Programming, numerics and optimization

Lecture C-1:
Basics of optimization (in structural engineering), sensitivity analysis

Łukasz Jankowski
ljank@ippt.pan.pl

Institute of Fundamental Technological Research
Room 4.32, Phone +22.8261281 ext. 428

April 27, 2021[1]

[1]Current version is available at http://info.ippt.pan.pl/~ljank.

Outline

Outline

1. Optimization problems
   - Optimization problem
   - Optimization in structural engineering
   - Prerequisites for optimization

Optimization problem

Optimization problem = objective function + domain

- *Objective function* $F : \mathbf{D} \to \mathbb{R}$, which is the function to be minimized or maximized
- *Domain* $\mathbf{D}$ of the objective function arguments
  - *Unconstrained*, $\mathbf{D} = \mathbb{R}^n$
  - *Constrained* with
    - equality constraints, $\mathbf{D} = \{\mathbf{x} \in \mathbb{R}^n | g_i(\mathbf{x}) = 0, i = 1, \ldots, m\}$
    - inequality constraints, $\mathbf{D} = \{\mathbf{x} \in \mathbb{R}^n | h_i(\mathbf{x}) \leq 0, i = 1, \ldots, m\}$
    - mixed type constraints,

  where g_i and h_i are called *constraint functions*.

Optimization problem

Find $\hat{\mathbf{x}} \in \mathbf{D}$ such that

- $\forall_{\mathbf{x} \in \mathbf{D}} F(\mathbf{x}) \geq F(\hat{\mathbf{x}})$ (minimization problem) or
- $\forall_{\mathbf{x} \in \mathbf{D}} F(\mathbf{x}) \leq F(\hat{\mathbf{x}})$ (maximization problem)

Optimization problem
Potential difficulties

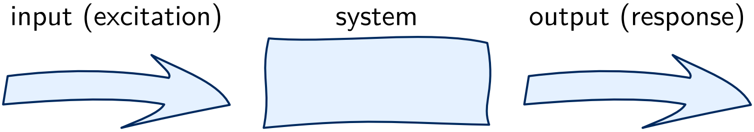
Objective function F

- several local extrema
- only F available, no sensitivity information (gradient, Hessian)
- F is a result of an experiment or simulation
  - no direct analytic formulation
  - long computation or measurement time
  - data contaminated with numerical or measurement errors
- extreme sensitivity
  - of $F(\mathbf{x})$ in the neighborhood of $\hat{\mathbf{x}}$
  - of $\hat{\mathbf{x}}$ with respect to external parameters

Domain $\mathbf{D}$:

- complex constraints defining the boundary of $\mathbf{D}$
- complex nature of $\mathbf{D}$, which can be an infinite-dimensional space of functions or a large discrete set

Optimization in structural engineering



Common optimization problems in structural engineering:

1. system identification (model updating)
2. system optimization/design
3. system control
4. input identification
5. sensor/actuator placement

Optimization in structural engineering

1. *System identification (model updating)*: Input and output known (measured), system characteristics unknown.
2. *System optimization/design*: Change the system so that it
   - responds (in some sense) optimally to a given set of inputs,
   - satisfies certain requirements (like minimum mass).
3. *System control*: Given system input (unknown or known in advance) and measured output, make the output satisfy certain requirements (e.g. follow the reference output) by
   - input forces (*active control*) or by
   - affecting selected characteristics of the system, such as stiffness, damping or yield stress (*semi-active control*)
4. *Input identification*: Identify (online or off-line) the input, given the output and system characteristics.
5. *Sensor/actuator placement*: Determine the "best" placement of sensors or actuators with respect to a given task (response measurement, input identification, damage identification, optimum control, etc.).

Prerequisites for optimization

Prerequisites for optimization

1. optimization problem
   - objective function
   - domain: variables and constraints
2. a method for sensitivity analysis (gradients and, if possible, Hessians of the objective function) — unless a zero-order method is used
3. optimization algorithm

Outline

2. Objective function
   - Typical objective functions
   - Local and global minima
   - Convex objective functions
   - Multicriteria optimization

Objective function

Objective function must reflect some computable (or measurable) system characteristics. It is often expressed in terms

- of the response
  - in the integral form (e.g. the mean-square acceleration)

$$F(\mathbf{x}) = \int_0^T h(t, \mathbf{u}_\mathbf{x}, \dot{\mathbf{u}}_\mathbf{x}, \ddot{\mathbf{u}}_\mathbf{x})\,\mathrm{d}t$$

  - as the extremum (of a function of) system response

$$F(\mathbf{x}) = \max_{0 \le t \le T} h(t, \mathbf{u}_\mathbf{x}, \dot{\mathbf{u}}_\mathbf{x}, \ddot{\mathbf{u}}_\mathbf{x})$$

- or of the structure itself: mass, volume, dimensions, number of elements, etc.

Typical objective functions

Examples

- System identification, input identification: fit between simulated and measured system characteristics (static or dynamic responses, eigenfrequencies, modal shapes/MAC, etc.),

$$F(\mathbf{x}) = \sum_{n \in A} \int_0^T \left[\ddot{u}_n - \ddot{u}_n^M\right]^2 \mathrm{d}t$$

  or

$$F(\mathbf{x}) = \sum_i \left[\frac{\omega_i - \omega_i^{\mathrm{M}}}{\omega_i^{\mathrm{M}}}\right]^2 .$$

- *System optimization*: e.g., total mass of the structure,

$$F(\mathbf{x}) = \sum_\alpha l_\alpha A_\alpha \rho_\alpha .$$

  for a truss structure.

