

Optimalizační metody v CFD – diferenciální evoluce

Ondřej Suchomel, ing. Tomáš Hyhlík

Abstrakt

Příspěvek popisuje využití jednokriteriální optimalizace pro řešení konkrétního problému Computational Fluid Dynamics (CFD). Použitou optimalizační metodou je diferenciální evoluce.

Klíčová slova

Jednokriteriální optimalizace, CFD, optimalizační metody, evoluční algoritmy, diferenciální evoluce, aplikace diferenciální evoluce v praxi

**Faculty of Mechanical Engineering at CTU
Department of Fluid Mechanics and Thermodynamics**

Optimisation methods in CFD – Differential Evolution

Ondřej Suchomel, ing. Tomáš Hyhlík

Abstract

Article describes use of monocriterial optimisation for solution of concrete problem in Computational Fluid Dynamics (CFD). Differential evolution is used as optimisation method.

Keywords

Monocriterial optimisation, CFD, Optimisation methods, Evolution algorithms, Differential evolution, Application of Differential evolution in practice

1. Úvod

Optimalizace je věc, která je v inženýrské praxi velmi potřebná. Chceme-li zvyšovat úroveň kvality navrženého výrobku, je ve většině případů použití optimalizace k vylepšení vlastností nezbytné.

Optimalizace je nalezení globálního extrému funkce f o více proměnných:

$$f = f(x_k), k = 1, \dots, n, \quad (1)$$

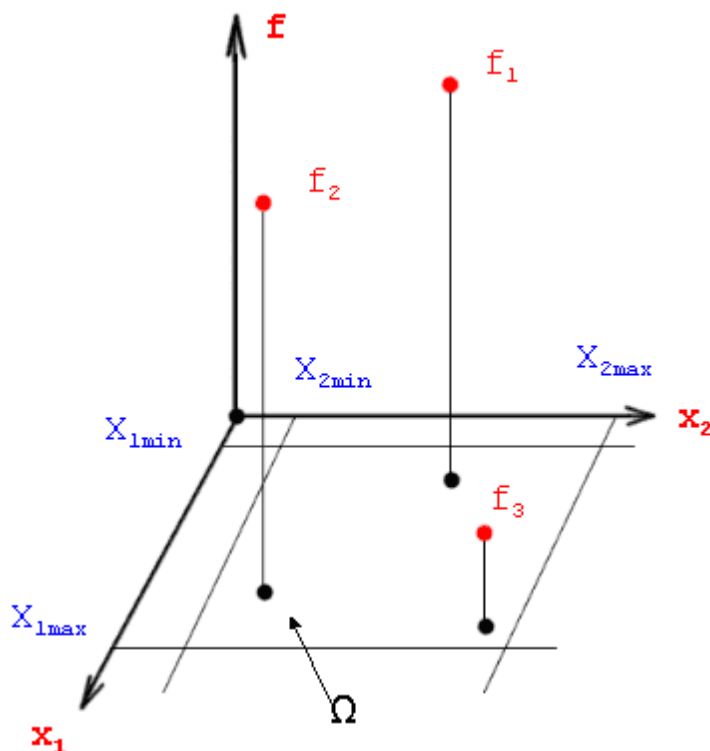
kde n je dimenze problému. Matematickou nutnou podmínkou extrému funkce f je nulová hodnota gradientu funkce f :

$$\frac{\partial f}{\partial x_k} = 0 \quad (2)$$

Inženýrské problémy, se kterými se lze v praxi setkat, mohou být svojí definicí velmi složité. Ve většině případů není možné vyjádřit optimalizovanou funkci analyticky. Toto platí dvojnásob u úloh s rozloženými parametry, které jsou popsány parciálními diferenciálními rovnicemi. Proto je nutné použít pro nalezení globálního extrému vhodnou optimalizační metodu. Optimalizační metodu je možné použít i tehdy, je-li známo analytické vyjádření funkce. To může být velmi složité a provedení parciálních derivací takřka nerealizovatelné. Pak se nabízí použití optimalizační metody ve spojení s počítačem. Lokalizace globálního extrému je pak otázkou několika vteřin výpočtového času.

Je možné optimalizovat více funkcí, které se navzájem vylučují (přítlak – rychlost vozu na rovině). Pak se sestavuje tzv. pareto množina možných řešení, tj. řešení, kdy pro konkrétní hodnotu jedné funkce nabývají ostatní funkce svého globálního extrému. Z pareto množiny se vybere na základě úvahy uživatele konkrétní řešení.

