Bose-Einstein Condensate Simulations with FreeFem++

Minimization of the Gross-Pitaevskii energy with the interior points optimizer IPOPT

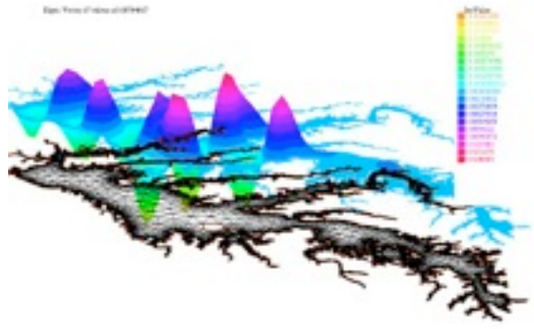
Sylvain Auliac - Laboratoire Jacques-Louis Lions

Outline

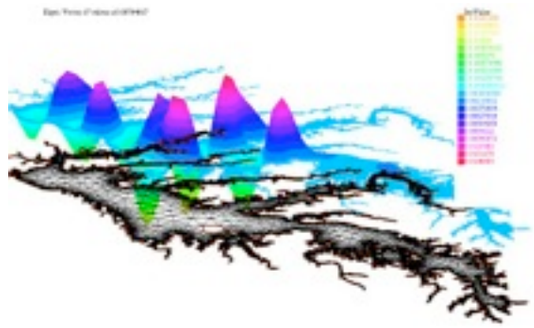
Outline

Introduction

- Optimization tools in FreeFem++

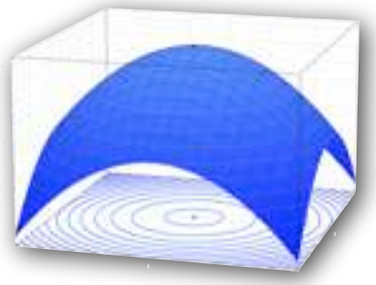


Outline



Introduction

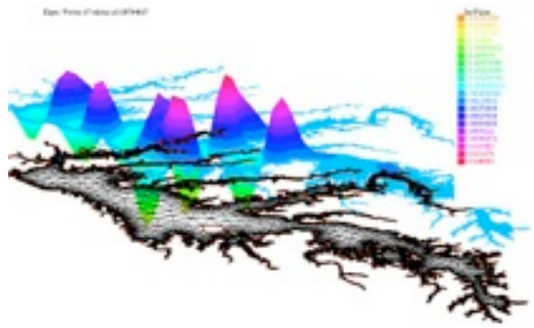
- Optimization tools in FreeFem++



Interior-Point Methods

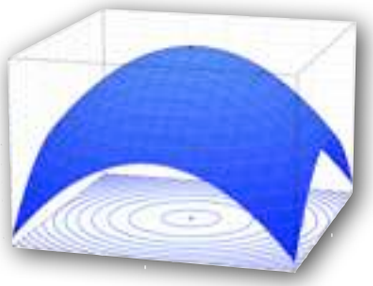
- A short overview
- The IPOPT software
- Using IPOPT through FreeFem++

Outline



Introduction

- Optimization tools in FreeFem++



Interior-Point Methods

- A short overview
- The IPOPT software
- Using IPOPT through FreeFem++

Bose-Einstein Condensate Simulations

- The Gross-Pitaevskii energy
- 2D Simulations
- Results in 3D

Closing...

Optimization Tools of FreeFem++ :

Optimization Tools of FreeFem++ :

Unconstrained Optimization :

Quadratic problems solvers: LinearCG, LinearGMRES

General non-linear optimizers:

- NLCG, BFGS
- CMAES (derivative free evolution strategy, MPI version available)
- NEWUOA (derivative free)

Optimization Tools of FreeFem++ :

Unconstrained Optimization :

Quadratic problems solvers: LinearCG, LinearGMRES

General non-linear optimizers:

- NLCG, BFGS
- CMAES (derivative free evolution strategy, MPI version available)
- NEWUOA (derivative free)

NLopt - Constrained non-linear optimizers :

- Derivative free / stochastic algorithms : DIRECT, CRS, ISRES, NEWUOA, BOBYQA, COBYLA, SBPLX, etc...
- Gradient-based algorithms : LBFGS, MMA, SLSQP, T-Newton, etc...
- Augmented Lagrangian Method

Whole list available on the NLopt web page

Optimization Tools of FreeFem++ :

Unconstrained Optimization :

Quadratic problems solvers: LinearCG, LinearGMRES

General non-linear optimizers:

- NLCG, BFGS
- CMAES (derivative free evolution strategy, MPI version available)
- NEWUOA (derivative free)

NLopt - Constrained non-linear optimizers :

- Derivative free / stochastic algorithms : DIRECT, CRS, ISRES, NEWUOA, BOBYQA, COBYLA, SBPLX, etc...
- Gradient-based algorithms : LBFGS, MMA, SLSQP, T-Newton, etc...
- Augmented Lagrangian Method

Whole list available on the NLopt web page

IPOPT :

Constrained non-linear optimization using sparse matrices

Part I

Interior Points Methods and the IPOPT Software

Interior Point Methods : a short overview

Classical minimization under constraints problem :

Objective function :

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

Constraints :

$$c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

Find : $$x_0 = \underset{\forall i,\ c_i(x) \geq 0}{\operatorname{argmin}} f(x)$$

Classical minimization under constraints problem :

Objective function :

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

Constraints :

$$c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

$$\text{Find : } x_0 = \underset{\forall i,\ c_i(x) \geq 0}{\operatorname{argmin}} f(x)$$

Barrier Function :

For a given $\mu > 0$, solve the unconstrained problem of finding

$$x_\mu = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}}\ f(x) - \mu \sum_{i=1}^{m} \ln c_i(x)$$

Classical minimization under constraints problem :

Objective function : Constraints :

$$f : \mathbb{R}^n \longrightarrow \mathbb{R} \qquad c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

$$\text{Find : } x_0 = \underset{\forall i,\ c_i(x) \geq 0}{\operatorname{argmin}} f(x)$$

Barrier Function :

For a given $\mu > 0$, solve the unconstrained problem of finding

$$x_\mu = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}}\ f(x) - \mu \sum_{i=1}^{m} \ln c_i(x)$$

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k \to +\infty}{\longrightarrow} 0$ find a a sequence (x_{μ_k}) of associated minimizers.

Barrier Function :

For a given $\mu > 0$, solve the unconstrained problem of finding

$$x_\mu = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; f(x) - \mu \sum_{i=1}^{m} \ln c_i(x)$$

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k \to +\infty}{\longrightarrow} 0$ find a a sequence (x_{μ_k}) of associated minimizers.

Barrier Function :
For a given $\mu > 0$, solve the unconstrained problem of finding

$$x_\mu = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; f(x) - \mu \sum_{i=1}^{m} \ln c_i(x)$$

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k \to +\infty}{\longrightarrow} 0$ find a a sequence (x_{μ_k}) of associated minimizers.

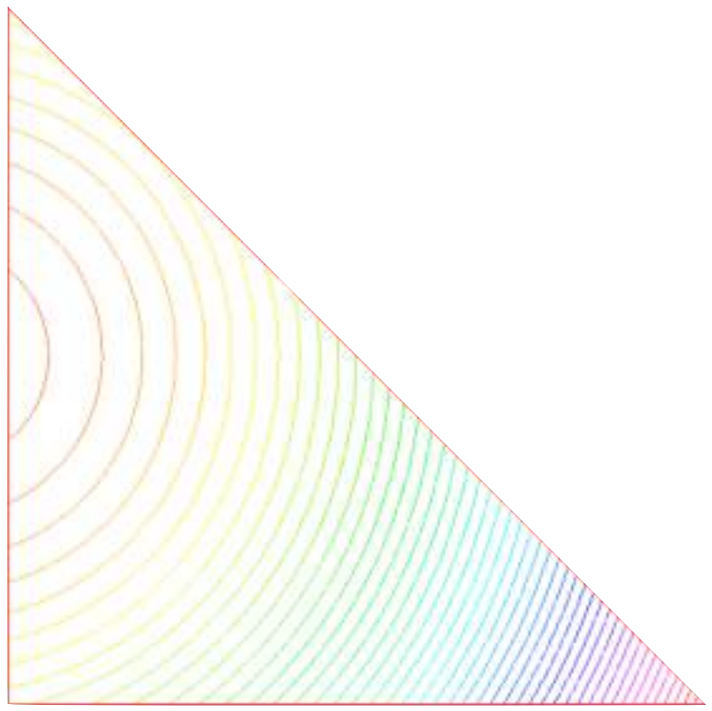
Illustration : minimize $f(x,y) = (x+0.1)^2 + (y-0.5)^2$ on a triangle

Barrier Function :
For a given $\mu > 0$, solve the unconstrained problem of finding

$$x_\mu = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; f(x) - \mu \sum_{i=1}^{m} \ln c_i(x)$$

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k \to +\infty}{\longrightarrow} 0$ find a a sequence (x_{μ_k}) of associated minimizers.

Illustration : minimize $f(x,y) = (x+0.1)^2 + (y-0.5)^2$ on a triangle



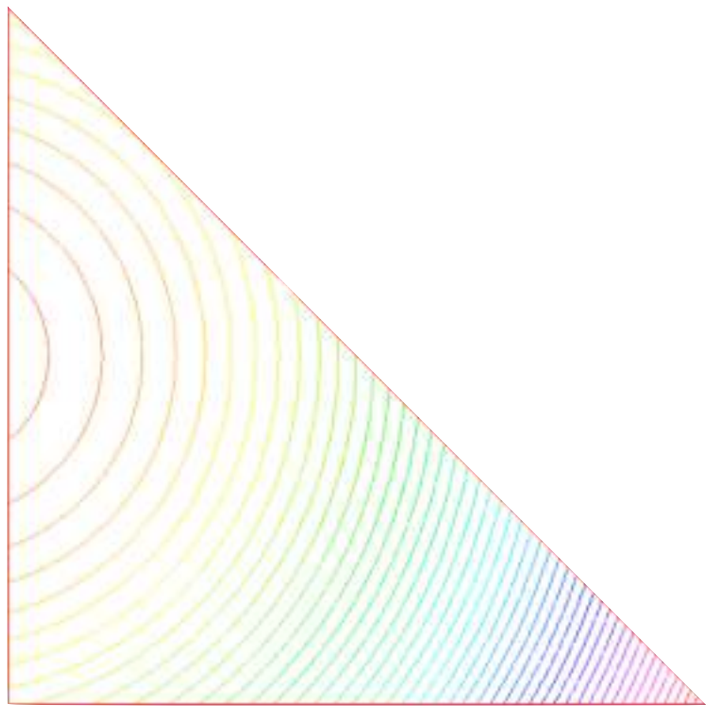
Isovalues of f

Barrier Function :
For a given $\mu > 0$, solve the unconstrained problem of finding

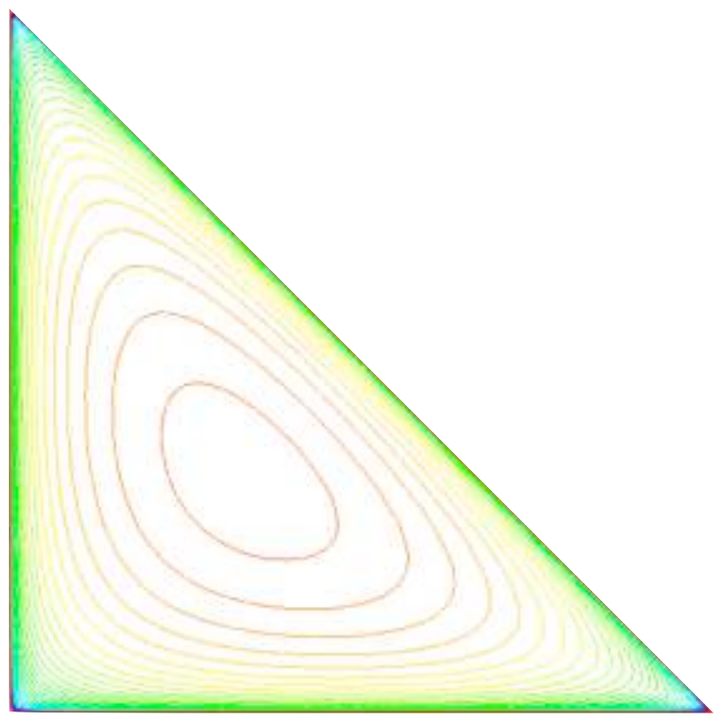
$$x_\mu = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; f(x) - \mu \sum_{i=1}^{m} \ln c_i(x)$$

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k \to +\infty}{\longrightarrow} 0$ find a a sequence (x_{μ_k}) of associated minimizers.

Illustration : minimize $f(x, y) = (x + 0.1)^2 + (y - 0.5)^2$ on a triangle



Isovalues of f



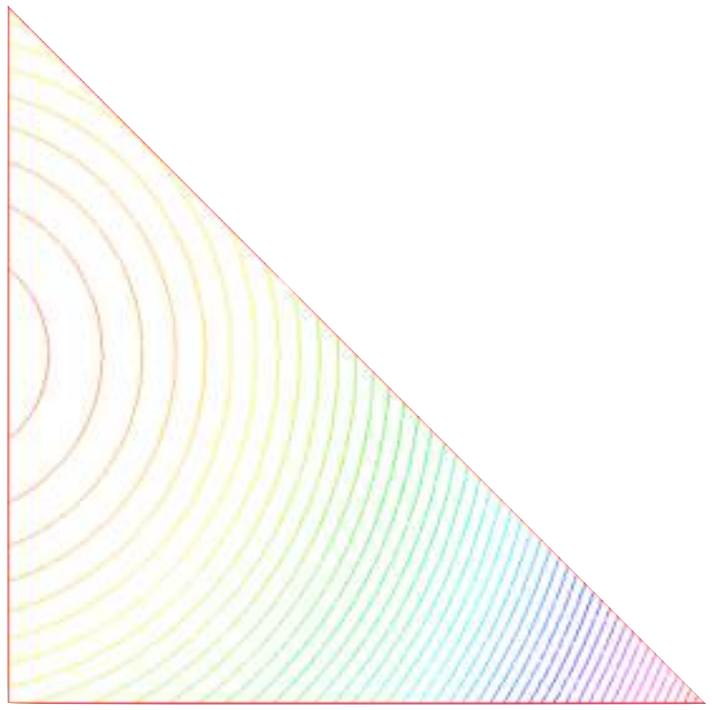
$\log(\mu) = 2$

Barrier Function :
For a given $\mu > 0$, solve the unconstrained problem of finding

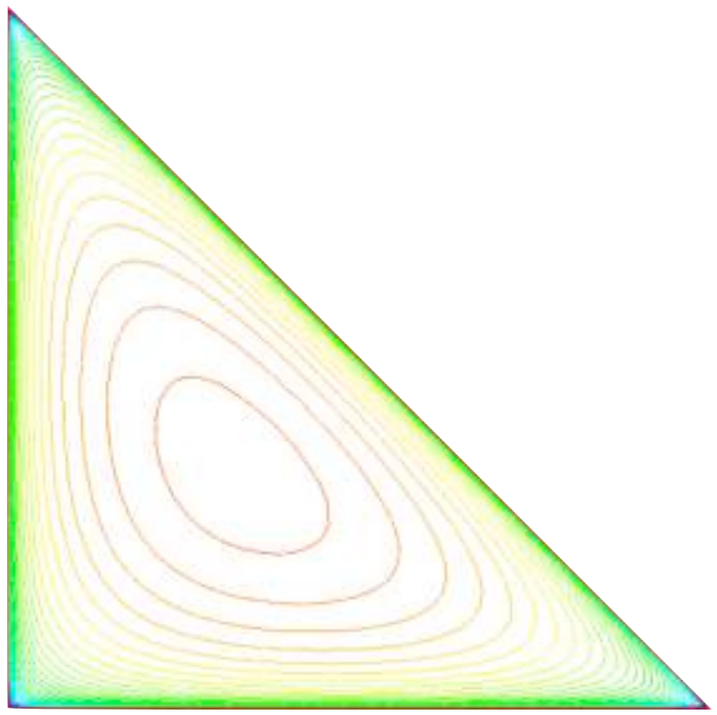
$$x_\mu = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; f(x) - \mu \sum_{i=1}^{m} \ln c_i(x)$$

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k \to +\infty}{\longrightarrow} 0$ find a a sequence (x_{μ_k}) of associated minimizers.

Illustration : minimize $f(x,y) = (x+0.1)^2 + (y-0.5)^2$ on a triangle



Isovalues of f



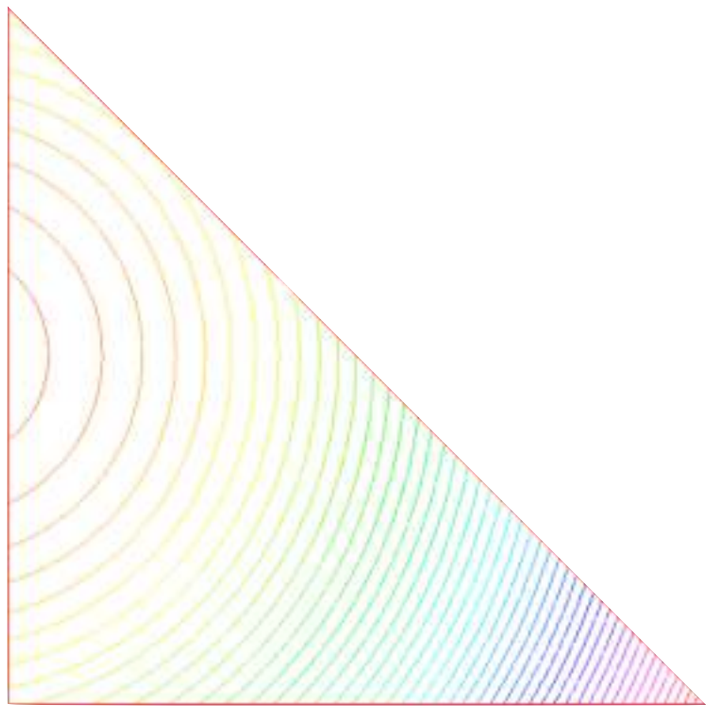
$\log(\mu) = 1$

Barrier Function :
For a given $\mu > 0$, solve the unconstrained problem of finding

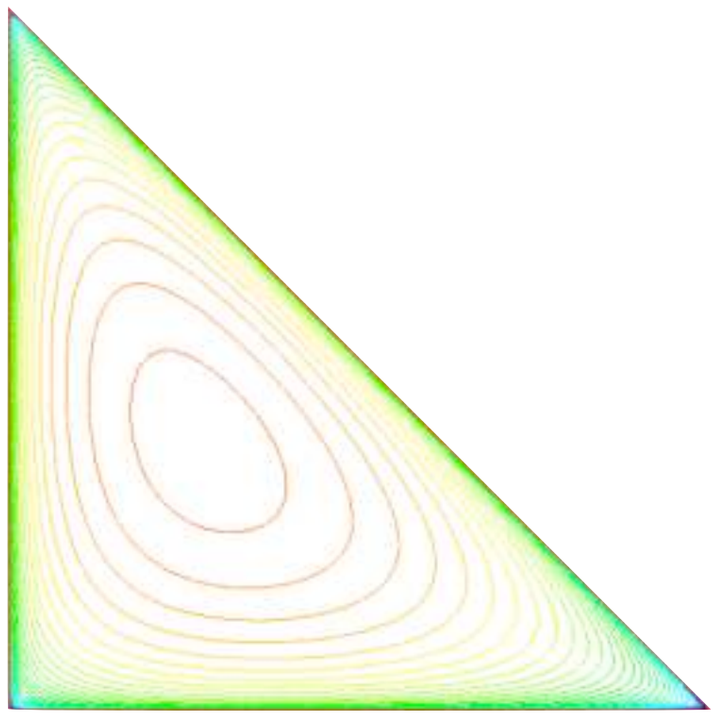
$$x_\mu = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; f(x) - \mu \sum_{i=1}^{m} \ln c_i(x)$$

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k \to +\infty}{\longrightarrow} 0$ find a a sequence (x_{μ_k}) of associated minimizers.