Typical objective functions

Examples

- System optimization and control:
    - mean square displacements or accelerations,

$$F(\mathbf{x}) = \sum_{n\in A} \int_0^T \ddot{u}_n^2 \mathrm{d}t,$$

    - maximum displacements or accelerations,

$$F(\mathbf{x}) = \max_{n\in A} \max_{0\leq t\leq T} \ddot{u}_n^2.$$

Typical objective functions

Examples

- Sensor/actuator placement:
  - measures related to observability/controllability of system/output
    - information content of the measured data
    - conditioning/sensitivity of the unknowns being identified with respect to the measurements
    - conditioning/sensitivity of the output with respect to the control variables
    - maximum force used for optimum control, etc.
  - number of sensors/actuators

Local and global minima

Minimization problem (finding a global minimum)

Find $\hat{\mathbf{x}} \in \mathbf{D}$ such that $\forall_{\mathbf{x} \in \mathbf{D}}\, F(\mathbf{x}) \geq F(\hat{\mathbf{x}})$.

An element $\hat{\mathbf{x}}$ is a *global minimum* of F, if there is no other $\mathbf{x} \in \mathbf{D}$ that has a smaller value of $F(\mathbf{x})$.

Local minimum

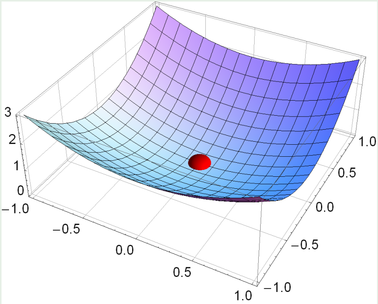
An element $\tilde{\mathbf{x}}$ is a *local minimum* of F, if there is no other $\mathbf{x}$ *in a neighborhood of* $\tilde{\mathbf{x}}$ that has a smaller value of $F(\mathbf{x})$, that is if

$$\exists_{\epsilon>0}\, \forall_{\mathbf{x} \in \mathbf{D}}\, \left[\|\mathbf{x} - \tilde{\mathbf{x}}\| < \epsilon \right] \Rightarrow \left[F(\tilde{\mathbf{x}}) \leq F(\mathbf{x}) \right].$$

Local and global minima

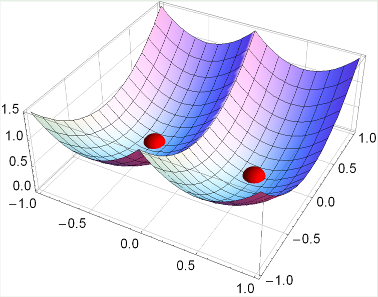
- A global minimum is also a local minimum.
- A function can have multiple local and multiple global minima.

$F(x, y) = x^2 + 2y^2$



Unique global minimum

$F(x, y) = (|x| - 0.5)^2 + y^2$

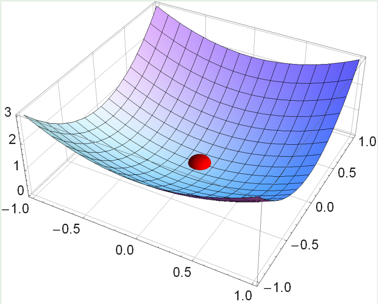


Multiple global minima

Local and global minima

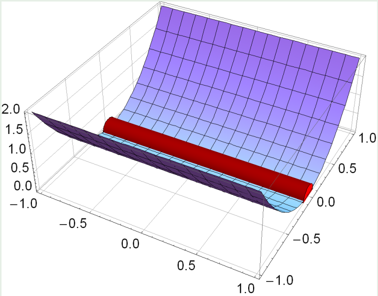
- A global minimum is also a local minimum.
- A function can have multiple local and multiple global minima.

$F(x, y) = x^2 + 2y^2$



Unique global minimum

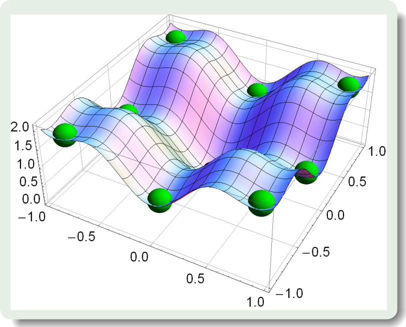
$F(x, y) = 2y^2$

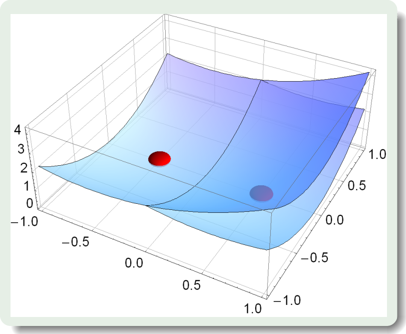


Multiple global minima

Local and global minima

- A function can have multiple local and multiple global minima.
- In general, there is no method to check if a given local minimum is also global[2].





[2] unless the objective function and the domain are both convex.

Convex objective functions

Convex function

A function $F : \mathbf{D} \to \mathbb{R}$, where $\mathbf{D} \subseteq \mathbb{R}^n$, is called *convex*, if

- The domain $\mathbf{D}$ is a convex set, that is if the line segment joining any two points in $\mathbf{D}$ lies completely in $\mathbf{D}$,

$$\forall_{\mathbf{x},\mathbf{y}\in\mathbf{D}}\,\forall_{\alpha\in[0,1]}\,(1-\alpha)\mathbf{x}+\alpha\mathbf{y}\in\mathbf{D}$$

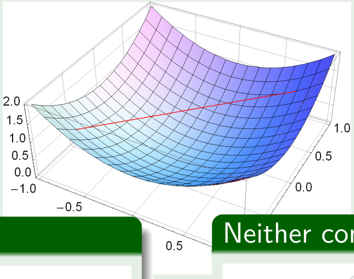
- and if

$$\forall_{\mathbf{x},\mathbf{y}\in\mathbf{D}}\,\forall_{\alpha\in[0,1]}\,F\left((1-\alpha)\mathbf{x}+\alpha\mathbf{y}\right)\leq(1-\alpha)F\left(\mathbf{x}\right)+\alpha F\left(\mathbf{y}\right). \tag{$\star$}$$

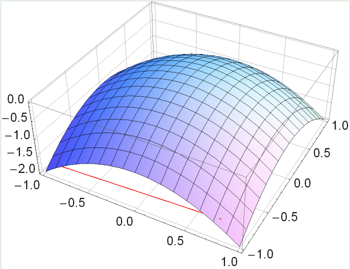
A *concave* function is defined in the same way, with the "$\geq$" sign in $(\star)$.

