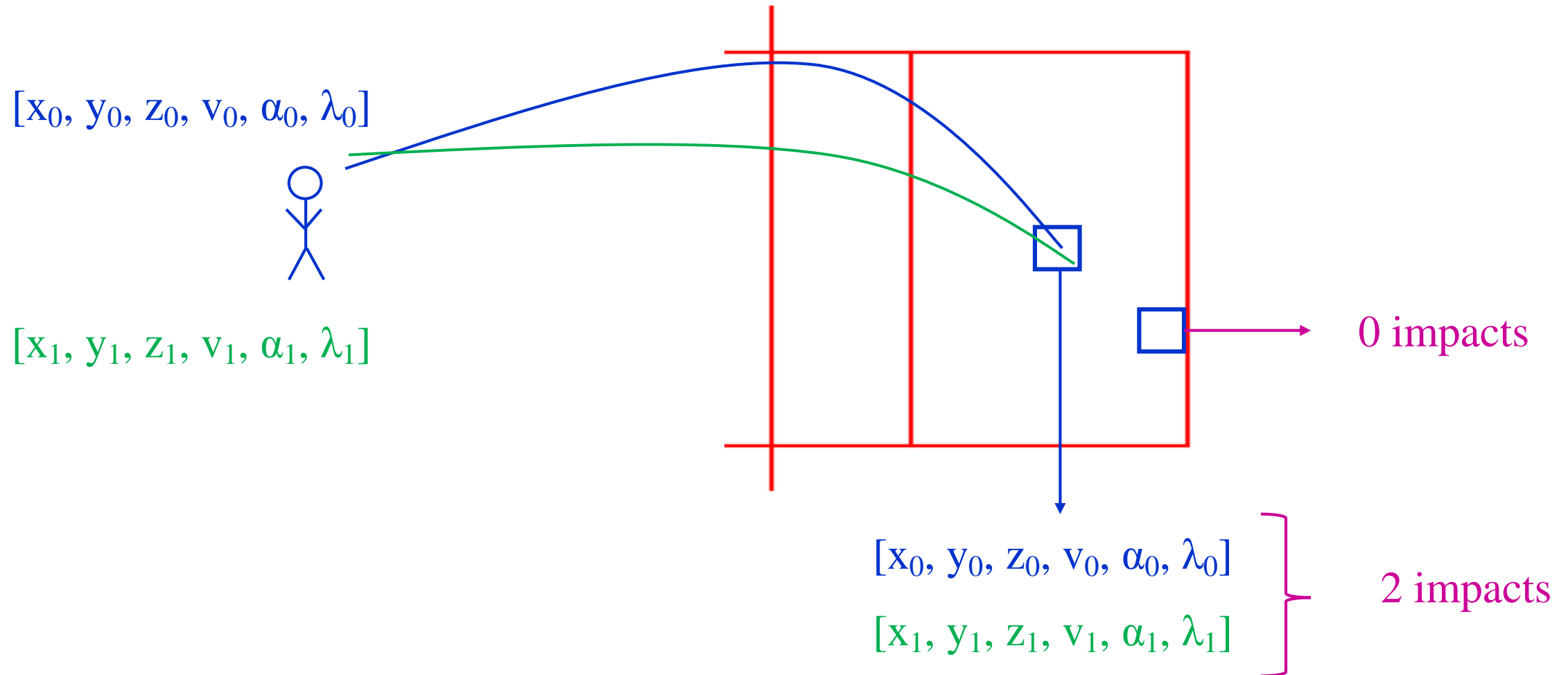


Programme : “remplissage_CI(paramètres , 1000) ”
→ Matrice Terrain remplie avec les conditions initiales



[[[], [], [], [], [], [], [], [], [], [], [0, 4.5, 2.2, 14.15, 24.6, 14.14]], [[0, 4.5, 2.2, 14.45, 24.12, 14.46]], [[0, 4.5, 2.2, 14.97, 22.51, 13.45]], [], [[0, 4.5, 2.2, 15.74, 21.08, 12.54]], [[0, 4.5, 2.2, 15.8, 21.63, 12.67]], [0, 4.5, 2.2, 15.38, 23.73, 12.89]], [[0, 4.5, 2.2, 15.81, 23.28, 12.16]], [0, 4.5, 2.2, 16.44, 20.16, 12.0]], [[0, 4.5, 2.2, 16.39, 21.59, 11.95]], [[0, 4.5, 2.2, 16.91, 20.39, 12.08]], [], [[0, 4.5, 2.2, 16.89, 22.01, 11.26]], [[0, 4.5, 2.2, 16.68, 24.66, 11.28]], [0, 4.5, 2.2, 17.31, 22.77, 10.78], [0, 4.5, 2.2, 17.85, 20.27, 10.53], [0, 4.5, 2.2, 17.08, 24.21, 10.42], [0, 4.5, 2.2, 17.18, 23.51, 11.1]], [], [], [[0, 4.5, 2.2, 18.58, 20.37, 9.96]], [0, 4.5, 2.2, 17.9, 23.79, 9.96]], [], [[0, 4.5, 2.2, 18.19, 24.28, 9.73]], [0, 4.5, 2.2, 18.13, 24.03, 9.58]], [], []]

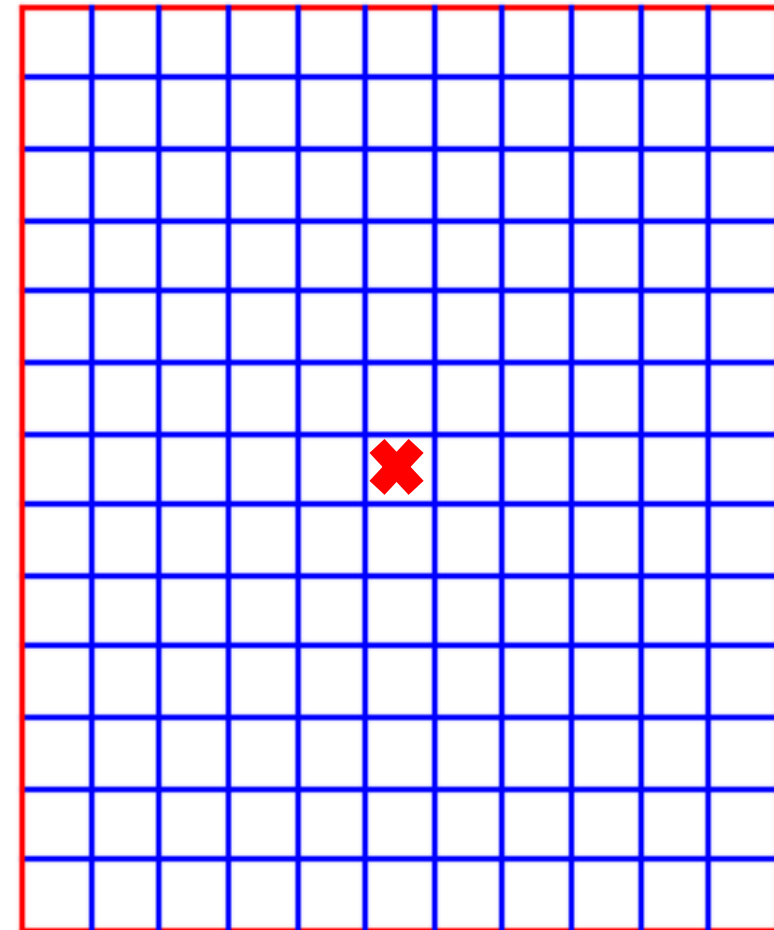
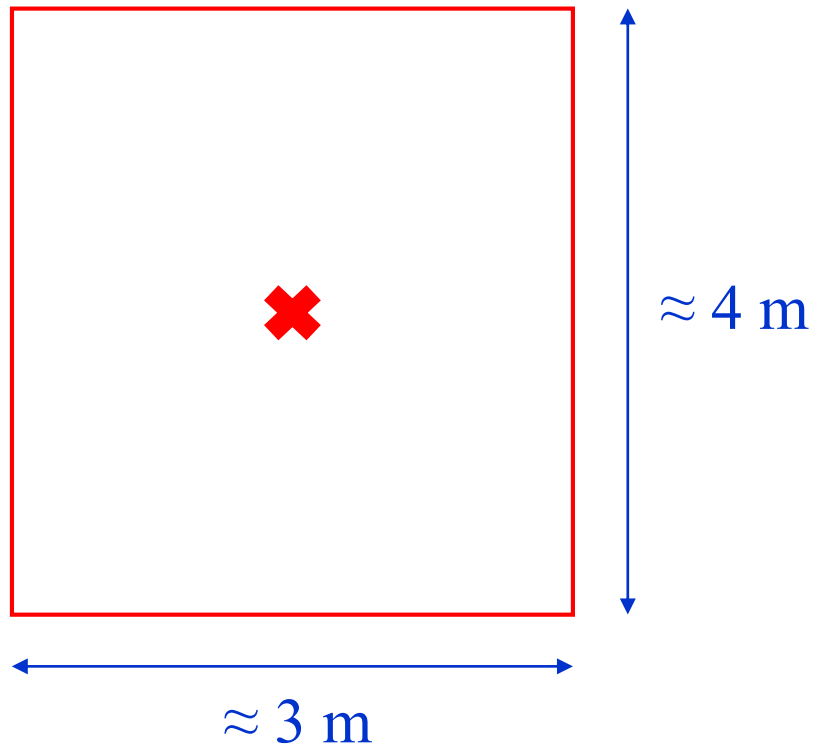
Étape 1 : faire des lancers aléatoires



