# Module « Optimisation et contrôle »

◇

# Projet sur les réseaux de distribution d'eau

Pierre Carpentier, Adrien Le Franc & Thomas Martin

pierre.carpentier@ensta-paris.fr

adrien.le-franc@enpc.fr

thomas.martin@enpc.fr

Année 2019/2020

**Cours** : `cermics.enpc.fr/~jpc/optimisation.html`

**Projet** : `perso.ensta-paris.fr/~pcarpent/TP_Reseau/ENPC/`

# Objectif et modalités du projet

Prendre prétexte d'un problème d'ingénierie (lié à l'hydraulique urbaine) pour faire apparaître un problème d'optimisation, puis mettre en œuvre sur ce problème les différentes approches et les différents algorithmes présentés dans le cours d'optimisation.

Dans la "vraie vie", il est rare d'écrire une méthode d'optimisation comme le gradient conjugué ou quasi-Newton, car c'est un travail de professionnel. Pour ce projet, vous êtes les professionnels...

**Travail en binôme**.

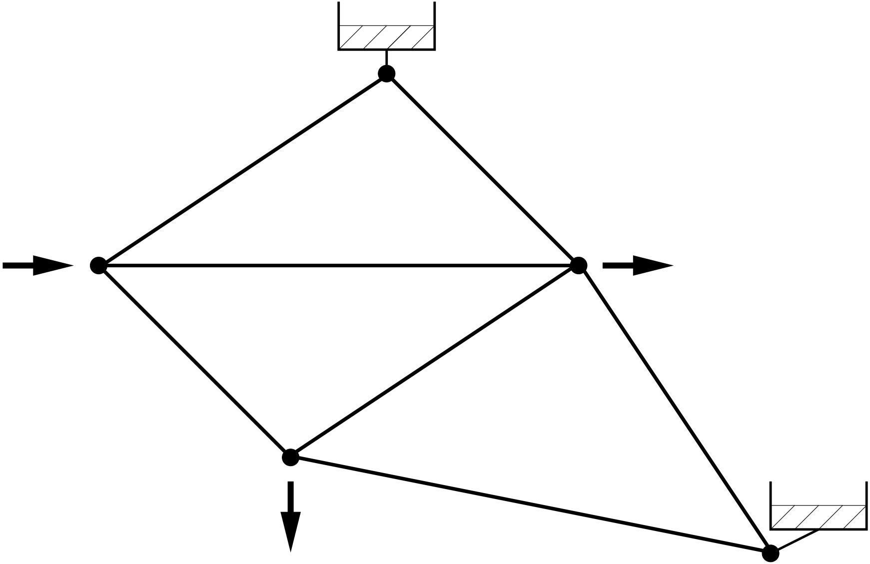
**Langage de programmation** : SCILAB ou PYTHON.

**Contrôle continu** pour l'évaluation du projet.

**Compte rendu** à l'issue de la 3-ème séance.

**Rapport final** à l'issue de la dernière séance.

# Schématique d'un réseau de distribution d'eau



# Variables décrivant le réseau et notations

**Tailles du réseau.**

- $n$ arcs (canalisations du réseau),
- $m$ nœuds : $m_r$ réservoirs, $m_d$ consommateurs $(m = m_r + m_d)$.

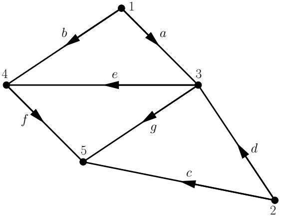
**Variables hydrauliques du réseau.**

- flux aux nœuds : $f = \begin{pmatrix} f_r \\ f_d \end{pmatrix} \in \mathbb{R}^m$ — $f_r$ à calculer, $f_d$ connus.
- pressions aux nœuds : $p = \begin{pmatrix} p_r \\ p_d \end{pmatrix} \in \mathbb{R}^m$ — $p_r$ connues, $p_d$ à calculer.
- résistances des arcs : $r \in \mathbb{R}^n$ connues.
- débits dans les arcs : $q \in \mathbb{R}^n$ à calculer.