Convex objective functions

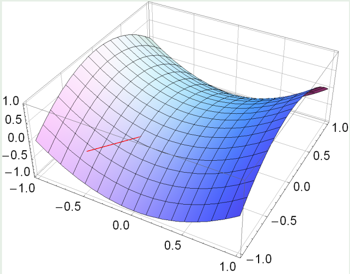
Convex



Concave



Neither convex nor concave



Convex objective functions

Convex (concave) objective functions have a very useful property, which makes them easier to optimize:

- Any local minimum of a convex objective function is also its global minimum;
- Any local maximum of a concave objective function is also its global maximum,

Unfortunately, not many practical objective functions in structural engineering are convex or concave.

The probably most important convex (concave) function is the quadratic form $\mathbf{x}^T\mathbf{A}\mathbf{x}$ with a positive (negative) (semi)definite matrix $\mathbf{A}$.

Multicriteria optimization

Not rarely there are multiple objective functions, which we would wish to optimize simultaneously. For example, we may wish

- a structure to have a low mass and high stiffness or
- minimize both deformations and accelerations.

The scalar concept of optimality does not apply directly to the case of multiple objective functions, since usually no point is optimum with respect to all objective functions simultaneously.

A trade-off between the functions is necessary.

Multicriteria optimization

In general, there are three ways to proceed:

1. Choose the most important objective function and impose limits on the others (make them constraints), for example
   - minimize the mass while keeping the compliance below a given level,
   - minimize the accelerations and keep the deformations below a given level.
2. Combine the functions in a single objective function, e.g.

$$F(\mathbf{x}) = \sum_i \alpha_i F_i(\mathbf{x}) \qquad \text{or} \qquad F(\mathbf{x}) = \sum_i \alpha_i F_i^2(\mathbf{x}).$$

3. The systematic approach of *Edgeworth-Pareto optimization*.

Multicriteria optimization

Edgeworth-Pareto optimization

Edgeworth-Pareto optimal point

Consider a minimization problem with multiple objective functions. A vector $\hat{\mathbf{x}} \in \mathbf{D}$ is called an *Edgeworth-Pareto optimal point* (non-dominated point, efficient point), if for all $\mathbf{x} \in \mathbf{D}$

$$\forall_i \, F_i(\mathbf{x}) = F_i(\hat{\mathbf{x}}) \qquad \text{or} \qquad \exists_i \, F_i(\mathbf{x}) \geq F_i(\hat{\mathbf{x}})$$

(that is if there is no other $\mathbf{x} \in \mathbf{D}$ with at least one objective function smaller and others equal).

In general, there are many Edgeworth-Pareto optimal points, which all lie on a *Edgeworth-Pareto curve* (plane, hyperplane).

In practice, a particular solution is chosen from this curve by combining all objective functions in a single compound function and optimizing it. Different combinations yield different points.

Multicriteria optimization

Edgeworth-Pareto optimization — simple example

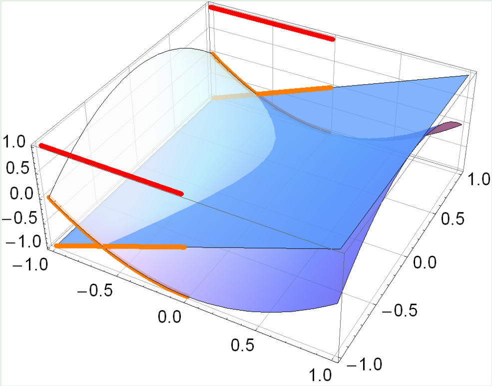
Objective functions

$$F_1(x, y) = x^2 - y^2$$
$$F_2(x, y) = x$$

Domain

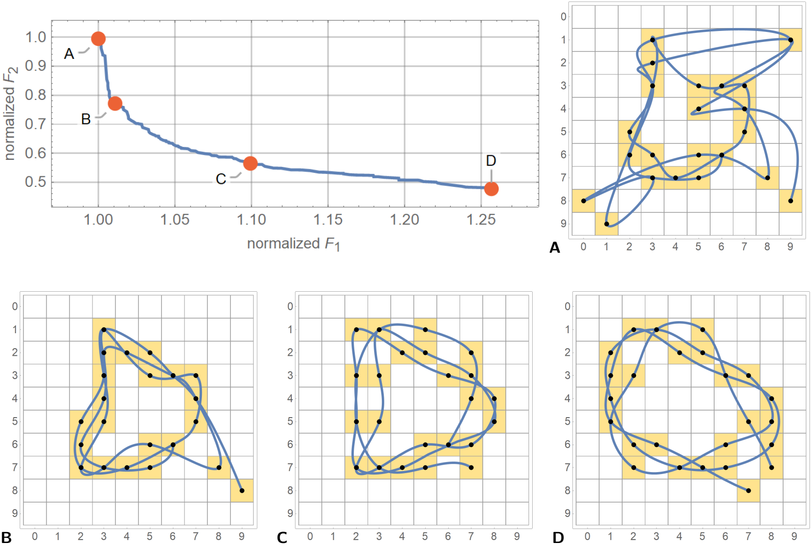
$$x, y \in [-1, 1]$$

Edgeworth-Pareto optimal points



Multicriteria optimization

Edgeworth-Pareto optimization — identification of a moving load trajectory



Multicriteria optimization

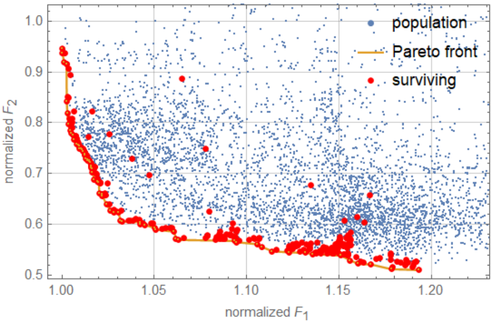
Edgeworth-Pareto optimization — identification of a moving load trajectory

An **evolutionary algorithm** is used

- intense mutation (local randomized modifications of trajectory points)
- no cross-over

The **initial population** is the naive solution based on F_1 only.