Etape 2 : étude de la réception adverse → Estimation des probabilités de succès en réception

Etape 2 : étude de la réception adverse → Estimation des probabilités de succès en réception

↪ Zone maximale « couverte » $\approx 12 \text{ m}^2$

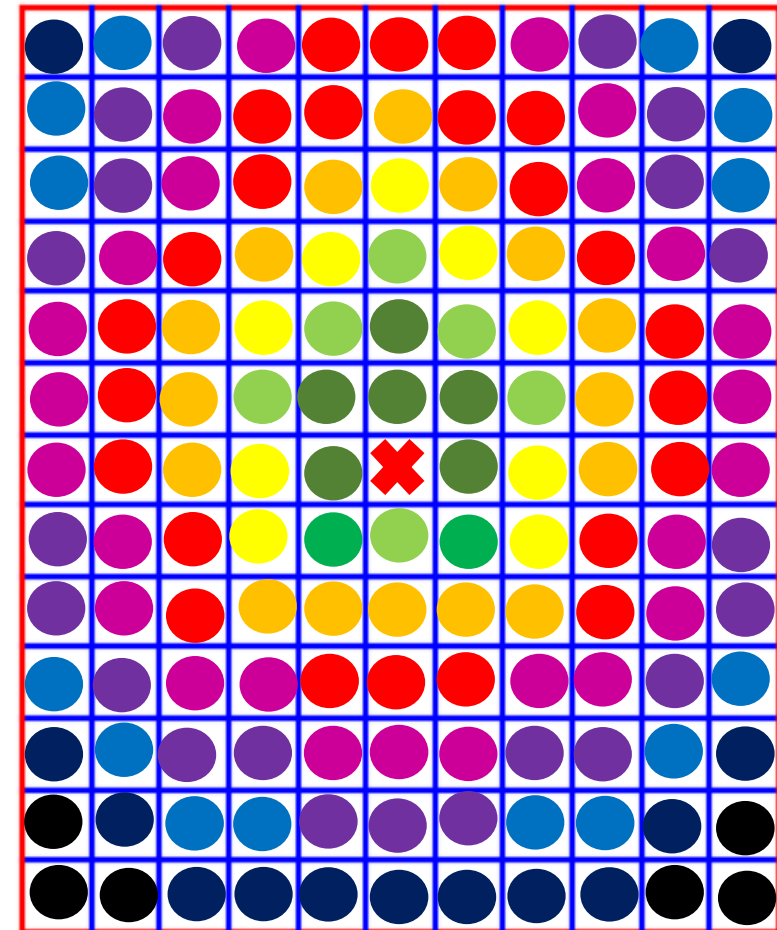


Etape 2 : étude de la réception adverse → Estimation des probabilités de succès en réception

↪ Zone maximale « couverte » $\approx 12 \text{ m}^2$

↪ Pourcentage de succès en réception

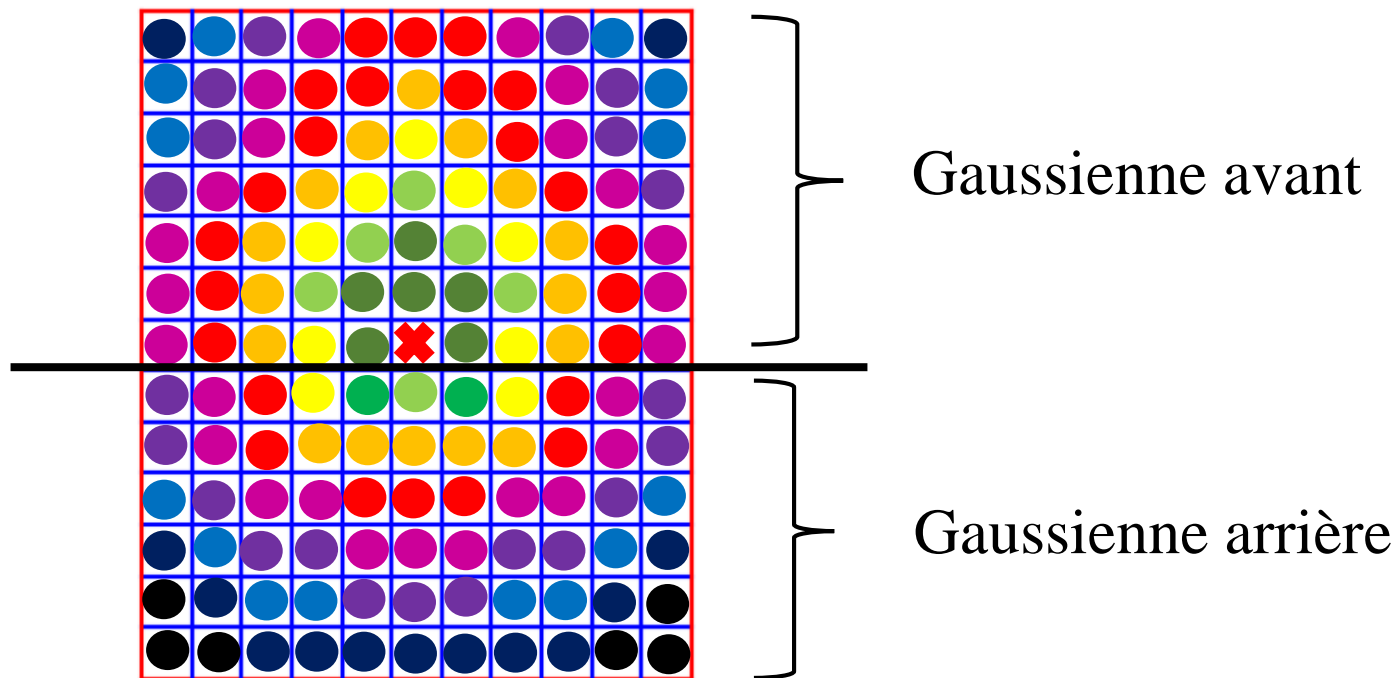
● 95 %	● 70 %	● 30 %
● 90 %	● 60 %	● 20 %
● 85 %	● 50 %	● 10 %
● 80 %	● 40 %	



Etape 2 : étude de la réception adverse → Estimation des probabilités de succès en réception

Modélisation de la réception adverse

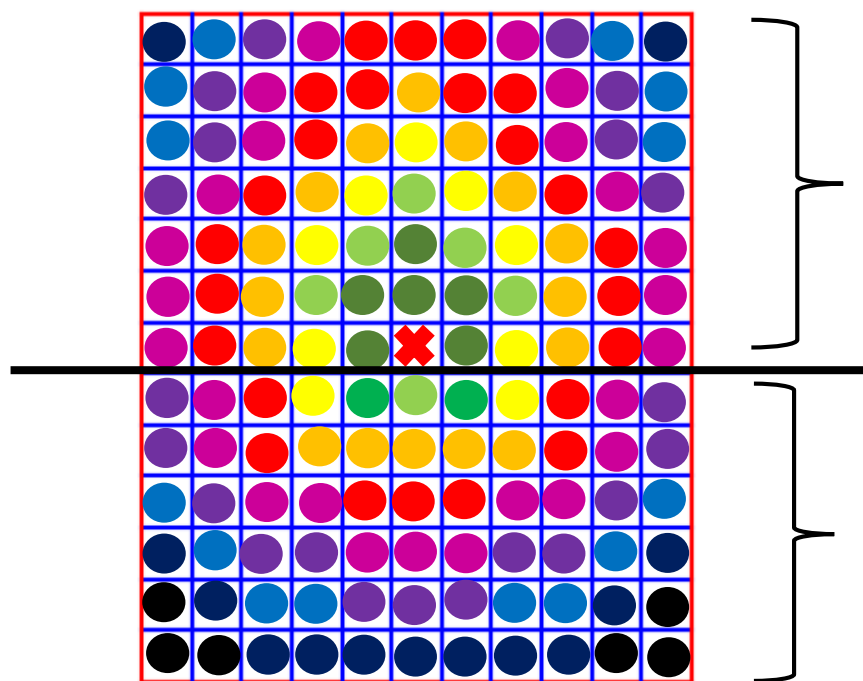
↪ Deux gaussiennes : une avant, une arrière



