Introduction to Gradient-Based Optimisation

Part 4: Constrained optimisation

Dr. J.-D. Müller
School of Engineering and Materials Science,
Queen Mary, University of London
j.mueller@qmul.ac.uk

UK Fluids Network SIG on Numerical Optimisation with Fluids
Cambridge, 8-10 August 2018

Notes

Organisation of the lectures

1. Univariate optimisation
   - Bisection, Steepest Descent, Newton's method
2. Multivariate optimisation
   - Steepest descent, Newton's method
   - and line-search methods: Wolfe and Armijo conditions,
   - Quasi-Newton methods,
3. Constrained Optimisation:
   - Projected gradient methods,
   - Penalty methods, exterior and interior point methods,
   - SQP
4. Adjoint methods
   - Reversing time, Automatic Differentiation
   - Adjoint CFD codes
5. Gradient computation
   - Manual derivation, Finite Differences
   - Algorithmic and automatic differentiation, fwd and bkwd.

Notes

Outline

Notes

Outline

Notes

Examples of constrained optimisation

A typical optimisation application in aerodynamics is to minimise the drag of a profile (aircraft wing, wind-turbine blade).
Aerodynamic drag increases due to wetted area (skin friction), but also due to generated lift (induced drag due to tip vortices).
Hence simply minimising drag would shrink the profile to a point with zero lift.
Constrained optimisation allows to prevent this:

$$\min c_D \quad \text{s.t.} \quad c_L = c_{L,Target}$$

Now we can minimise the drag, but ensure that we do not reduce the lift.

Notes

Tank example

Properties of a tank:

$$\text{Volume of a tank:} \quad V = x_1 x_2 x_3 \tag{1}$$

$$\text{Surface:} \quad S = 2x_1x_2 + 2x_1x_3 + x_2x_3 \tag{2}$$

Constrained optimisation:

$$\text{Minimise} \quad S \quad \text{subject to} \quad V = V^* \tag{3}$$

Notes

Outline

Notes

Optimality conditions

The system to optimise is

$$\min_x f(x) \quad \text{s.t.} \quad c_i(x) = 0, \quad i = 1, k$$

with k equality constraints and n controls x.

Consider a line-search as a constraint, i.e. find the minimum along that line:

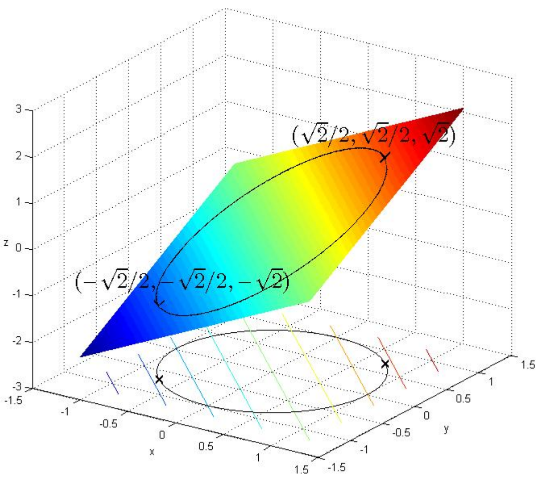
- The magnitude of the gradient is not zero in the minimum along the constraint direction,
- But the projection of the gradient along the constraint direction is zero. The gradient is perpendicular to the constraint.
- The constraints restrict the optimum to reside within a sub-manifold of f.

Notes

A very simple example of constrained optimisation

$$f(x, y) = x + y,$$
$$\text{s.t.} \quad x^2 + y^2 = 1$$

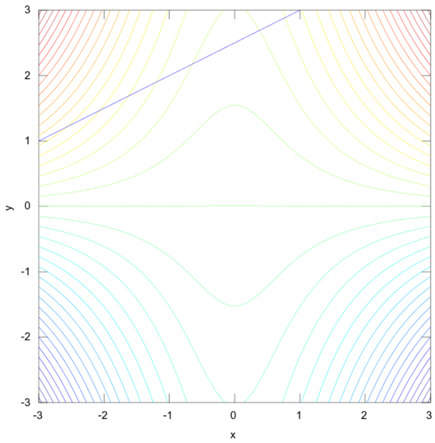
(Source: Wikipedia)