The **survival** is governed by the fit function that quantifies the distance to the Pareto front in the F_1–F_2 space.



Outline

3. Variables
   - Types of variables
   - Functions

Types of variables

Variables to be optimized can be

- continuous numbers
  - element cross-sections
  - damping coefficients
  - nodal co-ordinates and masses
  - material properties (stiffness, density, yield stress, etc.)
  - control gains/parameters
- discrete (numbers, subsets, sequences, etc.)
  - number of nodes, elements, sensors, actuators, supports, etc.
  - placement of elements, sensors, actuators, supports in discrete structures
  - sequence of graph vertices, etc.
- functions
  - time history of the optimum control
  - identified system excitation
  - continuous distribution of material properties

Functions

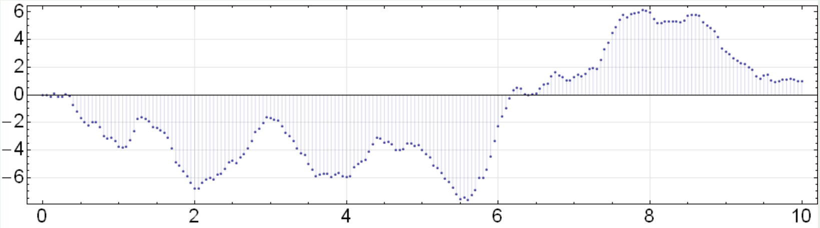
Typical optimization techniques require the variables to be (real, complex, integer) numbers. If they are functions:

- find directly the exact analytical solution, e.g. by the Calculus of Variations, Functional Analysis, etc. (beyond the scope of this lecture)
- reduce to a parametric (finite-dimensional) problem, e.g.
  1. discretize into time steps (finite elements)
  $$F \to [F(t_1), F(t_2), \ldots, F(t_n)]^{\mathrm{T}}$$
  2. use a finite combination of basis functions (harmonics, eigenfunctions, Bessels, etc.)
  $$F(t) \approx \sum_{i=1}^{N} a_i \hat{F}_i(t)$$
  3. In certain cases the function can be (directly or implicitely) implied by the optimization objective.

Functions

Discretization into time steps (finite elements)

Discretization $F \to [F(t_1), F(t_2), \ldots, F(t_n)]^T$



- Time discretization is natural, since ODEs are numerically solved in a finite number of time steps. Space discretization is natural in the context of the FEM.
- A huge number of variables for even simple problems.
- Danger of over-optimized solutions (extremely noisy and meaningless). Often it is better to have a more coarse distribution of variables than of time steps or finite elements.

Functions

Combination of basis function

- A natural way to have a more coarse distribution of variables is to express the unknown function in the form of a finite combination of some basis functions (harmonics, eigenfunctions, splines, etc.)

$$F(t) \approx \sum_{i=1}^{N} a_i \hat{F}_i(t),$$

  where N is much smaller than the number of time steps (finite elements).
- The coefficients a_i become the variables to be optimized.
- Dimensionality of the problem is significantly reduced and F is a linear function of the combination coefficients a_i.
- The choice of the basis and its dimension is not always obvious.

Functions

Implied directly

In certain cases the function can be (directly or implicitly) implied by the optimization objective.

- For example, the problem of optimum control for minimization of acceleration can be often simplified to the problem of immediate (or local) minimization of acceleration. The obtained solution is optimal only locally, but often reasonably close to the globally optimal solution.
- In certain bilinear control problems (e.g., control of damping), the optimum control function is implicitely implied by the optimization goal (via co-state/adjoint variables) in a very simple way:
  - bang-bang control is optimum,
  - (relatively) easy computation of derivatives,
  - (relatively) simple stationarity criterion ($\nabla F = 0$).

Outline

Constraints

In most practical continuous optimization problems, the domain **D** is a proper subset of $\mathbb{R}^n$ and the variables are constrained

- either directly (in terms of the variables)
  - Cross-sections of elements, nodal masses, damping coefficients, Young's moduli, etc. cannot be negative.
  - Hardening coefficients in plasticity cannot be greater than 1.
  - Constant volume in the problem of material redistribution

$$\sum_{\alpha} l_\alpha A_\alpha \rho_\alpha = V.$$

- or indirectly (in terms of the structural response)
  - Maximum stress, $\max_{0 \le t \le T} \sigma_i(t) \le \sigma_{\max}$.
  - Maximum mean square accelerations, $\int_0^T \ddot{x}_i^2 \mathrm{d}t \le C$.

Constraints

Constraints can be either

- natural constraints, like nonnegativity of masses, cross-sections, Young's moduli, or
- design constraints, which limit maximum (or mean-square) allowable stresses, displacements, accelerations, masses, etc.

Design constraints usually greatly influence the optimum solution $\hat{\mathbf{x}}$, so that it is often expected to lie on the boundary of the domain, $\hat{\mathbf{x}} \in \partial\mathbf{D}$.

In general, even if some variables can be in principle treated as unconstrained (like the excitation force to be identified), imposing some form of constraints may have a regularizing (stabilizing) effect on the solution.