Etape 2 : étude de la réception adverse → Estimation des probabilités de succès en réception

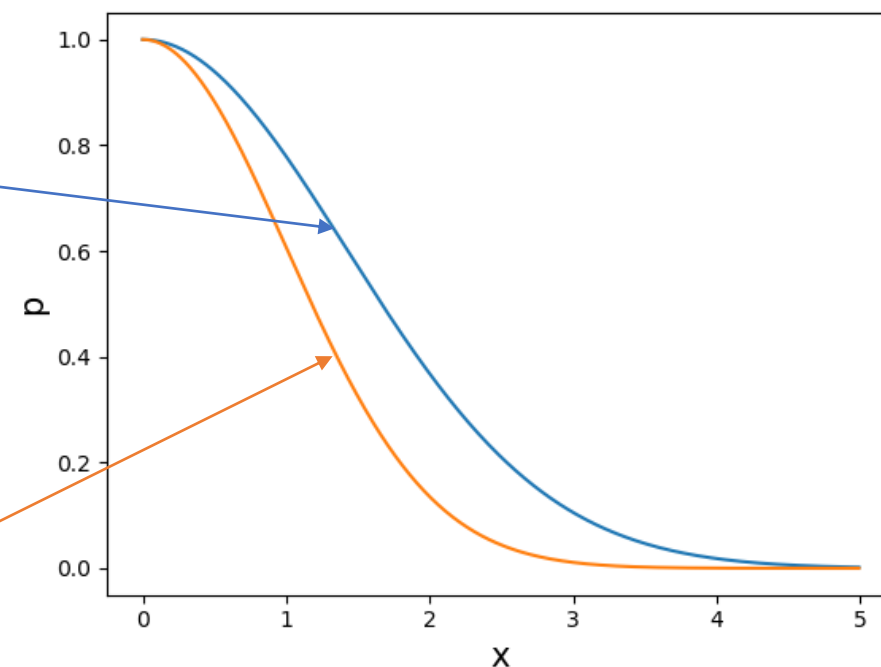
Modélisation de la réception adverse

↪ Deux gaussiennes : une avant, une arrière



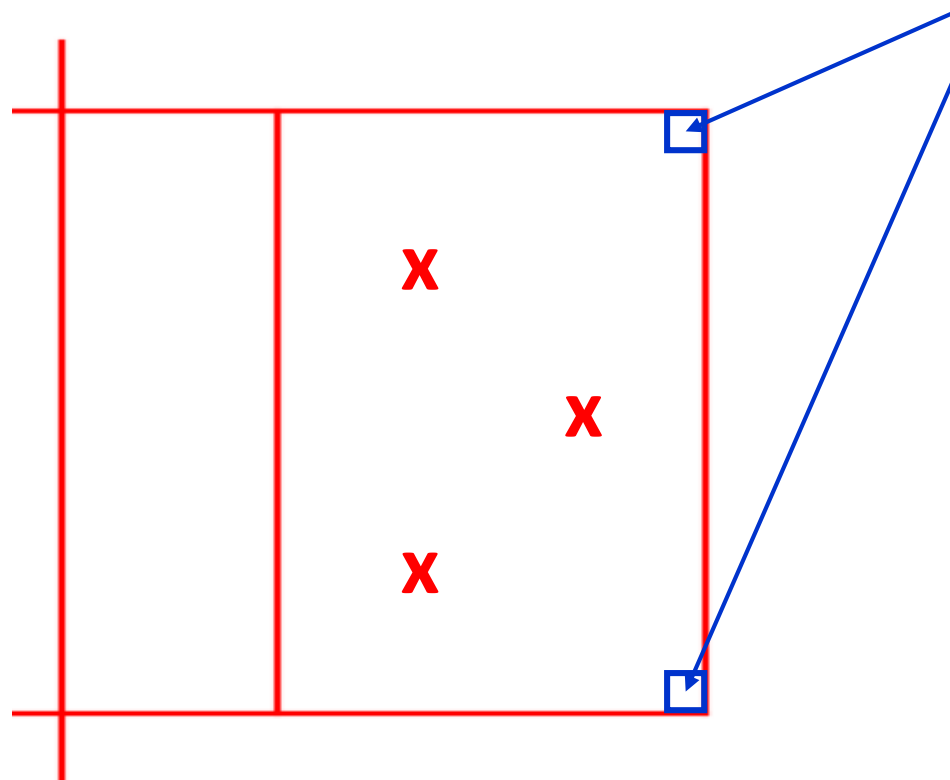
Gaussienne avant

Gaussienne arrière



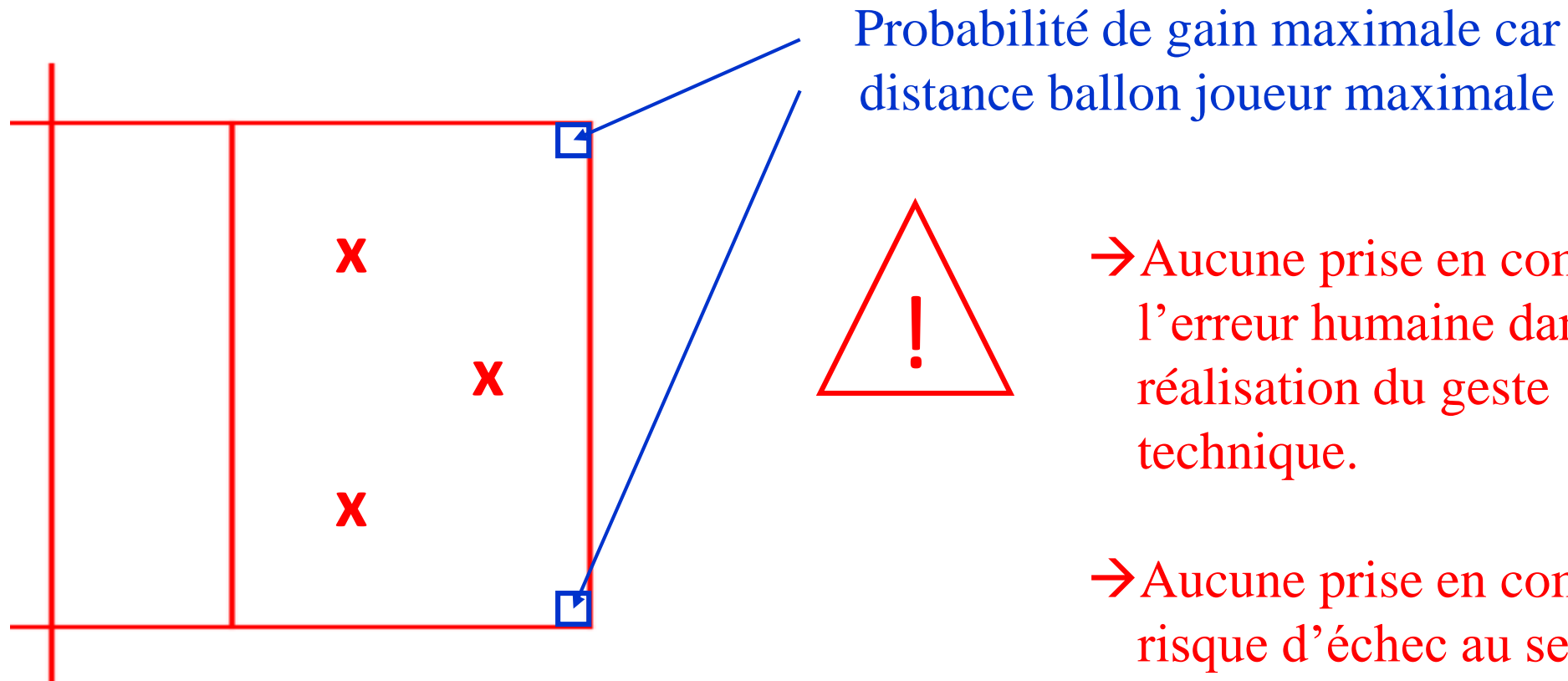
Rappel de l'objectif final : indiquer au serveur les zones dont la probabilité de gain est maximale.