Illustration : minimize $f(x, y) = (x + 0.1)^2 + (y - 0.5)^2$ on a triangle



Isovalues of f



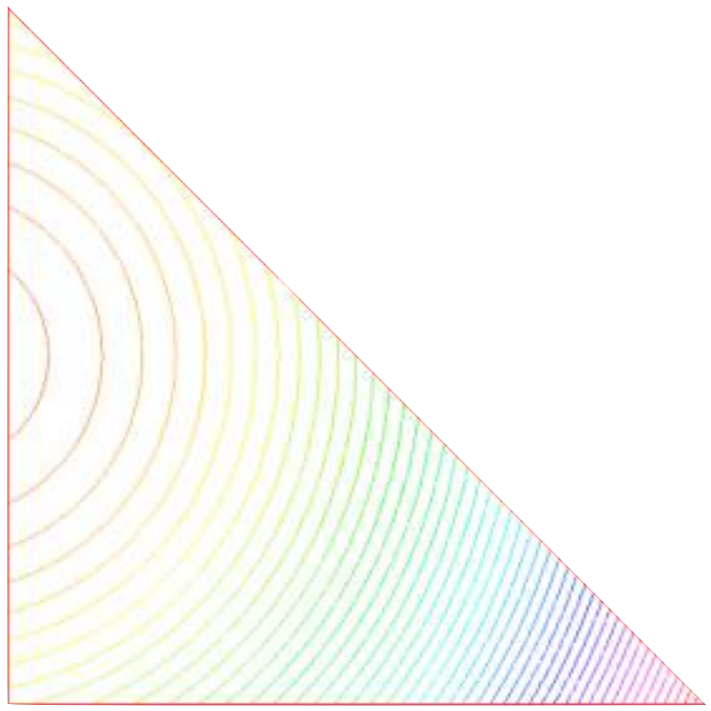
$\log(\mu) = 0$

Barrier Function :
For a given $\mu > 0$, solve the unconstrained problem of finding

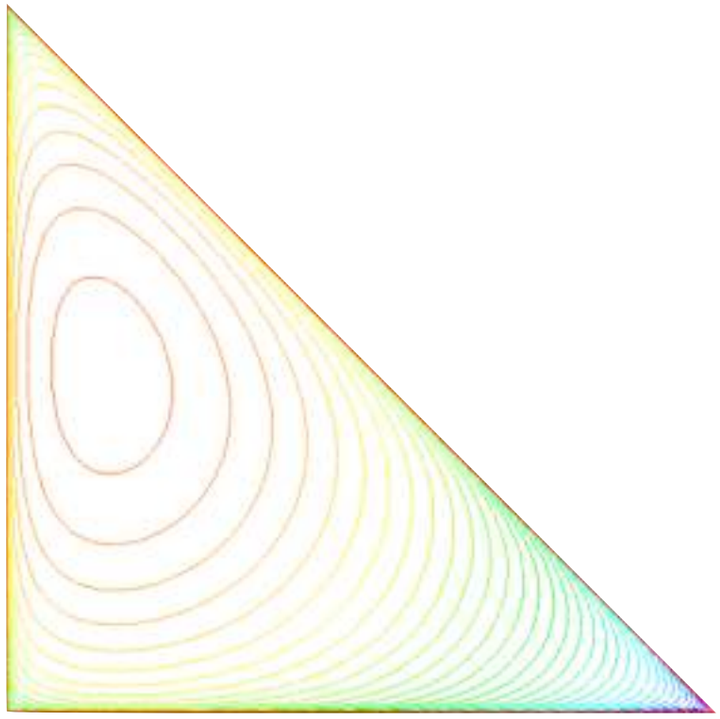
$$x_\mu = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; f(x) - \mu \sum_{i=1}^{m} \ln c_i(x)$$

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k \to +\infty}{\longrightarrow} 0$ find a a sequence (x_{μ_k}) of associated minimizers.

Illustration : minimize $f(x, y) = (x + 0.1)^2 + (y - 0.5)^2$ on a triangle



Isovalues of f



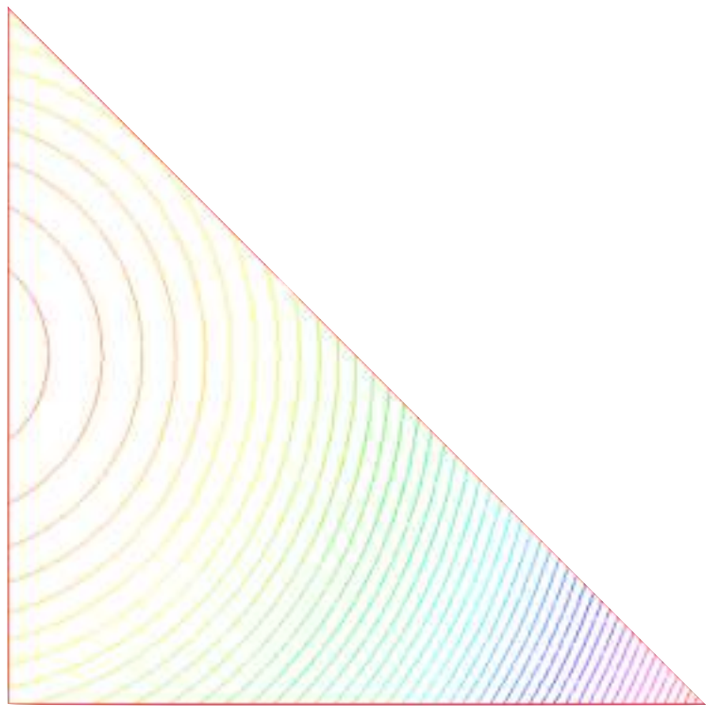
$\log(\mu) = -1$

Barrier Function :
For a given $\mu > 0$, solve the unconstrained problem of finding

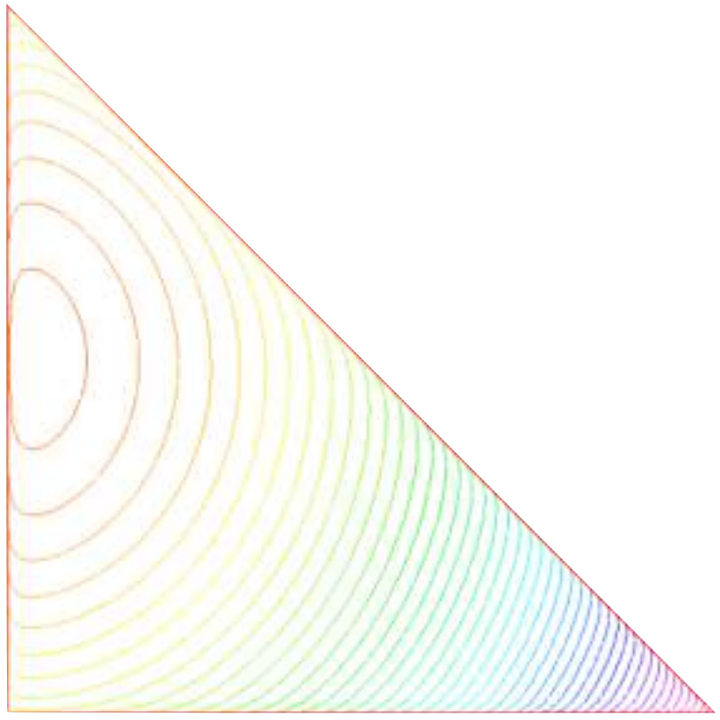
$$x_\mu = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; f(x) - \mu \sum_{i=1}^{m} \ln c_i(x)$$

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k \to +\infty}{\longrightarrow} 0$ find a a sequence (x_{μ_k}) of associated minimizers.

Illustration : minimize $f(x, y) = (x + 0.1)^2 + (y - 0.5)^2$ on a triangle



Isovalues of f



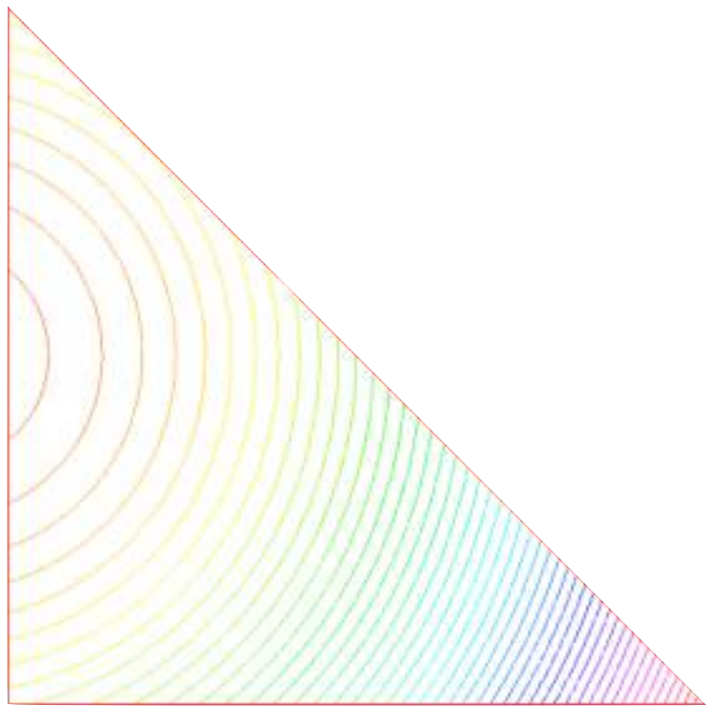
$\log(\mu) = -2$

Barrier Function :
For a given $\mu > 0$, solve the unconstrained problem of finding

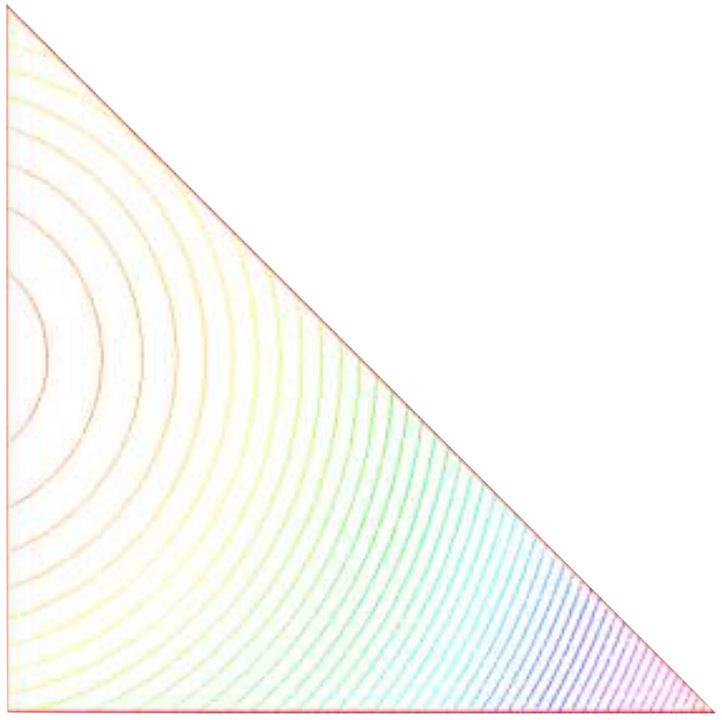
$$x_\mu = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; f(x) - \mu \sum_{i=1}^{m} \ln c_i(x)$$

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k \to +\infty}{\longrightarrow} 0$ find a a sequence (x_{μ_k}) of associated minimizers.

Illustration : minimize $f(x, y) = (x + 0.1)^2 + (y - 0.5)^2$ on a triangle



Isovalues of f



$\log(\mu) = -5$

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$
find a sequence (x_k) such that :

$$\forall k,\ x_k = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) - \mu_k \sum_{i=1}^{m} \ln c_i(x)$$

Links with the original problem :

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$ find a sequence (x_k) such that :

$$\forall k,\ x_k = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) - \mu_k \sum_{i=1}^{m} \ln c_i(x)$$

Links with the original problem :

Extremum point : $\nabla f(x_k) - \sum_{i=1}^{m} \frac{\mu_k}{c_i(x_k)} \nabla c_i(x_k) = 0$

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$ find a sequence (x_k) such that :

$$\forall k,\ x_k = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) - \mu_k \sum_{i=1}^{m} \ln c_i(x)$$

Links with the original problem :

«Dual variables»

Extremum point :

$$\nabla f(x_k) - J_c(x_k)^T \lambda_k = 0, \text{ where } \lambda_k = \mu_k / c(x_k)$$

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$ find a sequence (x_k) such that :

$$\forall k,\ \nabla f(x_k) - J_c(x_k)^T \lambda_k = 0,\ \lambda_k = \mu_k / c(x_k) \in \mathbb{R}^m$$

Links with the original problem :

Extremum point : $\nabla f(x_k) - J_c(x_k)^T \lambda_k = 0 \quad c_i(x_k)\lambda_{k,i} = \mu_k$

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$ find a sequence (x_k) such that :

$$\forall k,\ \nabla f(x_k) - J_c(x_k)^T \lambda_k = 0,\ \lambda_k = \mu_k / c(x_k) \in \mathbb{R}^m$$

Links with the original problem :

Extremum point : $\nabla f(x_k) - J_c(x_k)^T \lambda_k = 0 \quad c_i(x_k)\lambda_{k,i} = \mu_k$

KKT Conditions :

$$\begin{aligned} \nabla f(x^*) - J_c(x^*)^T \lambda^* &= 0 \\ c(x^*) &\geq 0 \\ \forall i,\ \lambda_i^* &\geq 0 \\ \forall i,\ c_i(x^*)\lambda_i^* &= 0 \end{aligned}$$

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$ find a sequence (x_k) such that :

$$\forall k,\ \nabla f(x_k) - J_c(x_k)^T \lambda_k = 0,\ \lambda_k = \mu_k / c(x_k) \in \mathbb{R}^m$$

Links with the original problem :

Extremum point : $\nabla f(x_k) - J_c(x_k)^T \lambda_k = 0 \quad c_i(x_k)\lambda_{k,i} = \mu_k$

KKT Conditions :

$$\begin{aligned} \nabla f(x^*) - J_c(x^*)^T \lambda^* &= 0 \\ c(x^*) &\geq 0 \\ \forall i,\ \lambda_i^* &\geq 0 \\ \forall i,\ c_i(x^*)\lambda_i^* &= 0 \end{aligned}$$

$\mu_k \longrightarrow 0$

IP Methods : given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$ find a sequence (x_k) such that :

$$\forall k,\ \nabla f(x_k) - J_c(x_k)^T \lambda_k = 0,\ \lambda_k = \mu_k / c(x_k) \in \mathbb{R}^m$$

Links with the original problem :

Extremum point : $\nabla f(x_k) - J_c(x_k)^T \lambda_k = 0 \quad c_i(x_k)\lambda_{k,i} = \mu_k$

$\exists (x^*, \lambda^*)$ KKT point and multipliers such that,

Under some favorable assumptions : $x_k \underset{k\to+\infty}{\longrightarrow} x^*$

With even more assumptions : $\lambda_k \underset{k\to+\infty}{\longrightarrow} \lambda^*$

IP main step: given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$ find a sequence (x_k) such that :

$$\forall k,\ \nabla f(x_k) - J_c(x_k)^T \lambda_k = 0,\ \lambda_k = \mu_k / c(x_k) \in \mathbb{R}^m$$

Links with the original problem :

Extremum point : $\nabla f(x_k) - J_c(x_k)^T \lambda_k = 0 \quad \lambda_k = \mu_k / c(x_k)$

$\exists (x^*, \lambda^*)$ KKT point and multipliers such that,

Under some favorable assumptions : $x_k \underset{k\to+\infty}{\longrightarrow} x^*$

With even more assumptions : $\lambda_k \underset{k\to+\infty}{\longrightarrow} \lambda^*$

IP main step: given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$ find a sequence (x_k) such that :

$$\forall k,\ \nabla f(x_k) - J_c(x_k)^T \lambda_k = 0,\ \lambda_k = \mu_k / c(x_k) \in \mathbb{R}^m$$

Links with the original problem :

Extremum point : $\nabla f(x_k) - J_c(x_k)^T \lambda_k = 0 \quad \lambda_k = \mu_k / c(x_k)$

$\exists (x^*, \lambda^*)$ KKT point and multipliers such that,

Under some favorable assumptions : $x_k \underset{k\to+\infty}{\longrightarrow} x^*$

With even more assumptions : $\lambda_k \underset{k\to+\infty}{\longrightarrow} \lambda^*$

IP main step: given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$ find a sequence (x_k) such that :

$$\forall k,\ \nabla f(x_k) - J_c(x_k)^T \lambda_k = 0,\ \lambda_k = \mu_k / c(x_k) \in \mathbb{R}^m$$

Links with the original problem :

$\exists (x^*, \lambda^*)$ KKT point and corresponding multipliers of the original problem, such that, under some assumptions :

$$x_k \underset{k\to+\infty}{\longrightarrow} x^* \qquad \lambda_k \underset{k\to+\infty}{\longrightarrow} \lambda^*$$

IP main step: given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$ find a sequence (x_k) such that :

$$\forall k,\ \nabla f(x_k) - J_c(x_k)^T \lambda_k = 0,\ \lambda_k = \mu_k / c(x_k) \in \mathbb{R}^m$$

Links with the original problem :

$\exists (x^*, \lambda^*)$ KKT point and corresponding multipliers of the original problem, such that, under some assumptions :

$$x_k \underset{k\to+\infty}{\longrightarrow} x^* \qquad \lambda_k \underset{k\to+\infty}{\longrightarrow} \lambda^*$$