Constraints

Constrained problems are usually harder than unconstrained, because the algorithm has to take care to stay within the domain.

In general, constraints are handled either by

1. *Penalty functions*, which transform the problem to an unconstrained problem by adding to the objective function a penalty term,
$$F_{\mathrm{p}}(\mathbf{x}) = F(\mathbf{x}) + \alpha F_{\mathbf{D}}(\mathbf{x}),$$
that "penalizes" the objective function for $\mathbf{x}$ close to, on or outside the boundaries of the domain $\mathbf{D}$.

Constraints

or by

2. *Lagrangian multipliers*, which are used to transform the problem of minimizing $F(\mathbf{x})$ subject to equality constraints $g_i(\mathbf{x}) = 0$ into the unconstrained problem of finding the stationary points of the Lagrangian,

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) = \mathbf{0} \qquad \text{where } L(\mathbf{x}, \lambda) = F(\mathbf{x}) + \sum_i \lambda_i g_i(\mathbf{x}),$$

which encodes both the objective function and the constraints. Generalization of the method of Lagrange multipliers to inequality constraints yields the Karush-Kuhn-Tucker (KKT) conditions.

Outline

5. Sensitivity analysis
   - Gradient and Hessian
   - Finite Difference Approximations (FDM)
   - Direct Differentiation Method (DDM)
   - Adjoint Method
   - Automatic Differentiation (AD)

Sensitivity analysis
Gradient

For continuous variables and sufficiently smooth objective functions, there exists gradient of the objective function, which is the vector of its first derivatives:

$$\nabla F(\mathbf{x}) = \frac{\partial F(\mathbf{x})}{\partial \mathbf{x}} = \left[\frac{\partial F(\mathbf{x})}{\partial x_1}, \frac{\partial F(\mathbf{x})}{\partial x_2}, \ldots, \frac{\partial F(\mathbf{x})}{\partial x_n}\right]^{\mathsf{T}}.$$

Since $-\nabla F(\mathbf{x})$ is the direction of the steepest descent, the gradient is extremely useful in optimization.

However, even if the gradient exists (continuous unknowns, smooth function/constraints), computing it can be numerically costly.

Sensitivity analysis

Gradient of the structural response

Assume the objective function (or a constraint) is of the form $F(\mathbf{u}_\mathbf{x}, \mathbf{x})$, where $\mathbf{u}_\mathbf{x}$ is the response, and depends on the variables $\mathbf{x}$. The gradient of F is then expressible in terms of the gradients of the structural response:

$$\frac{\mathrm{d}F(\mathbf{u}_\mathbf{x}, \mathbf{x})}{\mathrm{d}x_i} = \frac{\partial F}{\partial x_i} + \sum_j \frac{\partial F}{\partial u_j}\frac{\mathrm{d}u_j}{\mathrm{d}x_i},$$

so that

$$\nabla_\mathbf{x} F(\mathbf{u}_\mathbf{x}, \mathbf{x}) = \frac{\partial F}{\partial \mathbf{x}} + \frac{\partial F}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}_\mathbf{x}}{\mathrm{d}\mathbf{x}}.$$

Hence, if the derivatives of the response $\frac{\mathrm{d}\mathbf{u}_\mathbf{x}}{\mathrm{d}\mathbf{x}}$ are already computed, the gradient $\nabla_\mathbf{x} F(\mathbf{u}_\mathbf{x}, \mathbf{x})$ can be computed quickly.

Sensitivity analysis

Gradient of the structural response

For example

$$F(\mathbf{u},\mathbf{x}) = \sum_i \int_0^T \left[\ddot{u}_i - \ddot{u}_i^{\mathrm{M}}\right]^2 \mathrm{d}t.$$

$$\nabla_{\mathbf{x}} F(\mathbf{u},\mathbf{x}) = 2\sum_i \int_0^T \left[\ddot{u}_i - \ddot{u}_i^{\mathrm{M}}\right] \frac{\mathrm{d}\ddot{u}_i}{\mathrm{d}\mathbf{x}} \mathrm{d}t.$$

Sensitivity analysis

The gradient of the response expresses its sensitivity to the variables, hence calculating it is called *sensitivity analysis*. However, sensitivity analysis can be numerically costly, so some methods compute $\nabla_{\mathbf{x}} F(\mathbf{u},\mathbf{x})$ directly and not via $\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{x}}$.

Sensitivity analysis

Gradient of extremum-based objective functions

Objective functions (or constraints), which are based on the extremum value of the response, such as

$$\max_{0 \leq t \leq T} F(\mathbf{u}, \mathbf{x}, t) = F(\mathbf{u}, \mathbf{x}, t_{\mathrm{max}}),$$

have a useful property: when computing the derivative, the location of the extremum t_{max} of F can be assumed to be fixed in time, so that

$$\nabla_{\mathbf{x}} F = \frac{\partial F}{\partial \mathbf{x}} + \frac{\partial F}{\partial \mathbf{u}} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{x}} + \frac{\partial F}{\partial t} \frac{\mathrm{d}t_{\mathrm{max}}}{\mathrm{d}\mathbf{x}} = \frac{\partial F}{\partial \mathbf{x}} + \frac{\partial F}{\partial \mathbf{u}} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{x}},$$

since the third term is always zero:

- if t_{max} is an interior point, then $\left.\frac{\partial F}{\partial t}\right|_{t=t_{\mathrm{max}}} = 0$
- otherwise $t_{\mathrm{max}} \in \{0, T\}$ and $\frac{\mathrm{d}t_{\mathrm{max}}}{\mathrm{d}\mathbf{x}} = \mathbf{0}$.