2. Optimalizace



Obr. 2.1 Geometrická interpretace problému

Je zadán optimalizační problém, kde f je funkce 2 proměnných, viz. Obr. 2.1. Optimalizační parametry jsou x_1, x_2 . Tyto proměnné budou výsledkem, tj. řešením optimalizace. Oblast Ω , na které je definována funkce f , je obdélníková, ohraničená hodnotami parametrů $\langle x_{1\min}, x_{1\max} \rangle$, resp. $\langle x_{2\min}, x_{2\max} \rangle$. Konkrétními funkčními hodnotami optimalizované neboli cílové funkce jsou f_1, f_2, f_3 .

Oblast Ω nemusí být obdélníková, její tvar může být naprosto libovolný, např. kružnice, elipsa, trojúhelník či cokoli jiného. Pokud je dán problém, kde $n = 2$, a oblast Ω je křivka, pak se řeší úloha o vázaném extrému. Analogicky i pro více rozměrů.

Podle sestavení optimalizačního problému mohou být optimalizačními parametry x_i různé fyzikální i nefyzikální veličiny: *geometrické* (délky, úhly), *kinematické* (rychlosti, zrychlení), *technologické* (posuv, řezná rychlost), *ekonomické* (cena, náklady na provoz), *termodynamické* (teplota, tlak). Optimalizační parametry nemusí být spojité veličiny, ale mohou nabývat i diskrétních hodnot.

Cílovou funkcí může být rovněž cokoli, např. *celková hmotnost, objem, max. rychlost, čas nakolo, strojní čas, životnost, aerodynamický odpor, vztlak*.

O tvaru cílové funkce f na oblasti Ω většinou neexistují žádné korektní informace. Optimalizační metoda je nástroj, který pomůže extrém nalézt. Termín „nalezení extrému“ není přesný. Místo a hodnota extrému funkce, kterou vyhodnotí optimalizační metoda, je jen určitým přiblížením ke skutečnému extrému. Kde leží skutečný extrém, lze prohlásit jen při analytickém nalezení extrému matematicky zadané funkce. Přesnost, se kterou metoda lokalizuje extrém, je však většinou dostatečná.

2.1 Obecný postup hledání globálních extrémů optimalizační metodou

- 1) Určení prvních několika zkušebních bodů z oblasti Ω
 - 2) Testování cílové funkce f pro vybrané zkušební body
 - 3) Určení nových zkušebních bodů pomocí optimalizační metody
- Poté se cyklus 2. – 3. opakuje až do nalezení extrému cílové funkce f

3. Optimalizační metody

Druhů optimalizačních metod i metod samotných je nespočet. Stejně tak modifikací jednotlivých metod. Každý programátor si může kteroukoli metodu upravit tak, aby vyhovovala jeho představám nebo řešenému problému. Vybranými druhy metod jsou:

- 1) stochastické
- 2) geometrické – simplexová metoda
- 3) evoluční – diferenciální evoluce

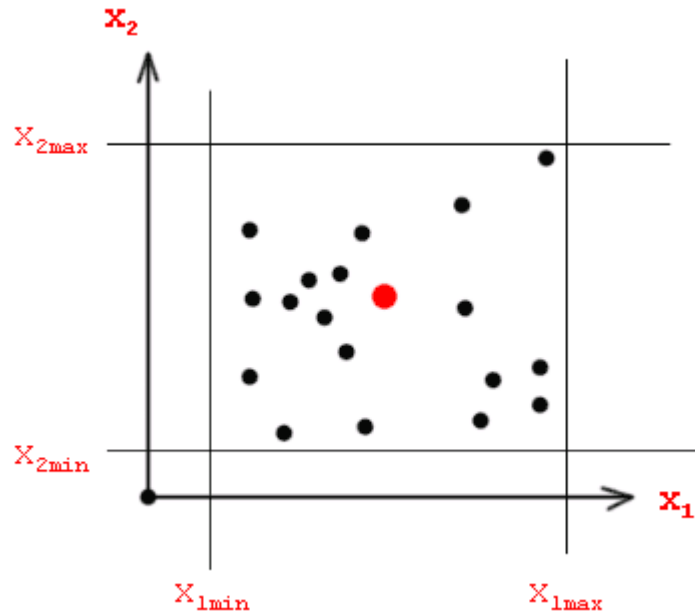
3.1 Stochastické metody

Nejjednodušší metoda stochastická využívá pro nalezení extrému náhody. Generuje dvojice čísel (tj. vektor neboli jedinec, viz. Obr. 3.1) z oblasti Ω , ve kterých se testuje cílová funkce f . Její maximum / minimum je poté prohlášeno za výsledek.