Rappel de l'objectif final : indiquer au serveur les zones dont la probabilité de gain est maximale.



Probabilité de gain maximale car distance ballon joueur maximale

Rappel de l'objectif final : indiquer au serveur les zones dont la probabilité de gain est maximale.



Imprécision humaine



Programme : “ modif_alea(paramètres) ”

→ Modifie les conditions initiales

$$x \rightarrow x \pm 0 \leq x_{er} \leq 0,5m$$

$$y \rightarrow y \pm 0 \leq y_{er} \leq 0,5m$$

$$z \rightarrow z \pm 0 \leq z_{er} \leq 0,2m$$

$$v \rightarrow v \pm 0 \leq v_{er} \leq 2 \text{ m.s}^{-1}$$

$$\alpha \rightarrow \alpha \pm 0 \leq \alpha_{er} \leq 2^\circ$$

$$\lambda \rightarrow \lambda \pm 0 \leq \lambda_{er} \leq 2,5^\circ$$

Imprécision humaine



Programme : “ modif_alea(paramètres) ”

→ Modifie les conditions initiales

$$x \rightarrow x \pm 0 \leq x_{er} \leq 0,5m$$

$$y \rightarrow y \pm 0 \leq y_{er} \leq 0,5m$$

$$z \rightarrow z \pm 0 \leq z_{er} \leq 0,2m$$

$$v \rightarrow v \pm 0 \leq v_{er} \leq 2 \text{ m.s}^{-1}$$

$$\alpha \rightarrow \alpha \pm 0 \leq \alpha_{er} \leq 2^\circ$$

$$\lambda \rightarrow \lambda \pm 0 \leq \lambda_{er} \leq 2,5^\circ$$

Risque de rater le service

Service faute (out ou filet) → -1

Service gagnant (non renvoyé) → +1

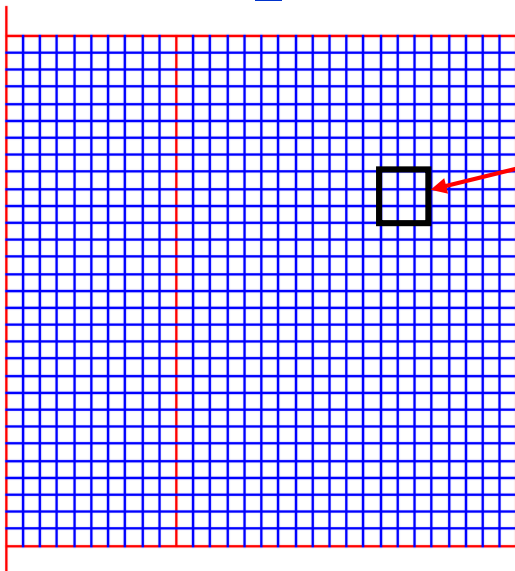
Sinon → 0

Etape 3 : remplir la matrice Terrain avec les espérances de gain associées à chaque case et trouver le maximum.

Programme : “case_a_viser_proba (paramètres, N) ”

Principe de fonctionnement :

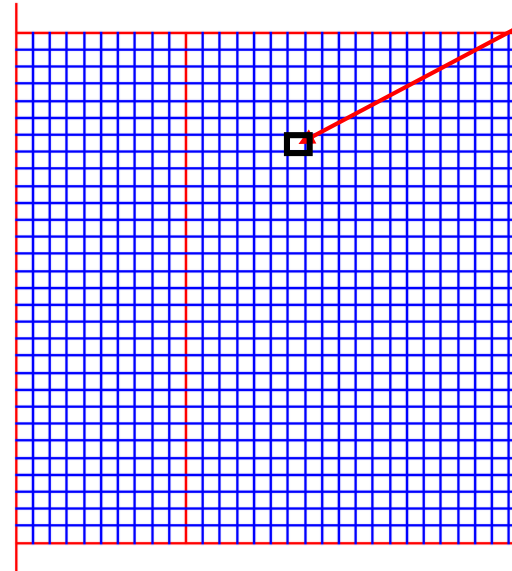
Tableau_reussite



0	0	0
0	0	0
0	0	0

Chaque case = 0

CI



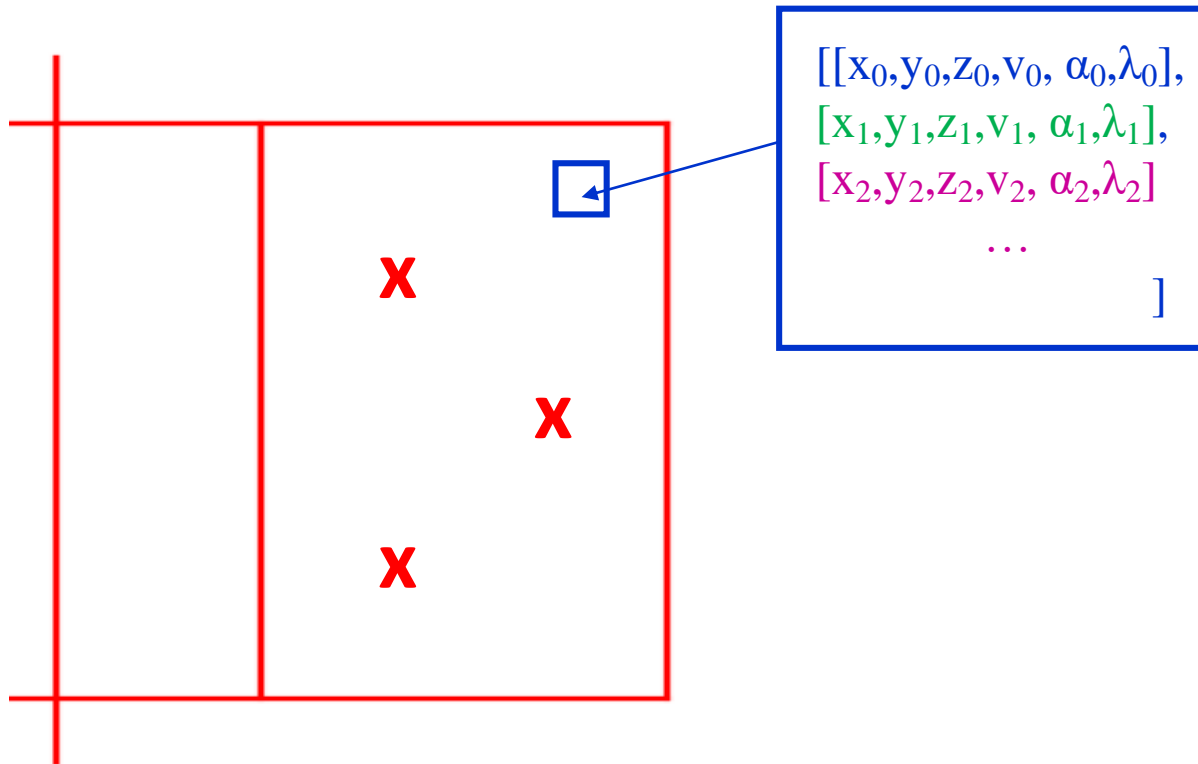
$[[x_0, y_0, z_0, v_0, \alpha_0, \lambda_0],$
 $[x_1, y_1, z_1, v_1, \alpha_1, \lambda_1],$
 $[x_2, y_2, z_2, v_2, \alpha_2, \lambda_2]]$

Chaque case = liste de conditions initiales

Via “remplissage_CI”