Sensitivity analysis

Hessian

Hessian of a function is the matrix of its second derivatives,

$$\nabla^2 F(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 F}{\partial x_1 \partial x_1} & \frac{\partial^2 F}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 F}{\partial x_1 \partial x_n} \\ \frac{\partial^2 F}{\partial x_2 \partial x_1} & \frac{\partial^2 F}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 F}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 F}{\partial x_n \partial x_1} & \frac{\partial^2 F}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 F}{\partial x_n \partial x_n} \end{bmatrix}.$$

- Full Hessian has n^2 elements, hence can be used only if the number of variables n is moderate. Otherwise the Hessian can be approximated by a sparse matrix.
- If the exact Hessian is not available (in a reasonable amount of time or memory), it can be approximated using the gradient
  - at few points (BFGS or SR1 algorithms, sparse approximation)
  - or at one point by exploiting special properties of the objective function (Gauss-Newton, Levenberg-Marquardt algorithms).

Methods of sensitivity analysis

In structural optimization, objective functions are rarely given explicitly in the analytical form. At least four basic methods can be considered to compute gradients:

1. Finite Difference Approximations (FDM)
2. Direct Differentiation Method (DDM)
3. Adjoint Method
4. Automatic Differentiation (AD)

Sensitivity analysis

Finite Difference Approximations (FDM)

The derivative of the objective function with respect to the variable x_i can be approximated with the finite difference:

$$\frac{\partial F(\mathbf{x})}{\partial x_i} \approx \frac{F(\mathbf{x} + h\mathbf{e}_i) - F(\mathbf{x})}{h}.$$

Computing the full gradient with respect to all n variables requires $n + 1$ computations of the objective function (one at $\mathbf{x}$ and n at $\mathbf{x} + h\mathbf{e}_i$ for all i), which usually amounts to $n + 1$ full system simulations.

Sensitivity analysis

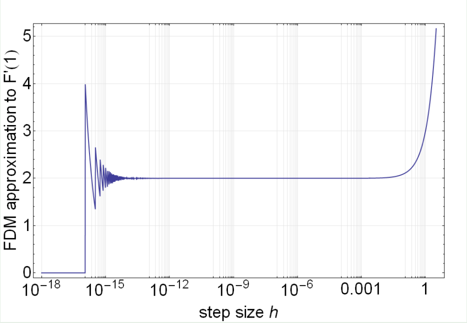
Finite Difference Approximations (FDM)

FDM can suffer from the problem of proper step size:

- If too large, the derivative is not accurate enough.
- If too small, numerical errors occur.

$$F(x) = x^2$$

$$F'(1) \approx \frac{(1+h)^2 - 1}{h}$$



In many problems, there can be no clear gap between the "large" and the "small" step sizes (the "plateau" disappears).

Sensitivity analysis

Direct Differentiation Method (DDM) — statics

The DDM is a method for computing the gradients of the response (sensitivity analysis), which can be further used in computations of the gradient of the objective function by $\nabla_{\mathbf{x}} F(\mathbf{u}_{\mathbf{x}}, \mathbf{x}) = \frac{\partial F}{\partial \mathbf{x}} + \frac{\partial F}{\partial \mathbf{u}} \frac{\mathrm{d}\mathbf{u}_{\mathbf{x}}}{\mathrm{d}\mathbf{x}}$.

The equation of equilibrium,

$$\mathbf{K}\mathbf{u} = \mathbf{f},$$

can be directly differentiated with respect to the variable x,

$$\mathbf{K}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x} = \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}x} - \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}x}\mathbf{u},$$

and solved to obtain the derivative of the response $\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x}$.

A separate solution (but with the same matrix $\mathbf{K}$) is necessary for each variable x.

Sensitivity analysis

Direct Differentiation Method (DDM) — frequency domain

Analysis in the frequency domain leads to the quasi-static formulation, hence the gradients of eigenvetors and eigenvalues can be computed in a similar way. Direct differentiation of

$$(\mathbf{K} - \lambda\mathbf{M})\,\mathbf{u} = \mathbf{0}, \qquad \mathbf{u}^{\mathsf{T}}\mathbf{M}\mathbf{u} = 1$$

with respect to the design variable x yields two equations, which can be combined together to form the following linear system:

$$\begin{bmatrix} \mathbf{K} - \lambda\mathbf{M} & -\mathbf{M}\mathbf{u} \\ -\mathbf{u}^{\mathsf{T}}\mathbf{M} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x} \\ \frac{\mathrm{d}\lambda}{\mathrm{d}x} \end{bmatrix} = \begin{bmatrix} -\left(\frac{\mathrm{d}\mathbf{K}}{\mathrm{d}x} - \lambda\frac{\mathrm{d}\mathbf{M}}{\mathrm{d}x}\right)\mathbf{u} \\ \frac{1}{2}\mathbf{u}^{\mathsf{T}}\frac{\mathrm{d}\mathbf{M}}{\mathrm{d}x}\mathbf{u} \end{bmatrix},$$

which has to be solved separately for each

- variable x and
- eigenpair $(\lambda, \mathbf{u})$.

Sensitivity analysis

Direct Differentiation Method (DDM) — dynamics

The equation of motion,

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f},$$

can be directly differentiated with respect to the variable x,

$$\mathbf{M}\frac{\mathrm{d}\ddot{\mathbf{u}}}{\mathrm{d}x} + \mathbf{C}\frac{\mathrm{d}\dot{\mathbf{u}}}{\mathrm{d}x} + \mathbf{K}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x} = \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}x} - \frac{\mathrm{d}\mathbf{M}}{\mathrm{d}x}\ddot{\mathbf{u}} - \frac{\mathrm{d}\mathbf{C}}{\mathrm{d}x}\dot{\mathbf{u}} - \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}x}\mathbf{u},$$

and integrated numerically using the same integration scheme as the original equation.

A separate analysis of the full structure is necessary for the derivative of response with respect to each variable.

