r21glauert()

Prandtl-Glauert metoda v octave

1. vydání

Ing. Martin Prokš
e-mail: martin.proks@proks-martin.cz

12. března 2021

r21glauert(): Prandtl-Glauert metoda v octave

1. vydání 12. března 2021
Copyright © 2021 Ing. Martin Prokš

Ing. Martin Prokš
Autor projektu.
e-mail: martin.proks@proks-martin.cz
Česká republika

Obsah

Úvod v
1. Instalace a návod k použití 1
2. Vzorové příklady 4
    2.1. Obdélníkové křídlo 5
    2.2. Obdélníkové křídlo, aerodynamicky proměnné 6
    2.3. Lichoběžníkové křídlo 7
    2.4. Eliptické křídlo 8
    2.5. Obdélníkové křídlo a křidélka 10
    2.6. Obdélníkové křídlo a vztlakové klapky 12
3. Poznámky a omezení metody 14
    3.1. Potenciální proudění - velké úhly 14
    3.2. Přímé křídlo - 1/4 čára 15
    3.3. Štíhlost křídla 16
    3.4. Spojitá geometrie - křidélka a klapky 16
    3.5. Počet řezů - přesnost řešení 18
    3.6. Lichý vs. sudý počet řezů 18
4. Teorie Prandtl-Glauertova řešení 19
5. Co dál, rozjímání nad teorií 23
6. Závěr 25
Literatura 26

Seznam obrázků

1. Geometrický souřadný systém letounu a křídla 3
2. Orientace úhlů profilu a křídla 3
3. Rozložení vztlaku - obdélníkové nezkroucené křídlo 5
4. Rozložení vztlaku - obdélníkové křídlo, aer. proměnné 7
5. Rozložení vztlaku - lichoběžníkové křídlo 8
6. Eliptické křídlo a náhradní tvar 9
7. Rozložení vztlaku - eliptické křídlo 10
8. Rozložení vztlaku - obdélníkové křídlo a křidélka 11
9. Rozložení vztlaku - obdélníkové křídlo a vztl. klapky 13
10. Vztlaková čára profilu - problém linearity 14
11. Rozdíl mezi geometrií a jak si to přebere P-G metoda 15
12. Prandtlova náhrada křídla (zdroj: lit. [aer-broz]) 19
13. Rozdělení křídla na řezy (když m = 9, potom M = 7). 20

Úvod

Toto je dokumentace k octave funkci `r21glauert()` v souboru `r21glauert.m`.

Octave [https://www.gnu.org/software/octave/] je programovací jazyk specializovaný na matematiku. Je to svobodná obdoba programu Matlab [https://www.mathworks.com/products/matlab.html]. Takže `r21glauert()` není celý hotový program, je to pouze funkce určená pro vložení do dalších větších programů psaných v `Octave`.

Funkce `r21glauert()` se vlastně jmenuje RozloženíVztlakuKřídlaGlauert, takže zkráceně „r 21 písmen glauert = r21glauert“. Zcela přesně a správně se jedná o Prandtl-Glauertovo řešení rozložení vztlaku po rozpětí křídla metodou podkovovitých potenciálních vírů - teorie vírové vztlakové čáry křídla.

Metoda je to již značně stará a nedokonalá, je jí již přes 100 let (Ludwig Prandtl: 1875-1953). Přesto má stále svou váhu při vývoji letadel, UAV, leteckých modelů a jejich kontrolách. Je to metoda relativně jednoduchá a rychlá na zadání i výpočet, takže je snadno použitelná i přes všechna omezení která má.

Metoda řeší přímé křídlo v nevazkém nestlačitelném podzvukovém proudění kontinua, popsatelného potenciální teorií zdrojů, propadů a potenciálních vírů. Oproti realitě má tedy významná omezené a je to jen hrubé přiblížení. Více o vlastnostech a omezeních metody v následujících kapitolách.

Řešení rozložení vztlaku (přesněji součinitele vztlaku) po rozpětí křídla lze provést několik různými metodami. Zde je jejich (neúplný) výčet od nejjednodušší po nejsložitější, poněkud letem světem:

- Odhad rozložení a vztlaku. Toto je jen velmi hrubá metoda, vhodná pro první rámcové nastřelení. Musíte být zkušený aerodynamik aby jste udělali realistický odhad, nebo vycházet např. z literatury [usaf-datcom], nebo [aap-roskam], nebo z něčeho podobného.

- Můžete použít tuto metodu zpracovanou např. do octave funkce - proč jinak by jste tuto dokumentaci četli, že. Popis teorie viz např. [aer-broz], nebo [wiki-llt]. Výhodou je, že zadáváte jen v podstatě minimum informací. Půdorysnou geometrii křídla, zkroucení křídla a aerodynamické charakteristiky profilů křídla ve význačných bodech.

- Můžete požít nějakou vyšší panelovou metodu. Ať již AVL [http://web.mit.edu/drela/Public/web/avl/], CMARC [http://www.aerologic.com/cmarc.html], VSAERO [https://www.aeroattack.com/vsaero] a další [https://en.wikipedia.org/wiki/Aerodynamic_potential-flow_code]. Zde již musíte definovat komplexní geometrii ve 3D, okrajové podmínky a pečlivě hlídat kvalitu a smysluplnost řešení.

- Můžete použít nějakou z metod konečných objemů - CFD. Ať již OpenFOAM, nebo Star-CCM +, nebo Ansys Fluent, nebo cokoli jiného. Zde se již ale jedná o extrémně komplexní záležitost a předpokládám, že pokud disponujete znalostmi, financemi a lidskými zdroji na zvládnutí něčeho takového, tak se nebudete zahazovat s něčím tak primitivním jako je přes sto let stará teorie vztlakové čáry křídla.

1. Instalace a návod k použití

Funkce `r21glauert()` je napsaná v souboru `r21glauert.m`. Tento soubor obsahuje celou funkci pro `Octave`, včetně nápovědy a vzorového použití.

Instalace je dle standardů `Octave`. Buďto tento soubor nahrajte do cest .m souborů které prohledává `Octave`, nebo si tento soubor dejte do pracovního adresáře s Vaším projektem, kde chcete tuto funkci používat.

Že se podařilo funkci vložit do `Octave` správně lze nejsnáze otestovat tak, že spustíte `Octave` a do příkazové řádky zadáte: `help r21glauert`. Pokud je vše v pořádku, měla by se zobrazit nápověda pro tuto funkci.

```
-- [vysl, errcode, errmsg] = r21glauert (geo, aer, m)
Rozložení vztlaku křídla po rozpětí metodou prandtl-glauert.

Viz 'Lifting-line theory' na vikipedii, nebo
'Aerodynamika nízkých rychlostí' od prof. Brože.

Poznámka: název je zkratkou 'RozloženíVztlakuKřídlaGlauert',
neboli 'r 21 písmen glauert'.

...
```

Vstupní data `geo`, `aer` a výstupní `vysl` jsou octave struktury. Vstupní proměnná `m` je celé číslo. Proměnná `errvarname` je číslo chyby - 0 = výpočet proběhl bez chyby. Proměnná `errmsg` je text případné chyby.

Vstupní proměnné a struktury:

- `geo` - Geometrie. Tato struktura obsahuje údaje o geometrii křídla:
  - `geo.S` - Referenční plocha křídla. Jednotky [m^2].
  - `geo.l` - Referenční rozpětí křídla. Jednotky [m].
  - `geo.z(:)` - Pole souřadnic po rozpětí křídla, kde jsou definovány geometrické údaje. Nula leží v rovině symetrie trupu, pravá polovina křídla má záporné hodnoty `z`, levá polovina křídla má kladné hodnoty `z`. Jednotky [m].
  - `geo.b(:)` - Pole lokálních hloubek křídla v místech řezů `geo.z`. Jednotky [m].
  - `geo.zkrouceni(:)` - Pole lokálních geometrických zkroucení v místech řezů `geo.z`. Kladné zkorucení znamená náběžná hrana nahoru, odtoková hrana dolů. Jednotky [rad].

