Proračunska aerodinamika

Nedelja 3

A. Simonović & J. Svorcan

Mašinski fakultet, Katedra za vazduhoplovstvo

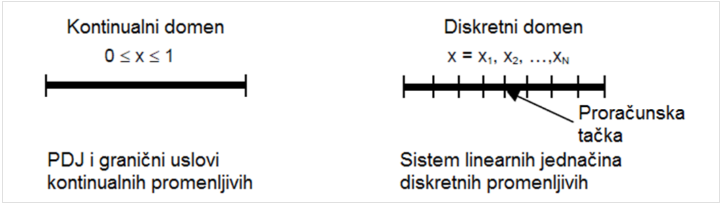
2023/2024.

Sadržaj

Uvod u Proračunsku aerodinamiku

Osnovni motiv Proračunske aerodinamike/mehanike fluida (CFD) je odredjivanje nepoznatih fizičkih veličina strujanja koje je ili nemoguće ili previše komlikovano analitički rešavati.
Uvode se izvesne aproksimacije i sprovode sledeći koraci:

- Kontinualni domen predstavlja se konačnim brojem tačaka/delića/zapreminica ...
- Veličine strujanja su definisane samo u kontrolnom elementu.
- Promene veličine predstavljene su linernim jednačinama.
- Rešava se formirani sistem jednačina.



Uvod u diskretni račun

- Potreban nam je odgovarajući matematički aparat za diskretni račun.
- Počinjemo od numeričke diferencijacije (vadjenja izvoda) i metoda konačnih razlika.
- Uglavnom razmatramo glatke funkcije, ali samo nad ograničenim skupom tačaka.
- Od ranije znamo da odredimo izvod eksplicitno zadatih/poznatih funkcija (elementarnih i složenih), a sad se bavimo f-jama nad diskretnim podacima.
- Tejlorovi polinomi su nam izuzetno korisni ...

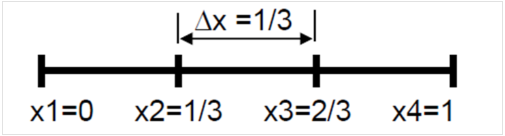
Uvod u Metod konačnih razlika

Osnovnu ideju Metoda konačnih razlika moguće je ilustrovati nad običnom diferencijalnom jednačinom 1. reda:

$$\frac{\mathrm{d}u}{\mathrm{d}x} + u^m = 0,\ 0 \le x \le 1,\ u(0) = 1.$$

1. Za $m = 1$ data jednačina je linearna.
2. Za $m = 2$ data jednačina je nelinearna.

Problem se može rešiti nad jednostavnom, malom mrežom prikazanom na slici.



Uvod u Metod konačnih razlika

Osnova Metoda konačnih razlika je aproksimacija izvoda funkcije:

$$\frac{\mathrm{d}u}{\mathrm{d}x} = \lim_{\Delta x \to 0} \frac{u(x+\Delta x) - u(x)}{\Delta x} \Rightarrow \frac{\mathrm{d}u}{\mathrm{d}x} \approx \frac{u(x+\Delta x) - u(x)}{\Delta x}.$$

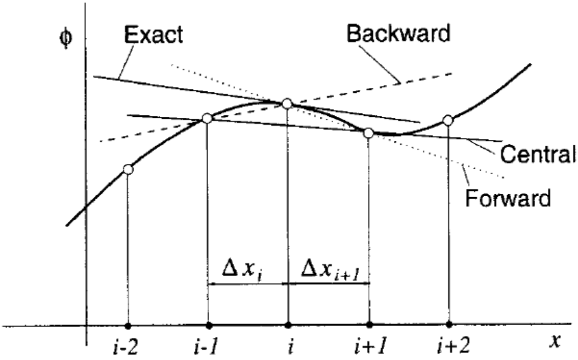
Greška ove aproksimacije je $o(\Delta x)$ (tačnost šeme unapred je 1). Analogno, može se koristiti šema unazad ili centralna šema (2. reda tačnosti):

$$\frac{\mathrm{d}u}{\mathrm{d}x} \approx \frac{u(x) - u(x-\Delta x)}{\Delta x}, \quad \frac{\mathrm{d}u}{\mathrm{d}x} \approx \frac{u(x+\Delta x) - u(x-\Delta x)}{2\Delta x}.$$

Sada se početna jednačina, za $m = 1$, može svesti na:

- "eksplicitni", $\frac{u_{i+1} - u_i}{\Delta x} + u_i = 0$, ili
- "implicitni" oblik, $\frac{u_{i+1} - u_i}{\Delta x} + u_{i+1} = 0$.

Uvod u Metod konačnih razlika



Slika: Ilustracija šema aproksimacije: unapred, unazad i centralne

Uvod u Metod konačnih razlika

Tejlorov red je jedan od načina da se izvrši aproksimacija izvoda f-je $f(x)$. Npr.

$$f(x+\Delta x) = f(x) + \Delta x f'(x) + \frac{\Delta x^2}{2!} f''(x) + \frac{\Delta x^3}{3!} f'''(x) + \dots$$

Šablon unapred direktno sledi:

$$f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x} \underbrace{- \frac{\Delta x}{2!} f''(x) - \frac{\Delta x^2}{3!} f'''(x) - \dots}_{o(\Delta x)}$$

Centralni šablon se izvodi iz razlike aproksimacija funkcije u tačkama $x+\Delta x$ i $x-\Delta x$, odnosno iz izraza:

$$f(x+\Delta x) - f(x-\Delta x) = 2\Delta x f'(x) + 2\frac{\Delta x^3}{3!} f'''(x) + \dots$$

Uvod u Metod konačnih razlika

Još jedan primer aproksimacije prvog izvoda fje šemom višeg reda tačnosti $o(\Delta x^2)$:

$$\frac{\mathrm{d}u}{\mathrm{d}x} \approx \frac{-u_{i+2} + 4u_{i+1} - 3u_i}{2\Delta x}.$$

Ili tačnosti $o(\Delta x^4)$:

$$\frac{\mathrm{d}u}{\mathrm{d}x} \approx \frac{u_{i-2} - 8u_{i-1} + 8u_{i+1} - u_{i+2}}{12\Delta x}.$$

Uočiti karakteristike razlomka ...
Koliko graničnih uslova je sada potrebno?