Notes

Saddlepoint example with line constraint

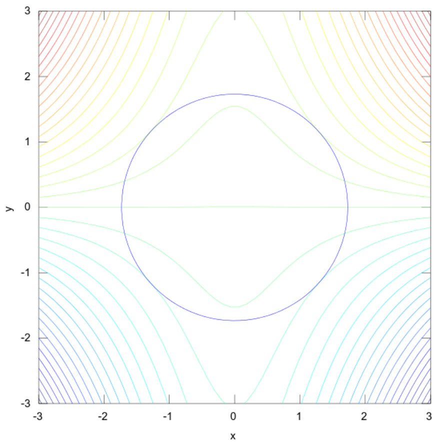
$$f(x,y) = x^2y + x$$
$$\text{s.t.} \quad x - 2y = -5$$



Notes

Saddlepoint example with circle constraint

$$f(x,y) = x^2y + x$$
$$\text{s.t.} \quad x^2 + y^2 = 3$$



Notes

Condition for a constrained minimum

- The gradient of the function is perpendicular to the direction of the constraint.
- The gradient of the function and the gradient of the constraint are parallel:

$$\begin{aligned} \frac{\partial f}{\partial x} &= \lambda \frac{\partial c}{\partial x} \\ \frac{\partial f}{\partial x} &- \lambda \frac{\partial c}{\partial x} = 0 \end{aligned} \tag{4}$$

  where λ is an "arbitrary" scalar.

Definition of *feasibility*:

- Any point x that satisfies the constraints is called a *feasible point*
- If x is a feasible point, any direction z for which $x + \varepsilon z$ is also feasible is called a *feasible direction*.

Notes

First-order optimality conditions: the Lagrangian

If the constraints are satisfied, i.e. $c_i = 0$, then we can add multiples of the constraints to the function to be minimised:

$$L(x,\lambda) = f(x) + \sum_k \lambda_i c_i(x) = f(x) + \lambda^T c$$

This is called the *Lagrangian* of the optimisation problem.

Consider x which satisfies the constraints. We need the constraints also to be satisfied for a change in design variables $x + \delta x$.

$$\begin{aligned} L(x+\delta x,\lambda) &= f(x+\delta x) + \sum_k \lambda_i c_i(x+\delta x) \\ &= f(x) + \frac{\partial f}{\partial x}\delta x + \sum_k \lambda_i (c_i(x) + \frac{\partial c_i}{\partial x}\delta x) + O(\delta x^2) \end{aligned}$$

Notes

First-order constrained optimality condition

The change in Lagrangian is

$$\begin{aligned} \delta L &= L(x+\delta x,\lambda) - L(x,\lambda) \\ &= \frac{\partial f}{\partial x}\delta x + \sum_k \lambda_i \frac{\partial c_i}{\partial x}\delta x + O(x^2) \\ &= \frac{\partial f}{\partial x}\delta x + \lambda^T \frac{\partial c}{\partial x}\delta x + O(\delta x^2) \end{aligned}$$

In the minimum no further reduction is possible,

$$\frac{\partial L(x^*)}{\partial x} = \frac{\partial f(x^*)}{\partial x} + \lambda^T \frac{\partial c(x^*)}{\partial x} = 0,$$

which corresponds to Eq. (4).

Notes

Second-order constrained optimality condition

In the minimum the gradient of the function is perpendicular to the direction of the constraint, so for any *feasible* direction z:

$$z^T \nabla F(x^*) = 0$$

In the minimum the constraint is satisfied and the Hessian along the constraint direction is positive (positive-definite):

$$z^T \nabla^2 F(x^*) z > 0$$

for any feasible direction z.

Notes

Example of constrained minimisation, first-order optimality

$$\min_x F(x) = x_1^2 + 3x_1x_2 \quad \text{s.t.} \quad c_1(x) = x_1 + 5x_2 - 1 = 0$$

Optimalitiy conditions:

$$\begin{aligned} \frac{\partial F}{\partial x_1} - \lambda_1 \frac{\partial c_1}{\partial x_1} &= 2x_1 + 3x_2 - \lambda_1 = 0 \\ \frac{\partial F}{\partial x_2} - \lambda_1 \frac{\partial c_1}{\partial x_2} &= 3x_1 - 5\lambda_1 = 0 \\ c_1 &= x_1 + 5x_2 - 1 = 0 \end{aligned}$$