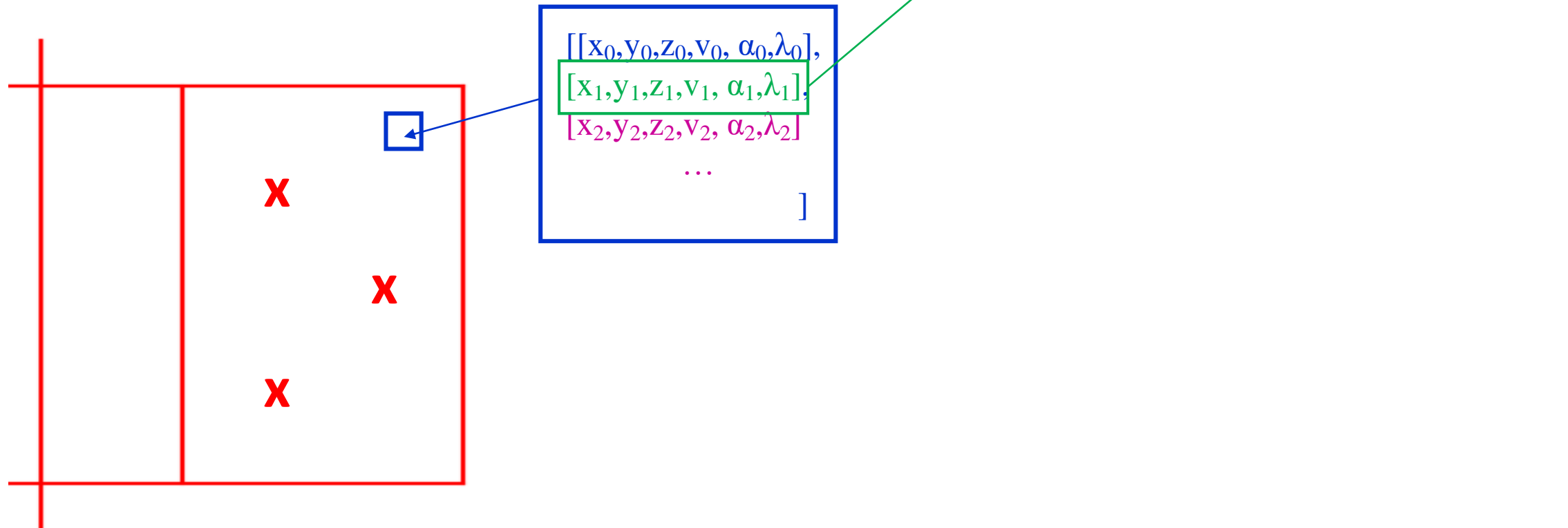
Principe de fonctionnement :

Pour chaque case : 1000 tests réalisés



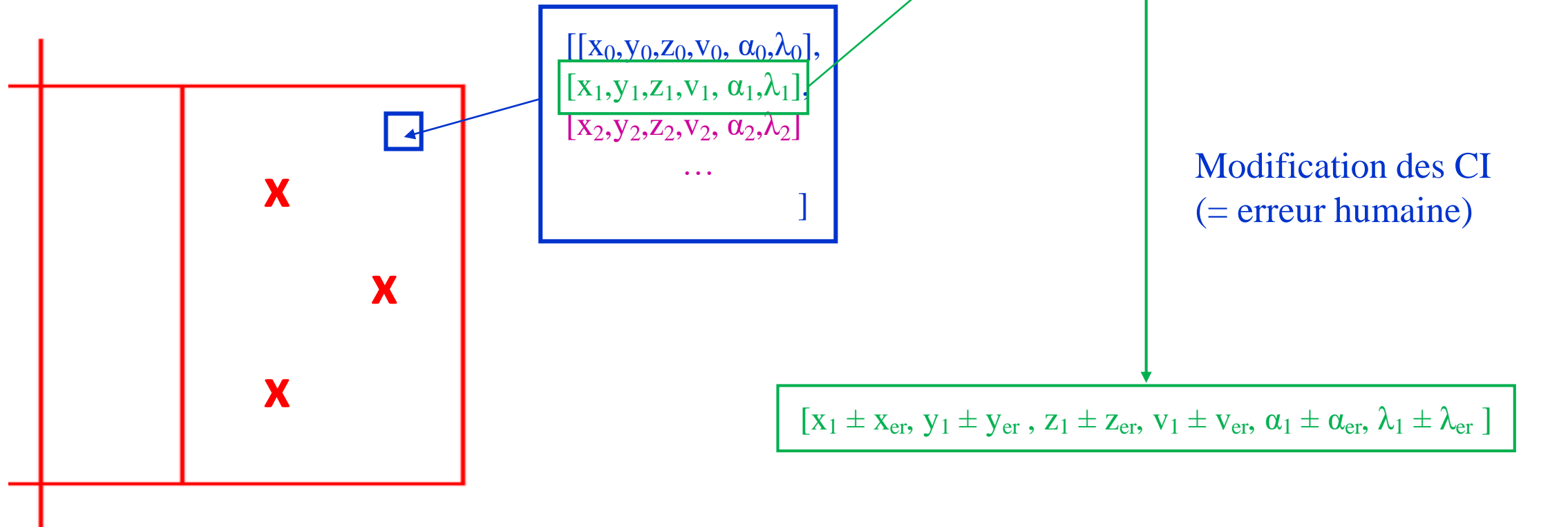
Principe de fonctionnement :

Pour chaque case : 1000 tests réalisés



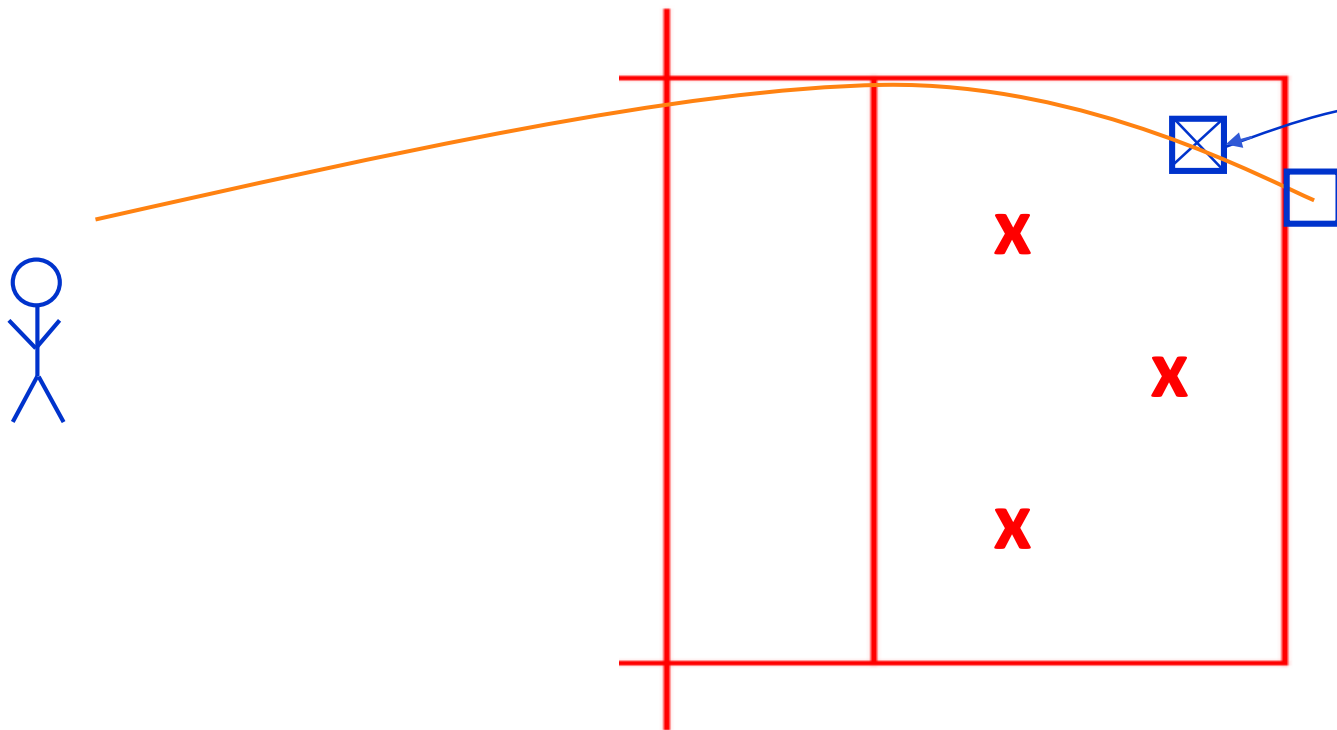
Principe de fonctionnement :

Pour chaque case : 1000 tests réalisés



Principe de fonctionnement :

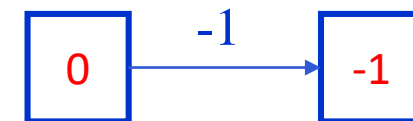
$[x_1 \pm x_{er}, y_1 \pm y_{er}, z_1 \pm z_{er}, v_1 \pm v_{er}, \alpha_1 \pm \alpha_{er}, \lambda_1 \pm \lambda_{er}]$



Si le service est out ou dans le filet \rightarrow -1 pour la case.