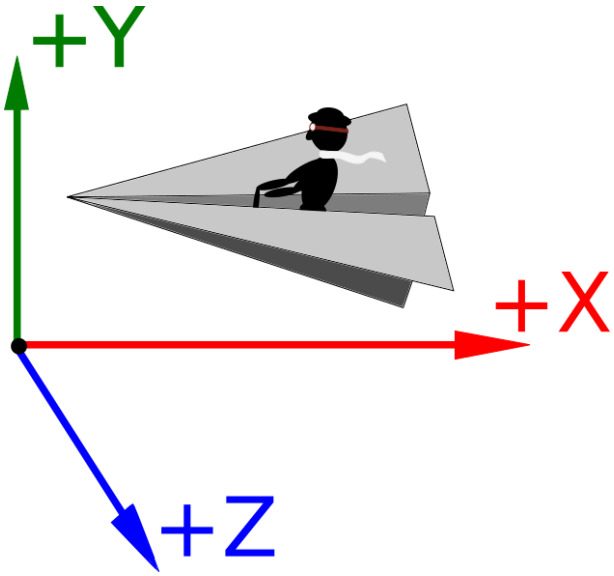
- `aer` - Aerodynamika. Tato struktura obsahuje aerodynamické údaje o křídle:
  - `aer.alfa` - Úhel nastavení celého křídla. Kladná hodnota znamená náběžná hrana nahoru, odtoková dolů. Jednotky [rad].
  - `aer.z(:)` - Pole souřadnic po rozpětí křídla, kde jsou definovány aerodynamické údaje. Počátek a orientace směrů stejná jako u geometrických souřadnic. Jednotky [m].
  - `aer.alfa0(:)` - Pole nulových úhlů náběhu profilu v místech řezů `aer.z`. Jednotky [rad].
  - `aer.CLalfa(:)` - Pole sklonů vztlakových čar profilu v místech řezů `aer.z`. Jednotky [1/rad].
- `m` - Celé číslo. Počet míst (řezů) ve kterých bude rozložení vztlaku vyhodnocováno. Toto číslo je včetně konců křídel. V podstatě se jedná o rozměr pole výstupních dat. `m` musí být více než 8 a je doporučeno liché číslo. Použitelné maximum bývá kolem 101 řezů, obvykle doporučené optimum mezi 41-61 řezy. Nadměrný počet řezů nezlepšuje přesnost, ale paradoxně vede k horší řešitelnosti a větší chybě. Diskuse o vlivu počtu řezů viz další kapitoly.

Výstupní proměnné a struktury:

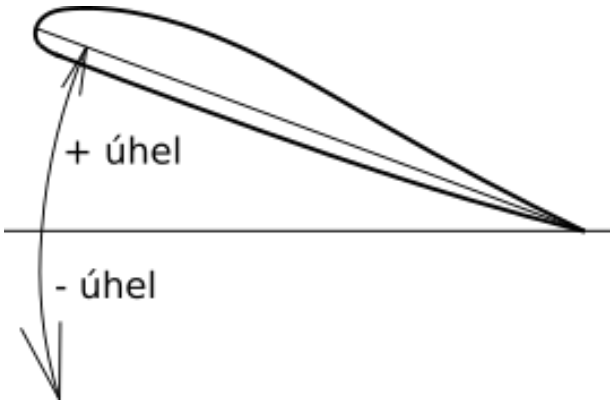
- `vys` - Struktura s výsledky výpočtů:
  - `vys.CL` - Výsledný součinitel vztlaku křídla. Jednotky [1].
  - `vys.CDi` - Výsledný indukovaný odpor křídla. Jednotky [1].
  - `vys.delta` - Glauertův opravný součinitel indukovaného odporu křídla. `CDi = (CL^2 / (3.14.. * ŠtíhlostKřídla)) * (1 + delta)` Jednotky [1].
  - `vys.z(:)` - Pole řezů kde jsou definované výsledky. Pole má rozměr `m`. Jednotky [m].
  - `vys.CL_fz(:)` - Pole lokálních součinitelů vztlaku, podle `z`. Jednotky [1].
  - `vys.CLb_fz(:)` - Pole lokálních součinitelů vztlaku krát lokální hloubka,podle `z`. Jednotky [1.m].
  - `vys.alfai_fz(:)` - Pole lokálních indukovaných úhlů náběhu, podle `z`. Rozměr [rad].
- `errcode` - Návratová hodnota z výpočtu. Celé číslo. Hodnota 0 znamená že vše proběhlo v pořádku. Hodnota 1 značí varování, výpočet proběhl ale je třeba věnovat zvýšenou pozornost výsledkům. Hodnota 2 znamená závažnou chybu při výpočtu, výpočet neproběhl.
- `errmsg` - Chybové a varovné hlášení blíže popisující co se stalo. Řetězec.

Poznámka: všechny délkové rozměry lze změnit z [m] na jakoukoli jinou délkovou jednotku, ale tato jednotka musí být dodržena konzistentně napříč všemi proměnými. Takže vše v [mm] a [mm^2], nebo vše v [cm] a [cm^2], nebo vše v [dm] a [dm^2], nebo vše ve [ft] a [ft^2]. Prostě konzistentně.

Doporučená jednotka je ale [m] aby to bylo v základní SI soustavě a předcházelo se zbytečným chybám při převodech a přepočtech jednotek.



Obrázek 1. Geometrický souřadný systém letounu a křídla



Obrázek 2. Orientace úhlů profilu a křídla

2. Vzorové příklady

Použití funkce v octave je teoreticky jednoduché. Připravíte vstupní data, zavoláte funkci a předáte ji vstupní data. Po skončení běhu funkce se vám vrátí výstupní data, která zpracujete. Hlavně to zpracování dat může vypadat různě, dle vkusu a individuálních potřeb uživatele.

Vzorový příklad jednoduchého použití:

```
## Zadání geometrie a aerodynamiky křídla
g.S = 10;                           #    [m^2]
g.l = 8.929;                        #      [m]
g.z =         [ -4.464, 4.464 ];    #      [m]
g.b =         [  1.120, 1.120 ];    #      [m]
g.zkrouceni = [  0.0,   0];         # [stupně]
g.zkrouceni = deg2rad(g.zkrouceni);# [rad]
a.alfa = deg2rad(3.0);              #    [rad]
a.z =      [ -4.464, 4.464 ];       #      [m]
a.alfa0 =  [ -5.125, -5.125 ];      # [stupně]
a.alfa0 = deg2rad(a.alfa0);         #    [rad]
a.CLalfa = [  6.12,  6.12 ];        #  [1/rad]

## Výpočet - volání výpočetní funkce
[ v, ecd, emsg ] = r21glauert (g, a, 51);

## Zobrazení výsledků
printf ("Error code = %d\n", ecd);
printf ("Error msg  = %s\n", emsg);
printf ("\n");
if (ecd < 2) # výsledky jsou k dispozici když ecd == 0 nebo 1
  printf ("Úhel náběhu kř. = %f deg\n", rad2deg(a.alfa));
  printf ("CL křídla       = %f /\n", v.CL);
  printf ("CDi křídla      = %f /\n", v.CDi);
  printf ("Souč. delta     = %f /\n", v.delta);
  ##
  plot (v.z, v.CL_fz);
  xlabel ("Rozpětí křídla [m]");
  ylabel ("CL [1]");
  title ("Rozložení součinitele vztlaku po rozpětí křídla");
endif
```

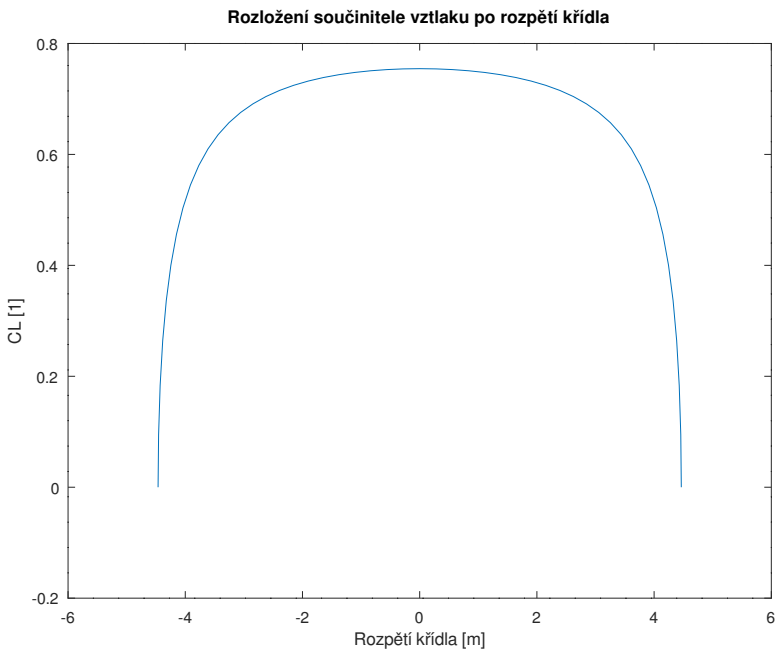
2.1. Obdélníkové křídlo

Obdélníkové křídlo, nijak nezkroucené, stejný profil po celém rozpětí. Rozpětí křídla 8,929 m, hloubka křídla = 1,12 m je konstantní po celém rozpětí křídla. Aerodynamický profil konstantní po celém rozpětí křídla, tedy od jednoho konce křídla až po druhý konec.

```
## Zadání geometrie a aerodynamiky křídla
g.S = 10;                          #    [m^2]
g.l = 8.929;                       #      [m]
g.z =         [ -4.464, 4.464 ]; #      [m] - obdélníkové křídlo
g.b =         [  1.120, 1.120 ]; #      [m] - obdélníkové křídlo
g.zkrouceni = [  0.0,   0];      #    [rad] - nijak nezkroucené
##
a.alfa = deg2rad(3.0);             #    [rad] - úhel náběhu křídla
a.z =      [ -4.464, 4.464 ];     #      [m] - aerodynamické řezy
a.alfa0 =  [ -5.125, -5.125 ];    # [stupně] - aer. nezkroucené
a.alfa0 = deg2rad(a.alfa0);        #    [rad] - převod na radiány
a.CLalfa = [  6.12,  6.12 ];      #  [1/rad]
```

Spuštění výpočtu a zpracování výsledků viz úvodní příklad v kapitole “2. *Vzorové příklady*” .