$$x_1^i = \text{rand}(x_{1\min}, x_{1\max}) \quad (3)$$

$$x_2^i = \text{rand}(x_{2\min}, x_{2\max}),$$

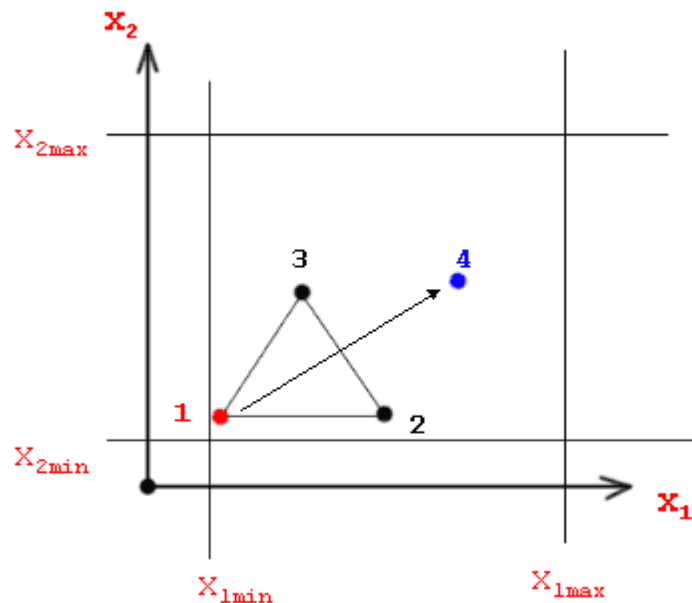
kde i je pořadové číslo jedince. Nevýhodou takovéto prosté metody je její závislost na náhodě a nemožnost cokoli o průběhu optimalizačního procesu předpovídat. Mezi známé stochastické optimalizační metody patří např. metoda Monte Carlo.



Obr. 3.1 Stochastická metoda

3.2 Geometrické metody

Geometrické metody (přesněji metody nultého řádu) aplikují geometrii pro nalezení místa extrému funkce f . Metoda simplexová využívá pro určení nových testovacích bodů (jedinců) jednoduchých geometrických útvarů – simplexů. Ve 2D se jedná o rovnostranný trojúhelník (v 3D o pravidelný čtyřstěn, viz. obr. 3.2).



Obr. 3.2 Simplexová metoda

Určování nově testovaných bodů se řídí se dvěma pravidly:

- 1) Bod s nejhorší hodnotou $f(1)$ se vypustí a nahradí se novým vrcholem, tak aby opět vznikl rovnostranný trojúhelník (4).
- 2) Není dovoleno se vracet do právě vypuštěného bodu.

Simplexová metoda nezohledňuje výhodnost směru, ve kterém se pohybuje. Toho např. využívá metoda Hooka a Jeavese, která při opakovaném postupu v daném směru zvětšuje krok a tím se snaží o urychlení nalezení extrému.

Nevýhodou je podobně jako u ostatních geometrických metody vysoká pravděpodobnost, že se zastaví v lokálním extrému, který nemusí být extrémem globálním.

3.3 Evoluční metody

Evoluční metody jsou velmi mladé optimalizační algoritmy. To souvisí především s nástupem počítačů, jejichž využití je při těchto metodách nevyhnutelné. První metodou tohoto typu je genetické žihání (1994, Price, Storn, USA).

Evoluční metody implementují do optimalizace zákonitosti přírody. Základním pravidlem je:

„Do nové generace se nedostane žádný slabý jedinec.“

Terminologie

jedinec J ... vektor o $n + 1$ složkách

$$J^{i,j} = [x_{k=1}^{i,j}, x_{k=2}^{i,j}, f^{i,j}],$$

populace ... množina NP jedinců

$$i = 1, \dots, NP$$

generace ... j -tá populace

$$j = 0, \dots, G$$

evoluce ... posloupnost G generací

$$k = 1, \dots, n$$

reprodukční cyklus ... algoritmus, který realizuje křížení jedinců a tvorbu další generace

4. Diferenciální evoluce

Diferenciální evoluce je jedním z evolučních algoritmů. Každý jedinec z j -té generace vytváří právě jednoho potomka, který je členem $j+1$ generace.