IP main step: given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$ find a sequence (x_k) such that :

$$\forall k,\ \nabla f(x_k) - J_c(x_k)^T \lambda_k = 0,\ \lambda_k = \mu_k / c(x_k) \in \mathbb{R}^m$$

Links with the original problem : $\exists (x^*, \lambda^*)$ KKT point and corresponding multipliers of the original problem, such that, under some assumptions :

$$x_k \underset{k\to+\infty}{\longrightarrow} x^* \qquad \lambda_k \underset{k\to+\infty}{\longrightarrow} \lambda^*$$

IP main step: given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$ find a sequence (x_k) such that :

$$\forall k,\ \nabla f(x_k) - J_c(x_k)^T \lambda_k = 0,\ \lambda_k = \mu_k / c(x_k) \in \mathbb{R}^m$$

Links with the original problem : $\exists (x^*, \lambda^*)$ KKT point and corresponding multipliers of the original problem, such that, under some assumptions :

$$x_k \underset{k\to+\infty}{\longrightarrow} x^* \qquad \lambda_k \underset{k\to+\infty}{\longrightarrow} \lambda^*$$

IP main step: given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$ find a sequence (x_k) such that :

$$\forall k,\ \nabla f(x_k) - J_c(x_k)^T \lambda_k = 0,\ \lambda_k = \mu_k / c(x_k) \in \mathbb{R}^m$$

Links with the original problem : $\exists (x^*, \lambda^*)$ KKT point and corresponding multipliers of the original problem, such that, under some assumptions :

$$x_k \underset{k\to+\infty}{\longrightarrow} x^* \qquad \lambda_k \underset{k\to+\infty}{\longrightarrow} \lambda^*$$

Primal-dual IP method - Newton's method is applied to the non-linear system :

$$\begin{cases} \nabla f(x) - J_c(x)^T \lambda = 0 \\ c_i(x)\lambda_i = \mu\ ,\ \forall i \leq m \end{cases}$$

IP main step: given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$ find a sequence (x_k) such that :

$$\forall k,\ \nabla f(x_k) - J_c(x_k)^T \lambda_k = 0,\ \lambda_k = \mu_k / c(x_k) \in \mathbb{R}^m$$

Links with the original problem : $\exists (x^*, \lambda^*)$ KKT point and corresponding multipliers of the original problem, such that, under some assumptions :

$$x_k \underset{k\to+\infty}{\longrightarrow} x^* \qquad \lambda_k \underset{k\to+\infty}{\longrightarrow} \lambda^*$$

Primal-dual IP method - Newton's method is applied to the non-linear system :

$$\begin{cases} \nabla f(x) - J_c(x)^T \lambda = 0 \\ c_i(x)\lambda_i = \mu\ ,\ \forall i \leq m \end{cases}$$

Newton update $(\Delta_x, \Delta_\lambda)$:

$$\begin{pmatrix} H & -J_c^T \\ \Lambda J_c & C \end{pmatrix} \begin{pmatrix} \Delta_x \\ \Delta_\lambda \end{pmatrix} = \begin{pmatrix} \nabla f - J_c^T \lambda \\ C\lambda - \mu 1 \end{pmatrix}$$

Where $H = \nabla^2 f - \sum_{i=1}^{m} \lambda_i \nabla^2 c_i$ and $C = (\delta_{ij} c_i)_{1 \leq i,j \leq m}$ (as for Λ).

IP main step: given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$ find a sequence (x_k) such that :

$$\forall k,\ \nabla f(x_k) - J_c(x_k)^T \lambda_k = 0,\ \lambda_k = \mu_k / c(x_k) \in \mathbb{R}^m$$

Links with the original problem : $\exists (x^*, \lambda^*)$ KKT point and corresponding multipliers of the original problem, such that, under some assumptions :

$$x_k \underset{k\to+\infty}{\longrightarrow} x^* \qquad \lambda_k \underset{k\to+\infty}{\longrightarrow} \lambda^*$$

Primal-dual IP method - Newton's method is applied to the non-linear system :

$$\begin{cases} \nabla f(x) - J_c(x)^T \lambda = 0 \\ c_i(x)\lambda_i = \mu\ ,\ \forall i \leq m \end{cases}$$

Newton update $(\Delta_x, \Delta_\lambda)$:

$$\begin{pmatrix} H & -J_c^T \\ \Lambda J_c & C \end{pmatrix} \begin{pmatrix} \Delta_x \\ \Delta_\lambda \end{pmatrix} = \begin{pmatrix} \nabla f - J_c^T \lambda \\ C\lambda - \mu 1 \end{pmatrix}$$

Where $H = \nabla^2 f - \sum_{i=1}^{m} \lambda_i \nabla^2 c_i$ and $C = (\delta_{ij} c_i)_{1 \leq i,j \leq m}$ (as for Λ).

IP main step: given a decreasing sequence (μ_k) with $\mu_k \underset{k\to+\infty}{\longrightarrow} 0$ find a sequence (x_k) such that :

$$\forall k,\ \nabla f(x_k) - J_c(x_k)^T \lambda_k = 0,\ \lambda_k = \mu_k / c(x_k) \in \mathbb{R}^m$$

Links with the original problem : $\exists (x^*, \lambda^*)$ KKT point and corresponding multipliers of the original problem, such that, under some assumptions :

$$x_k \underset{k\to+\infty}{\longrightarrow} x^* \qquad \lambda_k \underset{k\to+\infty}{\longrightarrow} \lambda^*$$

Primal-dual IP method - Newton's method is applied to the non-linear system :

$$\begin{cases} \nabla f(x) - J_c(x)^T \lambda = 0 \\ c_i(x)\lambda_i = \mu\ ,\ \forall i \leq m \end{cases}$$

Newton update $(\Delta_x, \Delta_\lambda)$:

$$\begin{pmatrix} H & -J_c^T \\ \Lambda J_c & C \end{pmatrix} \begin{pmatrix} \Delta_x \\ \Delta_\lambda \end{pmatrix} = \begin{pmatrix} \nabla f - J_c^T \lambda \\ C\lambda - \mu 1 \end{pmatrix}$$

Where $H = \nabla^2 f - \sum_{i=1}^{m} \lambda_i \nabla^2 c_i$ and $C = (\delta_{ij} c_i)_{1 \leq i,j \leq m}$ (as for Λ).

Equality constraints :

Objective function :

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

Constraints :

$$c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

$$\text{Find : } x^* = \underset{x \in \mathbb{R}^n \mid \forall i,\ c_i(x) \geq 0}{\operatorname{argmin}} f(x)$$

Equality constraints :

Objective function :

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

Constraints :

$$c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

$$\text{Find : } x^* = \underset{x \in V}{\operatorname{argmin}}\ f(x)$$

Where, given $\mathcal{E}$ and $\mathcal{I}$, such that $\mathcal{E} \cap \mathcal{I} = \emptyset$ and $\mathcal{E} \cup \mathcal{I} = [\![1, m]\!]$

$$V = \{x \in \mathbb{R}^n \ \ | \forall i \in \mathcal{I},\ c_i(x) \geq 0 \ and \ \forall i \in \mathcal{E},\ c_i(x) = 0\}$$

Equality constraints :

Objective function :

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

Find : $x^* = \underset{x \in V}{\operatorname{argmin}}\ f(x)$

Constraints :

$$c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

$$V = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \forall i \in \mathcal{I},\ c_i(x) \geq 0 \\ \forall i \in \mathcal{E},\ c_i(x) = 0 \end{array} \right\}$$

Equality constraints :

Objective function :

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

Find : $x^* = \underset{x \in V}{\operatorname{argmin}} \ f(x)$

Constraints :

$$c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

$$V = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \forall i \in \mathcal{I},\ c_i(x) \geq 0 \\ \forall i \in \mathcal{E},\ c_i(x) = 0 \end{array} \right\}$$

Equality constraints :

Objective function :

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

Find : $x^* = \underset{x \in V}{\operatorname{argmin}}\ f(x)$

Constraints :

$$c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

$$V = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \forall i \in \mathcal{I},\ c_i(x) \geq 0 \\ \forall i \in \mathcal{E},\ c_i(x) = 0 \end{array} \right\}$$

Barrier problem :

Find the minimizer x_μ of :

$$B(x, \mu) = f(x) - \mu \sum_{i \in \mathcal{I}} \ln c_i(x)$$

on $\{x \in \mathbb{R}^n \mid \forall i \in \mathcal{E},\ c_i(x) = 0\}$.

Equality constraints :

Objective function :

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

Find : $x^* = \underset{x \in V}{\operatorname{argmin}}\ f(x)$

Constraints :

$$c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

$$V = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \forall i \in \mathcal{I},\ c_i(x) \geq 0 \\ \forall i \in \mathcal{E},\ c_i(x) = 0 \end{array} \right\}$$

Barrier problem :

Find : $$x_\mu = \underset{x \in V'}{\operatorname{argmin}}\ f(x) - \mu \sum_{i \in \mathcal{I}} \ln c_i(x)$$

where $V' = \{x \in \mathbb{R}^n \mid \forall i \in \mathcal{E},\ c_i(x) = 0\}$

Equality constraints :

Objective function :

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

Find : $x^* = \underset{x \in V}{\operatorname{argmin}}\ f(x)$

Constraints :

$$c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

$$V = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \forall i \in \mathcal{I},\ c_i(x) \geq 0 \\ \forall i \in \mathcal{E},\ c_i(x) = 0 \end{array} \right\}$$

Barrier problem :

Find : $x_\mu = \underset{x \in V'}{\operatorname{argmin}}\ f(x) - \mu \sum_{i \in \mathcal{I}} \ln c_i(x)$

where $V' = \{x \in \mathbb{R}^n \mid \forall i \in \mathcal{E},\ c_i(x) = 0\}$

Interior Points Methods

Equality constraints :

Objective function :

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

Constraints :

$$c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

Find : $x^* = \underset{x \in V}{\operatorname{argmin}}\ f(x)$

$$V = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \forall i \in \mathcal{I},\ c_i(x) \geq 0 \\ \forall i \in \mathcal{E},\ c_i(x) = 0 \end{array} \right\}$$

Barrier problem :

Find : $x_\mu = \underset{x \in V'}{\operatorname{argmin}}\ f(x) - \mu \sum_{i \in \mathcal{I}} \ln c_i(x)$

where $V' = \{x \in \mathbb{R}^n \mid \forall i \in \mathcal{E},\ c_i(x) = 0\}$

Interior Points Methods

Equality constraints :

Objective function :

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

Constraints :

$$c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

Find : $x^* = \underset{x \in V}{\operatorname{argmin}}\ f(x)$

$$V = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \forall i \in \mathcal{I},\ c_i(x) \geq 0 \\ \forall i \in \mathcal{E},\ c_i(x) = 0 \end{array} \right\}$$

Barrier problem :

Find : $x_\mu = \underset{x \in V'}{\operatorname{argmin}}\ f(x) - \mu \sum_{i \in \mathcal{I}} \ln c_i(x)$

where $V' = \{x \in \mathbb{R}^n \mid \forall i \in \mathcal{E},\ c_i(x) = 0\}$

Equality constraints :

Objective function :

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

Constraints :

$$c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

Find : $x^* = \underset{x \in V}{\operatorname{argmin}}\ f(x)$

$$V = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \forall i \in \mathcal{I},\ c_i(x) \geq 0 \\ \forall i \in \mathcal{E},\ c_i(x) = 0 \end{array} \right\}$$

Barrier problem :

Find : $x_\mu = \underset{x \in V'}{\operatorname{argmin}}\ f(x) - \mu \sum_{i \in \mathcal{I}} \ln c_i(x)$

where $V' = \{x \in \mathbb{R}^n \mid \forall i \in \mathcal{E},\ c_i(x) = 0\}$

Lagrange's conditions for the barrier problem :

$\exists \lambda_\mu \in \mathbb{R}^{\mathcal{E}}$ such that :

$$\nabla f(x_\mu) + \sum_{i \in \mathcal{E}} \lambda_\mu^{(i)} \nabla c_i(x_\mu) - \sum_{i \in \mathcal{I}} \frac{\mu}{c_i(x_\mu)} \nabla c_i(x_\mu) = 0$$

and $\forall i \in \mathcal{E},\ c_i(x_\mu) = 0$

Equality constraints :

Objective function :

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

Constraints :

$$c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

Find : $x^* = \underset{x \in V}{\operatorname{argmin}}\ f(x)$

$$V = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \forall i \in \mathcal{I},\ c_i(x) \geq 0 \\ \forall i \in \mathcal{E},\ c_i(x) = 0 \end{array} \right\}$$

Barrier problem :

Find : $x_\mu = \underset{x \in V'}{\operatorname{argmin}}\ f(x) - \mu \sum_{i \in \mathcal{I}} \ln c_i(x)$

where $V' = \{x \in \mathbb{R}^n \mid \forall i \in \mathcal{E},\ c_i(x) = 0\}$

Lagrange's conditions for the barrier problem :

$\exists \lambda_\mu \in \mathbb{R}^{\mathcal{E}}$ such that :

$$\nabla f(x_\mu) + \sum_{i \in \mathcal{E}} \lambda_\mu^{(i)} \nabla c_i(x_\mu) - \sum_{i \in \mathcal{I}} \frac{\mu}{c_i(x_\mu)} \nabla c_i(x_\mu) = 0$$

and $\forall i \in \mathcal{E},\ c_i(x_\mu) = 0$

$\forall i \in \mathcal{I}$, we define $\lambda_\mu^{(i)}$ such that : $c_i(x_\mu)\lambda_\mu^{(i)} = \mu$

Equality constraints :

Objective function : $f : \mathbb{R}^n \longrightarrow \mathbb{R}$

Constraints : $c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$

Find : $x^* = \underset{x \in V}{\operatorname{argmin}}\ f(x)$ $\quad V = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \forall i \in \mathcal{I},\ c_i(x) \geq 0 \\ \forall i \in \mathcal{E},\ c_i(x) = 0 \end{array} \right\}$

Barrier problem :

Find : $x_\mu = \underset{x \in V'}{\operatorname{argmin}}\ f(x) - \mu \sum_{i \in \mathcal{I}} \ln c_i(x)$

where $V' = \{x \in \mathbb{R}^n \mid \forall i \in \mathcal{E},\ c_i(x) = 0\}$

Lagrange's conditions for the barrier problem :

$\exists \lambda_\mu \in \mathbb{R}^{\mathcal{E}}$ such that :

$$\nabla f(x_\mu) + \boxed{\sum_{i \in \mathcal{E}} \lambda_\mu^{(i)} \nabla c_i(x_\mu) - \sum_{i \in \mathcal{I}} \frac{\mu}{c_i(x_\mu)} \nabla c_i(x_\mu)} = 0$$

$J_c(x_\mu)^{\mathrm{T}} \lambda_\mu$

and $\forall i \in \mathcal{E},\ c_i(x_\mu) = 0$

$\forall i \in \mathcal{I}$, we define $\lambda_\mu^{(i)}$ such that : $c_i(x_\mu)\lambda_\mu^{(i)} = \mu$

Equality constraints :

Objective function :

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

Constraints :

$$c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

Find : $x^* = \underset{x \in V}{\operatorname{argmin}}\ f(x)$

$$V = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \forall i \in \mathcal{I},\ c_i(x) \geq 0 \\ \forall i \in \mathcal{E},\ c_i(x) = 0 \end{array} \right\}$$

Barrier problem :

Find : $x_\mu = \underset{x \in V'}{\operatorname{argmin}}\ f(x) - \mu \sum_{i \in \mathcal{I}} \ln c_i(x)$

where $V' = \{x \in \mathbb{R}^n \mid \forall i \in \mathcal{E},\ c_i(x) = 0\}$

Lagrange's conditions for the barrier problem :

$\exists \lambda_\mu \in \mathbb{R}^{\mathcal{E}}$ such that :

$$\nabla f(x_\mu) + \sum_{i \in \mathcal{E}} \lambda_\mu^{(i)} \nabla c_i(x_\mu) - \sum_{i \in \mathcal{I}} \frac{\mu}{c_i(x_\mu)} \nabla c_i(x_\mu) = 0$$

and $\forall i \in \mathcal{E},\ c_i(x_\mu) = 0$

$\forall i \in \mathcal{I}$, we define $\lambda_\mu^{(i)}$ such that : $c_i(x_\mu)\lambda_\mu^{(i)} = \mu$

Equality constraints :

Objective function :

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

Constraints :

$$c = (c_1, \ldots, c_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

Find : $x^* = \underset{x \in V}{\operatorname{argmin}}\ f(x)$

$$V = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \forall i \in \mathcal{I},\ c_i(x) \geq 0 \\ \forall i \in \mathcal{E},\ c_i(x) = 0 \end{array} \right\}$$

Barrier problem :

Find : $x_\mu = \underset{x \in V'}{\operatorname{argmin}}\ f(x) - \mu \sum_{i \in \mathcal{I}} \ln c_i(x)$

where $V' = \{x \in \mathbb{R}^n \mid \forall i \in \mathcal{E},\ c_i(x) = 0\}$

Resolution of the barrier problems :

Apply Newton's method to find $(x_\mu, \lambda_\mu) \in \mathbb{R}^n \times \mathbb{R}^m$, verifying :

$$\begin{cases} \nabla f(x_\mu) + J_c(x_\mu)^{\mathrm{T}} \lambda_\mu = 0 \\ \forall i \in \mathcal{E},\ c_i(x_\mu) = 0 \\ \forall i \in \mathcal{I},\ c_i(x_\mu)\lambda_\mu^{(i)} = \mu \end{cases}$$

The IPOPT software :

- C++ open source implementation of a primal-dual interior point method for non-linear smooth optimization, from the COIN-OR project
  - ☞ https://projects.coin-or.org/Ipopt
  - ☞ the mailing list
- Uses sparse matrices and sparse symetric linear solvers (mumps, PARDISO, etc...) for the Newton's steps.
- Handles both inequality and/or equality constraints
- A fortran variant for variationnal inequalities...?