Očekávaný výsledek běhu programu:



Obrázek 3. Rozložení vztlaku - obdélníkové nezkroucené křídlo

```
Error code = 0
Error msg  = OK vše v pořádku

Úhel náběhu kř. = 3.000000 deg
CL křídla       = 0.671268 /
CDi křídla      = 0.019242 /
Souč. delta     = 0.069339 /
```

2.2. Obdélníkové křídlo, aerodynamicky proměnné

Obdobný příklad, ale profil na konci křídla se liší od profilu v rovině symetrie letounu. Na koncích křídla je profil co má vyšší křivost než profil v rovině symetrie (na koncích křídla jsou profily s vyšší zápornou hodnotou alfa0).

```
## Zadání geometrie a aerodynamiky křídla
g.S = 10;                          #    [m^2]
g.l = 8.929;                       #      [m]
g.z =         [ -4.464, 4.464 ]; #      [m]
g.b =         [  1.120, 1.120 ]; #      [m]
g.zkrouceni = [  0.0,   0];      #    [rad]
##
a.alfa = deg2rad(3.0);                #    [rad]
a.z =      [ -4.464, 0, 4.464 ];     #      [m]
a.alfa0 =  [ -5.125, -4.0, -5.125 ]; # [stupně]
a.alfa0 = deg2rad(a.alfa0);           #    [rad]
a.CLalfa = [  6.12, 6.12,  6.12 ];   #  [1/rad]
```

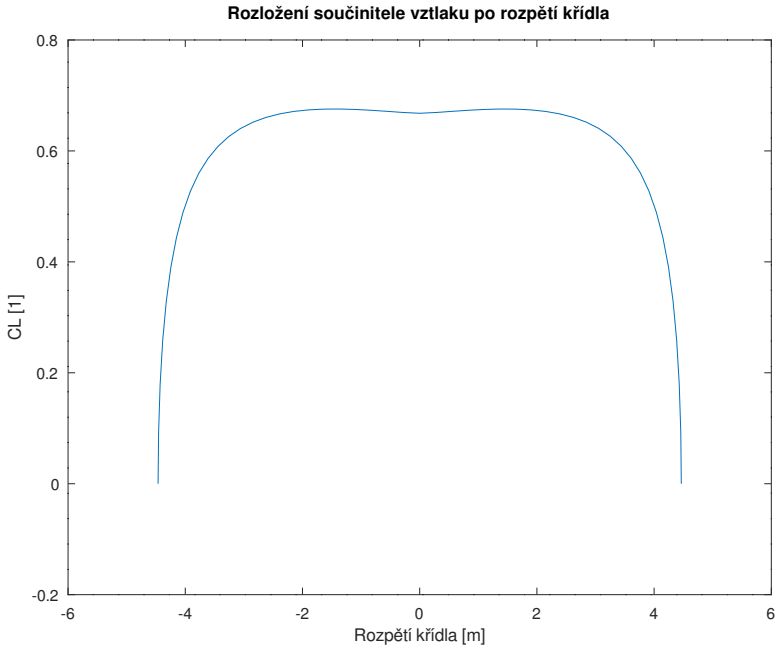
Spuštění výpočtu a zpracování výsledků viz úvodní příklad v kapitole “2. *Vzorové příklady*” .

Očekávaný výsledek běhu programu:

```
Error code = 0
Error msg  = OK vše v pořádku

Úhel náběhu kř. = 3.000000 deg
CL křídla       = 0.620973 /
CDi křídla      = 0.016927 /
```

```
Souč. delta      = 0.099254 /
```



Obrázek 4. Rozložení vztlaku - obdélníkové křídlo, aer. proměnné

2.3. Lichoběžníkové křídlo

Geometricky lichoběžníkové křídlo, zkroucené o -0,6 stupně na koncích křídla oproti rovině symetrie. Aerodynamicky proměnné též, na koncích jiný (tenčí) profil než v rovině symetrie, ale podobná křivost stř. čáry.

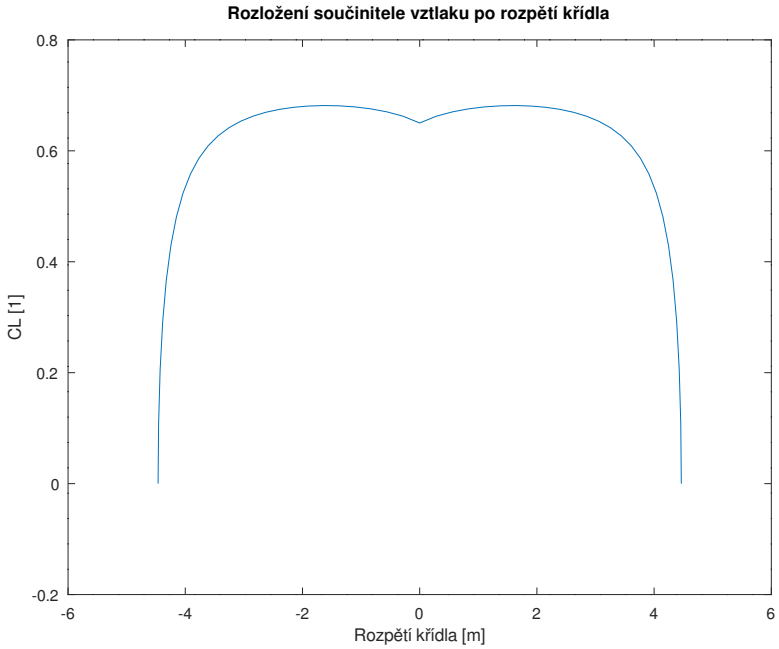
```
## Zadání geometrie a aerodynamiky křídla
g.S = 9.6;                                # [m^2]
g.l = 8.929;                              #   [m]
g.z =         [ -4.464, 0.0,   4.464 ]; #   [m]
g.b =         [  0.800, 1.350, 0.800 ]; #   [m]
g.zkrouceni = [  -0.6, 0.0, -0.6];      # [st.]
g.zkrouceni = deg2rad(g.zkrouceni);     # [rad]
##
a.alfa = deg2rad(3.0);             #    [rad]
a.z =      [ -4.464, 0, 4.464 ];   #      [m]
a.alfa0 =  [ -4.8, -4.9, -4.8 ];   # [stupně]
a.alfa0 = deg2rad(a.alfa0);        #    [rad]
a.CLalfa = [  6.12, 6.00,  6.12 ]; #  [1/rad]
```

Spuštění výpočtu a zpracování výsledků viz úvodní příklad v kapitole "2. *Vzorové příklady*" .

Očekávaný výsledek běhu programu:

```
Error code = 0
Error msg  = OK vše v pořádku

Úhel náběhu kř. = 3.000000 deg
CL křídla       = 0.640249 /
CDi křídla      = 0.016042 /
Souč. delta     = 0.020805 /
```



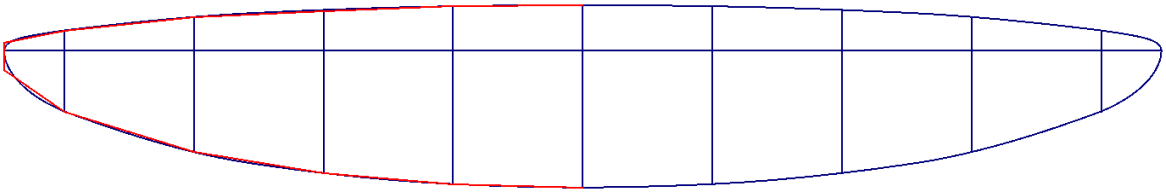
Obrázek 5. Rozložení vztlaku - lichoběžníkové křídlo

2.4. Eliptické křídlo

Geometricky eliptické křídlo, geometricky nezkroucené. Aerodynamicky nezkroucené (stejný profil po celém rozpětí).

Eliptické křídlo je křivková záležitost. Nicméně není nutné jej definovat přehnaně přesně, obvykle stačí 5 panelů na polorozpětí - 10 panelů po celém rozpětí, tedy 11 řezů. Také je dobré rezignovat na nulovou

hloubku na konci. Viz ilustrativní obrázek - modře ano je to eliptické křídlo, ale srovnané podle 1/4 čáry; červeně náhradní tvar pro zadání pro výpočet.