4.1 Vlastnosti diferenciální evoluce

- velmi jednoduchá na naprogramování, velmi rychlá, i pro více rozměrů
- velmi spolehlivě hledá globální extrém cílové funkce f (na rozdíl od geometrických optimalizačních metod)
- pokud má funkce více globálních extrémů, s velkou pravděpodobností je nalezne
- může pracovat s čísly typu *integer* i s binárními čísly a kombinacemi
- řešením je jedno (popř. více) nejlepší řešení, tj. bod, ve kterém cílová funkce f nabývá globálního extrému na oblasti Ω

Diferenciální evoluci lze použít i pro analyticky přesně definované funkce a hledání jejich globálních extrémů. Pokud je např. provedení parciálních derivací pro analytické určení extrému složité, pak je použití evolučního algoritmu vhodné a velmi rychlé. Testování evolučních algoritmů a ověřování jejich výkonosti probíhá na jednoduchých funkcích (rotační paraboloid), na funkcích s velmi nevýrazným (plochým) extrémem (Rosenbrockova funkce), i na funkcích, kde rozdíl mezi globálním extrémem a lokálními extrémy je velmi malý (Rastriginova funkce).

4.2 Diferenciální evoluce

Následující schéma je jádrem i ostatních evolučních algoritmů :

1. Stanovení parametrů

Před spuštěním evoluce je nutno stanovit parametry, kterými se bude řídit reprodukční cyklus evoluce. Těmito jsou:

- NP $NP > 3$ - počet jedinců, velikost populace
- F $F \in \langle 0,2 \rangle$ - mutační konstanta
- CR $CR \in \langle 0,1 \rangle$ - práh křížení
- G $G > 0$ - počet generací

Při stanovování vhodných parametrů záleží na zkušenosti programátora. Pokud se zvolí $F = 0$ nebo $CR = 1$, pak bude každá další generace přesnou kopií 1. generace. Více viz. (5) pro zkušební vektor.

Diferenciální evoluci lze použít i na stanovení parametrů, pak se mluví o metadiferenciální evoluci.

2. Tvorba populace

Nultá generace je generována zcela náhodně.

$$x_k^{i,0} = x_{k \min} + rand(x_{k \min}, x_{k \max}) \quad \text{for } i = 1, \dots, NP \quad (4)$$
$$k = 1, \dots, n$$

3. Reprodukční cyklus

Pro každého jedince (rodiče, tj. jedince j -té generace) je vygenerován zkušební jedinec zk .

Nejprve jsou vybráni tři různí jedinci téže generace. Rozdíl prvních dvou násobený konstantou F je přičten ke třetímu jedinci. Výsledkem je tzv. šumový vektor v ..

$$v_k^{i,j+1} = x_k^{r3,j} + F(x_k^{r1,j} - x_k^{r2,j}) \quad \text{for } i = 1, \dots, NP \quad (5)$$
$$k = 1, \dots, n$$

Poté je pro každý rozměr generováno náhodné číslo $\langle 0,1 \rangle$. Pokud je číslo menší než práh křížení CR , je zkušebnímu jedinci pro aktuální rozměr přiřazen parametr ze šumového vektoru. Je-li větší, pak parametr rodiče.

$$\text{if}(nahoda < CR) \longrightarrow zk_k^{i,j+1} = v_k^{i,j+1} \quad \text{for } i = 1, \dots, NP \quad (6)$$
$$\text{if}(nahoda \geq CR) \longrightarrow zk_k^{i,j+1} = J_k^{i,j} \quad k = 1, \dots, n$$

Ze vzorce (5) pro šumový vektor je patrný význam parametru F . Čím větší parametr F bude, tím „více jiný“ bude parametr šumového vektoru než parametr jedince č. 3. Parametr CR (práh křížení) určuje pravděpodobnost, se kterou se do zkušebního vektoru dostane parametr šumového vektoru. Důvod názvu diferenciální evoluce je nutno hledat rovněž v prvním vzorci. Název není odvozen od diferenciálního počtu, ale z prostého rozdílu vektorů.

4. Testování cílové funkce f pro zkušební jedince

Pro každého zkušebního jedince je testována cílová funkce f . Nástroj pro testování cílové funkce většinou není součástí evolučního algoritmu. Tímto nástrojem může být Fluent, Matlab, Ansys Excel apod.

5. Určení nové populace

Porovnáním hodnot cílových funkcí rodičů a zkušebních vektorů se určí nová generace jedinců. Pokud je cílová funkce f zkušebního vektoru zk lepší než jedince - rodiče, postoupí do nové generace zkušební vektor. Pokud ne, postoupí beze změny rodič. Tím je zaručeno, že se do nové generace nedostane horší jedinec než rodič.

$$\text{if}(f(zk^{i,j+1}) > f(J_k^{i,j})) \longrightarrow J^{i,j+1} = zk^{i,j+1} \quad \text{for } i = 1, \dots, NP \quad (7)$$

$$\text{if}(f(zk^{i,j+1}) \leq f(J^{i,j+1})) \longrightarrow J^{i,j+1} = J^{i,j}$$

Cyklus 3. – 5. se opakuje do splnění ukončovacího podmínky. Nejlepší jedinec je prohlášen řešením optimalizační úlohy.

