Université Grenoble Alpes
MAP350 – L3 MI

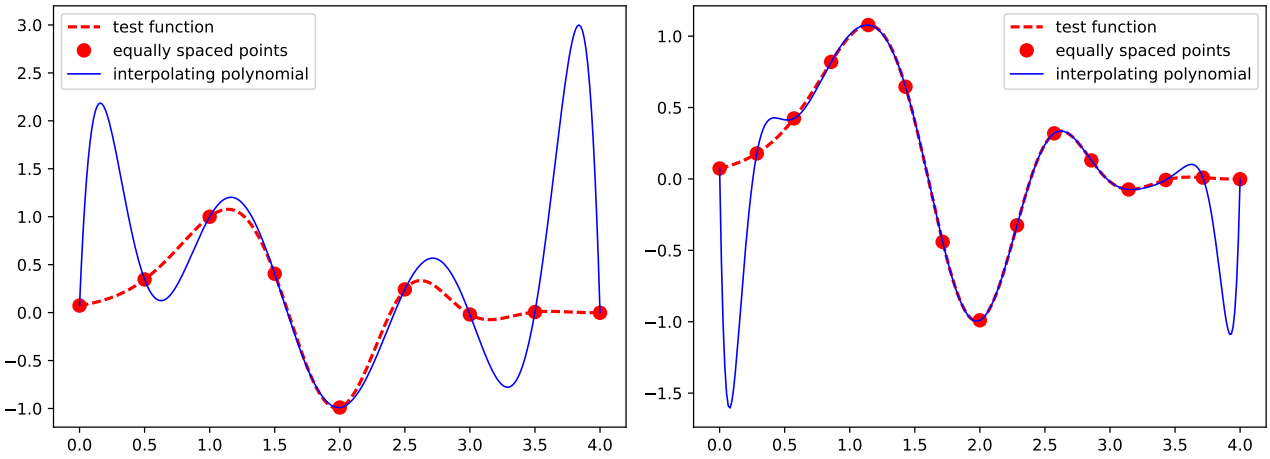
TP3 : Interpolation de Lagrange – base de Newton

---

1 séance de TP

L'interpolation de Lagrange consiste à faire "*passer*" un polynôme de degré adéquat par des points (x_i, y_i) où les abscisses x_i sont 2 à 2 distinctes. On considère ici la base de Newton. Dans la première question, on suppose que l'ensemble des $n+1$ données d'interpolation (x_i, y_i) sont connus initialement. Dans la deuxième question on considère *l'ajout dynamique* des données d'interpolation.

1) Reproduire la figure 1.5 du cours sur l'interpolation (et reproduite ci-dessous) à l'aide du script Python joint à ce document. La fonction test est précisée dans ce script Python.



Dans ce script Python on remarquera qu'on a utilisé la formule

$$P_n(x) = \delta[x_0] \; + \sum_{k=1}^{n} \delta[x_0, \ldots, x_k] \;\; (x - x_0) \cdots (x - x_{k-1}) \tag{1}$$

et que ce polynôme d'interpolation $P_n(x)$ est évalué selon le schéma de Horner sur la base de Newton. Par exemple, pour $n = 4$, et en notant $\delta_k = \delta[x_0, \ldots, x_k]$, on écrit

$$P_4(x) = \delta_0 + (x - x_0)\Big[\delta_1 + (x - x_1)\Big[\delta_2 + (x - x_2)\big[\delta_3 + (x - x_3)\big[\,\delta_4\,\big]\big]\Big]\Big]$$

L'algorithme de Horner est explicité en préambule du cours sur l'interpolation.

2) Modifier ce script Python afin de permettre une interpolation dynamique tel que décrite ci-dessous.