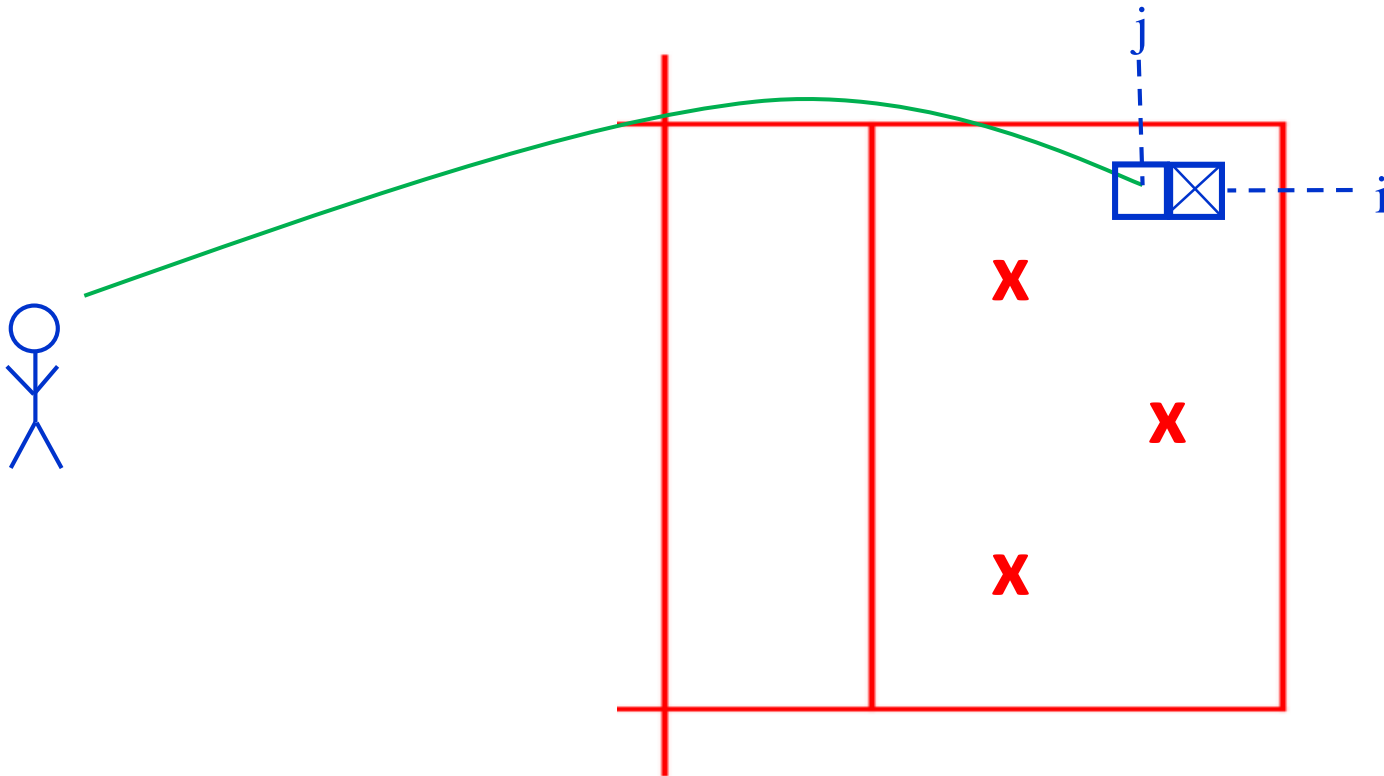
Modification de Tableau_reussite



Principe de fonctionnement :

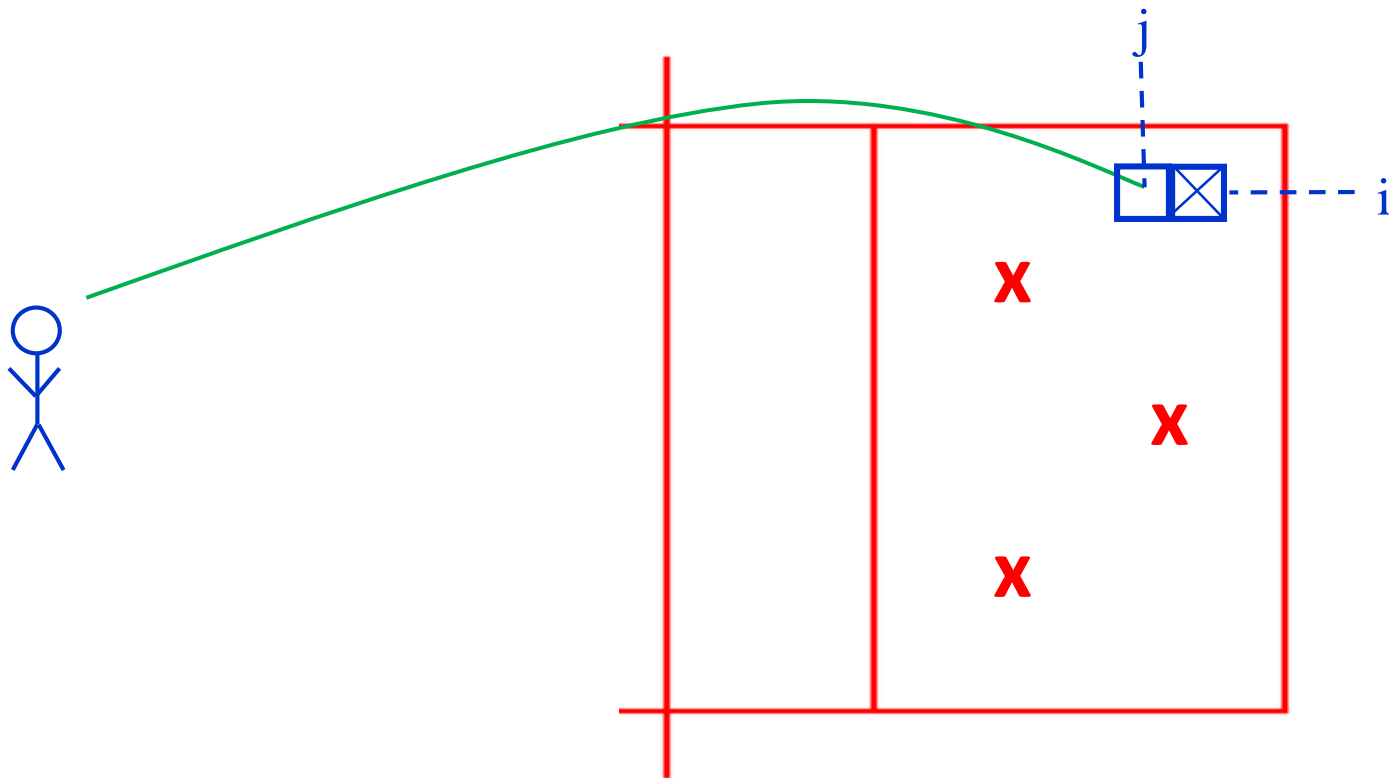
$[x_1 \pm x_{er}, y_1 \pm y_{er}, z_1 \pm z_{er}, v_1 \pm v_{er}, \alpha_1 \pm \alpha_{er}, \lambda_1 \pm \lambda_{er}]$

Si le service est dans le terrain $\rightarrow (i,j)$



Principe de fonctionnement :

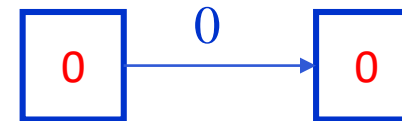
$[x_1 \pm x_{er}, y_1 \pm y_{er}, z_1 \pm z_{er}, v_1 \pm v_{er}, \alpha_1 \pm \alpha_{er}, \lambda_1 \pm \lambda_{er}]$