Poznámka

Při reprodukčním cyklu se velmi často stává, že nově generovaný zkušební jedinec zk je alespoň jedním parametrem mimo oblast Ω . Obvyklá řešení:

- 1) Zastavení jedince na hranici – pravděpodobnost nalezení lokálního extrému
 - 2) Náhodné generování překročeného parametru, stejně jako při tvorbě prvotní populace.
- Jedná se o standardní typ diferenciální evoluce.

Řešení problému, při němž jedinec alespoň jedním svým parametrem opustí oblast, může být libovolné. Např. navrácení jedince do první třetiny od překročené hranice. V případě č.1 však může nastat situace, kdy se jedinec zastaví v lokálním extrému na hranici oblast Ω a dojde ke zhoršení různorodosti populace. To může mít vliv na správné nalezení extrému. U řešení č.2 je toto nebezpečí nižší.

5. Řešení konkrétního problému

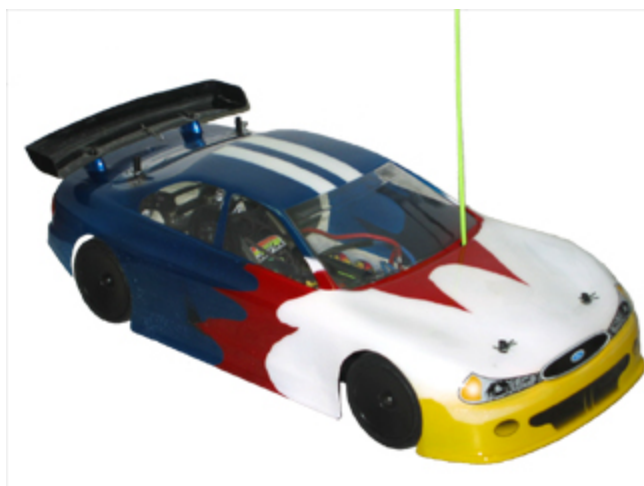
Cílem práce bylo testování využití evolučního optimalizačního algoritmu v aerodynamice za použití CFD softwaru. Tedy odpovědět na otázku, zda je diferenciální evoluce vhodná při řešení aerodynamiky při použití CFD simulací.

5.1 Definice problému

Oblast	- Vnější aerodynamika, užití CFD
Cílová funkce	- Vztlak
Vyšetřovaný objekt	- Zadní přitlačné křídlo modelu závodního vozu
Optimalizační metoda	- Diferenciální evoluce
Úkol	- Nalezení nejméně vhodného umístění a tvaru přitlačného křídla modelu závodního vozu
Program pro realizaci evoluce	- DevC++
Preprocesor	- Gambit
CFD solver	- Fluent

5.2 Modelovaný případ

CFD simulace byla prováděna pro model závodního vozu Ford Mondeo, měřítko 1/10, kategorie E 1/10 Touring Car, viz. Obr. 5.1. Rychlost proudění byla volena tak, aby se blížila rychlosti, které modely této třídy dosahují v rychlých zatáčkách závodních tratí, tj. 10 m.s^{-1} .



Obr. 5.1 Model vozu kategorie E 1/10 Touring cars

5.3 Parametry diferenciální evoluce

Parametry diferenciální evoluce byly voleny dle doporučení z literatury [2]:

- $NP = 4$ - počet jedinců, velikost populace
- $F = 1,23$ - mutační konstanta
- $CR = 0,45$ - práh křížení
- $G = 4$ - počet generací