Leading to the minimum with

$$x_1^* = -\frac{3}{4}, \quad x_2^* = \frac{7}{20}, \quad \lambda_1 = -\frac{9}{20}$$

(Source: Bartholomew-Biggs)

Notes

Lagrange multipliers

The λ_i are called the *Lagrange multipliers*. What is their meaning?

$$\begin{aligned} L(x, \lambda) &= f(x) + \sum_k \lambda_i c_i(x) = f(x) + \lambda^T c \\ \frac{\partial L}{\partial c_i} &= \lambda_i \end{aligned}$$

- the Lagrange multiplier measures the sensitivity of the objective with respect to changes in the constraint.
- It is the rate of change of the objective when the constraint is violated.

Notes

Example for the meaning of Lagrange multipliers

Tank example (Open-topped):

$$\min S(x_1, x_2, x_3) \qquad \text{s.t.} \qquad x_1x_2x_3 = V^* = 20$$

Solution:

$$x_1 = 1.71, x_2 = x_3 = 3.42, \quad S^* = 36.09, \quad \lambda = 1.17.$$

How does the minimal surface S^* change if we modify the target volume to, say, $V^* = 20.5$?

$$\delta S^* = \lambda \delta V^* = 1.17 * 0.5 = 0.585$$

(Source: Bartholomew-Biggs)

Notes

Outline

Notes

Jacobian of the constraints

- A feasible direction z does not violate the constraints:

$$c(x^* + \varepsilon z) = c(x^*) + \varepsilon \frac{\partial c}{\partial x} z = 0$$

- Consider the Jacobian of the constraints $A = \frac{\partial c}{\partial x}$:

$$A = \begin{bmatrix} \frac{\partial c_1}{\partial x_1} & \frac{\partial c_1}{\partial x_2} & \dots & \frac{\partial c_1}{\partial x_N} \\ \frac{\partial c_2}{\partial x_1} & \frac{\partial c_2}{\partial x_2} & \dots & \frac{\partial c_2}{\partial x_N} \\ \vdots & & & \\ \frac{\partial c_M}{\partial x_1} & \frac{\partial c_M}{\partial x_2} & \dots & \frac{\partial c_M}{\partial x_N} \end{bmatrix}$$

- Typically $N > M$, hence the matrix is singular and allows many solutions for feasible directions $Az = 0$.

Notes

Projected gradients method

- The feasible directions z are in the *nullspace* or *kernel* of A.
- We can compute a basis for the nullspace of A and project a search direction s onto the nullspace.
- This removes all components of s that would lead to a violation of the constraints and leaves us with a feasible direction z, tangent to the constraints.
- For non-linear constraints this is only a first-order approximation, we need to add steps normal to the constraint direction to recover feasibility.
- These steps will be in the *range* of A for which $Az' \neq 0$.

Notes

Outline

Notes

Adding the constraint as a penalty

We could add the value of the constraint c with some multiplicative factor $1/r$ to the function:

$$P(x, r) = F(x) + \frac{1}{r}\sum_{i=1}^{M} c_i(x)^2$$

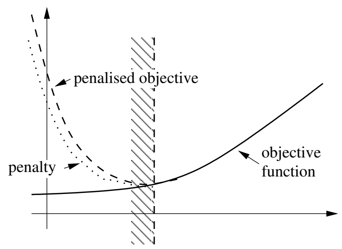
with the *penalty parameter* $r > 0$ and $r \to 0$ as we approach the solution.

- Choosing r to approach zero increases the value of P, so the optimiser chooses a new iterate x that reduces P, hence becomes more feasible.
- At x^*, the constraints are satisfied, so $c_i = 0$ and the penalty term vanishes, the optimiser follows $\nabla F(x)$.
- This is called an *exterior point* method as the penalty becomes non-zero only outside the feasible region, at *infeasible* points.

Notes

Penalty functions: exterior point methods

- Adding a positive multiple of the constraint is an *exterior* penalty function method:
- We start from any point, feasible or not, and iterate toward feasibility by increasing the penalty for constraint violation.
- The iteration may stop before a feasible point is reached.