Bibliography :

•A. Wächter and L.T. Biegler, **On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming**, *Mathematical Programming* 106(1), pp. 25-57, 2006
•Some IPOPT related papers : https://projects.coin-or.org/Ipopt/wiki/IpoptPapers

IPOPT - Handling Constraints

IPOPT - Handling Constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \ f(x) \\ \\ \forall i \leq n, \ x_i^{\mathrm{lb}} \leq x_i \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \ c_i^{\mathrm{lb}} \leq c_i(x) \leq c_i^{\mathrm{ub}} \end{cases}$$

IPOPT - Handling Constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \ f(x) \\ \\ \forall i \leq n, \ x_i^{\mathrm{lb}} \leq x_i \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \ c_i^{\mathrm{lb}} \leq c_i(x) \leq c_i^{\mathrm{ub}} \end{cases}$$

IPOPT - Handling Constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \ f(x) \\ \\ \forall i \leq n, \ x_i^{\mathrm{lb}} \leq x_i \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \ c_i^{\mathrm{lb}} \leq c_i(x) \leq c_i^{\mathrm{ub}} \end{cases}$$

Simple bounds :

- $x^{\mathrm{lb}}, \ x^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $\forall i \in [\![1, n]\!], \ x_i^{\mathrm{lb}} \leq x_i^{\mathrm{ub}}$

IPOPT - Handling Constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \ f(x) \\ \\ \forall i \leq n, \ x_i^{\mathrm{lb}} \leq x_i \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \ c_i^{\mathrm{lb}} \leq c_i(x) \leq c_i^{\mathrm{ub}} \end{cases}$$

Simple bounds :

- $x^{\mathrm{lb}}, \ x^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $\forall i \in [\![1, n]\!], \ x_i^{\mathrm{lb}} \leq x_i^{\mathrm{ub}}$
- $x_i^{\mathrm{lb}} = -\infty$ or $x_i^{\mathrm{ub}} = +\infty$ means unboundedness in the associated direction

IPOPT - Handling Constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \ f(x) \\ \\ \forall i \leq n, \ x_i^{\mathrm{lb}} \leq x_i \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \ c_i^{\mathrm{lb}} \leq c_i(x) \leq c_i^{\mathrm{ub}} \end{cases}$$

Simple bounds :

- $x^{\mathrm{lb}}, \ x^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $\forall i \in [\![1, n]\!], \ x_i^{\mathrm{lb}} \leq x_i^{\mathrm{ub}}$
- $x_i^{\mathrm{lb}} = -\infty$ or $x_i^{\mathrm{ub}} = +\infty$ means unboundedness in the associated direction

IPOPT - Handling Constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \ f(x) \\ \\ \forall i \leq n, \ x_i^{\mathrm{lb}} \leq x_i \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \ c_i^{\mathrm{lb}} \leq c_i(x) \leq c_i^{\mathrm{ub}} \end{cases}$$

Simple bounds :

- $x^{\mathrm{lb}}, \ x^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $\forall i \in [\![1, n]\!], \ x_i^{\mathrm{lb}} \leq x_i^{\mathrm{ub}}$
- $x_i^{\mathrm{lb}} = -\infty$ or $x_i^{\mathrm{ub}} = +\infty$ means unboundedness in the associated direction

Nonlinear constraints :

- $\forall i \in [\![1, m]\!], \ c_i^{\mathrm{lb}}, \ c_i^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $c_i^{\mathrm{lb}} \leq c_i^{\mathrm{ub}}$

IPOPT - Handling Constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \ f(x) \\ \\ \forall i \leq n, \ x_i^{\mathrm{lb}} \leq x_i \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \ c_i^{\mathrm{lb}} \leq c_i(x) \leq c_i^{\mathrm{ub}} \end{cases}$$

Simple bounds :

- $x^{\mathrm{lb}}, \ x^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $\forall i \in [\![1, n]\!], \ x_i^{\mathrm{lb}} \leq x_i^{\mathrm{ub}}$
- $x_i^{\mathrm{lb}} = -\infty$ or $x_i^{\mathrm{ub}} = +\infty$ means unboundedness in the associated direction

Nonlinear constraints :

- $\forall i \in [\![1, m]\!], \ c_i^{\mathrm{lb}}, \ c_i^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $c_i^{\mathrm{lb}} \leq c_i^{\mathrm{ub}}$
- $c_i^{\mathrm{lb}} = -\infty$ or $c_i^{\mathrm{ub}} = +\infty$ for unboundedness

IPOPT - Handling Constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \ f(x) \\ \\ \forall i \leq n, \ x_i^{\mathrm{lb}} \leq x_i \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \ c_i^{\mathrm{lb}} \leq c_i(x) \leq c_i^{\mathrm{ub}} \end{cases}$$

Simple bounds :

- $x^{\mathrm{lb}}, \ x^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $\forall i \in [\![1, n]\!], \ x_i^{\mathrm{lb}} \leq x_i^{\mathrm{ub}}$
- $x_i^{\mathrm{lb}} = -\infty$ or $x_i^{\mathrm{ub}} = +\infty$ means unboundedness in the associated direction

Nonlinear constraints :

- $\forall i \in [\![1, m]\!], \ c_i^{\mathrm{lb}}, \ c_i^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $c_i^{\mathrm{lb}} \leq c_i^{\mathrm{ub}}$
- $c_i^{\mathrm{lb}} = -\infty$ or $c_i^{\mathrm{ub}} = +\infty$ for unboundedness
- $c_i^{\mathrm{lb}} = c_i^{\mathrm{ub}} \Longrightarrow$ equality constraint $c_i(x) = c_i^{\mathrm{lb}} \left(= c_i^{\mathrm{ub}}\right)$

IPOPT - Handling Constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \ f(x) \\ \\ \forall i \leq n, \ x_i^{\mathrm{lb}} \leq x_i \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \ c_i^{\mathrm{lb}} \leq c_i(x) \leq c_i^{\mathrm{ub}} \end{cases}$$

Simple bounds :

- $x^{\mathrm{lb}}, \ x^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $\forall i \in [\![1, n]\!], \ x_i^{\mathrm{lb}} \leq x_i^{\mathrm{ub}}$
- $x_i^{\mathrm{lb}} = -\infty$ or $x_i^{\mathrm{ub}} = +\infty$ means unboundedness in the associated direction

Nonlinear constraints :

- $\forall i \in [\![1, m]\!], \ c_i^{\mathrm{lb}}, \ c_i^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $c_i^{\mathrm{lb}} \leq c_i^{\mathrm{ub}}$
- $c_i^{\mathrm{lb}} = -\infty$ or $c_i^{\mathrm{ub}} = +\infty$ for unboundedness
- $c_i^{\mathrm{lb}} = c_i^{\mathrm{ub}} \Longrightarrow$ equality constraint $c_i(x) = c_i^{\mathrm{lb}} \left(= c_i^{\mathrm{ub}}\right)$
- $c_i^{\mathrm{lb}} < c_i^{\mathrm{ub}} \Longrightarrow$ two inequality constraints $\begin{cases} c_i(x) - c_i^{\mathrm{lb}} \geq 0 \\ c_i^{\mathrm{ub}} - c_i(x) \geq 0 \end{cases}$

IPOPT - Handling Constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \ f(x) \\ \\ \forall i \leq n, \ x_i^{\mathrm{lb}} \leq x_i \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \ c_i^{\mathrm{lb}} \leq c_i(x) \leq c_i^{\mathrm{ub}} \end{cases}$$

Simple bounds :

- $x^{\mathrm{lb}}, \ x^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $\forall i \in [\![1, n]\!], \ x_i^{\mathrm{lb}} \leq x_i^{\mathrm{ub}}$
- $x_i^{\mathrm{lb}} = -\infty$ or $x_i^{\mathrm{ub}} = +\infty$ means unboundedness in the associated direction

Nonlinear constraints :

- $\forall i \in [\![1, m]\!], \ c_i^{\mathrm{lb}}, \ c_i^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $c_i^{\mathrm{lb}} \leq c_i^{\mathrm{ub}}$
- $c_i^{\mathrm{lb}} = -\infty$ or $c_i^{\mathrm{ub}} = +\infty$ for unboundedness
- $c_i^{\mathrm{lb}} = c_i^{\mathrm{ub}} \Longrightarrow$ equality constraint $c_i(x) = c_i^{\mathrm{lb}} \left(= c_i^{\mathrm{ub}}\right)$
- $c_i^{\mathrm{lb}} < c_i^{\mathrm{ub}} \Longrightarrow$ two inequality constraints $\begin{cases} c_i(x) - c_i^{\mathrm{lb}} \geq 0 \\ c_i^{\mathrm{ub}} - c_i(x) \geq 0 \end{cases}$

IPOPT - Handling Constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; f(x) \\ \\ \forall i \leq n, \; x_i^{\text{lb}} \leq x_i \leq x_i^{\text{ub}} \\ \forall i \leq m, \; c_i^{\text{lb}} \leq c_i(x) \leq c_i^{\text{ub}} \end{cases}$$

Simple bounds :

- $x^{\text{lb}}, \; x^{\text{ub}} \in \overline{\mathbb{R}}^n$, with $\forall i \in [\![1, n]\!], \; x_i^{\text{lb}} \leq x_i^{\text{ub}}$
- $x_i^{\text{lb}} = -\infty$ or $x_i^{\text{ub}} = +\infty$ means unboundedness in the associated direction

Nonlinear constraints :

- $\forall i \in [\![1, m]\!], \; c_i^{\text{lb}}, \; c_i^{\text{ub}} \in \overline{\mathbb{R}}^n$, with $c_i^{\text{lb}} \leq c_i^{\text{ub}}$
- $c_i^{\text{lb}} = -\infty$ or $c_i^{\text{ub}} = +\infty$ for unboundedness
- $c_i^{\text{lb}} = c_i^{\text{ub}} \Longrightarrow$ equality constraint $c_i(x) = c_i^{\text{lb}} \; (= c_i^{\text{ub}})$
- $c_i^{\text{lb}} < c_i^{\text{ub}} \Longrightarrow$ two inequality constraints $\begin{cases} c_i(x) - c_i^{\text{lb}} \geq 0 \\ c_i^{\text{ub}} - c_i(x) \geq 0 \end{cases}$

IPOPT - Handling Constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; f(x) \\ \\ \forall i \leq n, \; x_i^{\text{lb}} \leq x_i \leq x_i^{\text{ub}} \\ \forall i \leq m, \; c_i^{\text{lb}} \leq c_i(x) \leq c_i^{\text{ub}} \end{cases}$$

Simple bounds :

- $x^{\text{lb}}, \; x^{\text{ub}} \in \overline{\mathbb{R}}^n$, with $\forall i \in [\![1, n]\!], \; x_i^{\text{lb}} \leq x_i^{\text{ub}}$
- $x_i^{\text{lb}} = -\infty$ or $x_i^{\text{ub}} = +\infty$ means unboundedness in the associated direction

Nonlinear constraints :

- $\forall i \in [\![1, m]\!], \; c_i^{\text{lb}}, \; c_i^{\text{ub}} \in \overline{\mathbb{R}}^n$, with $c_i^{\text{lb}} \leq c_i^{\text{ub}}$
- $c_i^{\text{lb}} = -\infty$ or $c_i^{\text{ub}} = +\infty$ for unboundedness
- $c_i^{\text{lb}} = c_i^{\text{ub}} \Longrightarrow$ equality constraint $c_i(x) = c_i^{\text{lb}} \; (= c_i^{\text{ub}})$
- $c_i^{\text{lb}} < c_i^{\text{ub}} \Longrightarrow$ two inequality constraints $\begin{cases} c_i(x) - c_i^{\text{lb}} \geq 0 \\ c_i^{\text{ub}} - c_i(x) \geq 0 \end{cases}$

IPOPT - Handling Constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; f(x) \\ \\ \forall i \leq n, \; x_i^{\mathrm{lb}} \leq x_i \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \; c_i^{\mathrm{lb}} \leq c_i(x) \leq c_i^{\mathrm{ub}} \end{cases}$$

Simple bounds :

- $x^{\mathrm{lb}}, \; x^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $\forall i \in [\![1, n]\!], \; x_i^{\mathrm{lb}} \leq x_i^{\mathrm{ub}}$
- $x_i^{\mathrm{lb}} = -\infty$ or $x_i^{\mathrm{ub}} = +\infty$ means unboundedness in the associated direction

Nonlinear constraints :

- $\forall i \in [\![1, m]\!], \; c_i^{\mathrm{lb}}, \; c_i^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $c_i^{\mathrm{lb}} \leq c_i^{\mathrm{ub}}$
- $c_i^{\mathrm{lb}} = -\infty$ or $c_i^{\mathrm{ub}} = +\infty$ for unboundedness
- $c_i^{\mathrm{lb}} = c_i^{\mathrm{ub}} \Longrightarrow$ equality constraint $c_i(x) = c_i^{\mathrm{lb}} \; (= c_i^{\mathrm{ub}})$
- $c_i^{\mathrm{lb}} < c_i^{\mathrm{ub}} \Longrightarrow$ two inequality constraints $\begin{cases} c_i(x) - c_i^{\mathrm{lb}} \geq 0 \\ c_i^{\mathrm{ub}} - c_i(x) \geq 0 \end{cases}$

Internally, all inequality constraints are turned into an equality one and a simple bound by introducing *slack variables* .

$$c_i(x) - c_i^{\mathrm{lb}} \geq 0 \quad \Rrightarrow \quad \begin{cases} c_i(x) - c_i^{\mathrm{lb}} + s_i = 0 \\ s_i \geq 0 \end{cases}$$

IPOPT - Handling Constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; f(x) \\ \\ \forall i \leq n, \; x_i^{\mathrm{lb}} \leq x_i \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \; c_i^{\mathrm{lb}} \leq c_i(x) \leq c_i^{\mathrm{ub}} \end{cases}$$

Simple bounds :

- $x^{\mathrm{lb}}, \; x^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $\forall i \in [\![1, n]\!], \; x_i^{\mathrm{lb}} \leq x_i^{\mathrm{ub}}$
- $x_i^{\mathrm{lb}} = -\infty$ or $x_i^{\mathrm{ub}} = +\infty$ means unboundedness in the associated direction

Nonlinear constraints :

- $\forall i \in [\![1, m]\!], \; c_i^{\mathrm{lb}}, \; c_i^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $c_i^{\mathrm{lb}} \leq c_i^{\mathrm{ub}}$
- $c_i^{\mathrm{lb}} = -\infty$ or $c_i^{\mathrm{ub}} = +\infty$ for unboundedness
- $c_i^{\mathrm{lb}} = c_i^{\mathrm{ub}} \Longrightarrow$ equality constraint $c_i(x) = c_i^{\mathrm{lb}} \left(= c_i^{\mathrm{ub}}\right)$
- $c_i^{\mathrm{lb}} < c_i^{\mathrm{ub}} \Longrightarrow$ two inequality constraints $\begin{cases} c_i(x) - c_i^{\mathrm{lb}} \geq 0 \\ c_i^{\mathrm{ub}} - c_i(x) \geq 0 \end{cases}$

So, the IPOPT barrier function, $\mathbb{R}^n \times \mathbb{R}^{\mathcal{I}} \times \mathbb{R}^{\mathcal{I}} \times \mathbb{R} \longrightarrow \mathbb{R}$, is :

$$B(x, s^{\mathrm{lb}}, s^{\mathrm{ub}}; \mu) = f(x) - \mu \left[\sum_{i=1}^{n} \ln\left(x_i - x_i^{\mathrm{lb}}\right) + \sum_{i=1}^{n} \ln\left(x_i^{\mathrm{ub}} - x_i\right) + \sum_{i \in \mathcal{I}} \ln s_i^{\mathrm{lb}} + \sum_{i \in \mathcal{I}} \ln s_i^{\mathrm{ub}} \right]$$

IPOPT - Handling Constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; f(x) \\ \\ \forall i \leq n, \;\; x_i^{\mathrm{lb}} \leq x_i \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \;\; c_i^{\mathrm{lb}} \leq c_i(x) \leq c_i^{\mathrm{ub}} \end{cases}$$

Simple bounds :

- $x^{\mathrm{lb}}, \; x^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $\forall i \in [\![1, n]\!], \; x_i^{\mathrm{lb}} \leq x_i^{\mathrm{ub}}$
- $x_i^{\mathrm{lb}} = -\infty$ or $x_i^{\mathrm{ub}} = +\infty$ means unboundedness in the associated direction

Nonlinear constraints :

- $\forall i \in [\![1, m]\!], \; c_i^{\mathrm{lb}}, \; c_i^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $c_i^{\mathrm{lb}} \leq c_i^{\mathrm{ub}}$
- $c_i^{\mathrm{lb}} = -\infty$ or $c_i^{\mathrm{ub}} = +\infty$ for unboundedness
- $c_i^{\mathrm{lb}} = c_i^{\mathrm{ub}} \Longrightarrow$ equality constraint $c_i(x) = c_i^{\mathrm{lb}} \; (= c_i^{\mathrm{ub}})$
- $c_i^{\mathrm{lb}} < c_i^{\mathrm{ub}} \Longrightarrow$ two inequality constraints $\begin{cases} c_i(x) - c_i^{\mathrm{lb}} \geq 0 \\ c_i^{\mathrm{ub}} - c_i(x) \geq 0 \end{cases}$

So, the IPOPT barrier function, $\mathbb{R}^n \times \mathbb{R}^{\mathcal{I}} \times \mathbb{R}^{\mathcal{I}} \times \mathbb{R} \longrightarrow \mathbb{R}$, is :

$$B(x, s^{\mathrm{lb}}, s^{\mathrm{ub}}; \mu) = f(x) - \mu \left[\sum_{i=1}^{n} \ln \left(x_i - x_i^{\mathrm{lb}} \right) + \sum_{i=1}^{n} \ln \left(x_i^{\mathrm{ub}} - x_i \right) + \sum_{i \in \mathcal{I}} \ln s_i^{\mathrm{lb}} + \sum_{i \in \mathcal{I}} \ln s_i^{\mathrm{ub}} \right]$$

And the barrier problem consists in minimizing it under the equality constraints :

$$\begin{cases} c_i(x) - c_i^{\mathrm{b}} = 0 & \forall i \in \mathcal{E} \\ c_i(x) - c_i^{\mathrm{lb}} + s_i^{\mathrm{lb}} = 0 & \forall i \in \mathcal{I} \\ c_i^{\mathrm{ub}} - c_i(x) + s_i^{\mathrm{ub}} = 0 & \end{cases}$$

IPOPT - Handling Constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; f(x) \\ \\ \forall i \leq n, \;\; x_i^{\mathrm{lb}} \leq x_i \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \;\; c_i^{\mathrm{lb}} \leq c_i(x) \leq c_i^{\mathrm{ub}} \end{cases}$$

Simple bounds :

- $x^{\mathrm{lb}}, \; x^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $\forall i \in [\![1, n]\!], \; x_i^{\mathrm{lb}} \leq x_i^{\mathrm{ub}}$
- $x_i^{\mathrm{lb}} = -\infty$ or $x_i^{\mathrm{ub}} = +\infty$ means unboundedness in the associated direction

Nonlinear constraints :

- $\forall i \in [\![1, m]\!], \; c_i^{\mathrm{lb}}, \; c_i^{\mathrm{ub}} \in \overline{\mathbb{R}}^n$, with $c_i^{\mathrm{lb}} \leq c_i^{\mathrm{ub}}$
- $c_i^{\mathrm{lb}} = -\infty$ or $c_i^{\mathrm{ub}} = +\infty$ for unboundedness
- $c_i^{\mathrm{lb}} = c_i^{\mathrm{ub}} \Longrightarrow$ equality constraint $c_i(x) = c_i^{\mathrm{lb}} \; (= c_i^{\mathrm{ub}})$
- $c_i^{\mathrm{lb}} < c_i^{\mathrm{ub}} \Longrightarrow$ two inequality constraints $\begin{cases} c_i(x) - c_i^{\mathrm{lb}} \geq 0 \\ c_i^{\mathrm{ub}} - c_i(x) \geq 0 \end{cases}$

So, the IPOPT barrier function, $\mathbb{R}^n \times \mathbb{R}^{\mathcal{I}} \times \mathbb{R}^{\mathcal{I}} \times \mathbb{R} \longrightarrow \mathbb{R}$, is :

$$B(x, s^{\mathrm{lb}}, s^{\mathrm{ub}}; \mu) = f(x) - \mu \left[\sum_{i=1}^{n} \ln \left(x_i - x_i^{\mathrm{lb}} \right) + \sum_{i=1}^{n} \ln \left(x_i^{\mathrm{ub}} - x_i \right) + \sum_{i \in \mathcal{I}} \ln s_i^{\mathrm{lb}} + \sum_{i \in \mathcal{I}} \ln s_i^{\mathrm{ub}} \right]$$

And the barrier problem consists in minimizing it under the equality constraints :

$$\begin{cases} c_i(x) - c_i^{\mathrm{b}} = 0 & \forall i \in \mathcal{E} \\ c_i(x) - c_i^{\mathrm{lb}} + s_i^{\mathrm{lb}} = 0 & \forall i \in \mathcal{I} \\ c_i^{\mathrm{ub}} - c_i(x) + s_i^{\mathrm{ub}} = 0 & \end{cases}$$

The FreeFem++ Interface :

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \ f(x) \\ \\ \forall i \leq n, \ x_i^{\mathrm{lb}} \leq x_i \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \ c_i^{\mathrm{lb}} \leq c_i(x) \leq c_i^{\mathrm{ub}} \end{cases}$$

The FreeFem++ Interface :

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \ f(x) \\ \\ \forall i \leq n, \ x_i^{\mathrm{lb}} \leq x_i^* \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \ c_i^{\mathrm{lb}} \leq c_i(x^*) \leq c_i^{\mathrm{ub}} \end{cases}$$

The FreeFem++ Interface :

IPOPT needs $f,\ \nabla f,\ c,\ J_c, x^{\mathrm{lb}},\ x^{\mathrm{ub}},$ $c^{\mathrm{lb}},\ c^{\mathrm{ub}}$ and...