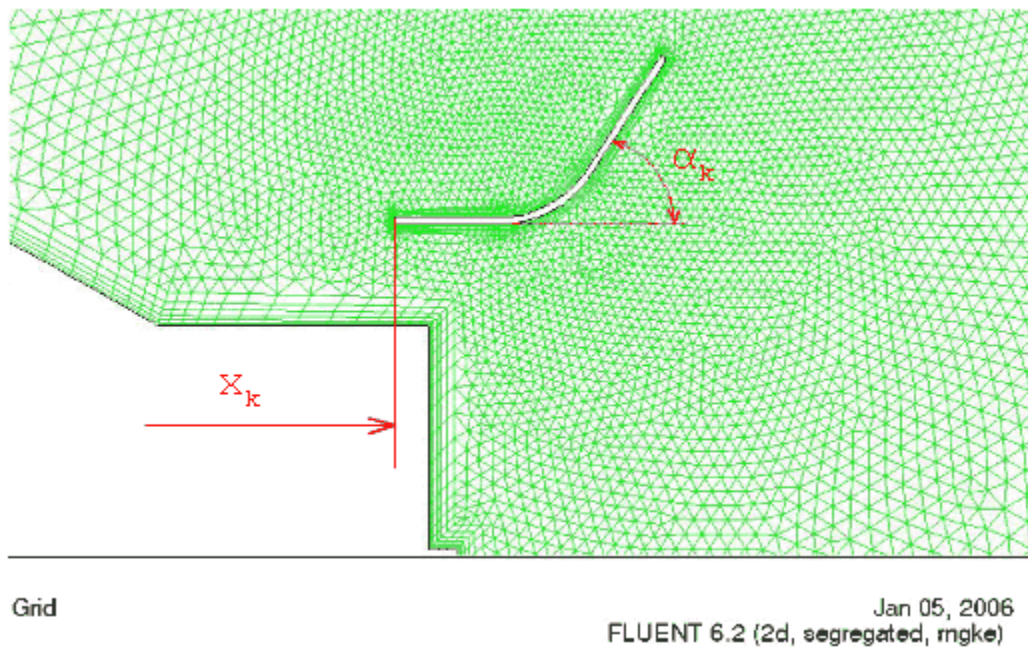
Časová náročnost CFD výpočtů umožnila provést 4 generace diferenciální evoluce. Pro nalezení extrému by však bylo potřeba daleko více výpočtů. Pro představu přesné stanovení extrému funkce rotačního paraboloidu vyžaduje řádově stovky generací o 4 jedincích.

5.4 Optimalizované parametry

Konkrétním objektem, pro který byl hledán globální extrém vzltaku, bylo zadní přítláčné křídlo. Optimalizované parametry byly x -ová poloha křídla ($k = 1$, měřena od přední části vozu) a úhel natočení svislé plochy křídla α_k ($k = 2$, měřeno od osy x , viz. Obr 5.2).

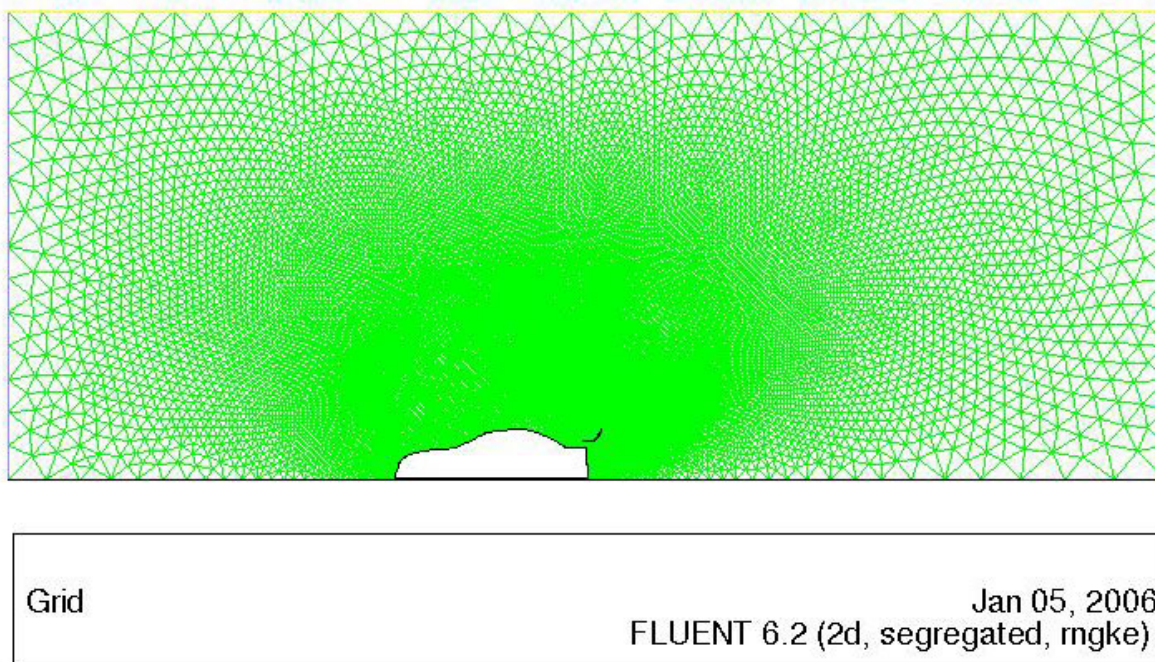
parametr $k = 1$ x -ová pozice křídla, x_k
 $x_k \in < 365, 416 > [\text{mm}]$

parametr $k = 2$ úhel natočení svislé plochy křídla α_k
 $\alpha_k \in < 45, 90 > [^\circ]$



