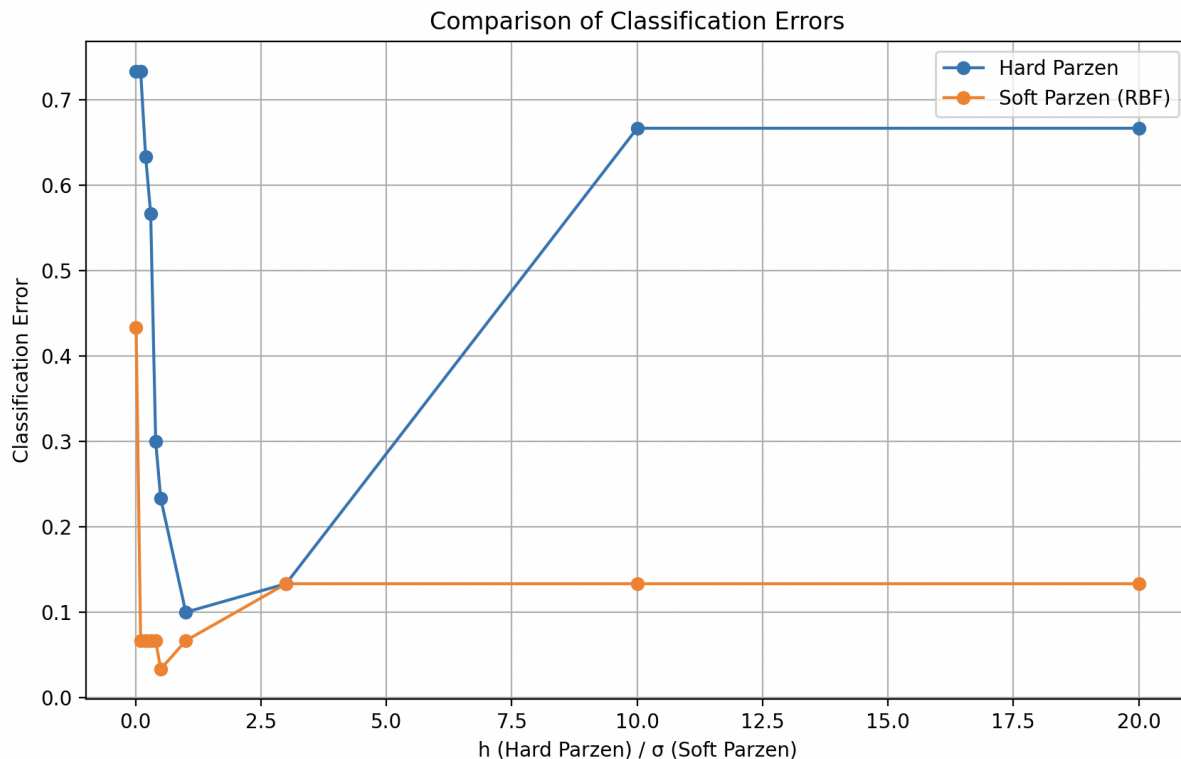


**Question 5** - graphe avec l'erreur de classification de Hard Parzen et l'erreur de classification de Soft Parzen (RBF Kernel)



#### Observations pour le Hard Parzen :

Pour de petites valeurs de  $h$  (entre 0.01 et 1.0), l'erreur de classification est relativement faible (environ 0.3), indiquant que le modèle parvient à bien capturer la structure locale des données. Autrement dit, **il parvient à utiliser efficacement les informations des points voisins pour faire des prédictions correctes**. Cependant, à mesure que  $h$  augmente au-delà de 1.0, l'erreur de classification s'accroît progressivement, atteignant un plateau à environ 0.7 pour  $h \geq 10.0$ . Cela suggère qu'un  **$h$  plus grand** inclut des points plus éloignés comme voisins, ce qui **dégrade les performances de classification**.

#### Observations pour le Soft Parzen (RBF Kernel) :

Pour des petites valeurs de  $\sigma$ , notamment à  $\sigma = 0.01$ , l'erreur de classification est très élevée ( $\sim 0.5$ ), indiquant que le noyau est trop étroit et **ne capte pas suffisamment d'informations des points proches**. À mesure que  $\sigma$  augmente, l'erreur diminue

rapidement, atteignant presque 0 autour de  $\sigma = 0.2$ , ce qui représente des performances optimales. Au-delà de  $\sigma = 1.0$ , l'erreur de classification augmente légèrement mais reste inférieure à 0.3 même pour  $\sigma = 20.0$ .

### **Comparaison entre Hard Parzen et Soft Parzen :**

La performance du Soft Parzen surpasse le Hard Parzen pour la majorité des valeurs de  $h$  et  $\sigma$ . Le Soft Parzen atteint une erreur de classification beaucoup plus faible à des valeurs optimales de  $\sigma$ , tandis que le Hard Parzen n'arrive pas à descendre en dessous de 0.3, même dans ses meilleures conditions. De plus, le Hard Parzen est plus sensible à l'augmentation de  $h$ , avec une dégradation rapide des performances, alors que le Soft Parzen montre une montée plus progressive de l'erreur, rendant ce dernier plus robuste et performant.

En conclusion, le **Soft Parzen avec un noyau RBF** offre de **meilleures performances globales et une plus grande flexibilité**, tandis que le **Hard Parzen** est **plus sensible au choix de  $h$**  et montre des erreurs plus élevées pour des valeurs plus grandes.

### Question 7 - discussion sur la complexité temporelle des deux méthodes

Les méthodes Hard Parzen et Soft Parzen partagent une complexité temporelle similaire. Dans les deux cas, la complexité théorique est de  $O(m \times n \times d)$ , où  $m$  est la taille de l'ensemble de test,  $n$  celle de l'ensemble d'entraînement, et  $d$  la dimension des données. Pour **Hard Parzen**, chaque point de l'ensemble de test est classé en fonction des points voisins situés dans un rayon  $h$ , ce qui nécessite de calculer les distances entre les points de test et ceux de l'entraînement, suivi d'une simple prise de décision en comptant les voisins. En revanche, **Soft Parzen** utilise un noyau gaussien paramétré par  $\sigma$ , pondérant les voisins selon leur distance, ce qui ajoute un léger coût computationnel pour chaque voisin.

Lorsque  $h$  ou  $\sigma$  changent, la complexité théorique reste inchangée, mais en pratique, **un  $h$  ou un  $\sigma$  plus élevé peut augmenter le nombre de voisins pris en compte**, ce qui pourrait **ralentir la prise de décision**. Hard Parzen tend à être plus rapide, car il n'y a pas de pondération par une fonction noyau, contrairement à Soft Parzen qui, en utilisant la fonction RBF, nécessite des calculs supplémentaires pour appliquer les pondérations aux voisins. En résumé, les deux méthodes évoluent linéairement avec la taille des ensembles de données, mais Soft Parzen introduit une légère surcharge en termes de calcul en raison de la fonction noyau.