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \leq n,\ x_i^{\mathrm{lb}} \leq x_i^* \leq x_i^{\mathrm{ub}} \\ \forall i \leq m,\ c_i^{\mathrm{lb}} \leq c_i(x^*) \leq c_i^{\mathrm{ub}} \end{cases}$$

$$\begin{array}{ccc} H : \mathbb{R}^n \times \mathbb{R}^m & \longrightarrow & \mathcal{M}_n(\mathbb{R}) \\ (x\ ,\ \lambda) & \longmapsto & \nabla^2 f(x) - \sum_{i=1}^{m} \lambda_i \nabla^2 c_i(x) \end{array}$$

The FreeFem++ Interface :

IPOPT needs $f,\ \nabla f,\ c,\ J_c, x^{\mathrm{lb}},\ x^{\mathrm{ub}},$ $c^{\mathrm{lb}},\ c^{\mathrm{ub}}$ and...

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \leq n,\ x_i^{\mathrm{lb}} \leq x_i^* \leq x_i^{\mathrm{ub}} \\ \forall i \leq m,\ c_i^{\mathrm{lb}} \leq c_i(x^*) \leq c_i^{\mathrm{ub}} \end{cases}$$

$$H : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^m \longrightarrow \mathcal{M}_n(\mathbb{R})$$

$$(x\ ,\ \sigma\ ,\ \lambda) \longmapsto \sigma\nabla^2 f(x) + \sum_{i=1}^{m} \lambda_i \nabla^2 c_i(x)$$

The FreeFem++ Interface :

IPOPT needs

f

$\nabla f,\ c$

J_c

H

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \le n,\ x_i^{\mathrm{lb}} \le x_i^* \le x_i^{\mathrm{ub}} \\ \forall i \le m,\ c_i^{\mathrm{lb}} \le c_i(x^*) \le c_i^{\mathrm{ub}} \end{cases}$$

$$x^{\mathrm{lb}},\ x^{\mathrm{ub}},\ c^{\mathrm{lb}},\ c^{\mathrm{ub}}$$

The FreeFem++ Interface :

IPOPT needs

f : **func real** f(**real**[**int**] &X) {...}

$$\nabla f,\ c$$

$$J_c$$

$$H$$

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \le n,\ x_i^{\mathrm{lb}} \le x_i^* \le x_i^{\mathrm{ub}} \\ \forall i \le m,\ c_i^{\mathrm{lb}} \le c_i(x^*) \le c_i^{\mathrm{ub}} \end{cases}$$

$$x^{\mathrm{lb}},\ x^{\mathrm{ub}},\ c^{\mathrm{lb}},\ c^{\mathrm{ub}}$$

The FreeFem++ Interface :

IPOPT needs

f : **func real** f(**real**[**int**] &X) {...}

$\nabla f,\ c$: **func real**[**int**] df(**real**[**int**] &X) {...}
func real[**int**] c(**real**[**int**] &X) {...} (optional)

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \le n,\ x_i^{\mathrm{lb}} \le x_i^* \le x_i^{\mathrm{ub}} \\ \forall i \le m,\ c_i^{\mathrm{lb}} \le c_i(x^*) \le c_i^{\mathrm{ub}} \end{cases}$$

$x^{\mathrm{lb}},\ x^{\mathrm{ub}},\ c^{\mathrm{lb}},\ c^{\mathrm{ub}}$

J_c

H

The FreeFem++ Interface :

IPOPT needs

f : **func real** f(**real**[**int**] &X) {...}

$\nabla f,\ c$: **func real**[**int**] df(**real**[**int**] &X) {...}
func real[**int**] c(**real**[**int**] &X) {...} (optional)

J_c : **func matrix** Jc(**real**[**int**] &X) {...}
Needed only if there are constraint functions

H

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \le n,\ x_i^{\mathrm{lb}} \le x_i^* \le x_i^{\mathrm{ub}} \\ \forall i \le m,\ c_i^{\mathrm{lb}} \le c_i(x^*) \le c_i^{\mathrm{ub}} \end{cases}$$

$$x^{\mathrm{lb}},\ x^{\mathrm{ub}},\ c^{\mathrm{lb}},\ c^{\mathrm{ub}}$$

The FreeFem++ Interface :

IPOPT needs

f : **func real** f(**real**[**int**] &X) {...}

$\nabla f,\ c$: **func real**[**int**] df(**real**[**int**] &X) {...}
func real[**int**] c(**real**[**int**] &X) {...} (optional)

J_c : **func matrix** Jc(**real**[**int**] &X) {...}
Needed only if there are constraint functions

H : **func matrix** H(**real**[**int**] &X, **real** s, **real**[**int**] &L)
In case of affine constraints, the prototype may be :
func matrix H(**real**[**int**] &X) {...}
If H is omitted Newton is replaced by a BFGS algorithm.

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \le n,\ x_i^{\mathrm{lb}} \le x_i^* \le x_i^{\mathrm{ub}} \\ \forall i \le m,\ c_i^{\mathrm{lb}} \le c_i(x^*) \le c_i^{\mathrm{ub}} \end{cases}$$

$$x^{\mathrm{lb}},\ x^{\mathrm{ub}},\ c^{\mathrm{lb}},\ c^{\mathrm{ub}}$$

The FreeFem++ Interface :

IPOPT needs

f : **func real** f(**real**[**int**] &X) {...}

∇f, c : **func real**[**int**] df(**real**[**int**] &X) {...}
func real[**int**] c(**real**[**int**] &X) {...} (optional)

J_c : **func matrix** Jc(**real**[**int**] &X) {...}
Needed only if there are constraint functions

H : **func matrix** H(**real**[**int**] &X, **real** s, **real**[**int**] &L)
In case of affine constraints, the prototype may be :
func matrix H(**real**[**int**] &X) {...}
If H is omitted Newton is replaced by a BFGS algorithm.

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \ f(x) \\ \\ \forall i \leq n, \ x_i^{\mathrm{lb}} \leq x_i^* \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \ c_i^{\mathrm{lb}} \leq c_i(x^*) \leq c_i^{\mathrm{ub}} \end{cases}$$

x^{lb}, x^{ub}, c^{lb}, c^{ub} :
real[**int**] arrays
- size n for x bounds
- size m for c bounds
- x bounds are optionnal
- Set components to ±1e19 for unboundedness in particular directions.

The FreeFem++ Interface :

IPOPT needs

f : **func real** f(**real**[**int**] &X) {...}

$\nabla f,\ c$: **func real**[**int**] df(**real**[**int**] &X) {...}
func real[**int**] c(**real**[**int**] &X) {...} (optional)

J_c : **func matrix** Jc(**real**[**int**] &X) {...}
Needed only if there are constraint functions

H : **func matrix** H(**real**[**int**] &X, **real** s, **real**[**int**] &L)
In case of affine constraints, the prototype may be :
func matrix H(**real**[**int**] &X) {...}
If H is omitted Newton is replaced by a BFGS algorithm.

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \le n,\ x_i^{\mathrm{lb}} \le x_i^* \le x_i^{\mathrm{ub}} \\ \forall i \le m,\ c_i^{\mathrm{lb}} \le c_i(x^*) \le c_i^{\mathrm{ub}} \end{cases}$$

x^{lb}, x^{ub}, c^{lb}, c^{ub} :
real[**int**] arrays
- size n for x bounds
- size m for c bounds
- x bounds are optionnal
- Set components to ±1e19 for unboundedness in particular directions.

Call IPOPT : full featured

int status = **IPOPT**(f, dF, c, Jc, H, xstart, **lb**=xlb,**ub**=xub,**clb**=clb,**cub**=cub, ...);

Remark: requires **load** "ff-Ipopt"; earlier in the script

The FreeFem++ Interface :

IPOPT needs

f : **func real** f(**real**[**int**] &X) {...}

$\nabla f,\ c$: **func real**[**int**] df(**real**[**int**] &X) {...}
func real[**int**] c(**real**[**int**] &X) {...} (optional)

J_c : **func matrix** Jc(**real**[**int**] &X) {...}
Needed only if there are constraint functions

H : **func matrix** H(**real**[**int**] &X, **real** s, **real**[**int**] &L)
In case of affine constraints, the prototype may be :
func matrix H(**real**[**int**] &X) {...}
If H is omitted Newton is replaced by a BFGS algorithm.

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \le n,\ x_i^{\mathrm{lb}} \le x_i^* \le x_i^{\mathrm{ub}} \\ \forall i \le m,\ c_i^{\mathrm{lb}} \le c_i(x^*) \le c_i^{\mathrm{ub}} \end{cases}$$

$x^{\mathrm{lb}},\ x^{\mathrm{ub}},\ c^{\mathrm{lb}},\ c^{\mathrm{ub}}$:
real[**int**] arrays
- size n for x bounds
- size m for c bounds
- x bounds are optionnal
- Set components to ±1e19 for unboundedness in particular directions.

Call IPOPT : unconstrained

int status = **IPOPT**(f, dF, H, xstart, **lb**=xlb,**ub**=xub, ...);

Remark: requires **load** "ff-Ipopt"; earlier in the script

The FreeFem++ Interface :

IPOPT needs

f : **func real** f(**real**[**int**] &X) {...}

∇f, c : **func real**[**int**] df(**real**[**int**] &X) {...}
func real[**int**] c(**real**[**int**] &X) {...} (optional)

J_c : **func matrix** Jc(**real**[**int**] &X) {...}
Needed only if there are constraint functions

H : **func matrix** H(**real**[**int**] &X, **real** s, **real**[**int**] &L)
In case of affine constraints, the prototype may be :
func matrix H(**real**[**int**] &X) {...}
If H is omitted Newton is replaced by a BFGS algorithm.

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \ f(x) \\ \\ \forall i \leq n, \ x_i^{\mathrm{lb}} \leq x_i^* \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \ c_i^{\mathrm{lb}} \leq c_i(x^*) \leq c_i^{\mathrm{ub}} \end{cases}$$

x^{lb}, x^{ub}, c^{lb}, c^{ub} :
real[**int**] arrays
- size n for x bounds
- size m for c bounds
- x bounds are optionnal
- Set components to ±1e19 for unboundedness in particular directions.

Call IPOPT : with BFGS

int status = **IPOPT**(f, dF, c, Jc, xstart, **lb**=xlb,**ub**=xub,**clb**=clb,**cub**=cub ... ...);

Remark: requires **load** "ff-Ipopt"; earlier in the script

The FreeFem++ Interface :

IPOPT needs

f : **func real** f(**real**[**int**] &X) {...}

$\nabla f,\ c$: **func real**[**int**] df(**real**[**int**] &X) {...}
func real[**int**] c(**real**[**int**] &X) {...} (optional)

J_c : **func matrix** Jc(**real**[**int**] &X) {...}
Needed only if there are constraint functions

H : **func matrix** H(**real**[**int**] &X, **real** s, **real**[**int**] &L)
In case of affine constraints, the prototype may be :
func matrix H(**real**[**int**] &X) {...}
If H is omitted Newton is replaced by a BFGS algorithm.

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \le n,\ x_i^{\mathrm{lb}} \le x_i^* \le x_i^{\mathrm{ub}} \\ \forall i \le m,\ c_i^{\mathrm{lb}} \le c_i(x^*) \le c_i^{\mathrm{ub}} \end{cases}$$

$x^{\mathrm{lb}},\ x^{\mathrm{ub}},\ c^{\mathrm{lb}},\ c^{\mathrm{ub}}$:
real[**int**] arrays
- size n for x bounds
- size m for c bounds
- x bounds are optionnal
- Set components to ±1e19 for unboundedness in particular directions.

Call IPOPT : unconstrained with BFGS

int status = **IPOPT**(f, dF, xstart, **lb**=xlb,**ub**=xub, ...);

Remark: requires **load** "ff-Ipopt"; earlier in the script

The FreeFem++ Interface :

case of quadratic objective and affine constraints

$$\begin{cases} x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \ f(x) \\ \\ \forall i \leq n, \ x_i^{\mathrm{lb}} \leq x_i^* \leq x_i^{\mathrm{ub}} \\ \forall i \leq m, \ c_i^{\mathrm{lb}} \leq c_i(x^*) \leq c_i^{\mathrm{ub}} \end{cases}$$

The FreeFem++ Interface :

case of quadratic objective and affine constraints

$$\forall x \in \mathbb{R}^n, \quad \begin{array}{rcl} f(x) & = & \frac{1}{2}\langle Ax, x\rangle + \langle b, x\rangle \\ c(x) & = & Cx + d \end{array}$$

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \le n,\ x_i^{\mathrm{lb}} \le x_i^* \le x_i^{\mathrm{ub}} \\ \forall i \le m,\ c_i^{\mathrm{lb}} \le c_i(x^*) \le c_i^{\mathrm{ub}} \end{cases}$$

$$(A, b) \in \mathcal{M}_{n,n}(\mathbb{R}) \times \mathbb{R}^n,\ (C, d) \in \mathcal{M}_{n,m}(\mathbb{R}) \times \mathbb{R}^m$$

The FreeFem++ Interface :

case of quadratic objective and affine constraints

$$\forall x \in \mathbb{R}^n, \quad \begin{aligned} f(x) &= \frac{1}{2}\langle Ax, x\rangle + \langle b, x\rangle \\ c(x) &= Cx + d \end{aligned}$$

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \le n,\ x_i^{\mathrm{lb}} \le x_i^* \le x_i^{\mathrm{ub}} \\ \forall i \le m,\ c_i^{\mathrm{lb}} \le c_i(x^*) \le c_i^{\mathrm{ub}} \end{cases}$$

$$(A, b) \in \mathcal{M}_{n,n}(\mathbb{R}) \times \mathbb{R}^n,\ (C, d) \in \mathcal{M}_{n,m}(\mathbb{R}) \times \mathbb{R}^m$$

➡ Possibility to directly pass the matrices and vectors

The FreeFem++ Interface :

case of quadratic objective and affine constraints

$$\forall x \in \mathbb{R}^n, \quad \begin{aligned} f(x) &= \frac{1}{2}\langle Ax, x\rangle + \langle b, x\rangle \\ c(x) &= Cx + d \end{aligned}$$

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \le n,\ x_i^{\mathrm{lb}} \le x_i^* \le x_i^{\mathrm{ub}} \\ \forall i \le m,\ c_i^{\mathrm{lb}} \le c_i(x^*) \le c_i^{\mathrm{ub}} \end{cases}$$

$$(A, b) \in \mathcal{M}_{n,n}(\mathbb{R}) \times \mathbb{R}^n,\ (C, d) \in \mathcal{M}_{n,m}(\mathbb{R}) \times \mathbb{R}^m$$

➡ Possibility to directly pass the matrices and vectors

Quadratic objective and affine constraints :

```
... //begining of the script
matrix A = ... ;
matrix C = ... ;
real[int] b = ... , d = ... ;
IPOPT([A,b] , [C,d] , xstart ,  /*named parameters*/  );
```

The FreeFem++ Interface :

case of quadratic objective and affine constraints

$$\forall x \in \mathbb{R}^n, \quad \begin{aligned} f(x) &= \frac{1}{2}\langle Ax, x\rangle + \langle b, x\rangle \\ c(x) &= Cx + d \end{aligned}$$

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \le n,\ x_i^{\mathrm{lb}} \le x_i^* \le x_i^{\mathrm{ub}} \\ \forall i \le m,\ c_i^{\mathrm{lb}} \le c_i(x^*) \le c_i^{\mathrm{ub}} \end{cases}$$

$$(A, b) \in \mathcal{M}_{n,n}(\mathbb{R}) \times \mathbb{R}^n,\ (C, d) \in \mathcal{M}_{n,m}(\mathbb{R}) \times \mathbb{R}^m$$

➡ Possibility to directly pass the matrices and vectors

Homogeneous quadratic objective and affine constraints :

```
... //begining of the script
matrix A = ... ;
matrix C = ... ;
real[int] d = ... ;
IPOPT(A , [C,d] , xstart ,  /*named parameters*/  );
```

The FreeFem++ Interface :

case of quadratic objective and affine constraints

$$\forall x \in \mathbb{R}^n, \quad \begin{aligned} f(x) &= \frac{1}{2}\langle Ax, x\rangle + \langle b, x\rangle \\ c(x) &= Cx + d \end{aligned}$$