Obr. 5.2 Optimalizované parametry

5.5 CFD výpočet



Obr. 5.3 Výpočtová síť

Výpočtová síť (viz. Obr. 5.3) byla generována automaticky pomocí programu Gambit v textovém režimu. Vstupními daty byly souřadnice jednotlivých bodů přítlačného křídla (Vertex Data).

CFD výpočet probíhal externě ve Fluentu. Výstupem byla hodnota zobecněné y -nové síly F_y^* působící na karoserii vozu, tj. vztlak. Po dokončení CFD výpočtu pro všechny jedince dané generace byla data vložena do programu napsaném v jazyce C, který generoval nové zkušební jedince s vypočtenými souřadnicemi bodů křídla.

Nastavení Fluentu

- Rychlost 10 m.s^{-1}
- Model turbulence $k - \varepsilon$
- Počet iterací 3000
- Schéma Upwind 1. řádu

Průměrný výpočtový čas pro jednu síť při volných výpočtových serverech byl 35 minut.

5.6 Realizace diferenciálních evolucí

Byly provedeny celkem 2 evoluce o 4 generacích:

1. Zastavení jedince na hranici
2. Standard

Při evoluci č.1 byl zkušební jedinec z_k překročivší hranici oblast Ω na této hranici zastaven, protože již nebylo matematicky možné, aby hranici opustil. Jedná se o jednodušší typ evoluce, u kterého hrozí, že lokalizuje lokální extrém.

Při evoluci č.2 bylo programově ošetřeno překročení hranice tím, že jedinec byl vrácen na místo uvnitř oblasti, které bylo zcela náhodně generováno. To mělo za důsledek zachování různorodosti populace nutné pro postupnou lokalizaci globálního extrému.

Nejlepší jedinec v případě 1.) byl znám až v poslední generaci, zatímco v evoluci č.2. již v první. To je dílem náhody, která hraje v diferenciální evoluci, podobně jako v přírodě, významnou roli. Lepší výsledek poskytla evoluce č.1. - právě díky zastavení v extrému. Zda-li byl tento extrém globálním, by poskytlo pokračování v evoluci.

5.7 Výsledky diferenciálních evolucí

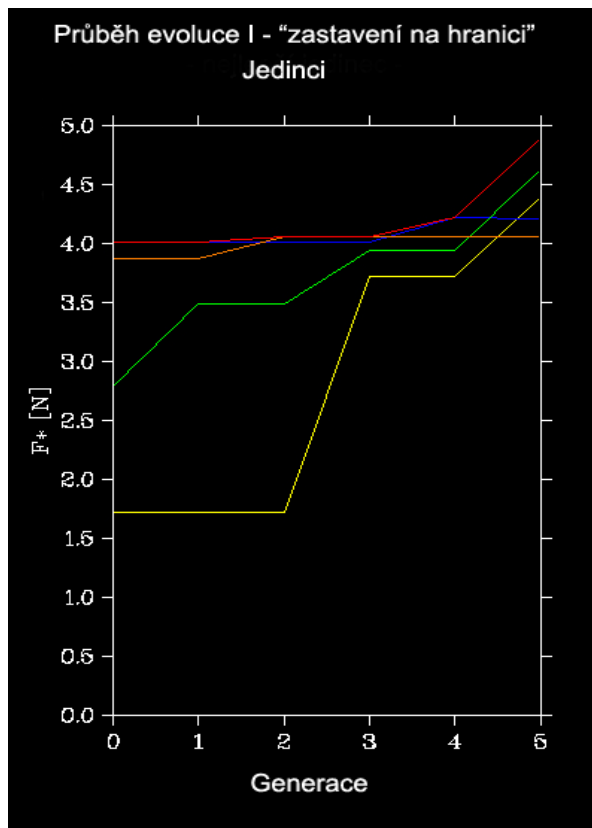
G	X_k [mm]	α_k [mm]	F_Y^* [N]	G	X_k [mm]	α_k [mm]	F_Y^* [N]
0	376.00	45.00	4.01	0	367.00	60.00	3.73
1	376.00	45.00	4.01	1	367.00	60.00	3.73
2	365.00	45.00	4.01	2	367.00	60.00	3.73
3	365.00	45.00	4.06	3	367.00	60.00	3.73
4	365.00	45.00	4.22	4	367.00	60.00	3.73