Notes

Example of penalised objective functions

Example: minimise drag c_D subject to constant lift c_L at $c_{L,target}$.
Penalised cost function P:

$$P = c_D + \frac{1}{r}\left(c_L - c_{L,target}\right)^2$$

Inequality constrained: minimise weight W of a structure, subject to keeping stress σ below maximal stress σ_{max}:

$$P = W + \frac{1}{r}\left(\max(0, |\sigma| - \sigma_{max})\right)^k$$

where k is some positive, even, constant, the larger k, the more rapidly the penalty increases when the constraint is violated.

Notes

Summary of Exterior point methods

Advantages:

- This formulation does not compute Lagrange multipliers.
- We only deal with an unconstrained minimisation of a modified function.

Disadvantages:

- We need to start with large values of r, the constraint $c_i = 0$ will not be closely satisfied.
- for small r the Jacobian and Hessian of P can become very *ill conditioned*, resulting in erratic convergence.
- The c_i need to be near zero when using small values of r.
- Choice of r is not simple, poor choice can lead to lack of convergence.

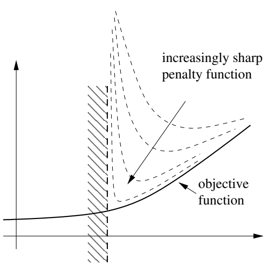
Notes

Barrier methods: for inequality constraints

- Start from a feasible point: $c_i \geq 0$.
- Add a penalty function that penalises when approaching the feasibility boundary, i.e. start to violate the constraint:

$$B(x) = f(x) + r\sum \frac{1}{c_i(x)}$$

$$B(x) = f(x) - r\sum \log(c_i(x)).$$



- Sharpen the penalty function $r \to 0$, making it "steeper", to let the solution approach the feasibility boundary as we converge.
- The current iterate is always feasible! This is called an **interior point method.**

Notes

Summary of Interior point methods

Advantages:

- This formulation does not compute Lagrange multipliers.
- We only deal with an unconstrained minimisation of a modified function.
- Iterates always remain feasible

Disadvantages:

- Needs an algorithm for adaptive control of barrier term
- If barrier function is not 'sharp', optimisation may stop far from the constraint boundary.

Notes

Augmented Lagrangian

Consider the alternative penalty formulation:

$$\begin{aligned} M(x, v, r) &= F(x) + \frac{1}{r} \sum_{i=1}^{M} \left(c_i(x) - \frac{r}{2} v_i \right)^2 \\ &= F(x) + \frac{1}{r} \sum_{i=1}^{M} c_i(x)^2 - \sum_{i=1}^{M} v_i c_i(x) + \frac{r}{4} \sum_{i=1}^{M} v_i^2 \end{aligned}$$

- The function M is called the *Augmented Lagrangian*.
- If v approaches the Lagrange multipliers λ, we recover the first-order constrained optimality conditions.
- As we explicitly consider feasibility through the Lagrangian, we need only make r "sufficiently small", but not near zero, which results in better conditioning.

Notes

Outline

Notes

Approximating the function with a quadratic

Similar to Newton methods, we can approximate the function F as a quadratic and the constraints as linear:

$$\min_x \frac{1}{2} x^T G x + h^T x + c \qquad \text{s.t.} \qquad Ax + b = 0$$

The Hessian G, the Jacobian of the constraints A and the vectors h, b are considered constant.
First-order optimality is then

$$\begin{aligned} Ax^* + b &= 0 \\ Gx^* + h - A^T \lambda^* &= 0. \end{aligned}$$

Which is equivalent to:

$$\begin{pmatrix} G & -A^T \\ -A & 0 \end{pmatrix} \begin{pmatrix} x^* \\ \lambda^* \end{pmatrix} = \begin{pmatrix} -h \\ b \end{pmatrix}$$

Notes

Equality-constrained quadratic programming

$$\begin{pmatrix} G & -A^T \\ -A & 0 \end{pmatrix} \begin{pmatrix} x^* \\ \lambda^* \end{pmatrix} = \begin{pmatrix} -h \\ b \end{pmatrix} \tag{5}$$

Eq. (5) is called an *equality-constrained quadratic programming problem*, EQP.

- The system matrix has zeros in the lower right diagonal block, hence the system matrix is not positive-definite.
- This means that standard iterative methods to solve this system will fail.
- A number of alternative methods to find solutions for Eq. (5) have been devised.
- But the equations in this formulation remain very difficult to be solved. This is an ongoing field of research.