Obrázek 6. Eliptické křídlo a náhradní tvar

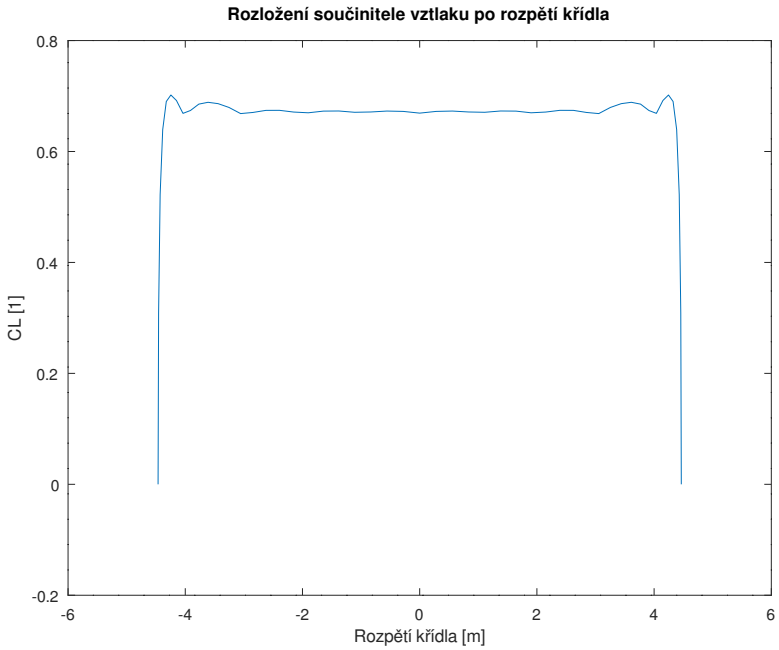
```
## Zadání geometrie a aerodynamiky křídla
g.S = 9.54;                    # [m^2]
g.l = 8.928;                   #   [m]
g.z = [ -4.464, -4.000, -3.000, -2.000, -1.000, ...
                              0.0, ...
         1.000,  2.000,  3.000,  4.000,  4.464 ]; # [m]
g.b = [  0.209,  0.622,  1.037,  1.252,  1.364, ...
                              1.400 ...
         1.364,  1.252,  1.037,  0.622,  0.209];  # [m]
g.zkrouceni = [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]; #[rad]
##
a.alfa = deg2rad(3.0);         #    [rad]
a.z =      [ -4.464, 4.464 ]; #      [m]
a.alfa0 =  [ -4.8, -4.8 ];     # [stupně]
a.alfa0 = deg2rad(a.alfa0);    #    [rad]
a.CLalfa = [  6.12, 6.12 ];    #  [1/rad]
```

Spuštění výpočtu a zpracování výsledků viz úvodní příklad v kapitole “2. *Vzorové příklady*” .

Očekávaný výsledek běhu programu:

```
Error code = 1
Error msg  =  Varování: Křídlo, jeho geometrie a počet panelů
vedou na špatně řešitelné rovnice. Zkontrolujte si přesnost
výsledků prosím.

Úhel náběhu kř. = 3.000000 deg
CL křídla       = 0.685193 /
CDi křídla      = 0.017893 /
Souč. delta     = 0.000381 /
```



Obrázek 7. Rozložení vztlaku - eliptické křídlo

Dle teorie by rozložení součinitele vztlaku eliptického křídla mělo být konstantní po celém rozpětí. Ty vlnovky na konci jsou vliv nedokonalé náhrady zadávání geometrie křídla. Při pečlivější náhradě by se výsledek více blížil teorii.

2.5. Obdélníkové křídlo a křidélka

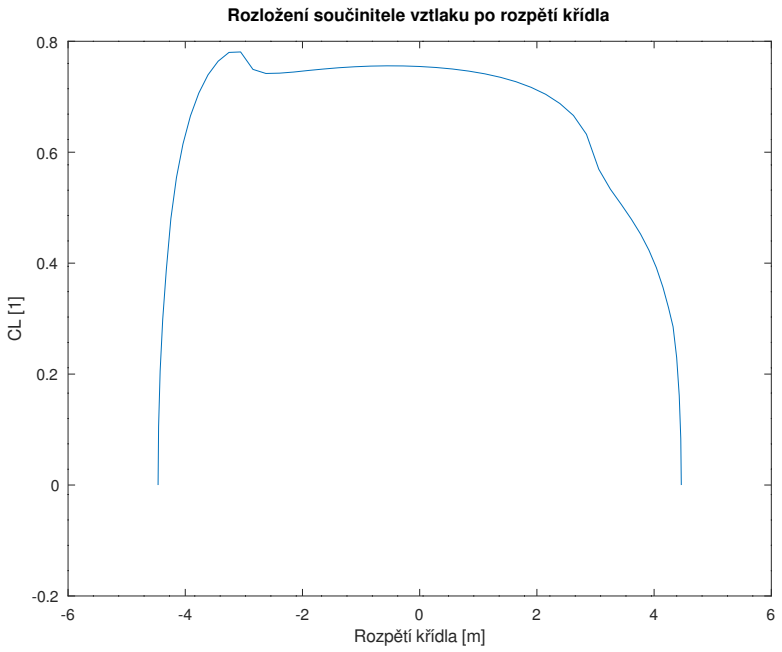
Geometricky pro jednoduchost obdélníkové křídlo, ale s vychýlenými křidélky. Křidélka se projeví změnou aerodynamických charakteristik profilů - změna úhlu nulového vztlaku. Samozřejmě musíte znát (nebo alespoň odhadnout) efekt výchylky, hloubky a Re křidélka na aerodynamické hodnoty - úhel nulového vztlaku profilu a sklon vztlakové čáry profilu. V tom může být nápomocna lit. [usaf-datcom], nebo [aap-roskam].

```
## Zadání geometrie a aerodynamiky křídla
g.S = 10;                           #    [m^2]
g.l = 8.929;                        #      [m]
g.z =         [ -4.464, 4.464 ];    #      [m]
g.b =         [  1.120, 1.120 ];    #      [m]
g.zkrouceni = [  0.0,   0];         # [stupně]
g.zkrouceni = deg2rad(g.zkrouceni);# [rad]
##
```

```
a.alfa = deg2rad(3.0);            #    [rad]
a.z =      [ -4.464, -4.310, ... # vnější konec křídla
             -4.300, -3.000, ... # pravé křidélko
             -2.990,  2.990, ... # vnitřní část křídla
              3.000,  4.300, ... # levé křidélko
              4.310,  4.464 ];   # vnější konec kř. [m]
a.alfa0 =  [ -5.125, -5.125, ...
             -7.5,   -7.5, ...   # efekt výchylky dolů
             -5.125, -5.125, ...
             -2.75,  -2.75, ...  # efekt výchylky nahoru
             -5.125, -5.125];    # [stupně]
a.alfa0 = deg2rad(a.alfa0);      #    [rad]
a.CLalfa = [  6.12,  6.12, ...
              6.10,  6.10, ...
              6.12,  6.12, ...
              6.12,  6.12, ...
              6.12,  6.12 ];     #  [1/rad]
```

Spuštění výpočtu a zpracování výsledků viz úvodní příklad v kapitole “2. *Vzorové příklady*” .

Očekávaný výsledek běhu programu:



Obrázek 8. Rozložení vztlaku - obdélníkové křídlo a křidélka

```
Error code = 0
Error msg  = OK vše v pořádku

Úhel náběhu kř. = 3.000000 deg
CL křídla       = 0.671032 /
CDi křídla      = 0.019924 /
Souč. delta     = 0.108047 /
```

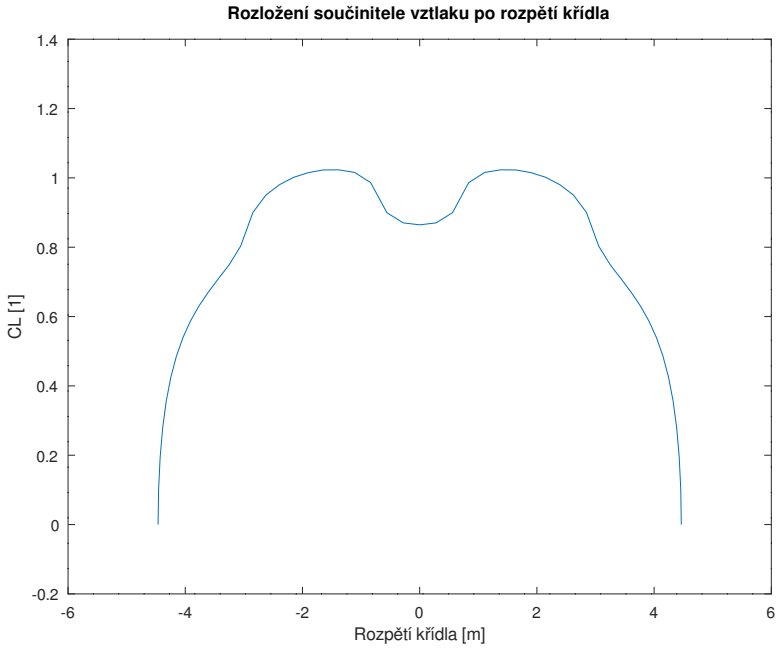
2.6. Obdélníkové křídlo a vztlakové klapky