Tabulka 5.1. Nejlepší jedinec

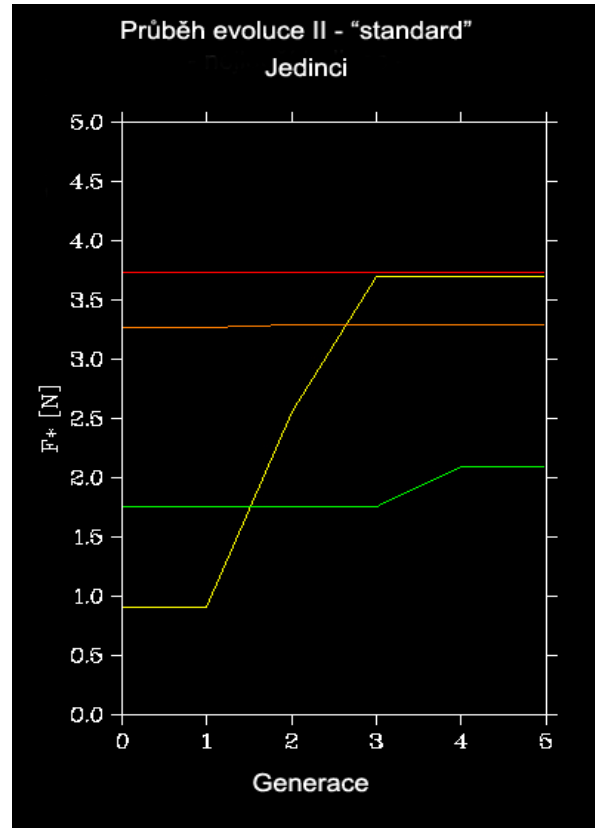
v levé části tabulky jsou výsledky evoluce č.1, v pravé části výsledky evoluce č.2

Z dosažených výsledku(viz. Tabulka 5.1.) je patrné, že nejméně vhodné umístění přitlačného křídla je pro $x_k = 365\text{mm}$ a $\alpha_k = 45^\circ$. To odpovídá realitě, neboť při umístění přitlačného křídla na přední část zavazadlového prostoru leží přitlačné plochy v oblasti odtržení proudu vzduchu, což má za následek zvýšení vztlaku neboli snížení přitlaku. Výsledek optimalizačního problému tedy můžeme považovat za pravdivý.

Průběh obou evolucí je na Obr. 5.4 a 5.5. Vývoj jedinců evoluce č.1 byl oproti evoluci č.2 zajímavější. Populace se neustále vylepšovala a až ve svém posledním kroku generovala nejlepší řešení. Naopak evoluce č.2 byla ve svém vývoji velmi konzervativní. To lze přičíst z části náhodě a z části omezení uvedenému v odstavci 5.6.



Obr. 5.4 Průběh Evoluce č.1



Obr. 5.5 Průběh Evoluce č.2

5.7 Závěr

Díky omezeným možnostem testování cílových funkcí v CFD programu a časové náročnosti výpočtu nebylo možné optimalizaci dovést až do konce. Pro jednoduché matematické funkce (rotační paraboloid) je potřeba stovky až tisíce testování cílové funkce.

Diferenciální evoluce je velmi mocný prostředek pro optimalizaci. Avšak velké procento výpočtů zkušebních jedinců je neúspěšné, tj. není generován lepší jedinec. Data získaná ze špatných jedinců nepřijdou nazmar. Mohou být využita pro studium problému samotného, který je optimalizační metodou řešen. To však se samotnou optimalizací již nespojuje.

Pro časově náročné aerodynamické optimalizační problémy bych doporučil spíše metody, které berou v potaz rovněž výsledky z předchozích výpočtů. Jednou z možností je např. proložení regresní plochy testovanými body, výpočet gradientu a pokračování ve výpočtu ve směru nejvyššího stoupání.

Seznam použité literatury

- [1] P. Lederer: Syntéza a optimalizace mechanických systémů, Fakulta strojní ČVUT, ČVUT, Praha, 1979
- [2] Mařík a kolektiv: Umělá inteligence (4), Academia, Praha, 2003
- [3] S. Kračmar, J. Vogel: Programovací jazyk C, Fakulta strojní ČVUT, ČVUT, Praha, 2002

Domovská stránka

<http://www.suchomel.org/academics/optimalizace>

nejnovější verze článku, prezentace, grady, animace vývoje populace, možnost diskutovat