Sensitivity analysis

Adjoint Method

The DDM, as all methods based on the sensitivity of the response, requires one full system simulation *for each design variable* to obtain the sensitivity of the response.

If there are many variables, but only few objective functions (constraints), the Adjoint Method can be quicker, as it requires one full system simulation *for each objective function (constraint)* instead of each variable.

Sensitivity analysis

Adjoint Method — statics

Differentiation of the objective function $F(x, \mathbf{u})$ yields

$$\frac{\mathrm{d}F}{\mathrm{d}x} = \frac{\partial F}{\partial x} + \frac{\partial F}{\partial \mathbf{u}} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x},$$

which includes the sensitivity of the response that should be eliminated. Direct differentiation of the equation of equilibrium,

$$\mathbf{K}\mathbf{u} = \mathbf{f},$$

yields, as in the DDM,

$$\mathbf{K}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x} + \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}x}\mathbf{u} - \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}x} = 0,$$

which can be premultiplied by the vector of adjoint variables $\boldsymbol{\lambda}$ and (as vanishing) added to the derivative of the objective function:

$$\frac{\mathrm{d}F}{\mathrm{d}x} = \frac{\partial F}{\partial x} + \frac{\partial F}{\partial \mathbf{u}} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x} + \boldsymbol{\lambda}^{\mathsf{T}} \left[\mathbf{K}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x} + \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}x}\mathbf{u} - \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}x}\right].$$

Sensitivity analysis
Adjoint Method — statics

The terms with the derivatives of the response can be collected together,

$$\frac{\mathrm{d}F}{\mathrm{d}x} = \frac{\partial F}{\partial x} + \boldsymbol{\lambda}^{\mathsf{T}} \left(\frac{\mathrm{d}\mathbf{K}}{\mathrm{d}x}\mathbf{u} - \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}x} \right) + \left(\frac{\mathrm{d}F}{\mathrm{d}\mathbf{u}} + \boldsymbol{\lambda}^{\mathsf{T}}\mathbf{K} \right) \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x}.$$

The last term can be made vanishing, if $\boldsymbol{\lambda}$ is the solution of

$$\boldsymbol{\lambda}^{\mathsf{T}}\mathbf{K} = -\frac{\mathrm{d}F}{\mathrm{d}\mathbf{u}} \qquad \text{or} \qquad \mathbf{K}\boldsymbol{\lambda} = -\left(\frac{\mathrm{d}F}{\mathrm{d}\mathbf{u}}\right)^{\mathsf{T}},$$

which should be solved once for each objective function (constraint) F. The derivative of the objective function can be obtained for every unknown x by substituting this solution into

$$\frac{\mathrm{d}F}{\mathrm{d}x} = \frac{\partial F}{\partial x} + \boldsymbol{\lambda}^{\mathsf{T}} \left(\frac{\mathrm{d}\mathbf{K}}{\mathrm{d}x}\mathbf{u} - \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}x} \right).$$

Sensitivity analysis

Adjoint Method — dynamics

Let the objective function (constraint) be expressed as

$$F = \int_0^T g(t, x, \mathbf{u})\, \mathrm{d}t,$$

which also includes extremum-based functions by

$$F = h(t_{\mathrm{max}}, x, \mathbf{u}(t_{\mathrm{max}})) = \int_0^T h(t, x, \mathbf{u})\delta(t - t_{\mathrm{max}})\, \mathrm{d}t.$$

Differentiation with respect to the variable x yields

$$\frac{\mathrm{d}F}{\mathrm{d}x} = \int_0^T \left[\frac{\partial g}{\partial x} + \frac{\partial g}{\partial \mathbf{u}} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x}\right] \mathrm{d}t,$$

which still includes the sensitivity of the response that should be eliminated.

Sensitivity analysis
Adjoint Method — dynamics

Direct differentiation of the equation of motion,

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f},$$

yields, as in the DDM,

$$\mathbf{M}\frac{\mathrm{d}\ddot{\mathbf{u}}}{\mathrm{d}x} + \mathbf{C}\frac{\mathrm{d}\dot{\mathbf{u}}}{\mathrm{d}x} + \mathbf{K}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x} + \frac{\mathrm{d}\mathbf{M}}{\mathrm{d}x}\ddot{\mathbf{u}} + \frac{\mathrm{d}\mathbf{C}}{\mathrm{d}x}\dot{\mathbf{u}} + \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}x}\mathbf{u} - \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}x} = 0,$$

which can be premultiplied by the vector of adjoint variables $\boldsymbol{\lambda}$ and (as vanishing) added to the derivative of the objective function:

$$\frac{\mathrm{d}F}{\mathrm{d}x} = \int_0^T \left[\frac{\partial g}{\partial x} + \frac{\partial g}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x} + \boldsymbol{\lambda}^{\mathsf{T}}\left(\mathbf{M}\frac{\mathrm{d}\ddot{\mathbf{u}}}{\mathrm{d}x} + \mathbf{C}\frac{\mathrm{d}\dot{\mathbf{u}}}{\mathrm{d}x} + \mathbf{K}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x} \right.\right.$$
$$\left.\left. + \frac{\mathrm{d}\mathbf{M}}{\mathrm{d}x}\ddot{\mathbf{u}} + \frac{\mathrm{d}\mathbf{C}}{\mathrm{d}x}\dot{\mathbf{u}} + \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}x}\mathbf{u} - \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}x}\right)\right] \mathrm{d}t.$$