Geometricky pro jednoduchost obdélníkové křídlo, ale s vychýlenými vztlakovými klapkami. Klapky se projeví změnou aerodynamických charakteristik profilů - změna úhlu nulového vztlaku. Stejně jako v případě výchylky křidélek, i u vztlak. klapek musíte znát efekt výchylky, hloubky a Re klapky na aerodynamické hodnoty profilu. Opět viz např. literatura [usaf-datcom], nebo [aap-roskam].

```
## Zadání geometrie a aerodynamiky křídla
g.S = 10;                           #    [m^2]
g.l = 8.929;                        #      [m]
g.z =         [ -4.464, 4.464 ];    #      [m]
g.b =         [  1.120, 1.120 ];    #      [m]
g.zkrouceni = [  0.0,   0];         # [stupně]
g.zkrouceni = deg2rad(g.zkrouceni);# [rad]
##
a.alfa = deg2rad(3.0);            #    [rad]
a.z =      [ -4.464, -3.000, ... # vnější konec křídla
             -2.990, -0.805, ... # pravá vztl. kl.
             -0.795,  0.795, ... # vnitřní část křídla
              0.805,  2.990, ... # levá vztl. kl.
              3.000,  4.464 ];   # vnější konec kř. [m]
a.alfa0 =  [ -5.125, -5.125, ...
             -9.12,   -9.12, ... # efekt výchylky klapky
             -5.125, -5.125, ...
             -9.12,  -9.12, ...  # efekt výchylky klapky
             -5.125, -5.125];    # [stupně]
a.alfa0 = deg2rad(a.alfa0);      #    [rad]
a.CLalfa = [  6.12,  6.12, ...
              6.00,  6.00, ...
              6.12,  6.12, ...
              6.00,  6.00, ...
              6.12,  6.12 ];     #  [1/rad]
```

Spuštění výpočtu a zpracování výsledků viz úvodní příklad v kapitole “2. *Vzorové příklady*” .

Očekávaný výsledek běhu programu:



Obrázek 9. Rozložení vztlaku - obdélníkové křídlo a vztl. klapky

```
hel náběhu kř. = 3.000000 deg
CL křídla       = 0.843870 /
CDi křídla      = 0.030069 /
Souč. delta     = 0.057367 /
```

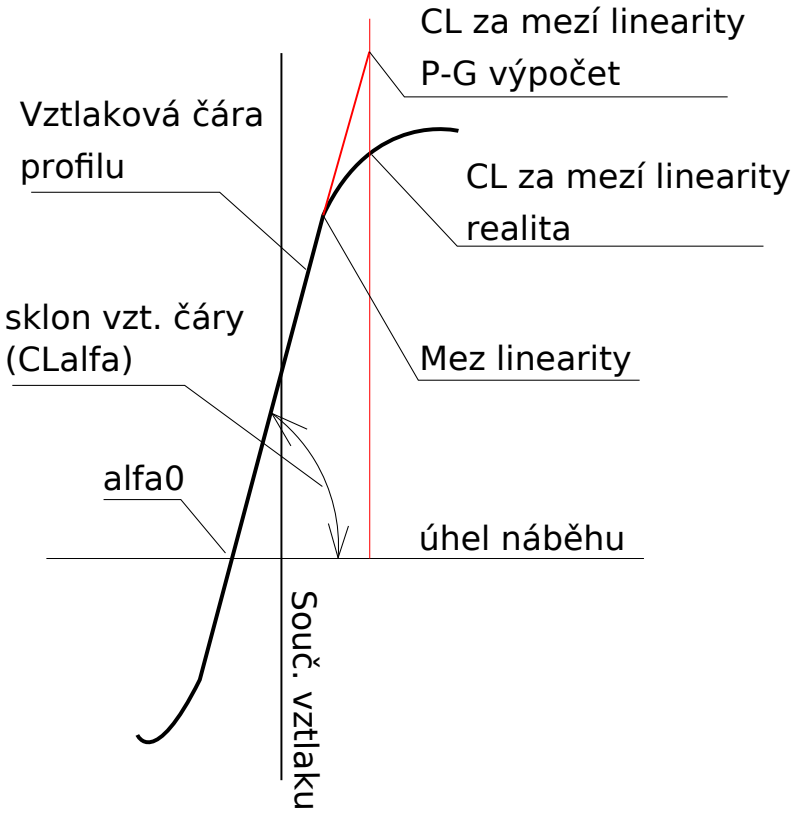
3. Poznámky a omezení metody

Je velmi dobré znát omezení a specifika použité teorie a metody s dopady do výsledků. Zde je (jistě neúplný) soupis těchto omezení a vlastností.

3.1. Potenciální proudění - velké úhly

Prandtl-Glauertovo řešení stojí na teorii vztlakové čáry křídla. Tato teorie je v základu potenciální proudění, které celou 3D geometrii křídla redukuje do čáry konečné délky ve které se realizuje potenciální vírový vztlak. To používá linearizovaný problém vztlakové čáry. Tento dlouhý popis má za úkol vysvětlit, že i řešení je přísně linearizované a nelze jinak.

Reálný profil ale nemá plně lineární charakteristiku. Většina aerodynamických profilů má lineární hlavní pracovní oblast, ale v blízkosti mezního vztlaku (ať již maximálního kladného, nebo záporného) se chovají nelineárně. Některé profily přísně vzato ani lineární pracovní oblast nemají a chovají se nelineárně v celém rozsahu.



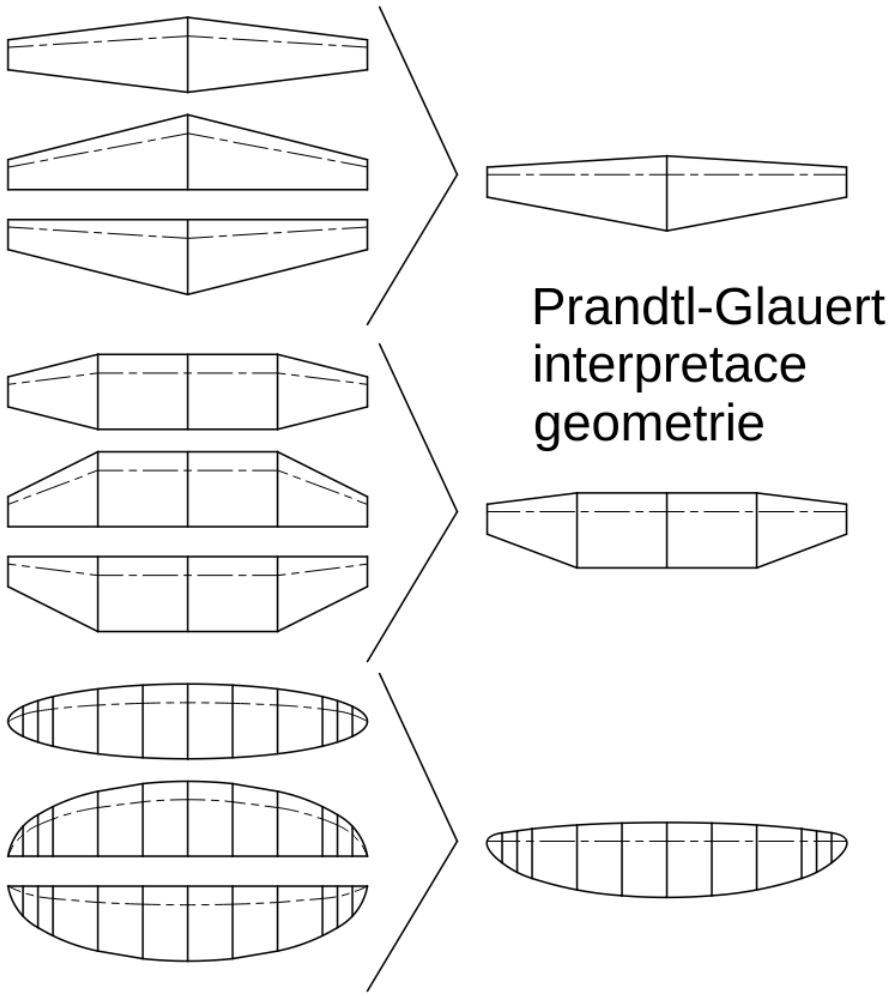
Obrázek 10. Vztlaková čára profilu - problém linearity

Na velkých úhlech náběhu křídla dojde k tomu, že řešení začne lokálně překračovat mez linearity součinitele vztlaku profilu. V ten moment se řešení podle této metody začne odchylovat od reality. Metoda pak nadhodnocuje lokální hodnoty souč. vztlaku a jeho rozložení oproti realitě.

3.2. Přímé křídlo - 1/4 čára

Metoda nahrazuje křídlo vírovým vztlakem v konečně dlouhé přímé čáře - spojnici 1/4 bodů profilů po rozpětí křídla. Tato kudrnatá věta říká, že ať vypadá křídlo jakkoli, s jakýmkoli šípem nebo půdorysným tvarem, metoda si křídlo narovná podle 1/4 bodů. Tečka.

V případě obdélníkového křídla není problém. Obdélník je prostě obdélník, šípovitost nulová. V případě lichoběžníkového křídla již odlišnost být může. Kombinované křídlo obdélník-lichoběžník je další dost obvyklá varianta a je třeba si být vědomý rozdílu. Eliptické křídlo obdobně.



