1 Analyse de la convergence

On va essayer de construire des polynômes qui passent par un ensemble (nuage) de points donnés.

Si ces points sont les valeurs d'une fonction, on aimerait :

- avoir un polynôme construit et d'autant plus proche de la fonction que le nombre de points est grand.

C'est-à-dire, est-ce que la suite des "meilleurs" polynômes tend vers la fonction lorsque le nombre de points tend vers l'infini?

Question : Comment quantifier cette convergence? C'est-à-dire à quelle vitesse (ordre) cette convergence a lieu.

1.1 Approches

- **Approche 1 :** Approximation linéaire
  - Moindre carré de degré 1
- **Approche 2 :** Polynôme d'ordre 1
  - Interpolation polynomiale (Lagrange)
- **Approche 3 :** Autres approches
  - Splines, ondelettes, etc.

1.2 Valeur approchée par itération

1.2.1 Définition de convergence

Definition 1.1. Soit $(x_n)_{n\in\mathbb{N}} \subset \mathbb{R}^d$ une suite qui converge vers $x^* \in \mathbb{R}^d$, pour une norme $\|\cdot\|$ de $\mathbb{R}^d$.

1.2.2 Vitesse (ordre) de convergence

- **Convergence linéaire (ordre 1)**: Si $K_1 = \lim\limits_{n\to+\infty} \dfrac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|}$ existe et $K_1 \in [0, 1[$, on dit que la suite converge **linéairement** vers x^*, ou que la convergence est d'ordre 1.
- **Convergence quadratique (ordre 2)**: Si $K_1 = 0$, $K_2 = \lim\limits_{n\to+\infty} \dfrac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^2}$ existe et non nul, on dit que la suite converge **quadratiquement** vers x^*, ou que la convergence est d'ordre 2.
- **Convergence d'ordre q**: Si $K_q = \lim\limits_{n\to+\infty} \dfrac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^q}$ existe et $\neq 0$, la convergence est d'ordre q.

Remark 1.2. La constante K_1 est appelée constante asymptotique d'erreur pour la convergence linéaire.

1.2.3 Exemples

Example 1.3. Soit $x_n = (0.2)^n$. On a $\lim\limits_{n\to+\infty} x_n = 0$. La convergence est vers $x^* = 0$.

$$\lim_{n\to+\infty} \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|} = \lim_{n\to+\infty} \frac{(0.2)^{n+1}}{(0.2)^n} = 0.2 \in [0, 1[$$

D'où, x_n converge à l'ordre 1. Sa constante asymptotique est $K_1 = 0.2$.

Example 1.4. Soit $y_n = (0.2)^{2^n}$.

$$y_{n+1} = (0.2)^{2^{n+1}} = (0.2)^{2^n \cdot 2} = ((0.2)^{2^n})^2 = (y_n)^2$$

$$\lim_{n\to+\infty} \frac{\|y_{n+1} - x^*\|}{\|y_n - x^*\|^2} = \lim_{n\to+\infty} \frac{y_{n+1}}{(y_n)^2} = \lim_{n\to+\infty} \frac{(y_n)^2}{(y_n)^2} = 1$$

D'où, convergence d'ordre 2, de constante $K_2 = 1$.

1.2.4 Définition formelle de la convergence d'ordre q

Definition 1.5. On dit que x_n converge vers x^* à l'ordre q s'il existe un entier $N \in \mathbb{N}$ et des constantes $A, B \in \mathbb{R}$ telles que pour tout $n > N$,

$$0 < A \leq \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^q} \leq B < +\infty$$

Remark 1.6. La convergence est au moins d'ordre q si seulement on a

$$\limsup_{n\to+\infty} \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^q} < +\infty$$

1.3 Interprétation pratique de la vitesse de convergence

1.3.1 Nombre de chiffres significatifs

Remark 1.7. Si $|x_n - x^*| = 4 \cdot 10^{-8} = 0.\underbrace{00000004}_{\text{8 digits}}$, on dit que x_n et x^* ont 8 chiffres exacts après la virgule.

$$\log_{10} |x_n - x^*| = \log_{10}(4 \cdot 10^{-8}) = \log_{10} 4 - 8 \approx -8$$

On pose $d_n = -\log_{10} |x_n - x^*|$. d_n mesure approximativement le nombre de chiffres décimaux exacts entre x_n et x^*.

$$\lim_{n\to+\infty} \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^q} = K_q \Rightarrow \|x_{n+1} - x^*\| \approx K_q \|x_n - x^*\|^q$$

$$\begin{aligned} \log_{10} \|x_{n+1} - x^*\| &\approx \log_{10}(K_q \|x_n - x^*\|^q) \\ &= \log_{10} K_q + q \log_{10} \|x_n - x^*\| \end{aligned}$$

$$\begin{aligned} -d_{n+1} &\approx \log_{10} K_q + q(-d_n) \\ d_{n+1} &\approx q d_n - \log_{10} K_q \end{aligned}$$

Si $q = 1$, $d_{n+1} \approx d_n - \log_{10} K_1$. À chaque itération, on gagne environ $-\log_{10} K_1$ chiffres significatifs.

Si $q > 1$, $d_{n+1} \approx q d_n$. Le nombre de chiffres significatifs est approximativement multiplié par q à chaque itération.

1.4 Nombre d'itérations pour gagner un chiffre en convergence linéaire

Proposition 1.8. Si x_n converge à l'ordre 1 vers x^* avec une constante asymptotique K_1. Alors, le nombre d'itérations nécessaires pour gagner un chiffre significatif est approximativement $-\dfrac{1}{\log_{10}(K_1)}$.