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \le n,\ x_i^{\mathrm{lb}} \le x_i^* \le x_i^{\mathrm{ub}} \\ \forall i \le m,\ c_i^{\mathrm{lb}} \le c_i(x^*) \le c_i^{\mathrm{ub}} \end{cases}$$

$$(A, b) \in \mathcal{M}_{n,n}(\mathbb{R}) \times \mathbb{R}^n,\ (C, d) \in \mathcal{M}_{n,m}(\mathbb{R}) \times \mathbb{R}^m$$

➡ Possibility to directly pass the matrices and vectors

Standard objective and affine constraints :

```
... //begining of the script
func real f(real[int] &X) {...}
func real[int] df(real[int] &X) {...}
func matrix H(real[int] &X) {...} //no need for the full prototype
matrix C = ... ;
real[int] d = ... ;
IPOPT(f, df, H, [C,d] , xstart ,  /*named parameters*/  );
```

The FreeFem++ Interface :

case of quadratic objective and affine constraints

$$\forall x \in \mathbb{R}^n, \quad \begin{array}{rcl} f(x) & = & \frac{1}{2}\langle Ax, x\rangle + \langle b, x\rangle \\ c(x) & = & Cx + d \end{array}$$

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \le n,\ x_i^{\mathrm{lb}} \le x_i^* \le x_i^{\mathrm{ub}} \\ \forall i \le m,\ c_i^{\mathrm{lb}} \le c_i(x^*) \le c_i^{\mathrm{ub}} \end{cases}$$

$$(A, b) \in \mathcal{M}_{n,n}(\mathbb{R}) \times \mathbb{R}^n,\ (C, d) \in \mathcal{M}_{n,m}(\mathbb{R}) \times \mathbb{R}^m$$

➡ Possibility to directly pass the matrices and vectors

Quadratic objective and standard constraints : hazardous case

```
... //begining of the script
func real[int] c(real[int] &X) {...}
func matrix Jc(real[int] &X) {...}
matrix A = ... ;
real[int] b = ... ;

IPOPT([A,b] , c, Jc, xstart ,  /*named parameters*/  );
```

The FreeFem++ Interface :

case of quadratic objective and affine constraints

$$\forall x \in \mathbb{R}^n, \quad \begin{aligned} f(x) &= \frac{1}{2}\langle Ax, x\rangle + \langle b, x\rangle \\ c(x) &= Cx + d \end{aligned}$$

$$\begin{cases} x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) \\ \\ \forall i \le n,\ x_i^{\mathrm{lb}} \le x_i^* \le x_i^{\mathrm{ub}} \\ \forall i \le m,\ c_i^{\mathrm{lb}} \le c_i(x^*) \le c_i^{\mathrm{ub}} \end{cases}$$

$$(A, b) \in \mathcal{M}_{n,n}(\mathbb{R}) \times \mathbb{R}^n,\ (C, d) \in \mathcal{M}_{n,m}(\mathbb{R}) \times \mathbb{R}^m$$

➡ Possibility to directly pass the matrices and vectors

Quadratic objective and standard constraints : hazardous case

```
... //begining of the script
func real[int] c(real[int] &X) {...}
func matrix Jc(real[int] &X) {...}
matrix A = ... ;
real[int] b = ... ;

IPOPT([A,b] , c, Jc, xstart ,  /*named parameters*/  );
```

Here, $H(x, \sigma, \lambda)$ is always equal to A
➡ erroneous hessian in some cases

The FreeFem++ Interface : some hints and tips

The FreeFem++ Interface : some hints and tips

Matrix returning functions : must return a global object

```
func matrix H(real[int] &X)
{
    matrix M;
    ...              //compute M
    return M;
}
```

The FreeFem++ Interface : some hints and tips

Matrix returning functions : must return a global object

```
func matrix H(real[int] &X)
{
    matrix M;
    ...              //compute M
    return M;
}
```

M is cleaned after closing the function block, so the returned object does not exist anymore...

The FreeFem++ Interface : some hints and tips

Sparse matrix returning functions : must return a global object

```
matrix M; //just declare a global matrix
func matrix H(real[int] &X)
{
    ...              //do something to fill M
    return M;  //M will not be deleted before H goes out of scope
}
```

The FreeFem++ Interface : some hints and tips

Sparse matrix returning functions : must return a global object

```
matrix M; //just declare a global matrix
func matrix H(real[int] &X)
{
    ...              //do something to fill M
    return M;  //M will not be deleted before H goes out of scope
}
```

The structure of the matrices : has to be constant through the optimization process...

- Use **varf** as much as possible to build matrices, matrices built with the same varf always have the same structure even if some coefficients becomes null.
- Remember that zeros are discarded when building sparse matrices from arrays :

```
real[int] c=...;
int[int] I=..., J=...;
matrix M = [I, J ,c];
```

```
real[int,int] tmp=...;
matrix M = tmp;
//two dangerous methods
```

The FreeFem++ Interface : some hints and tips (2)

The FreeFem++ Interface : some hints and tips (2)

Named parameters : subset of the IPOPT parameters, plus a few FreeFem++ specific ones, which can be used for :

- Stopping criteria (**tol**, **maxiter**, **maxcputime**, etc...)
- Getting final values (**objvalue**, **lm**, **uz**, **lz**, etc...)
- Perform a warm start
- Call a derivatives checker
- Force the BFGS mode (add **bfgs=1**)
- Specify an extern option file (see the IPOPT documentation)
- etc...

Check the documentation for more informations.

The FreeFem++ Interface : some hints and tips (2)

Named parameters : subset of the IPOPT parameters, plus a few FreeFem++ specific ones, which can be used for :

- Stopping criteria (**tol**, **maxiter**, **maxcputime**, etc...)
- Getting final values (**objvalue**, **lm**, **uz**, **lz**, etc...)
- Perform a warm start
- Call a derivatives checker
- Force the BFGS mode (add **bfgs=1**)
- Specify an extern option file (see the IPOPT documentation)
- etc...

Check the documentation for more informations.

Returned value : the IPOPT function returns an **int** revealing what happens during the optimization. Positive values often means IPOPT performed well and negative values are relevant of troubles.

Part II

Bose-Einstein Condensate Simulations using FreeFem++ and IPOPT

The Gross-Pitaevskii energy : complex version

$\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, and $u \in H^1(\mathcal{D}, \mathbb{C})$

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}.\,(iu, \nabla u)$$

The Gross-Pitaevskii energy : complex version

$\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, and $u \in H^1(\mathcal{D}, \mathbb{C})$

Trapping potential

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}.\,(iu, \nabla u)$$

Trapping potential :

The Gross-Pitaevskii energy : complex version

$\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, and $u \in H^1(\mathcal{D}, \mathbb{C})$

Trapping potential

- harmonic

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}.\,(iu, \nabla u)$$

Trapping potential :

- Harmonic : $V = \frac{1}{2}\left(\omega_x^2 x^2 + \omega_y^2 y^2 + \omega_z^2 z^2\right)$

  Limits the magnitude of the rotation Ω by $\min\left(\omega_x, \omega_y, \omega_z\right)$.

The Gross-Pitaevskii energy : complex version

$\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, and $u \in H^1(\mathcal{D}, \mathbb{C})$

Trapping potential
- harmonic
- harmonic + quartic

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}.\,(iu, \nabla u)$$

Trapping potential :

- Harmonic : $V = \frac{1}{2}\left(\omega_x^2 x^2 + \omega_y^2 y^2 + \omega_z^2 z^2\right)$

  Limits the magnitude of the rotation Ω by $\min\left(\omega_x, \omega_y, \omega_z\right)$.
- Harmonic + quartic : $V = \frac{1}{2}\omega_\perp^2 r^2 + \frac{1}{4}\kappa r^4 + \frac{1}{2}\omega_z^2 z^2$

  Allows any rotation speed.

The Gross-Pitaevskii energy : complex version

$\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, and $u \in H^1(\mathcal{D}, \mathbb{C})$

Trapping potential
- harmonic
- harmonic + quartic

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}.\,(iu, \nabla u)$$

Trapping potential :

- Harmonic : $V = \frac{1}{2}\left(\omega_x^2 x^2 + \omega_y^2 y^2 + \omega_z^2 z^2\right)$

  Limits the magnitude of the rotation Ω by $\min(\omega_x, \omega_y, \omega_z)$.
- Harmonic + quartic : $V = \frac{1}{2}\omega_\perp^2 r^2 + \frac{1}{4}\kappa r^4 + \frac{1}{2}\omega_z^2 z^2$

  Allows any rotation speed.

The Gross-Pitaevskii energy : complex version

$\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, and $u \in H^1(\mathcal{D}, \mathbb{C})$

Trapping potential
- harmonic
- harmonic + quartic

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}.\,(iu, \nabla u)$$

Trapping potential :
- Harmonic : $V = \frac{1}{2}\left(\omega_x^2 x^2 + \omega_y^2 y^2 + \omega_z^2 z^2\right)$

  Limits the magnitude of the rotation Ω by $\min(\omega_x, \omega_y, \omega_z)$.
- Harmonic + quartic : $V = \frac{1}{2}\omega_\perp^2 r^2 + \frac{1}{4}\kappa r^4 + \frac{1}{2}\omega_z^2 z^2$

  Allows any rotation speed.

The Gross-Pitaevskii energy : complex version

$\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, and $u \in H^1(\mathcal{D}, \mathbb{C})$

Trapping potential
- harmonic
- harmonic + quartic

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}.\,(iu, \nabla u)$$

Trapping potential :
- Harmonic : $V = \frac{1}{2}\left(\omega_x^2 x^2 + \omega_y^2 y^2 + \omega_z^2 z^2\right)$

  Limits the magnitude of the rotation Ω by $\min(\omega_x, \omega_y, \omega_z)$.
- Harmonic + quartic : $V = \frac{1}{2}\omega_\perp^2 r^2 + \frac{1}{4}\kappa r^4 + \frac{1}{2}\omega_z^2 z^2$

  Allows any rotation speed.

The Gross-Pitaevskii energy : complex version

$\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, and $u \in H^1(\mathcal{D}, \mathbb{C})$

Trapping potential
- harmonic
- harmonic + quartic

Atomic coupling constant

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}.\,(iu, \nabla u)$$

Constants :

- Atomic coupling constant : g

  Higher values result in more numerous and smaller vortices.

Trapping potential :
- Harmonic : $V = \frac{1}{2}\left(\omega_x^2 x^2 + \omega_y^2 y^2 + \omega_z^2 z^2\right)$

  Limits the magnitude of the rotation Ω by $\min(\omega_x, \omega_y, \omega_z)$.
- Harmonic + quartic : $V = \frac{1}{2}\omega_\perp^2 r^2 + \frac{1}{4}\kappa r^4 + \frac{1}{2}\omega_z^2 z^2$

  Allows any rotation speed.

The Gross-Pitaevskii energy : complex version

$\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, and $u \in H^1(\mathcal{D}, \mathbb{C})$

Trapping potential
- harmonic
- harmonic + quartic

Atomic coupling constant

Rotation vector

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}.\,(iu, \nabla u)$$

Constants :

- Atomic coupling constant : g

  Higher values result in more numerous and smaller vortices.
- Dimensionless angular velocity of the rotation : Ω

  Increase the size and the number of vortices.

Trapping potential :
- Harmonic : $V = \frac{1}{2}\left(\omega_x^2 x^2 + \omega_y^2 y^2 + \omega_z^2 z^2\right)$

  Limits the magnitude of the rotation Ω by $\min(\omega_x, \omega_y, \omega_z)$.
- Harmonic + quartic : $V = \frac{1}{2}\omega_\perp^2 r^2 + \frac{1}{4}\kappa r^4 + \frac{1}{2}\omega_z^2 z^2$

  Allows any rotation speed.

The Gross-Pitaevskii energy : complex version

$\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, and $u \in H^1(\mathcal{D}, \mathbb{C})$

Trapping potential
- harmonic
- harmonic + quartic

Atomic coupling constant

Rotation vector

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}.\,(iu, \nabla u)$$

Constants :

- Atomic coupling constant : g
  Higher values result in more numerous and smaller vortices.
- Dimensionless angular velocity of the rotation : Ω
  Increase the size and the number of vortices.

Trapping potential :
- Harmonic : $V = \frac{1}{2}\left(\omega_x^2 x^2 + \omega_y^2 y^2 + \omega_z^2 z^2\right)$
  Limits the magnitude of the rotation Ω by $\min(\omega_x, \omega_y, \omega_z)$.
- Harmonic + quartic : $V = \frac{1}{2}\omega_\perp^2 r^2 + \frac{1}{4}\kappa r^4 + \frac{1}{2}\omega_z^2 z^2$
  Allows any rotation speed.

The Gross-Pitaevskii energy : complex version

$\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, and $u \in H^1(\mathcal{D}, \mathbb{C})$

Trapping potential
- harmonic
- harmonic + quartic

Atomic coupling constant

Rotation vector

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}.\,(iu, \nabla u)$$

Constants :

- Atomic coupling constant : g
  Higher values result in more numerous and smaller vortices.
- Dimensionless angular velocity of the rotation : Ω
  Increase the size and the number of vortices.

Trapping potential :
- Harmonic : $V = \frac{1}{2}\left(\omega_x^2 x^2 + \omega_y^2 y^2 + \omega_z^2 z^2\right)$
  Limits the magnitude of the rotation Ω by $\min(\omega_x, \omega_y, \omega_z)$.
- Harmonic + quartic : $V = \frac{1}{2}\omega_\perp^2 r^2 + \frac{1}{4}\kappa r^4 + \frac{1}{2}\omega_z^2 z^2$
  Allows any rotation speed.

The Gross-Pitaevskii energy : complex version

$\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, and $u \in H^1(\mathcal{D}, \mathbb{C})$

Trapping potential
- harmonic
- harmonic + quartic

Atomic coupling constant

Rotation vector

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}.\,(iu, \nabla u)$$

Trapping potential :
- Harmonic : $V = \frac{1}{2}\left(\omega_x^2 x^2 + \omega_y^2 y^2 + \omega_z^2 z^2\right)$
  Limits the magnitude of the rotation Ω by $\min(\omega_x, \omega_y, \omega_z)$.
- Harmonic + quartic : $V = \frac{1}{2}\omega_\perp^2 r^2 + \frac{1}{4}\kappa r^4 + \frac{1}{2}\omega_z^2 z^2$
  Allows any rotation speed.

Constants :
- Atomic coupling constant : g
  Higher values result in more numerous and smaller vortices.
- Dimensionless angular velocity of the rotation : Ω
  Increase the size and the number of vortices.

The Gross-Pitaevskii energy : complex version

$\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, and $u \in H^1(\mathcal{D}, \mathbb{C})$

Trapping potential
- harmonic
- harmonic + quartic

Atomic coupling constant

Rotation vector

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}.\,(iu, \nabla u)$$

$\mathbf{x}$: current point

$\mathbf{\Omega} = \Omega \mathbf{e}_z$

$(a, b) = a\overline{b} + \overline{a}b$

Trapping potential :
- Harmonic : $V = \frac{1}{2}\left(\omega_x^2 x^2 + \omega_y^2 y^2 + \omega_z^2 z^2\right)$
  Limits the magnitude of the rotation Ω by $\min(\omega_x, \omega_y, \omega_z)$.
- Harmonic + quartic : $V = \frac{1}{2}\omega_\perp^2 r^2 + \frac{1}{4}\kappa r^4 + \frac{1}{2}\omega_z^2 z^2$
  Allows any rotation speed.

Constants :
- Atomic coupling constant : g
  Higher values result in more numerous and smaller vortices.
- Dimensionless angular velocity of the rotation : Ω
  Increase the size and the number of vortices.

The Gross-Pitaevskii energy : real/imag. part formulation

$\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, and $u \in H^1(\mathcal{D}, \mathbb{C})$

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}. \, (iu, \nabla u)$$

The Gross-Pitaevskii energy : real/imag. part formulation

$\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, and $u \in H^1(\mathcal{D}, \mathbb{C})$

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}. \, (iu, \nabla u)$$

The Gross-Pitaevskii energy : real/imag. part formulation

$\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, and $u \in H^1(\mathcal{D}, \mathbb{C})$

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}. (iu, \nabla u)$$

Optimizers works with real variables

$\Longrightarrow$ We need an expression of E as a function of the real and the imaginary parts $(u_r, u_i) \in H^1(\mathcal{D})^2$.