Obrázek 11. Rozdíl mezi geometrií a jak si to přebere P-G metoda

Všechny výše uvedené příklady ilustrují, že metoda počítá s nulovou šípovitostí křídla vůči 1/4 bodům profilů. Pokud však máte výrazněji šípovité křídlo, svou roli to již hrát může. Šípovitost do cca 15 stupňů prakticky nehraje roli, chyba je přibližně (1 - cos(šípovitost). Čili pro 15 stupňů je chyba vlivem šípovitosti přibližně do 5 procent. Pak ale začne chyba rycle růst. Při šípovitosti 20 stupňů je chyba přibližně 6 procent a při šípovistosti 30 stupňů je chyba již přes 13 procent a rychle roste.

Samozřejmě jestliže drtivá většina křídla má šípovitost pod 15 stupňů a pouze koneček křídla vybočuje (např. eliptické křídlo narovnané podle odtokové hrany), pak to v podstatě nehraje roli. Podstatná je ta drtivá většina křídla.

3.3. Štíhlost křídla

Metoda redukuje křídlo o reálné hloubce do nekonečně tenké čáry, čili nekonečně vysoké štíhlosti. Metoda zcela ignoruje problematiku 3D geometrie, konců křídla, 3D koncových vírů a podobně. Tomu musí odpovídat i vstupní geometrie. Limitní štíhlost křídla od které jsou výsledky použitelné se uvádí cca štíhlost 5 až 7.

Metoda se tedy hodí pro podzvukové letouny s přímým křídlem o vyšší nebo vysoké štíhlosti (štíhlost vyšší než 7).

Metoda se nehodí pro transsonické letouny s nízkou štíhlostí, nebo exotické konfigurace s nízkou štíhlostí křídla (a velkým šípem ať již dopředným nebo zadním).

Metoda je absolutně nepoužitelná pro krátká hluboká křídla supersoniků, často dokonce trojúhelníkového charakteru jako má Mig-21, Mirage a podobně.

3.4. Spojitá geometrie - křidélka a klapky

Jak již bylo několikrát zmíněno, metoda považuje křídlo za spojitou čáru. Ale reálná geometrie má nenulovou hloubku a ne vždy je dokonale spojitá. Například vychýlené křidélko znamená, že se otevřela štěrbina a konce křidélek mají své vlastní víry a přefuky. Tím se mění proudění. Metoda však nic takového nedokáže zachytit. Vztlakové klapky jsou ještě výraznější příklad, fowlerova typu obzvláště.

Nicméně to má dopad i do zadávání geometrie. Již vstupní geometrie musí být spojitá. To znamená, že mezi např. koncem křidélka a navazující geometrií musí být nenulová vzdálenost z.

Příklad jak NELZE definovat geometrii křidélek:

```
## Zadání geometrie a aerodynamiky křídla
..
a.z =       [ -4.464, -4.300, ... # vnější konec křídla
              -4.300, -3.000, ... # pravé křidélko
```

```
             -3.000,  3.000, ... # vnitřní část křídla
              3.000,  4.300, ... # levé křidélko
              4.300,  4.464 ];   # vnější konec kř. [m]
a.alfa0 =  [ -5.125, -5.125, ...
             -7.5,   -7.5, ...   # efekt výchylky dolů
             -5.125, -5.125, ...
             -2.75,  -2.75, ...  # efekt výchylky nahoru
             -5.125, -5.125];    # [stupně]
a.alfa0 = deg2rad(a.alfa0);      #    [rad]
a.CLalfa = [  6.12,  6.12, ...
              6.10,  6.10, ...
              6.12,  6.12, ...
              6.12,  6.12, ...
              6.12,  6.12 ];     #  [1/rad]
```

Problém je v těch souřadnicích `a.z(2) = a.z(3) = -4.300`, dále v souřadnicích `a.z(4) = a.z(5) = -3.000`, dále v souřadnicích `a.z(6) = a.z(7) = 3.000` a dále v souřadnicích `a.z(8) = a.z(9) = 4.300`, i když by to tak geometricky na první pohled dávalo smysl. V takových místech je nespojitost a metoda s nespojitostí neumí počítat. Žádné souřadnice `g.z` nebo `a.z` se nesmí opakovat.

Příklad jak DEFINOVAT geometrii křidélek:

```
## Zadání geometrie a aerodynamiky křídla
..
a.alfa = deg2rad(3.0);           #    [rad]
a.z =      [ -4.464, -4.310, ... # vnější konec křídla
             -4.300, -3.000, ... # pravé křidélko
             -2.990,  2.990, ... # vnitřní část křídla
              3.000,  4.300, ... # levé křidélko
              4.310,  4.464 ];   # vnější konec kř. [m]
a.alfa0 =  [ -5.125, -5.125, ...
             -7.5,   -7.5, ...   # efekt výchylky dolů
             -5.125, -5.125, ...
             -2.75,  -2.75, ...  # efekt výchylky nahoru
             -5.125, -5.125];    # [stupně]
a.alfa0 = deg2rad(a.alfa0);      #    [rad]
a.CLalfa = [  6.12,  6.12, ...
              6.10,  6.10, ...
              6.12,  6.12, ...
              6.12,  6.12, ...
              6.12,  6.12 ];     #  [1/rad]
```

Za povšimnutí stojí to, že ony problematické místa se liší v souřadnici `a.z` o 10 mm. Nemusí to být zrovna 10 mm, ale je to vhodná hodnota. Vhodná hodnota je přehledně zaokrouhlené číslo mezi 1 promilem a 1 procentem z rozpětí křídla.

3.5. Počet řezů - přesnost řešení

Na první pohled by se zdálo, že čím hustější panelizace (více řezů) ve kterých se počítají rovnice, tím lépe. Není tomu tak úplně a to ze dvou provázaných důvodů.

V kapitole "4. *Teorie Prandtl-Glauertova řešení*" je uvedeno, že Fourierova transformace má stejný počet kořenů jako je řezů (mínus konce křídel). Na jednu stranu je velký počet řezů a tím kořenů transformace dobře, protože čím více kořenů, tím přesnější transformace bude. Ale na druhou stranu počet kořenů rovná se počet lineárních rovnic, které je nutné vyřešit. Numericky vyřešit, protože nad tři lin. rovnice již neexistuje obecné analytické řešení. A to je problém. S rostoucím počtem rovnic se zhoršuje matematická řešitelnost a začínají se projevovat numerické chyby řešení. Nelze obecně stanovit hranici, protože hodně záleží i na samotných rovnicích, jak jsou podmíněné. A to se obecně nedá určit. Variabilita geometrie a aerodynamiky ze kterých se ty rovnice určují je vysoká, tedy i rovnice mají veliký rozptyl podmíněnosti a tím řešitelnosti.

Druhý úzce související problém vysokého počtu řezů je to, že v každém z těch řezů se sestavuje rovnice, která se bude řešit. Ale pokud jsou řezy hustě vedle sebe, pak rovnice které se tam sestaví jsou si velmi podobné. Sousední rovnice popisují prakticky stejná místa se stejnými geometrickými a fyzikálními hodnotami. Takže při velkém a hustém počtu řezů dostáváme velké množství velmi podobných rovnic, které je třeba vyřešit. To vede opět na špatnou matematickou podmíněnost a špatnou řešitelnost - začnou se projevovat chyby řešění. Opět strašně špatně se dá obecně říci exaktní číslo.

Na druhou stranu je potřeba aby počet řezů nebyl příliš malý. Jak bylo řečeno, počet řezů rovná se počet kořenů Fourierovy transformace. Při malém počtu kořenů pak transformace jen velmi hrubě, a tedy špatně, reprezentuje původní fyzikální problém. Minimální jakš také použitelný počet řezů lze odhadnout cca na 9. Pro 9 řezů (a tedy 4 panely na polokřídle, 8 panelů po celém rozpětí) již bude docela sedět celkový souč. vztlaku, ale rozložení po rozpětí bude velmi velmi hrubé.

Vzato ze všech stran, rozumný kompromis lze odhadnout někde mezi 31 až 61 řezy, optimum spíše k těm 51 řezům. Ale pro některé specifické geometrie či aerodynamické konfigurace to nemusí platit a může být vhodný jiný počet řezů.

3.6. Lichý vs. sudý počet řezů

Není zcela bezvýznamné, zda je počet řezů lichý nebo sudý. LICHÝ počet řezů vede k tomu, že jeden z řezů BUDE umístěn přesně V POLOVINĚ ROZPĚTÍ křídla. Naproti tomu SUDÝ počet řezů znamená, že POLOVINĚ ROZPĚTÍ křídla NEBUDE umístěn řez.