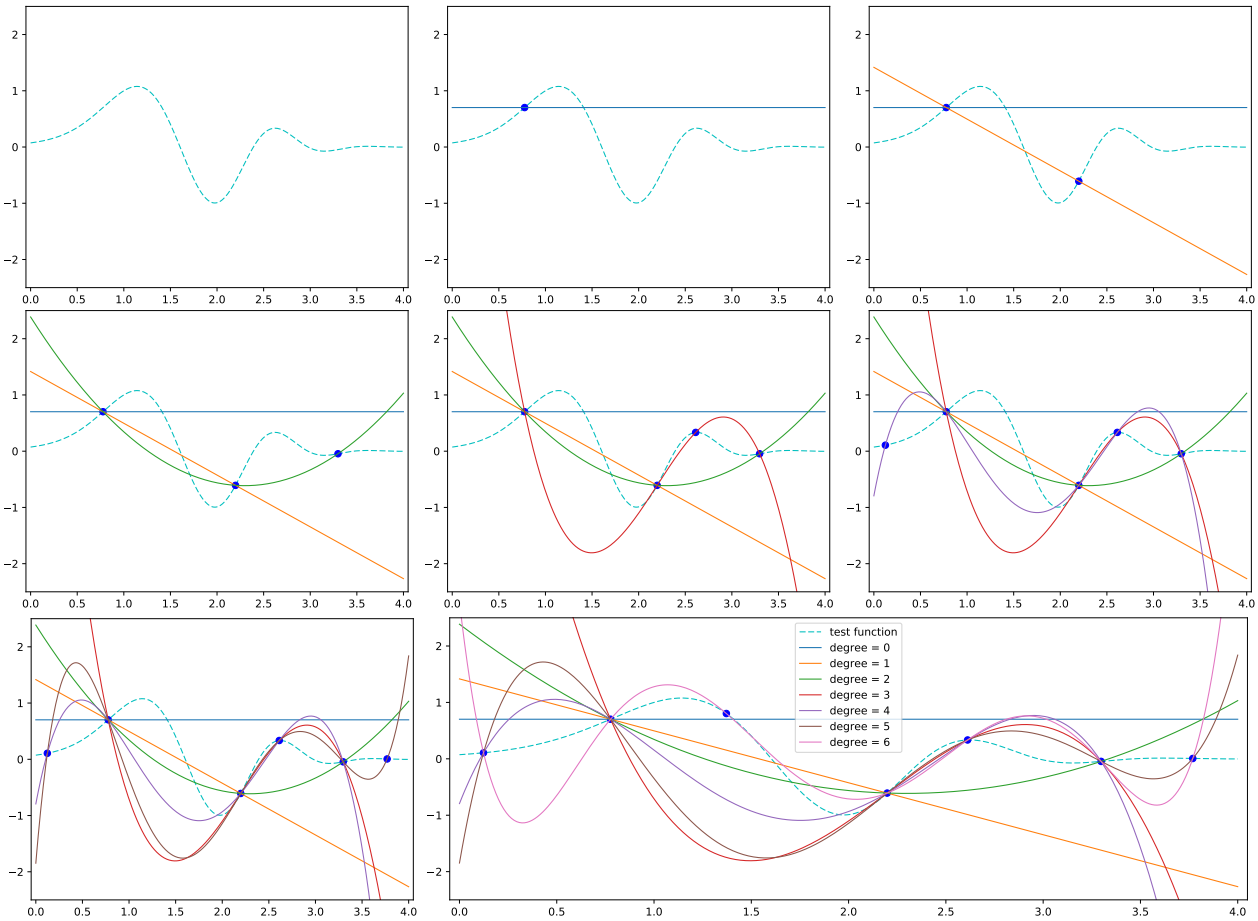
- Les données d'interpolation $(x_i, y_i = f(x_i))$, $i = 0, 1, 2, \ldots$ sont acquises dynamiquement à l'aide de la fonction `AcquisitionOnePoint()`.

- Pour chaque nouveau point (x_{k+1}, y_{k+1}) on détermine et trace sur la même figure le polynôme d'interpolation $P_{k+1}(x)$ des données $(x_0, y_0), \ldots, (x_k, y_k), (x_{k+1}, y_{k+1})$, comme illustré sur la suite des figures ci-dessous.

- Pour le calcul de ce polynôme d'interpolation $P_{k+1}(x)$ sur les $k+2$ données $x_0, \ldots, x_k, x_{k+1}$, on considérera la relation du cours

$$\begin{aligned} P_{k+1}(x) &= P_k(x) \ + \ \delta_{k+1} \ N_{k+1}(x) \\ &= P_k(x) \ + \ \delta_{k+1} \ N_k(x) \ (x - x_k) \end{aligned} \tag{2}$$

qui permet d'obtenir ce polynôme $P_{k+1}(x)$ par mise à jour des calculs précédents (différences divisées et base de Newton) ce qui en particulier contraint à ne plus utiliser l'algorithme de Horner.



Newton interpolation of the function $f(x) = \cos(1-x^2)\ e^{-x^2+3x-2}$ *over 7 points* x_i *dynamically acquired in the interval* $[0, 4]$

3) Déterminer le coût (nombre de multiplications et/ou divisions) de chacune de ces 2 approches. Serait-il intéressant dans le deuxième cas (données dynamiques) d'utiliser l'algorithme de Horner pour l'évaluation, comme dans le premier cas, sur l'ensemble des données d'interpolation connues (autrement dit en *oubliant* les calculs précédemment effectués) ?

Le compte rendu de ce TP consistera en un fichier PYTHON dont le nom sera `TP3_NOM1_NOM2.py`
Il s'agira du fichier donné initialement, complété et renommé, précisément :
— la partie 2 (interpolation dynamique) aura été complétée,
— la partie 3 (coût des 2 approches....) sera détaillée en commentaire.