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right)$$

$$+ \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x}. (u_r \nabla u_i - u_i \nabla u_r)$$

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right)$$

$$+ \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x}. \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right)$$

$$+ \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x}. \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

Optimization method : IPOPT

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right)$$

$$+ \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x}. \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

Optimization method : IPOPT

No inequality constraint or simple bound

$\Longrightarrow$
- no intermediary barrier problem
- IPOPT acts as a smart Newton method with filter line search

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right)$$

$$+ \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x}. \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

Optimization method : IPOPT

- IPOPT acts as a smart Newton method with filter line search for this problem (no intermediary barrier problem)
- we need at least the first derivatives of $E$and constraint, and hessians for an optimal operating

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right)$$

$$+ \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x}. \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

Optimization method : IPOPT

- IPOPT acts as a smart Newton method with filter line search for this problem (no intermediary barrier problem)
- we need at least the first derivatives of $E$and constraint, and hessians for an optimal operating

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right)$$

$$+ \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x}. \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

Optimization method : IPOPT

- IPOPT acts as a smart Newton method with filter line search for this problem (no intermediary barrier problem)
- we need at least the first derivatives of $E$and constraint, and hessians for an optimal operating

BEC Simulations using IPOPT

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right) + \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x}. \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

Discretization / Implementation :

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right) + \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x} . \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

Discretization / Implementation :

P^1 finite elements :

$$V_h = \{v \in H^1(\mathcal{T}_h) \text{ s.t. } \forall T \in \mathcal{T}_h,\ v|_T \in P^1(T)\}$$

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right) + \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x} . \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

Discretization / Implementation :

P1 finite elements :

$$V_h = \{v \in H^1(\mathcal{T}_h) \text{ s.t. } \forall T \in \mathcal{T}_h,\ v|_T \in P^1(T)\}$$

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right) + \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x} . \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

FreeFem implementation of the functional :

```
func real J(real[int] & u)
{
  Vh2 [ur,ui] ; ur[]=u;
  real E = int2d(Th) (
        ( Grad(ur)'*Grad(ur) + Grad(ui)'*Grad(ui) ) * 0.5
     + Vtrap* ( ur*ur + ui * ui)
     + g* square( ur*ur + ui * ui) * 0.5
     + Omega*(  y *( - dx(ur)*ui + dx(ui)*ur )
              + x *(   dy(ur)*ui - dy(ui)*ur ) )
              );
   return E;
}
```

Discretization / Implementation :

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right) + \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x} . \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

$$V_h = \left\{ v \in H^1(\mathcal{T}_h) \text{ s.t. } \forall T \in \mathcal{T}_h,\ v|_T \in P^1(T) \right\}$$

FreeFem implementation of the functional :

```
func real J(real[int] & u)
{
  Vh2 [ur,ui] ; ur[]=u;
  real E = int2d(Th) (
       ( Grad(ur)'*Grad(ur) + Grad(ui)'*Grad(ui) ) * 0.5
     + Vtrap* ( ur*ur + ui * ui)
     + g* square( ur*ur + ui * ui) * 0.5
     + Omega*(  y *( - dx(ur)*ui + dx(ui)*ur )
              + x *(   dy(ur)*ui - dy(ui)*ur ) )
              );
   return E;
}
```

Discretization / Implementation : derivatives

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right) + \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x} . \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

$$V_h = \left\{ v \in H^1(\mathcal{T}_h) \text{ s.t. } \forall T \in \mathcal{T}_h,\ v|_T \in P^1(T) \right\}$$

Discretization / Implementation : derivatives

Let $(u_r, u_i) \in H^1(\mathcal{D})^2$, for all functions $(v_r, v_i) \in H^1(\mathcal{D})^2$, vanishing on the border $\partial\mathcal{D}$:

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right) + \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x}. \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

$$V_h = \left\{ v \in H^1(\mathcal{T}_h) \text{ s.t. } \forall T \in \mathcal{T}_h,\ v|_T \in P^1(T) \right\}$$

$$dE(u_r, u_i)(v_r, v_i) = \int_{\mathcal{D}} (\nabla u_r . \nabla v_r + \nabla u_i . \nabla v_i) + \int_{\mathcal{D}} 2V \left(u_r v_r + u_i v_i \right) + \int_{\mathcal{D}} g \left(u_r^2 + u_i^2 \right) \left(u_r v_r + u_i v_i \right) - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x}. \left(v_r \nabla u_i + u_r \nabla v_i - v_i \nabla u_r - u_i \nabla v_r \right)$$

Discretization / Implementation : derivatives

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right) + \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x} . \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

$$V_h = \left\{ v \in H^1(\mathcal{T}_h) \text{ s.t. } \forall T \in \mathcal{T}_h,\ v|_T \in P^1(T) \right\}$$

Corresponding FreeFem++ code :

```
func real[int] dJ(real[int] & u)
{
  Vh2 [ur,ui] ; ur[]=u;
  varf vdj([aa,bb],[vr,vi]) =
    int2d(Th) (
      ( Grad(ur)'*Grad(vr) +  Grad(ui)'*Grad(vi) )
    + Vtrap* ( ur*vr + ui * vi)*2.
    + g*  ( ur*vr + ui*vi) *( ur*ur + ui*ui)
    + Omega*(  y *( - dx(ur)*vi + dx(ui)*vr )
             + x *(   dy(ur)*vi - dy(ui)*vr ) )
    + Omega*(  y *( - dx(vr)*ui + dx(vi)*ur )
             + x *(   dy(vr)*ui - dy(vi)*ur ) )
             );
    real [int] du= vdj(0,Vh2);
  return du ;
}
```

Discretization / Implementation : hessian

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right) + \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x} . \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

$$V_h = \left\{ v \in H^1(\mathcal{T}_h) \text{ s.t. } \forall T \in \mathcal{T}_h,\ v|_T \in P^1(T) \right\}$$

Discretization / Implementation : hessian

Let $(u_r, u_i) \in H^1(\mathcal{D})^2$, for all functions (v_r, v_i) and (w_r, w_i), vanishing on the border $\partial\mathcal{D}$:

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right) + \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x}. \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

$$V_h = \left\{ v \in H^1(\mathcal{T}_h) \text{ s.t. } \forall T \in \mathcal{T}_h,\ v|_T \in P^1(T) \right\}$$

$$d^2 E(u_r, u_i)\left[(v_r, v_i), (w_r, w_i)\right] =$$

$$\int_{\mathcal{D}} \left(\nabla w_r . \nabla v_r + \nabla w_i . \nabla v_i \right) + 2V \left(w_r v_r + w_i v_i \right)$$

$$+ \int_{\mathcal{D}} g \left(u_r^2 + u_i^2 \right) \left(w_r v_r + w_i v_i \right) + 2g \left(w_r u_r + w_i u_i \right) \left(v_r u_r + v_i u_i \right)$$

$$- \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x}. \left(v_r \nabla w_i + w_r \nabla v_i + v_i \nabla w_r + w_i \nabla v_r \right)$$

Discretization / Implementation : hessian

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right) + \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x} . \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

$$V_h = \left\{ v \in H^1(\mathcal{T}_h) \text{ s.t. } \forall T \in \mathcal{T}_h,\ v|_T \in P^1(T) \right\}$$

Corresponding FreeFem++ code :

```
matrix Hessian;

func matrix hJ(real[int] & u,real obj,real[int] & l)
{
  Vh2 [ur,ui] ; ur[]=u;
  varf vhj([wr,wi],[vr,vi]) =
     int2d(Th) (
      + (obj*Vtrap+ l[0]) * ( wr*vr + wi*vi)*2.
      + obj*(
        ( Grad(wr)'*Grad(vr) +  Grad(wi)'*Grad(vi) )
      + g*      ( wr*vr + wi*vi) * ( ur*ur + ui*ui)
      + g*2.*   ( ur*vr + ui*vi) * ( wr*ur + wi*ui)

      + Omega*(  y *( - dx(wr)*vi + dx(wi)*vr )
               + x *(   dy(wr)*vi - dy(wi)*vr ) )
      + Omega*(  y *( - dx(vr)*wi + dx(vi)*wr )
               + x *(   dy(vr)*wi - dy(vi)*wr ) )
            ));
  Hessian = vhj(Vh2,Vh2);
  return Hessian;
}
```

Discretization / Implementation : hessian

Corresponding FreeFem++ code :

```
matrix Hessian;

func matrix hJ(real[int] & u,real obj,real[int] & l)
{
  Vh2 [ur,ui] ; ur[]=u;
  varf vhj([wr,wi],[vr,vi]) =
     int2d(Th) (
      + (obj*Vtrap+ l[0]) * ( wr*vr + wi*vi)*2.
      + obj*(
        ( Grad(wr)'*Grad(vr) +  Grad(wi)'*Grad(vi) )
      + g*      ( wr*vr + wi*vi) * ( ur*ur + ui*ui)
      + g*2.*   ( ur*vr + ui*vi) * ( wr*ur + wi*ui)

      + Omega*(  y *( - dx(wr)*vi + dx(wi)*vr )
                + x *(   dy(wr)*vi - dy(wi)*vr ) )
      + Omega*(  y *( - dx(vr)*wi + dx(vi)*wr )
                + x *(   dy(vr)*wi - dy(vi)*wr ) )
              ));
  Hessian = vhj(Vh2,Vh2);
  return Hessian;
}
```

Minimization problem : Given a domain $\mathcal{D} \subset \mathbb{R}^d$, with $d = 2$ or 3, a potential $V \in L^2(\mathcal{D})$ and $g, \Omega \in \mathbb{R}^+$, minimize

$$\widetilde{E}(u_r, u_i) = \int_{\mathcal{D}} \left(\frac{1}{2} |\nabla u_r|^2 + \frac{1}{2} |\nabla u_i|^2 \right) + \int_{\mathcal{D}} V \left(u_r^2 + u_i^2 \right) + \int_{\mathcal{D}} \frac{g}{2} \left| u_r^2 + u_i^2 \right|^2 - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x} . \left(u_r \nabla u_i - u_i \nabla u_r \right)$$

over $H^1(\mathcal{D})^2$, under the constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$.

$$V_h = \left\{ v \in H^1(\mathcal{T}_h) \text{ s.t. } \forall T \in \mathcal{T}_h,\ v|_T \in P^1(T) \right\}$$

Constraint :

$$\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$$

$\Rightarrow$ quadratic

We just build the matrix with varf.

Minimizing with IPOPT :

Minimizing with IPOPT :

- Energy functional

$$E(u) = \int_{\mathcal{D}} \frac{1}{2}|\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}.(iu, \nabla u)$$

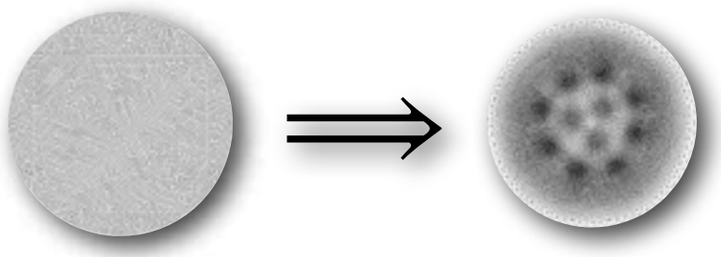
- Constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$

- Derivatives

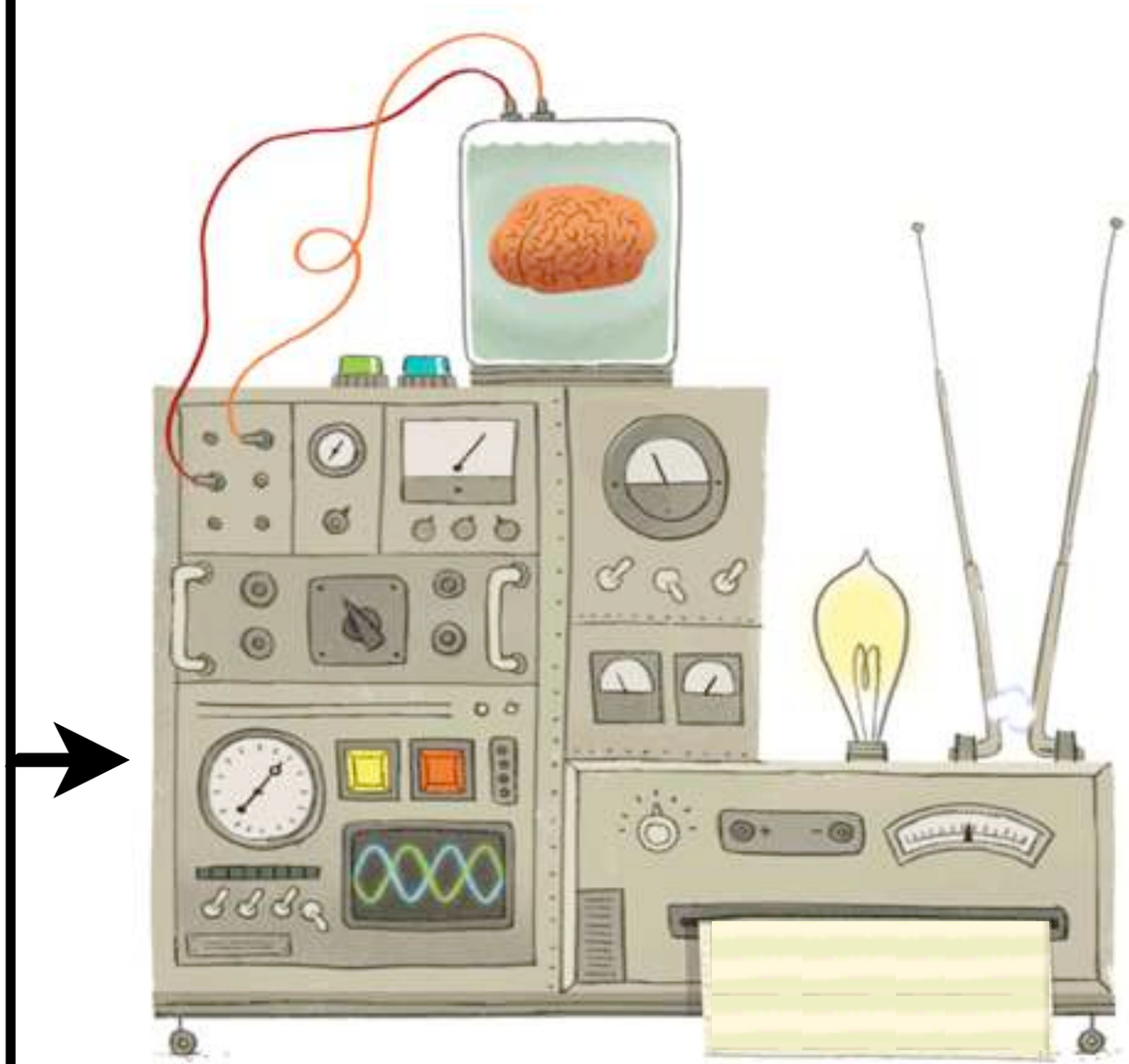
$$\begin{aligned} d^2E(u_r,u_i)[(v_r,v_i),(w_r,w_i)] = \\ \int_{\mathcal{D}} (\nabla w_r.\nabla v_r + \nabla w_i.\nabla v_i) + 2V\,(w_r v_r + w_i v_i) \\ + \int_{\mathcal{D}} g\,(u_r^2 + u_i^2)\,(w_r v_r + w_i v_i) + 2g\,(w_r u_r + w_i u_i)\,(v_r u_r + v_i u_i) \\ - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x}.(v_r \nabla w_i + w_r \nabla v_i + v_i \nabla w_r + w_i \nabla v_r) \end{aligned}$$

- Initialization with Thomas-Fermi density $\rho_{\mathrm{TF}} = \dfrac{\mu - V}{g}$

- Mesh adaptation



- Optimizations for 3D



Minimizing with IPOPT :

- Energy functional

$$E(u) = \int_{\mathcal{D}} \frac{1}{2}|\nabla u|^2 + V|u|^2 + \frac{g}{2}|u|^4 - \frac{1}{2}\mathbf{\Omega} \wedge \mathbf{x}.(iu, \nabla u)$$

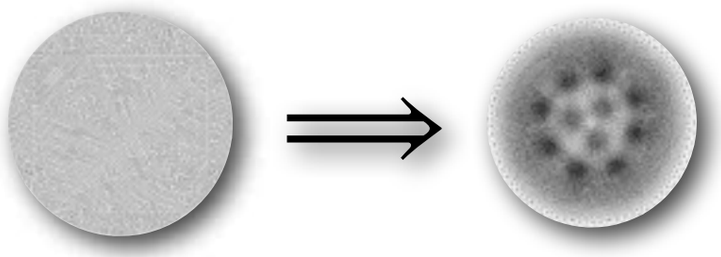
- Constraint $\int_{\mathcal{D}} u_r^2 + u_i^2 = 1$

- Derivatives

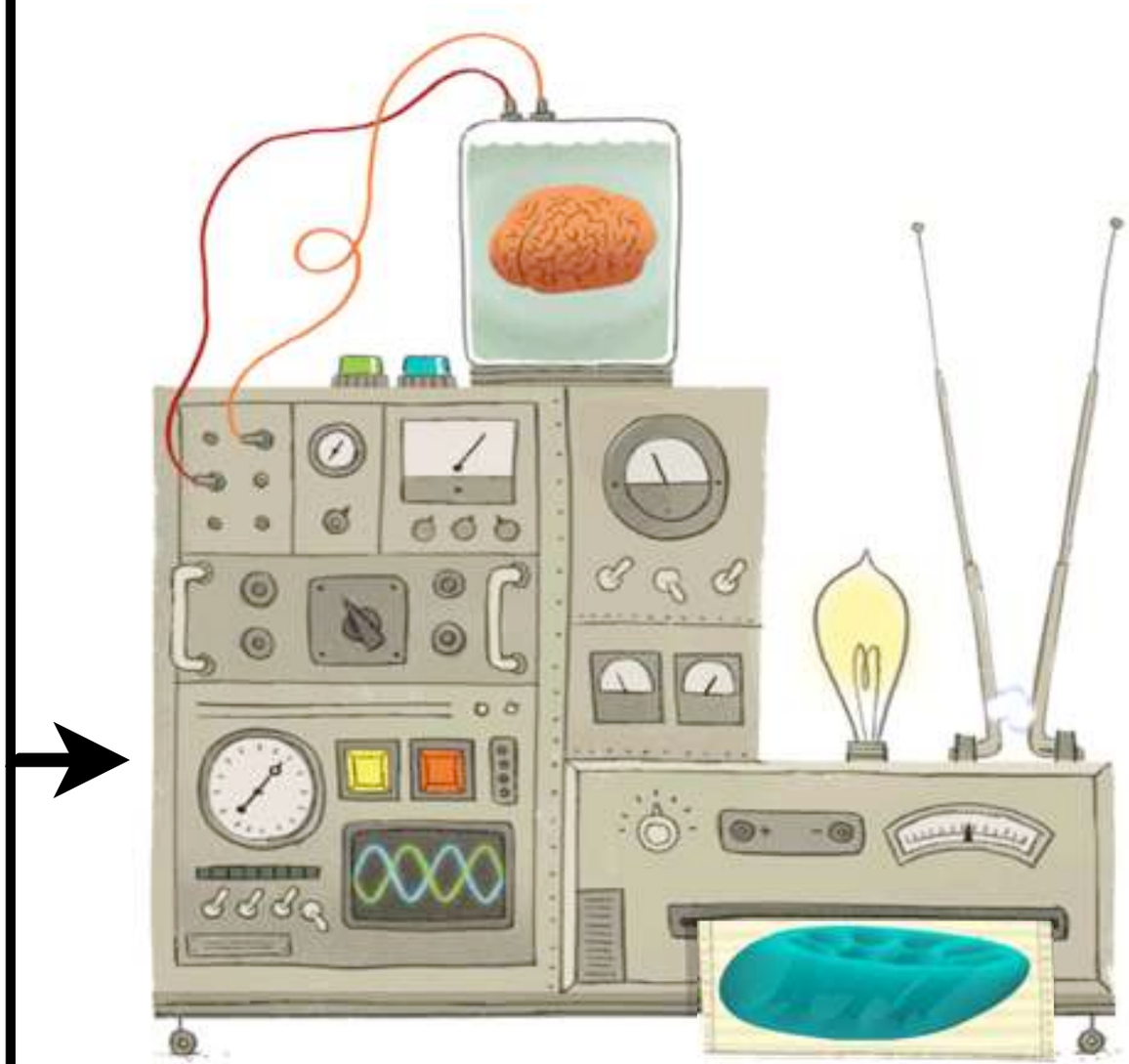
$$\begin{aligned} d^2E(u_r,u_i)[(v_r,v_i),(w_r,w_i)] = \\ \int_{\mathcal{D}} (\nabla w_r.\nabla v_r + \nabla w_i.\nabla v_i) + 2V\,(w_r v_r + w_i v_i) \\ + \int_{\mathcal{D}} g\,(u_r^2+u_i^2)\,(w_r v_r + w_i v_i) + 2g\,(w_r u_r + w_i u_i)\,(v_r u_r + v_i u_i) \\ - \int_{\mathcal{D}} \mathbf{\Omega} \wedge \mathbf{x}.(v_r \nabla w_i + w_r \nabla v_i + v_i \nabla w_r + w_i \nabla v_r) \end{aligned}$$

- Initialization with Thomas-Fermi density $\rho_{\mathrm{TF}} = \dfrac{\mu - V}{g}$