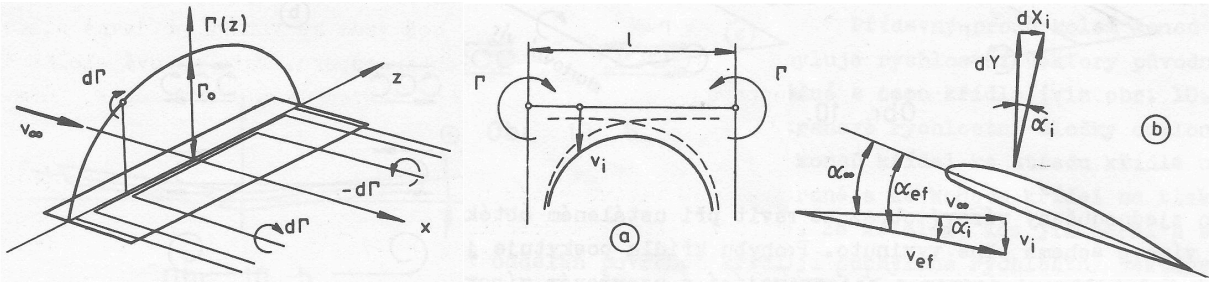
Lze asi předpokládat, že většina uživatelů bude chtít mít rozložení vztlaku s tím, že chce řez/hodnotu v rovině symetrie křídla. Čili potřebují LICHÝ počet řezů. Proto je doporučeno používat prostě lichý počet řezů.

4. Teorie Prandtl-Glauertova řešení

Nebudu předstírat, že té teorii rozumím a umím to. Je to již 20 let co jsem ze školy a tehdá jsem se to naučil na zkoušku a pak zase pustil z hlavy, protože tu teorii jako takovou jsem pak už nikdy nepotřeboval. Dnes už pouze umím číst ta scripta a podle nich postavit program. A snad ještě chápu omezení a důsledky dané teorie, které popisuji v kapitole s poznámkami a omezeními.

Nicméně aby bylo vidět že to není žádná černá skříňka, popisuji zde teorii tak jak jsem ji použil.

Křídlo je nahrazeno sérií podkovovitých vírů dle Prandtla.



Obrázek 12. Prandtlova náhrada křídla (zdroj: lit. [aer-broz])

Lokální efektivní úhel náběhu profilu v místě se souřadnicí z je dán rozdílem úhlu náběhu volného proudu a indukovaného úhlu náběhu.

$$\alpha_{ef}(z) = \alpha_\infty(z) - \alpha_i(z)$$
$$\alpha_a(z) = \alpha_\infty(z) = \alpha_{kridla} + \alpha_{zkrouceni}(z) - \alpha_0(z)$$

Cirkulace v lokálním místě se souřadnicí z a rozepsáním pomocí sklonu vztlakové čáry profilu:

$$\Gamma(z) = \frac{1}{2}.V_\infty.C_L(z).b(z)$$
$$\Gamma(z) = \frac{1}{2}.V_\infty.b(z).C_L^\alpha(z).\alpha_e f(z)$$
$$\Gamma(z) = \frac{1}{2}.V_\infty.b(z).C_L^\alpha(z).(\alpha_a(z) - \alpha_i(z))$$

A z toho dalšími složitými úpravami vypadne Prandtlova integro-diferenciální rovnice křídla:

$$\Gamma(z) = \frac{1}{2}.b(z).V_\infty.C_L^\alpha(z).\left[\alpha_a(z) + \frac{1}{4.\pi.V_\infty}.\int_{-l/2}^{l/2}\frac{d\Gamma}{dz_1}.\frac{dz_1}{(z_1 - z)}\right]$$

Na to byla aplikována Glauertova transformace lineárního rozměru do úhlového, což umožnilo rovnici přepsat do tvaru:

$$z = -\frac{l}{2}.\cos(\theta)$$
$$\Gamma(\theta) = \frac{1}{2}.b(\theta).V_\infty.C_L^\alpha(\theta).\left[\alpha_a(\theta) + \frac{1}{2.\pi.V_\infty.l}.\int_0^\pi\frac{d\Gamma}{d\theta_1}.\frac{d\theta_1}{(\cos\theta - \cos\theta_1)}\right]$$

Pak byla cirkulace nahrazena Fourierovou řadou o rozměru `M` plus doplněny okrajové podmínky pro konce křídla. (Ve skriptech [aer-broz] je trochu jiný formální zápis, zde to je přepsané do formy vhodné pro zápis do programového kódu)

$$\Gamma(\theta) = 2.\,l.\,V_\infty.\sum_{n=1}^{M} X(n).\sin(n.\,\theta)$$
$$\Gamma(0) = \Gamma(\pi) = 0$$

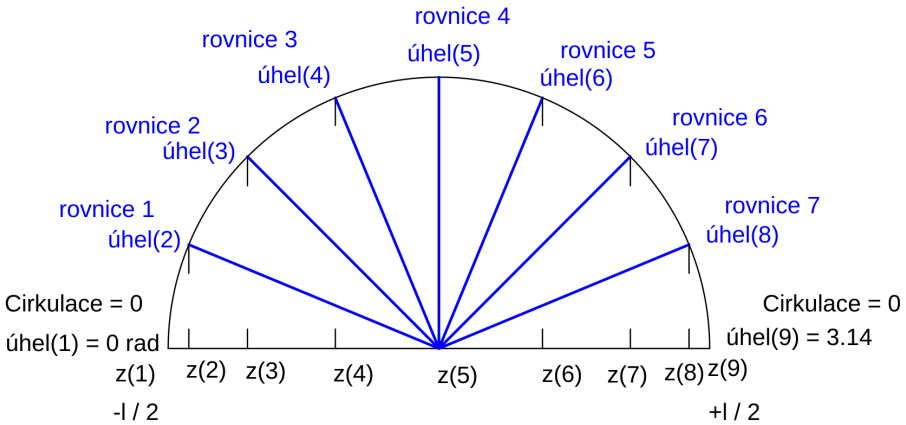
A dalšími úpravami vypadla zdánlivě ošklivá, ve skutečnosti ale již naprogramovatelná rovnice:

$$\sum_{n=1}^{M}\left(\sin(\theta) + n.\,\frac{b(\theta).\,C_L^\alpha(\theta)}{4.l}\right).\sin(n.\,\theta).\,X(n) = \frac{b(\theta).\,C_L^\alpha(\theta)}{4.l}.\,\alpha_a(\theta).\sin(\theta)$$

Tím došlo k tomu, že jsme neznámou funkci cirkulace nahradili lineární rovnicí s vektorem neznámých `X` o rozměru `M`. To znamená, že potřebujeme sestavit `M` rovnic abychom mohli vyřešit `M` neznámých (prvků vektoru `X`). Formálně lze přepsat:

$$A.\,\vec{X} = \vec{B}$$

Takže jak z té jedné rovnice udělat soustavu rovnic? S tím pomůže nařezání křídla podle úhlu po rozpětí a v každém tomto bodě vyčíslit jednu rovnici. Celkem bude potřeba `M` řezů *uvnitř* křídla. Konce křídla se do toho nezapočítávají, v nich jsou definované okrajové podmínky. Nicméně to znamená, že křídlo je fakticky nařezáno na `m = M + 2` řezů včetně konců křídla. Uživatele zajímá rozložení po celém rozpětí včetně konců křídla, takže pole úhlů a všch výsledků bude mít rozměr `m`. Na to je třeba dát pozor, že se liší počet řezů výsledků a úhlů od rozměru výpočetních matic a polí `A`, `X` a `B`.



Obrázek 13. Rozdělení křídla na řezy (když m = 9, potom M = 7).

Takže index úhlů je o jeden posunutý oproti indexu kořenů Fourierovy řady, na to je třeba dávat pozor. Prvky matice `A` a vektoru `B` lze pak sestavit:

$$i = \langle 2; m-1 \rangle \quad ; \quad j = i-1 \quad ; \quad n = \langle 1; m-2 \rangle$$

$$A(j,n) = \left(\sin(\theta_i) + n.\, \frac{b(\theta_i).\, CL^{\alpha}(\theta_i)}{4.\, l} \right). \sin(n.\, \theta_i)$$

$$B(j) = \alpha_a(\theta_i). \sin(\theta_i).\, \frac{b(\theta_i).\, CL^{\alpha}(\theta_i)}{4.\, l}$$

Nyní je třeba soustavu rovnic numericky vyřešit. To je samostatná kapitola, to zde popisovat nebudu. V tomto jsem využil řešič lineárních rovnic v Octave a postupoval jsem dle manuálu.

Když jsou rovnice vyřešeny, je třeba určit výstupní důležité hodnoty. Důležitými údaji co metoda Prandtl-Glauert poskytuje jsou:

- Součinitel vztlaku celého křídla - CL
- Opravný Glauertův součinitel indukovaného odporu křídla - delta
- Indukovaný součinitel odporu celého křídla - CDi
- Rozložení místního vztlaku po rozpětí - CL(z) . b(z)
- Rozložení součinitele vztlaku po rozpětí křídla - CL(z)
- Rozložení indukovaného úhlu náběhu po rozpětí křídla - alfa_i(z)

$$CL = X(1).\, \pi.\, \lambda$$

$$\delta = \sum_{n=2}^{m-2} n.\, \left(\frac{X(n)}{X(1)} \right)^2$$

$$CD_i = \lambda.\, \pi.\, X(1)^2.\, (1+\delta)$$

$$i = \langle 1; m \rangle \qquad CLb(\theta_i) = 4.l. \sum_{n=1}^{m-2} X(n). \sin(n.\, \theta_i)$$

$$i = \langle 1; m \rangle \qquad CL(\theta_i) = \frac{4.l}{b(\theta_i)} . \sum_{n=1}^{m-2} X(n). \sin(n.\, \theta_i)$$

$$\alpha_i(1) = \sum_{n=1}^{m-2} n^2.\, X(n) \qquad ; \qquad \alpha_i(m) = \sum_{n=1}^{m-2} n^2.\, X(n)$$

$$i = \langle 2; m-1 \rangle \qquad \alpha_i(\theta_i) = \sum_{n=1}^{m-2} n.\, X(n).\, \frac{\sin(n.\, \theta_i)}{\sin(\theta_i)}$$

A poslední poznámka, je třeba ošetřit případy dělení nulou, když se dělí místní hloubkou křídla (ostrý konec křídla např.).