Sensitivity analysis

Adjoint Method — dynamics

The terms with the derivatives of the response can be collected together,

$$\frac{\mathrm{d}F}{\mathrm{d}x} = \int_0^T \left[\frac{\partial g}{\partial x} + \boldsymbol{\lambda}^\mathsf{T}\left(\frac{\mathrm{d}\mathbf{M}}{\mathrm{d}x}\ddot{\mathbf{u}} + \frac{\mathrm{d}\mathbf{C}}{\mathrm{d}x}\dot{\mathbf{u}} + \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}x}\mathbf{u} - \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}x}\right)\right]\mathrm{d}t$$
$$+ \int_0^T \left[\left[\frac{\partial g}{\partial \mathbf{u}} + \boldsymbol{\lambda}^\mathsf{T}\mathbf{K}\right]\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x} + \boldsymbol{\lambda}^\mathsf{T}\left(\mathbf{M}\frac{\mathrm{d}\ddot{\mathbf{u}}}{\mathrm{d}x} + \mathbf{C}\frac{\mathrm{d}\dot{\mathbf{u}}}{\mathrm{d}x}\right)\right]\mathrm{d}t,$$

and integrated by parts,

$$\int_0^T \boldsymbol{\lambda}^\mathsf{T}\mathbf{C}\frac{\mathrm{d}\dot{\mathbf{u}}}{\mathrm{d}x}\,\mathrm{d}t = \boldsymbol{\lambda}^\mathsf{T}\mathbf{C}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x}\Big|_0^T - \int_0^T \dot{\boldsymbol{\lambda}}^\mathsf{T}\mathbf{C}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x}\,\mathrm{d}t,$$

$$\int_0^T \boldsymbol{\lambda}^\mathsf{T}\mathbf{M}\frac{\mathrm{d}\ddot{\mathbf{u}}}{\mathrm{d}x}\,\mathrm{d}t = \boldsymbol{\lambda}^\mathsf{T}\mathbf{M}\frac{\mathrm{d}\dot{\mathbf{u}}}{\mathrm{d}x}\Big|_0^T - \dot{\boldsymbol{\lambda}}^\mathsf{T}\mathbf{M}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x}\Big|_0^T + \int_0^T \ddot{\boldsymbol{\lambda}}^\mathsf{T}\mathbf{M}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x}\,\mathrm{d}t.$$

Sensitivity analysis
Adjoint Method — dynamics

Substituting the results back we obtain

$$\frac{\mathrm{d}F}{\mathrm{d}x} = \int_0^T \left[\frac{\partial g}{\partial x} + \boldsymbol{\lambda}^\mathsf{T}\left(\frac{\mathrm{d}\mathbf{M}}{\mathrm{d}x}\ddot{\mathbf{u}} + \frac{\mathrm{d}\mathbf{C}}{\mathrm{d}x}\dot{\mathbf{u}} + \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}x}\mathbf{u} - \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}x}\right)\right]\mathrm{d}t$$
$$+ \int_0^T \left[\frac{\partial g}{\partial \mathbf{u}} + \boldsymbol{\lambda}^\mathsf{T}\mathbf{K} - \dot{\boldsymbol{\lambda}}^\mathsf{T}\mathbf{C} + \ddot{\boldsymbol{\lambda}}^\mathsf{T}\mathbf{M}\right]\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x}\,\mathrm{d}t$$
$$+ \boldsymbol{\lambda}^\mathsf{T}\mathbf{C}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x}\bigg|_0^T + \boldsymbol{\lambda}^\mathsf{T}\mathbf{M}\frac{\mathrm{d}\dot{\mathbf{u}}}{\mathrm{d}x}\bigg|_0^T - \dot{\boldsymbol{\lambda}}^\mathsf{T}\mathbf{M}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}x}\bigg|_0^T.$$

All terms but the first vanish if the initial conditions are zero and

$$\ddot{\boldsymbol{\lambda}}^\mathsf{T}\mathbf{M} - \dot{\boldsymbol{\lambda}}^\mathsf{T}\mathbf{C} + \boldsymbol{\lambda}^\mathsf{T}\mathbf{K} = -\frac{\partial g}{\partial \mathbf{u}}, \qquad \boldsymbol{\lambda}(T) = \dot{\boldsymbol{\lambda}}(T) = \mathbf{0},$$

which can be backward integrated and substituted to the first line to obtain the derivative of F with respect to every x without additional full system simulations.

Sensitivity analysis

Adjoint Method — dynamics

Finally,

$$\frac{\mathrm{d}F}{\mathrm{d}x} = \int_0^T \left[\frac{\partial g}{\partial x} + \boldsymbol{\lambda}^\mathsf{T}\left(\frac{\mathrm{d}\mathbf{M}}{\mathrm{d}x}\ddot{\mathbf{u}} + \frac{\mathrm{d}\mathbf{C}}{\mathrm{d}x}\dot{\mathbf{u}} + \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}x}\mathbf{u} - \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}x}\right)\right]\mathrm{d}t,$$

where the adjoint variables $\boldsymbol{\lambda}$ are obtained via a backward integration of

$$\mathbf{M}\ddot{\boldsymbol{\lambda}} - \mathbf{C}\dot{\boldsymbol{\lambda}} + \mathbf{K}\boldsymbol{\lambda} = -\left(\frac{\partial g}{\partial \mathbf{u}}\right)^\mathsf{T}$$

with the following endpoint conditions

$$\boldsymbol{\lambda}(T) = \dot{\boldsymbol{\lambda}}(T) = \mathbf{0}.$$

Sensitivity analysis

Automatic Differentiation (AD)

Automatic Differentiation

- disregards the physical meaning of the structural response, objective function, constraints etc. in order to just
- break down the algorithm for calculating the response into elementary arithmetic operations, which can be easily differentiated by the chain rule.

For example the Euler formula for integrating ODEs,

$$u_{t+1} = u_t + \Delta t\, f(t, u_t),$$

where $f(t, u) = \dot{u}$, can be mechanically differentiated to obtain

$$\frac{du_{t+1}}{dx} = \frac{du_t}{dx} + \Delta t \frac{\partial f}{\partial u} \frac{du_t}{dx}.$$

Since objective functions in structural optimization are often complicated, Automatic Differentiation is usually used in combination with other methods in hybrid approaches.