Notes

Outline

Notes

Principle of Sequential Quadratic Programming I

Sequential quadratic programming (SQP) is widely accepted as the most powerful method to solve constrained optimisation problems. Different variants are used.

- Express an improvement in the first-order optimality condition moving from x to x^* by $\delta x = x^* - x$:

$$\frac{\partial F}{\partial x}(x+\delta x) - \sum_{i=1}^{M} \lambda^* \frac{\partial c_i(x+\delta x)}{\partial x} = 0$$
$$c_i(x+\delta x) = 0 \qquad \text{for} \qquad i = 1, \cdots, M$$

- Using Taylor expansion with $g = \frac{\partial F}{\partial x}, G = \frac{\partial^2 f}{\partial x^2}, \nabla = \frac{\partial \cdot}{\partial x}$, we find:

$$g(x) + G(x)\delta x - \sum_{i=1}^{M} \lambda^* \left[\nabla c_i(x) + \nabla^2 c_i(x)\delta x\right] = 0$$
$$c_i(x+\delta x) = 0 \qquad \text{for} \qquad i = 1, \cdots, M$$

Notes

Principle of Sequential Quadratic Programming II

- Defining

$$\hat{G} = G(x) - \lambda^* \nabla^2 c_i(x)$$

- we find the conditions for the step δx to be:

$$\begin{aligned} \hat{G}\delta x - A^T \lambda^* &= -g \\ -A\delta x &= c \end{aligned}$$

- This is equivalent to solving the following EQP for δx:

$$\min_x \hat{F}(x) = \frac{1}{2}(\delta x^T \hat{G} \delta x) + g^T \delta x$$
$$\text{s.t.} \quad c + A\delta x = 0$$

Notes

SQP Algorithms

$$\min_x \hat{F}(x) = \frac{1}{2}(\delta x^T \hat{G} \delta x) + g^T \delta x \tag{6}$$
$$\text{s.t.} \quad c + A\delta x = 0 \tag{7}$$

- Although the constraint is considered linear in (7), the Hessian of the constraint is included in (6).
- We can either ensure improvement in feasibility by including conditions on c_i in the line search (Wilson-Han-Powell SQP)
- or include a penalty function in the Augmented Lagrangian to ensure feasibility.
- The penalty variant has better convergence properties.

Notes

Outline

Notes

Summary of constrained optimisation

- Projected gradient methods are very effective where constraints are not too non-linear.
- Penalty and barrier methods are simplest to implement, need careful adjustment of penalty parameters to achieve optimality.
- The Augmented Lagrangian reduces the stiffness of the penalty problem by including approximations to the Lagrange multipliers.
- The SQP algorithm is the state of the art, considers a quadratic model for the function.
- SQP leads to stiff saddle-point problem, many variants exist which may be better/worse for a particular problem.

Notes

Organisation of the lectures

1. Univariate optimisation
   - Bisection, Steepest Descent, Newton's method
2. Multivariate optimisation
   - Steepest descent, Newton's method
   - and line-search methods: Wolfe and Armijo conditions,
   - Quasi-Newton methods,
3. Constrained Optimisation:
   - Projected gradient methods,
   - Penalty methods, exterior and interior point methods,
   - SQP
4. Adjoint methods
   - Reversing time, Automatic Differentiation
   - Adjoint CFD codes
5. Gradient computation
   - Manual derivation, Finite Differences
   - Algorithmic and automatic differentiation, fwd and bkwd.

Notes

Exercises on multi-variate, constrained optimisation

Consider the provided `multivar_opt.m` code for the Rosenbrock function f.
Consider the constraint $x \leq 0.5$.

1. Add an exterior type penalty $c = \max(0, x - 0.5)$ to f in the form $f' = f + \epsilon(c)^2$. Find a suitable value of ϵ.
2. Add a barrier type penalty $c = \min(0, 0.5 - x)$ to f in the form $f' = f + \frac{\epsilon}{c}$. Find a suitable value of ϵ.
3. Use a starting value of $x = 0.5, y = -2$, and modify the computed gradient to implement a Projected Gradient Method.

Assess the performance of the 3 methods.

Notes

Notes

Notes