Si le service est dans le terrain $\rightarrow (i,j)$

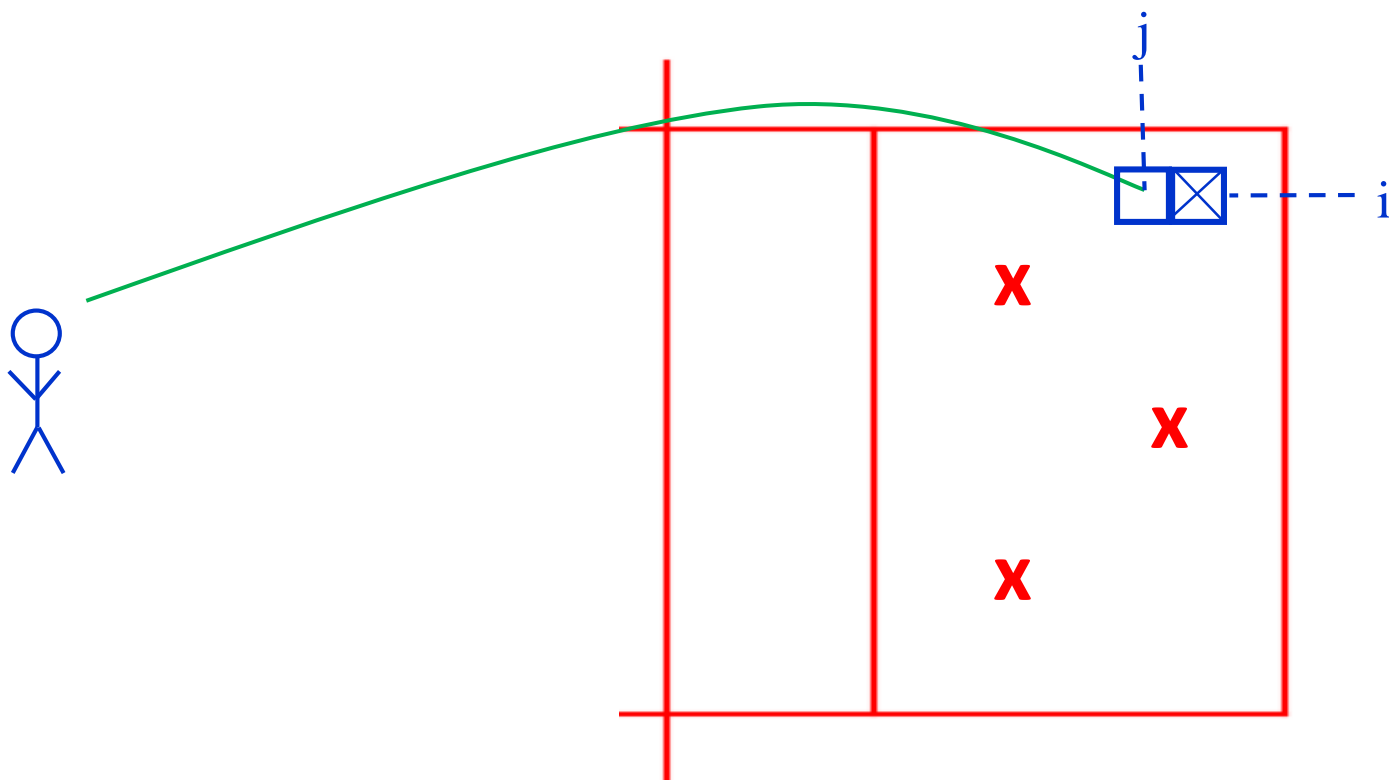
Modification de
Tableau_reussite

Ballon renvoyé



Principe de fonctionnement :

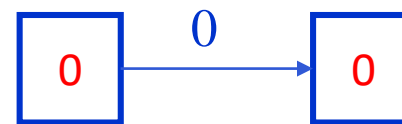
$[x_1 \pm x_{er}, y_1 \pm y_{er}, z_1 \pm z_{er}, v_1 \pm v_{er}, \alpha_1 \pm \alpha_{er}, \lambda_1 \pm \lambda_{er}]$



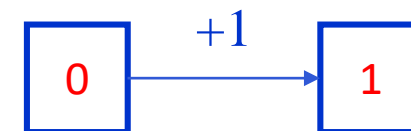
Si le service est dans le terrain $\rightarrow (i,j)$

Modification de
Tableau_reussite

Ballon renvoyé



Ballon non renvoyé

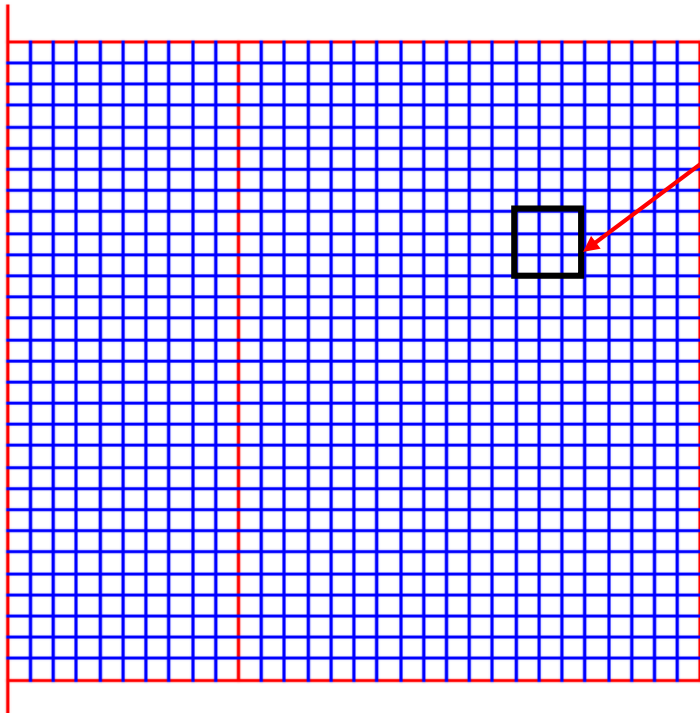


Point marqué par le
serveur

Principe de fonctionnement :

Après 1000 tests :

Tableau_reussite



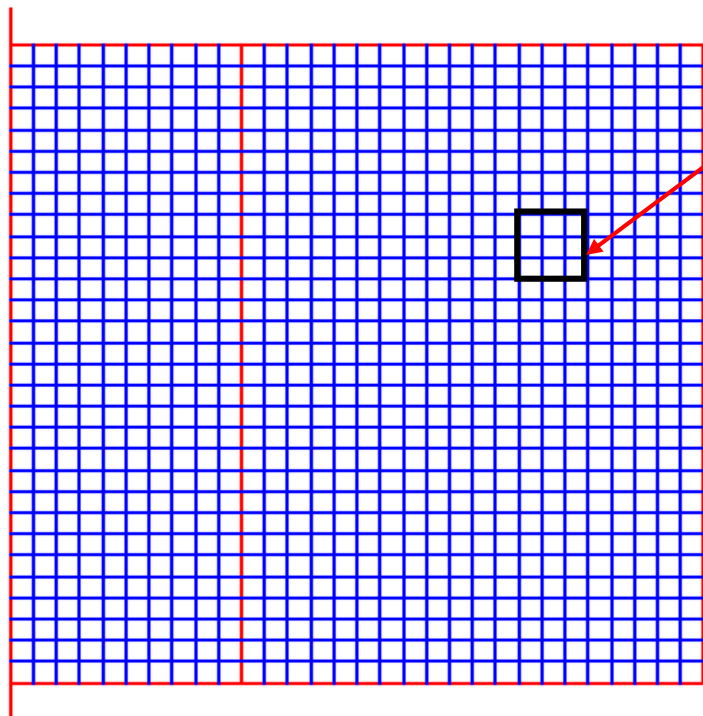
58	980	-32
-340	12	489
-890	266	555

Chaque case : $-1000 \leq n \leq 1000$

Principe de fonctionnement :

Après 1000 tests :