$\rightsquigarrow$ $(m+n)$ inconnues à déterminer.

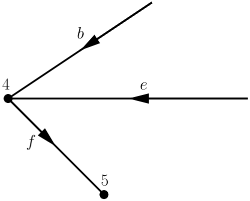
# Matrice d'incidence et équations d'équilibre du réseau

## Matrice d'incidence nœuds–arcs.




$$A = \begin{array}{c} \begin{array}{ccccccc} a & b & c & d & e & f & g \end{array} \\ \begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ +1 & 0 & 0 & +1 & -1 & 0 & -1 \\ 0 & +1 & 0 & 0 & +1 & -1 & 0 \\ 0 & 0 & +1 & 0 & 0 & +1 & +1 \end{pmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}$$

## Équations décrivant l'état d'équilibre.




$$\begin{array}{lcl} \textit{Noeud } 4 & : & q_b + q_e - q_f = f_4. \\ \rightsquigarrow & & Aq - f = 0. \\[2ex] \textit{Arc } f & : & p_4 - p_5 = r_f q_f |q_f|. \\ \rightsquigarrow & & A^\top p + r \bullet q \bullet |q| = 0. \end{array}$$

$\rightsquigarrow \quad (m+n)$ équations.

# Problème d'optimisation associé à l'équilibre

Ces équations d'équilibre sont en fait les **conditions d'optimalité** du problème d'optimisation suivant :

$$\min_{(q\in\mathbb{R}^n, f_r\in\mathbb{R}^{m_r})} \quad \frac{1}{3}\Big\langle q \,, r \bullet q \bullet |q| \Big\rangle + \Big\langle p_r \,, f_r \Big\rangle \,,$$

sous la contrainte :

$$Aq - f = 0 \,,$$

qui s'écrit après réduction (par élimination de variables) comme un problème d'optimisation sans contrainte :

$$\min_{q_C\in\mathbb{R}^{n-m_d}} \; \frac{1}{3}\Big\langle q^{(0)} + Bq_C \,, r \bullet \big(q^{(0)} + Bq_C\big) \bullet \big|q^{(0)} + Bq_C\big| \Big\rangle + \Big\langle p_r \,, A_r\big(q^{(0)} + Bq_C\big) \Big\rangle \,.$$

**C'est ce problème que l'on va résoudre** *(pour commencer...)* **!**

# But du projet

Écrire des méthodes génériques d'optimisation (recherche linéaire, gradient conjugué, quasi-Newton, Newton) qui ne communiquent avec le problème que par l'intermédiaire d'un oracle associé à :

$$F : q_c \mapsto \frac{1}{3}\left\langle q^{(0)} + Bq_c \, , r \bullet \left(q^{(0)} + Bq_c\right) \bullet \left|q^{(0)} + Bq_c\right| \right\rangle + \left\langle p_r \, , A_r\left(q^{(0)} + Bq_c\right) \right\rangle .$$

## Oracle

Fonction qui pour tout $q_c$ calcule $F(q_c)$, $\nabla F(q_c)$, $\nabla^2 F(q_c)$ :

```
[F,G,H] = Oracle(qc,ind).
```

## Algorithme d'optimisation

Fonction qui minimise $F$ en partant d'un point initial $q_{\text{ini}}$ :

```
[Fopt,qopt,Gopt] = Minimise(Oracle,qini).
```

## Recherche linéaire

Fonction qui assure la décroissance de $F$ dans une direction $d$.

# L'algorithme du gradient à pas fixe

```
function [Fopt,qopt,Gopt]=Gradient_F(Oracle,qini)

   iter = 5000 ; tol = 0.000001 ; alpha = 0.0005; qc = qini;

   for k = 1:iter

      [F,G] = Oracle(qc,ind);
      if norm(G) <= tol then
         kstar = k; break
      end
      qc = qc - (alpha*G);

      logG = [ logG ; log10(norm(G)) ]; Cout = [ Cout ; F ];

   end

   Fopt = F; qopt = x; Gopt = G;

   Visualg(logG,Cout);

endfunction
```