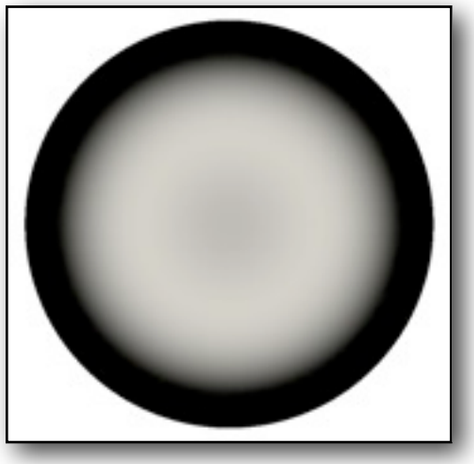
- Mesh adaptation



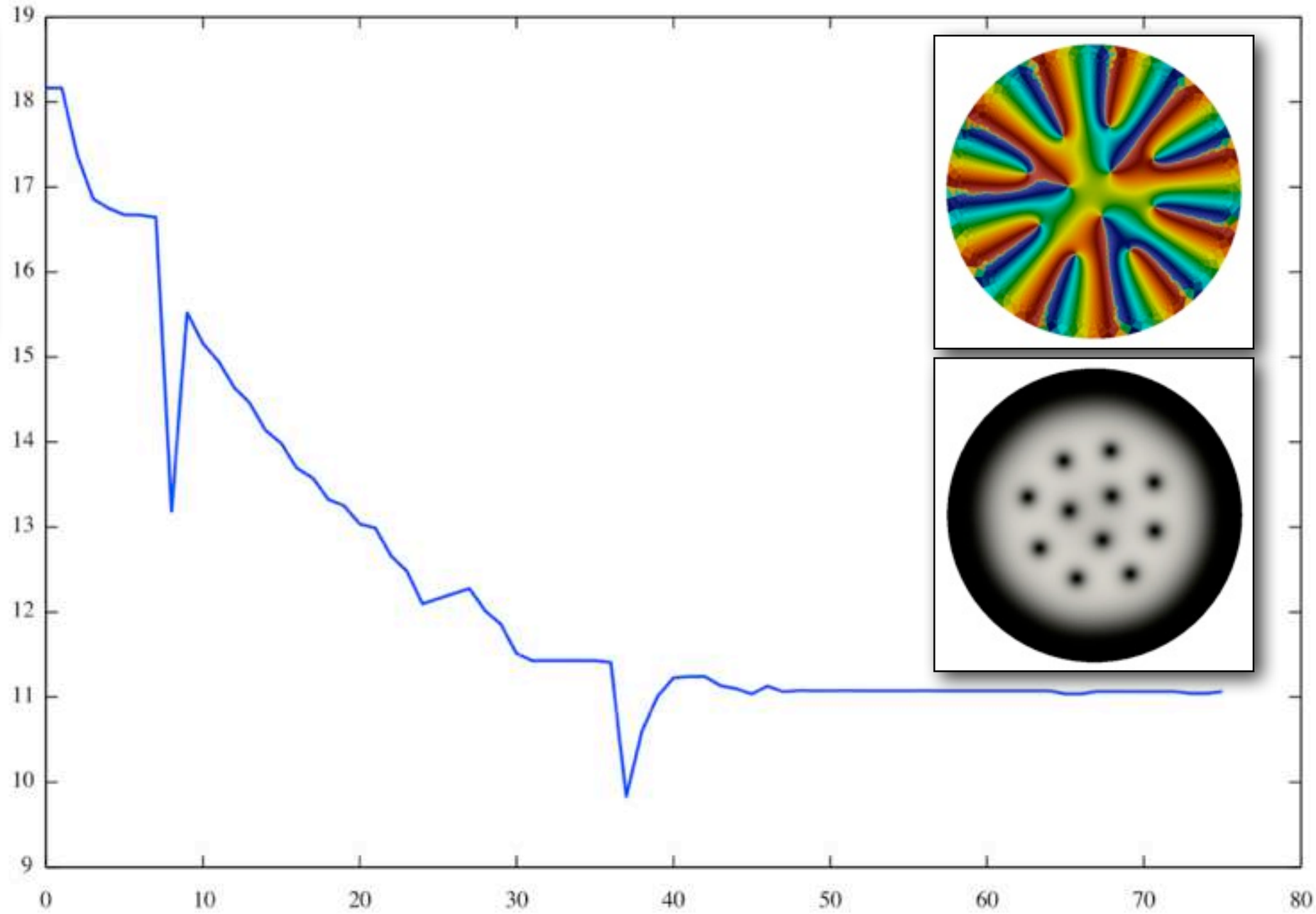
- Optimizations for 3D



Results in 2D : harmonic+quartic potential - $g = 500$ - $\Omega = 2$



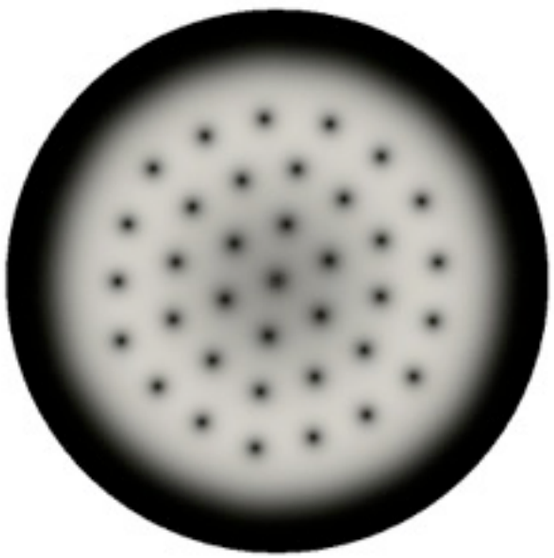
Initialisation



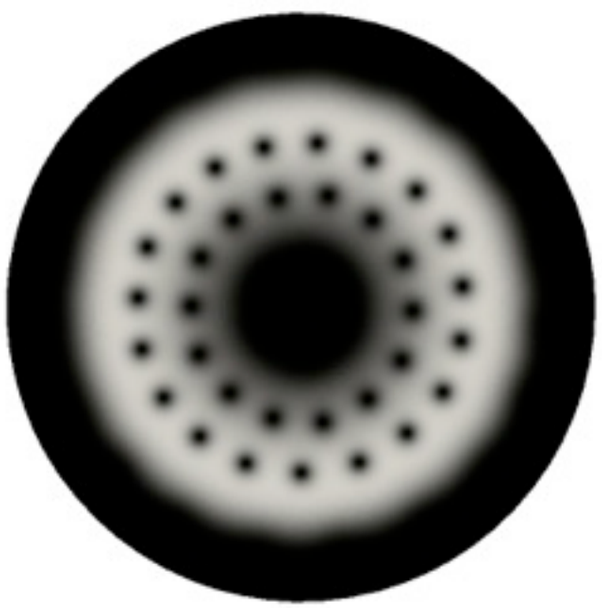
Results in 2D : harmonic+quartic potential - $g = 1000$

Results in 2D : harmonic+quartic potential - $g = 1000$

$\Omega = 3$

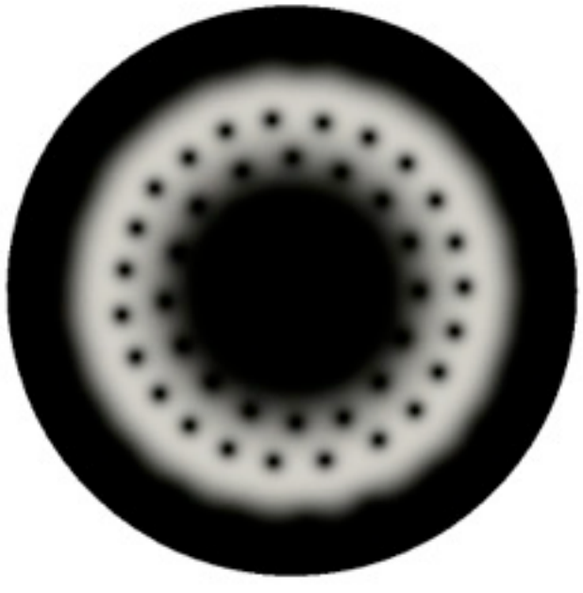


$\Omega = 5$



$\Omega = 4$

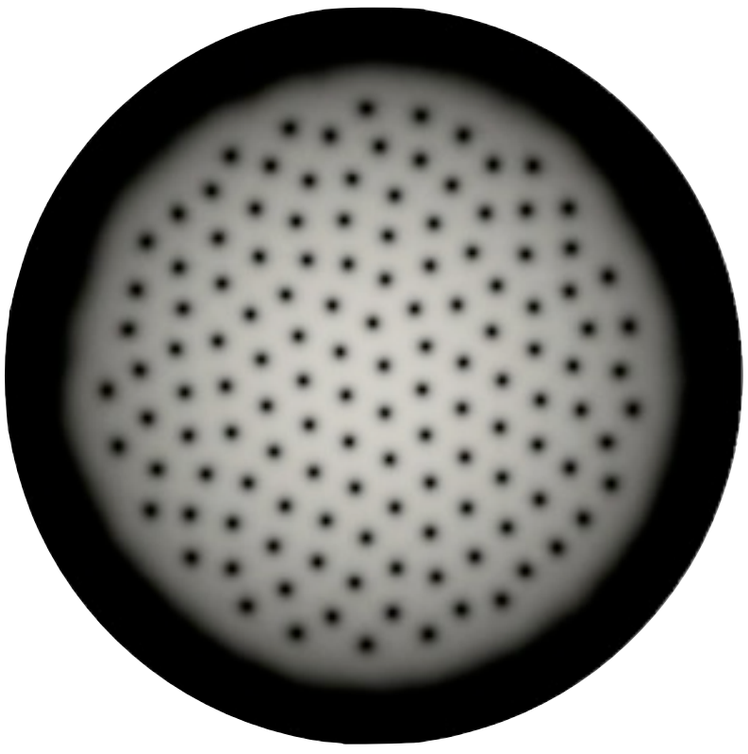
$\Omega = 3{,}5$



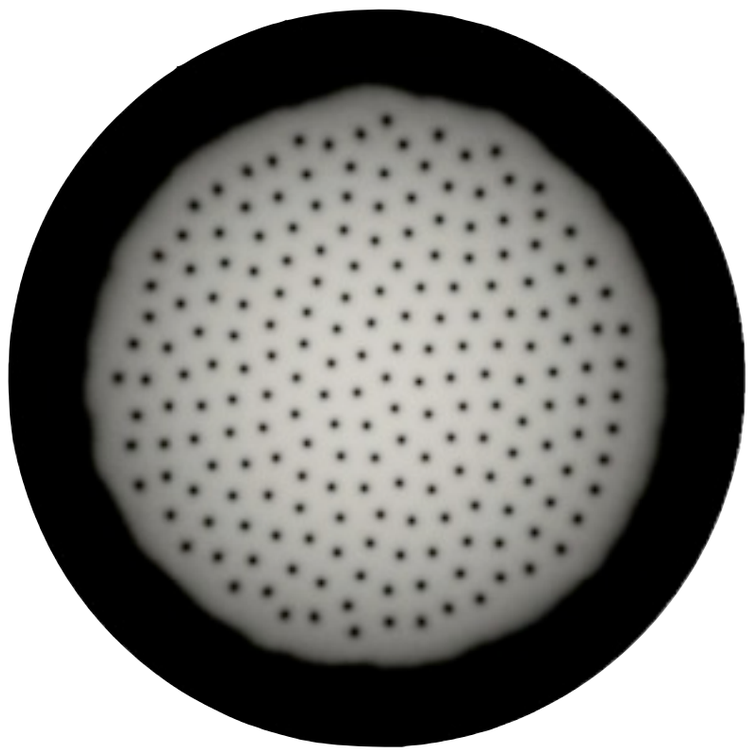
	Ω	M	N_t/N_v	E(u)	Itérations
IPOPT	3	200	33558/67038	2.72	151
	3.5	200	45605/91141	−12.03	164
	4	200	89110/178142	−35.95	155
	5	200	157069/314050	−122.3	318

Results in 2D : harmonic potential - $\Omega = 0.95$

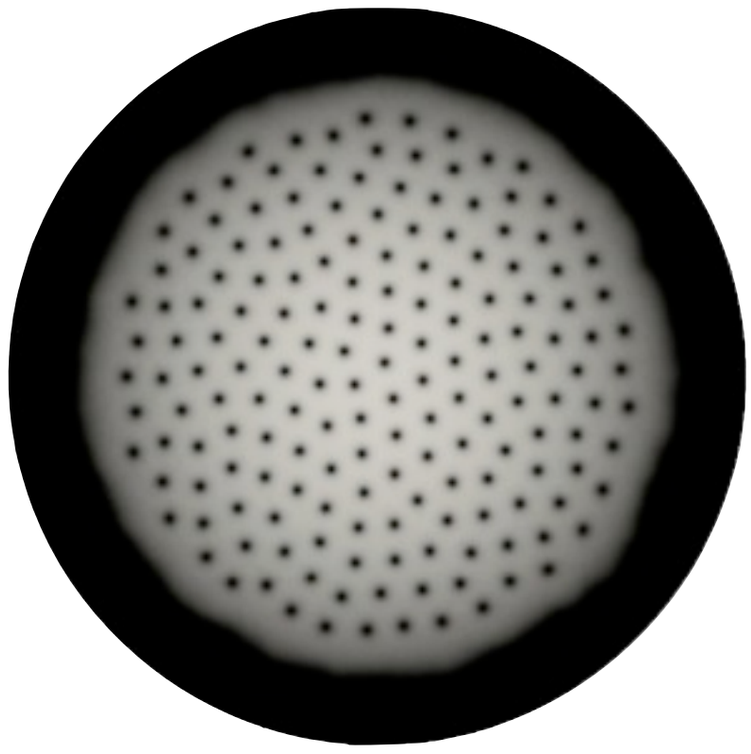
Results in 2D : harmonic potential - $\Omega = 0.95$



$g = 5000$



$g = 10000$

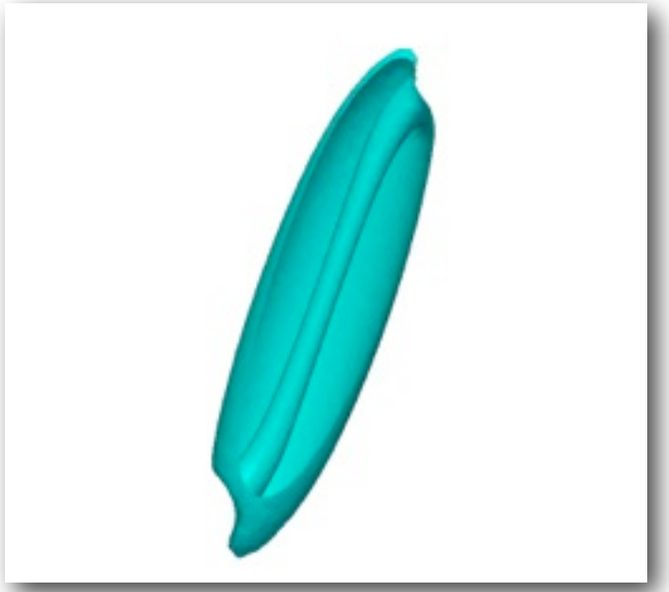


$g = 15000$

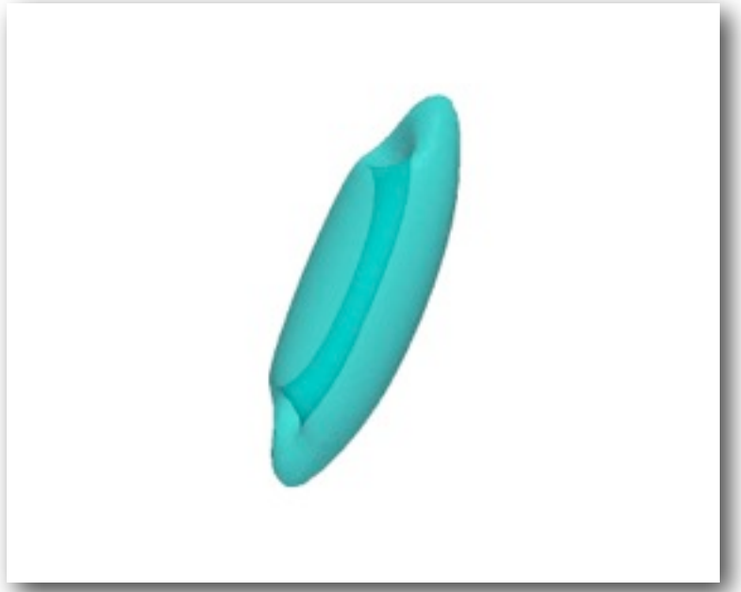
	g	M	N_t/N_v	E(u)	Itérations	CPU (s)
IPOPT	5000	200	114830/229568	10.28	615	16658 (4.6h)
	10000	200	144172/288264	14.38	952	44206 (12.2h)
	15000	200	170490/340912	17.67	880	35483 (9.8h)

Results in 3D :

Results in 3D :



S Shape Vortex
Anisotropic harmonic potential
$g = 1250 \quad \Omega = 0.5$



U Shape Vortex
Anisotropic harmonic potential
$g = 1250 \quad \Omega = 0.5$

Wasp Nest Vortex
Anisotropic harmonic potential
$g = 5000 \quad \Omega = 0.95$

Isotropic (w.r.t x y) hormonic potential
$g = 5000 \quad \Omega = 0.95$

Anisotropic harmonic potential
$g = 2000 \quad \Omega = 0.8$

Isotropic (w.r.t x y) hormonic+quartic potential $g = 2500$ and $\Omega = 1.5$.

Results in 3D :

S Shape Vortex
Anisotropic harmonic potential
$g = 1250 \quad \Omega = 0.5$

U Shape Vortex
Anisotropic harmonic potential
$g = 1250 \quad \Omega = 0.5$

Wasp Nest Vortex
Anisotropic harmonic potential
$g = 5000 \quad \Omega = 0.95$

Isotropic (w.r.t x y) hormonic potential
$g = 5000 \quad \Omega = 0.95$

Anisotropic harmonic potential
$g = 2000 \quad \Omega = 0.8$

Isotropic (w.r.t x y) hormonic+quartic potential $g = 2500$ and $\Omega = 1.5$.

Perspectives :

Perspectives :

- Deeper exploration of 3D computations

Perspectives :

- Deeper exploration of 3D computations

- Parallelization of the code

  ➡ by using parallelized linear solver (PARDISO?)

  ➡ parallel hessians and gradients computation can help

Perspectives :

- Deeper exploration of 3D computations

- Parallelization of the code

  ➡ by using parallelized linear solver (PARDISO?)

  ➡ parallel hessians and gradients computation can help

- Avoiding jumps between local minimizers

  ➡ rewrite the optimizer to keep only the needed features (barrier problems not used, avoid the *feasibility restoration phase*)