5. Co dál, rozjímání nad teorií

Metoda Prandtl-Glauertova řešení vztlakové čáry křídla má dost omezení. Je to už dost let co jsem naprogramoval první verzi (tehdy v jazycích C a Tcl+Tk) a co mi leželo v hlavě, jestli by to šlo vylepšit. Je dobré si roztřídit jednotlivé oblasti problémů a na každý z nich se podívat zvlášť. V podstatě jsou dvě skupiny problémů, každá obsahuje své dílčí podproblémy.

- Fyzikální problémy pramenící ze zjednodušení metody oproti realitě. Toto je dominantní problém a omezení metody.
- Matematické problémy pramenící z ryze matematických zjednodušení a numerických chyb. Toto je sekundární zdroj chyb a nepřesností.

První a dominantní fyzikální problém je náhrada *vazkého stlačitelného proudění kolem komplexní prostorové geometrie křídla* zjednodušenou teorií *nevazkého nestlačitelného potenciálního proudění kolem křídla redukovaného do vztlakové úsečky*. To v první řadě zcela smaže veškeré geometrické nuance, nespojitosti geometrie a aerodynamiky. A s tím se nedá nic dělat. Tato nevýhoda je ale vyvážena jednoduchostí řešení i uživatelského zadání a tím použití.

Druhý fyzikální problém, a s tím by se už něco dělat dalo, je linearizace metody. Prandtlova rovnice vztlakové čáry, a následně Prandtl-Glauertova rovnice, obsahují vztah sklon vztlakové čáry krát aerodynamický úhel náběhu. To je jinými slovy místní hodnota vztlaku při místním úhlu náběhu. Od toho se odečítá vliv indukovaného úhlu náběhu - srážka vztlaku vlivem indukovaného úhlu náběhu. Prandtl-Glauert tím násobením používá lineárizovanou vztlakovou čáru. Nahrazením této čáry kopletní nelineární čarou by šlo, prostě by se určoval místní souč. vztlaku a v daném místě by se určoval místní sklon pro redukci vlivem ind. úhlu náběhu. Přidáním opět Fourierova rozvoje z toho dostaneme soustavu rovnic, kterou by šlo opět řešit numericky. Např. Newtonovým schématem, nebo nějakým lepším (např. Runge-Kutta). Tím by se teorie rozšířila i na velké úhly náběhu, potud skvělé. Problém tohoto přístupu je ale v mnohem komplexnějším zadání křídla. Zatímco klasický Prandtl-Glauert si aerodynamicky vystačí s úhlem nulového vztlaku a sklonem vztlakové čáry v místě aerodynamického řezu, zde by to vyžadovalo plnou a dost přesnou definici vztlakové čáry. A to je komplikace kterou bych sám nebyl ochoten používat, proto jsem zůstal jen u teoretizování.

První matematický problém - Fourierův rozvoj. Jak již bylo řečeno v předchozím, pro Fourierův rozvoj by bylo dobré mít co nejvíce kořenů. Ale když jich je moc, tak je i moc rovnic a numerické řešení začíná mít problémy s přesností. Optimum se mi podle zkušenosti jeví někde mezi 30 až 60 kořeny. Také bylo řečeno, že vyšší počet kořenů se rovná vyšší počet řezů a tam ještě přichází problém že jsou ty řezy blízko a tak jsou tam hodně podobné rovnice a to dělá prolbémy při řešení těch rovnic. Ale kdyby jsme nechali počet kořenů Fourierova rozvoje konstantní a zvedli počet řezů, tak dostaneme víc rovnic než neznámých. Přímo to řešit nejde. Ale lze pak poštvat na řezy statistiku a udělat nějakou optimalizaci. Konkrétně se přímo nabízí metoda nejmenších čtverců pro optimální výpočet parametrů s nejmenší odchylkou. V takovém případě naopak platí, že čím víc řezů (přebytek rovnic nad počtem kořenů) tím lépe, metoda nejmenších čtverců pak má z čeho optimalizovat. Nicméně jsem si nikdy neudělal čas a nesebral sílu toto realizovat. Nevím jestli by ta 1-2 procenta zisku lepšího výsledku za tu námahu stálo.

Druhý matematický problém - Fourierova transformace samotná. Dobře nahrazuje funkce typu vlna (proto je tak důležitá ve zpracování zvuku), ale ostré zlomy a hrany jí dělají problémy. Zvážit i jiné náhradní funkce jako Taylorův rozvoj, nebo ještě exotičtější věci. Ale kdo by se s tím patlal? Zisk na zlepšení přesnosti je malý a spíše jen v nějakých speciáních případech.

Třetí matematický problém - přístup přes přímé řešení soustavy `A.X = B`. Toto přímé numerické řešení nebude nikdy přesné. Možná by bylo vhodnější to zkusit numerickými odhady řešení - Newtonem nebo Runge-Kutta přímo výchozích fyzikálních rovnic. Možná by se dosáhlo lepších přesností. Ale zase, kdo by se kvůli zpřesnění o jedno až dvě procenta s tím tak patlal.

Věřím tomu, že vše toto už někoho napadlo a určitě to vyřešil. Jen mi to nestálo za to hledat a provádět na toto téma „hystorický výzkum“. Benefity mi přijdou příliš malé, takže mi to za tu námahu nestálo.

6. Závěr

Snad toto vše někomu k něčemu bude a neskončí to zapomenuté jako datový bordel. Dalo to dost práce to naprogramovat čistě abych se za to nemusel stydět a hlavně to pak pořádně zdokumentovat. Ta dokumentace mě stála jako obvykle více práce a energie než samotné programování.

Pokud budete mít nějaké dotazy nebo náměty, tak e-mail na mě je v tomto dokumentu uvedený. Nicméně se těmito věcmi zabývám jen bokem jako své hobby, takže nemůžu nic slíbit. Ale přinejmenším mě potěší vědomí že to asi někdo opravdu používá.

Pokud najdete chybu ať již v kódu nebo v této dokumentaci, neváhejte mě prosím kontaktovat. Jak se říká, nikdo není dokonalý a v každém programu je nejméně jedna chyba.

Mám v plánu postupně zveřejnit více nástrojů co jsem si za ta léta co dělám do ULL a modelářské aerodynamiky udělal. Asi největší balík mám zpracovaný program pro stanovení aerodynamického a setrvačného zatížení částí letounu dle předpisu ULL-600kg, ale to je nezveřejnitelné v podobě jak to je teď. Jednak tam jsou nasázena privátní data o letounu, druhak to je na ten letoun hodně napasované (není to moc univerzální) a hlavně to je dost naprasené jak jsem to dělal pod tlakem a neustále rozšiřoval a upravoval jak se měnily požadavky co vše ještě musíme dodat k certifikaci. Ale chtěl bych to pomalu v klidu přepisovat, modularizovat a alespoň částečně zveřejnit. No mám nejméně na 2 roky co dělat po večerech.

A v neposlední řadě jsem tento dokument pojal jako ostrý test pro mě nového systému pro sazbu textů - DocBook s XSLT. Už nějakou dobu se to učím a zkouším, tak teď jsem to poprvé reálně použil místo dosavadního TeXu. Výsledek vypadá snad použitelně. DocBook samotný je pro mě použitelný (byť s výhradami ohledně rovnic a matematiky), ale to XSLT mi dává hodně zahulit, to je peklo.

Literatura

[aer-broz] *Aerodynamika nízkých rychlostí*; prof. Ing. Václav Brož CSc.; vydání třetí; České vysoké učení v Praze; v lednu 1990; Ediční středisko ČVUT, Zikova 4 Praha 6 ;

[wiki-llt] *WIKIPEDIA: Lifting-line theory*; ---; 2021; Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Lifting-line_theory; 2021;

[usaf-datcom] *USAF Stabilty and Control DATCOM*; R. Fink; Final Report for Period September 1975 - September 1977; McDonnell Douglas Corporation, Douglas Aircraft Division; April 1978; Long Beach California 90846 ;

[aap-roskam] *Airplane Aerodynamics and Performance*; Dr. Jan RoskamDr. Chaun-TauEdward Lan; ISBN 1-884885-44-6; Design, Analysis and Research Corporation (DARcorporation); 1997; 120 East Ninth Street, Suite 2 Lawrence, Kansas, USA 66044 ;