Preuve. Soit m le nombre d'itérations pour gagner 1 chiffre significatif. Pour une convergence linéaire, $d_{n+1} \approx d_n - \log_{10} K_1$. En partant de d_n, après m itérations, on aura:

$$d_{n+m} \approx d_n - m \log_{10} K_1$$

On souhaite gagner 1 chiffre significatif, donc $d_{n+m} \approx d_n + 1$.

$$d_n + 1 = d_n - m \log_{10} K_1 \Leftrightarrow 1 = -m \log_{10} K_1 \Leftrightarrow m = -\frac{1}{\log_{10} K_1}$$

□

1.5 Linéarisation pour estimation graphique de q

$$\begin{aligned}
\frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^q} &\approx K_q \\
\log_{10} \|x_{n+1} - x^*\| &\approx \log_{10}(K_q \|x_n - x^*\|^q) \\
\log_{10} \|x_{n+1} - x^*\| &\approx \underbrace{q \log_{10} \|x_n - x^*\|}_{x} + \underbrace{\log_{10} K_q}_{b}
\end{aligned}$$

C'est de la forme $y = qx + b$, où $y = \log_{10} \|x_{n+1} - x^*\|$, $x = \log_{10} \|x_n - x^*\|$, q est la pente et $b = \log_{10} K_q$ est l'ordonnée à l'origine.

1.6 Procédure pour déterminer q graphiquement

1. Calculer les erreurs $\|x_n - x^*\|$ et $\|x_{n+1} - x^*\|$ pour les itérations disponibles.
2. Calculer les logarithmes (base 10) de ces erreurs: $\log_{10} \|x_n - x^*\|$ et $\log_{10} \|x_{n+1} - x^*\|$.
3. Tracer le nuage de points $(\log_{10} \|x_n - x^*\|, \log_{10} \|x_{n+1} - x^*\|)$.
4. Estimer graphiquement ou par régression linéaire la pente q de la droite qui approxime au mieux ce nuage de points. Cette pente q est une estimation de l'ordre de convergence.

2 Python code pour l'estimation de la convergence (exemple)

Listing 1: Code Python pour l'estimation de la convergence

```python
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import numpy as np

# Exemple de suite xn (remplacez avec votre suite)
xn = np.array([0.7**(i) for i in range(1, 20)])
x_star = 0 # Valeur limite de la suite

errors_n = np.abs(xn - x_star) # Erreur absolue |xn - x*|
log_errors_n = np.log10(errors_n) # Logarithme base 10 des erreurs
```

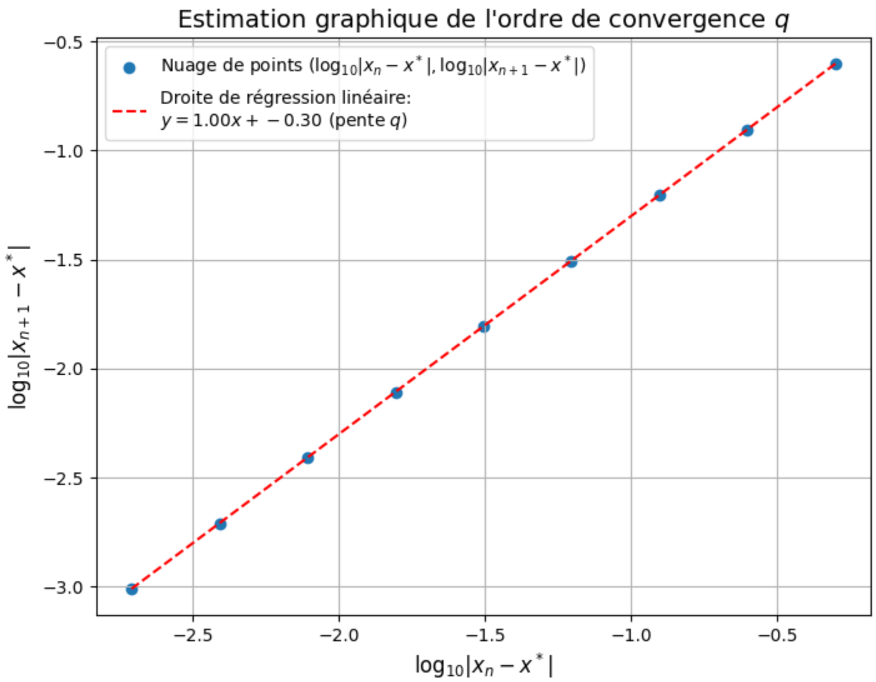

Figure 1: Estimation graphique de l'ordre de convergence q

```
ex = log_errors_n[:-1] # $\log_{10} \|x_n - x^*\|$
ey = log_errors_n[1:]  # $\log_{10} \|x_{n+1} - x^*\|$

plt.figure(figsize=(8, 6))
plt.scatter(ex, ey, label="Nuage de points")

ab = np.polyfit(ex, ey, 1) # Regression lineaire (y = ax + b)
y_fit = ab[0]*ex + ab[1]
plt.plot(ex, y_fit, color='red', linestyle='--', label=f"Droite de r gression: $y = {ab[0


plt.xlabel("$\log_{10} \|x_n - x^*\|$", fontsize=12)
plt.ylabel("$\log_{10} \|x_{n+1} - x^*\|$", fontsize=12)
plt.title("Estimation graphique de l'ordre de convergence $q$", fontsize=14)
plt.legend()
plt.grid(True)
plt.show() # Affiche le graphique (pour execution hors LaTeX)
```