Tableau_reussite



58	980	-32
-340	12	489
-890	266	555

Chaque case : $-1000 \leq n \leq 1000$

$$\frac{\bullet}{\bullet} 1000$$

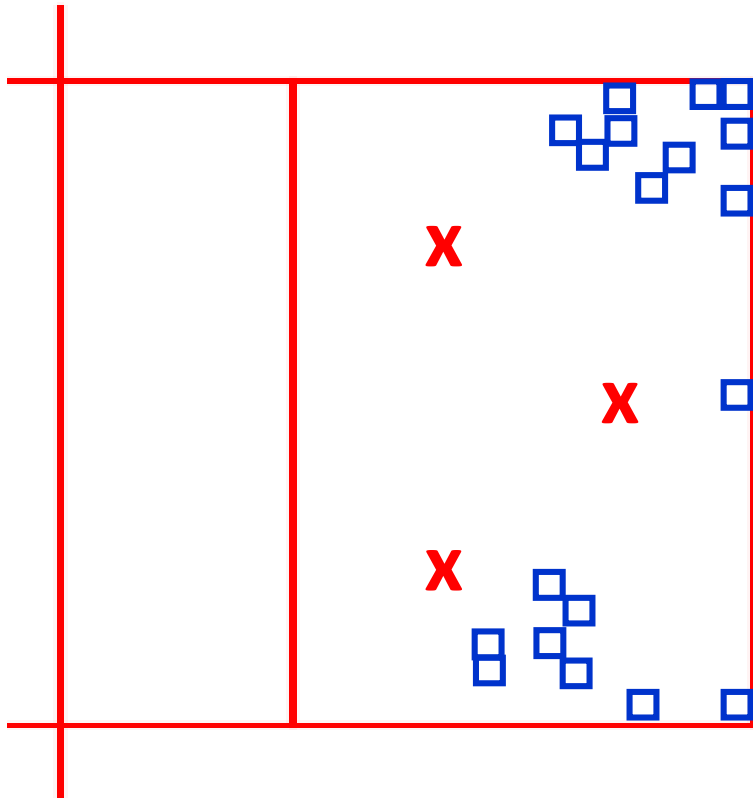


0,058	0,980	-0,032
-0,340	0,012	0,489
-0,890	0,266	0,555

$$-1 \leq p \leq 1$$

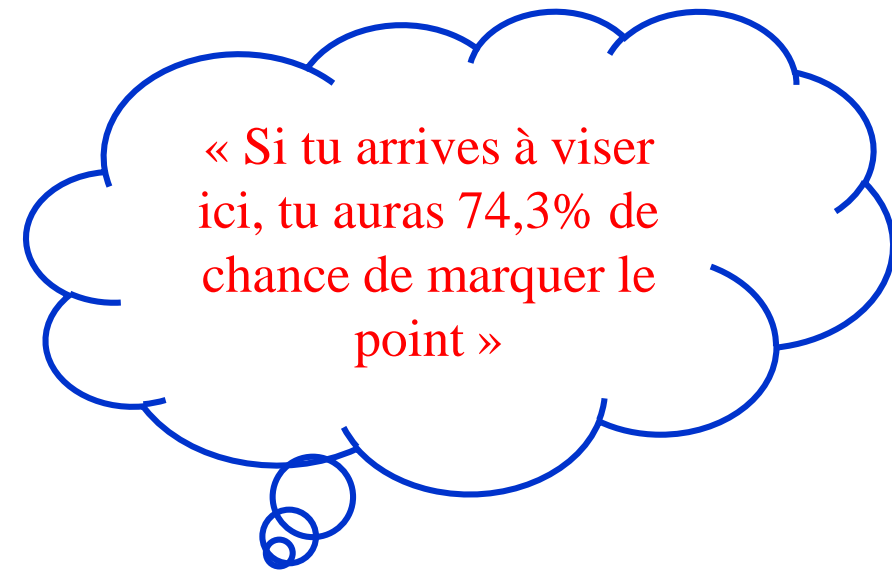
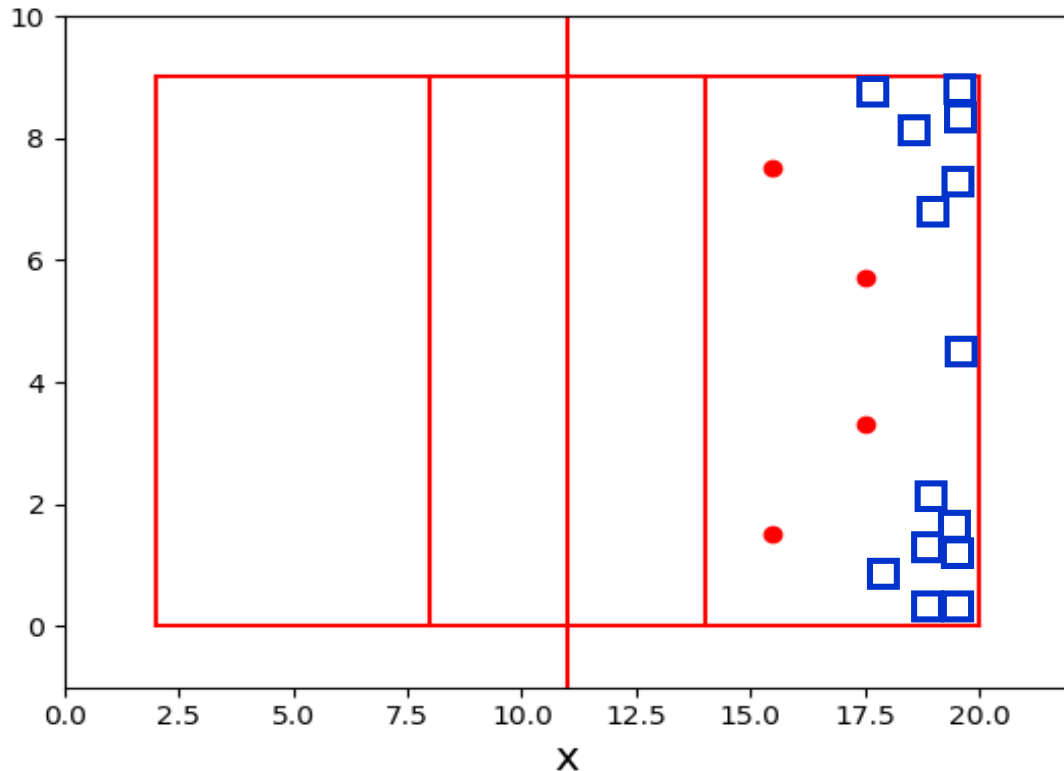
\Leftrightarrow Espérance de gain

→ Stocke les cases où l'espérance de gain est maximale



[[[0, 22], [1, 15], [1, 19], [2, 15], [2, 18], [3, 19], [3, 23], [4, 18], [4, 19],
[5, 20], [6, 21], [8, 21], [19, 17], [19, 19], [20, 17], [20, 19], [20, 20],
[21, 18], [22, 17]], 19, 0.899]

Réalisation du même processus mais pour 4 récepteurs :



[[[1, 18], [1, 22], [1, 23], [2, 21], [4, 17], [4, 22], [7, 19], [9, 18], [13, 20], [14, 17], [18, 17], [21, 17], [22, 20], [24, 15]], 14, 0.743]

Observation :

Pour 3 réceptionneurs : $p=0,899$

Pour 4 réceptionneurs : $p=0,743$

→ Réception à 4 plus efficace

MAIS en pratique : la réception à 3 joueurs est la plus courante

Idée : tester toutes les configurations possibles de réception à 3 et 4 joueurs pour obtenir la réception idéale.

Exemple (pour 3 joueurs) :



Programme : “meilleure_recep (paramètres, N)”

→ 6 boucles imbriquées faisant varier les positions x et y des joueurs en réception entre 0 et 30.

→ Programme interminable car complexité trop élevée.

CONCLUSION

→ Objectif global du TIPE atteint

→ Objectif global du TIPE atteint

→ Mais modèle qui reste à améliorer car :

- la modélisation par une gaussienne reste une approximation.
- le programme ne prend pas en compte le niveau individuel des joueurs.
- le programme sur l'optimisation de la réception est à revoir.

→ Objectif global du TIPE atteint

→ Mais modèle qui reste à améliorer car :

- la modélisation par une gaussienne reste une approximation.
- le programme ne prend pas en compte le niveau individuel des joueurs.
- le programme sur l'optimisation de la réception est à revoir.

→ Idées d'amélioration :

- études de vidéos des réceptions.
- réussir à modéliser le niveau de chaque joueur en ajoutant des paramètres.
- réduire la complexité du programme concernant la réception en prenant en compte les fautes de positions entre les joueurs, ou en imaginer un autre.