# Enchaînement des tâches : Moniteur_Skel.sce

```
// Donnees du probleme
exec('Probleme_R.sce'); exec('Structures_N.sce');

// Fonction de visualisation du deroulement de l'algorithme
exec('Visualg.sci');

// Fonctions de verification des resultats
exec('HydrauliqueP.sci'); exec('Verification.sci');

// Oracle et algorithme d'optimisation
exec('Oracle.sci'); exec('Gradient_F.sci');

// Optimisation
qini = 0.1 * rand(n-md,1);
[Fopt,qopt,Gopt] = Gradient_F(Oracle,qini);

// Verification des resultats
[q,z,f,p] = HydrauliqueP(qopt); Verification(q,z,f,p);
```

# Variables disponibles dans l'environnement Scilab

| Description de la variable | Nom math. | Variable info. | Espace |
|---|---|---|---|
| Nombre total d'arcs | $n$ | `n` | $\mathbb{N}$ |
| Nombre total de nœuds | $m$ | `m` | $\mathbb{N}$ |
| Nombre de nœuds de demande | $m_d$ | `md` | $\mathbb{N}$ |
| Nombre de nœuds réservoir | $m_r$ | `mr` | $\mathbb{N}$ |
| Flux aux nœuds de demande | $f_d$ | `fd` | $\mathcal{M}(m_d, 1)$ |
| Pressions aux nœuds réservoir | $p_r$ | `pr` | $\mathcal{M}(m_r, 1)$ |
| Résistances des arcs | $r$ | `r` | $\mathcal{M}(n, 1)$ |
| Vecteur initial des débits | $q^{(0)}$ | `q0` | $\mathcal{M}(n, 1)$ |
| Matrice d'incidence nœuds-arcs | $A$ | `A` | $\mathcal{M}(m, n)$ |
| Sous-matrice "demande" de $A$ | $A_d$ | `Ad` | $\mathcal{M}(m_d, n)$ |
| Sous-matrice "réservoir" de $A$ | $A_r$ | `Ar` | $\mathcal{M}(m_r, n)$ |
| Sous-matrice "arbre" de $A_d$ | $A_{d,T}$ | `AdT` | $\mathcal{M}(m_d, m_d)$ |
| Sous matrice "coarbre" de $A_d$ | $A_{d,C}$ | `AdC` | $\mathcal{M}(m_d, n - m_d)$ |
| Matrice inverse de $A_{d,T}$ | | `AdI` | $\mathcal{M}(m_d, m_d)$ |
| Matrice d'incidence arcs-cycles | $B$ | `B` | $\mathcal{M}(n, n - m_d)$ |

**Attention** : en Scilab, les variables sont globales !

# Programmation du projet

1. **06 mars (09h45 — 11h15)**

   ⇝ écriture de l'oracle

2. **13 mars (09h45 — 11h15)**

   ⇝ recherche linéaire

3. **20 mars (09h45 — 11h15)**

   ⇝ algorithmes d'optimisation

4. **10 avril (08h30 — 11h15)**

   ⇝ dualité

# Programme de travail de cette première séance

1. Lire attentivement le document descriptif du TP ...

2. Récupérer les documents et les codes SCILAB du TP :

   http://perso.ensta-paris.fr/~pcarpent/TP_Reseau/ENPC/

3. Calculer analytiquement le gradient de la fonction :

$$
\begin{aligned}
F : \mathbb{R}^{n-m_d} &\longrightarrow \mathbb{R} \\
q_c \quad &\mapsto \quad \frac{1}{3}\left\langle q^{(0)} + Bq_c \,, r \bullet \left(q^{(0)} + Bq_c\right) \bullet \left|q^{(0)} + Bq_c\right| \right\rangle + \\
&\qquad\qquad \left\langle p_r \,, A_r\left(q^{(0)} + Bq_c\right) \right\rangle .
\end{aligned}
$$

4. Écrire un oracle codant les expressions obtenues (en SCILAB), et tester cet oracle avec l'algorithme du gradient à pas fixe.

Pour la **prochaine séance** : avoir un oracle en état de marche !

**Ce projet. . .**

# Le vrai problème. . .