Uvod u Metod konačnih razlika

Kako odrediti kako će članovi učestvovati, za željenu najveću tačnost?
Npr, ukoliko želimo da uključimo članove u_j, u_{j+1} i u_{j+2} moguće je formirati sistem jednačina:

$$u'_j \Delta x + \sum_{k=0}^{2} a_k u_{j+k} = o(?),$$

gde je potrebno odrediti koeficijente a_k ...
Sistem je zapisan tako da koeficijenti ne zavise od Δx.

Linearna jednačina, eksplicitni metod

Nepoznata veličina u_{i+1} može se izraziti preko poznate veličine u_i:

$$u_{i+1} = u_i(1 - \Delta x).$$

Za zadati uslov $u(0) = 1$ može se dobiti numeričko rešenje (u Matlab-u) i uporediti sa analitičkim:

```
N = 4;
x = linspace(0,1,N);
dx = x(2) - x(1);
u = zeros(1,N); u(1) = 1;    % granicni uslov
for i = 2:N
    u(i) = u(i-1)*(1-dx);
end
figure, plot(x,u,'-',x,exp(-x),'-.','linewidth',2);
```

Linearna jednačina, eksplicitni metod, viši red tačnosti

Iz diskretizovanog izraza polazne jne:

$$\frac{u_{i+1} - u_{i-1}}{2\Delta x} + u_i = 0,$$

Nepoznata veličina u_{i+1} može se izraziti kao:

$$u_{i+1} = u_{i-1} - 2\Delta x u_i.$$

Za proračun vrednosti u_{i+1} potrebno je poznavati vrednosti u_i i u_{i-1} . . .

```
% 2. reda tacnosti
us = zeros(1,N);
% granicni uslovi
us(1) = 1;
us(2) = exp(-x(2));
% us(2) = us(1)*(1-dx);
for i = 3:N
    us(i) = us(i-2) - 2*dx*us(i-1);
end
hold on, plot(x,us,':','linewidth',2);
```

Linearna jednačina, implicitni metod

Malo drugačijim zapisom:

$$-u_i + (1 + \Delta x)u_{i+1} = 0.$$

Sistem algebarskih jednačina može se zapisati matrično:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1+\Delta x & 0 & 0 \\ 0 & -1 & 1+\Delta x & 0 \\ 0 & 0 & -1 & 1+\Delta x \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Kako u_o nije definisano prvi red matrice koeficijenata se razlikuje! Promenljive su definisane jedne preko drugih, pa sve jednačine moraju biti rešene istovremeno, odatle **implicitni** metod.

Linearna jednačina, implicitni metod

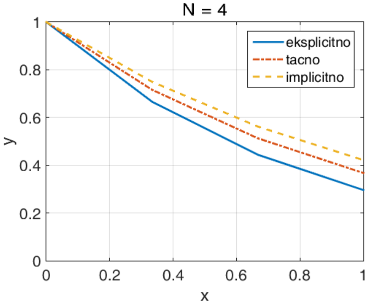
Kod u Matlab-u:

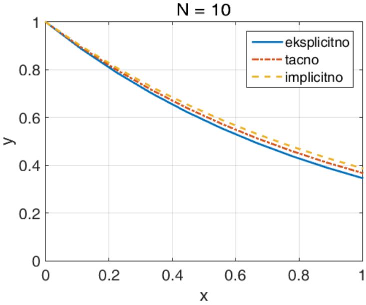
```
% implicitno
% matrica koeficijenata A
e = ones(N,1);
A = spdiags([-e (1+dx)*e],[-1 0],N,N);
% korekcija, zbog granicnog uslova
A(1,1) = 1;
% rezultujuca matrica B
B = zeros(N,1); B(1) = 1;
u2 = A\B;
hold on, plot(x,u2,'--','linewidth',2);
xlabel('x'), ylabel('y'), axis([0 1 0 1])
```

Linearna jednačina, analiza i poredjenje rezultata

- Naći analitičko rešenje.
- Uporediti rezultate, oceniti grešku korišćenih metoda.
- Proveriti kvalitet proračunske mreže (sprovesti studiju konvergencije mreže). Šta se dobija za različit (veći) broj tačaka proračunskog domena N?

Linearna jednačina, analiza i poredjenje rezultata





Nelinearna jednačina, eksplicitni metod

Polazni problem

$$\frac{\mathrm{d}u}{\mathrm{d}x} + u^2 = 0,\ 0 \leq x \leq 1,\ u(0) = 1,$$

možemo zapisati u sledećem obliku:

$$\frac{u_{i+1} - u_i}{\Delta x} + u_i^2 = 0 \Rightarrow u_{i+1} = u_i(1 - \Delta x \cdot u_i).$$

Primer koda u Matlab-u:

```
% eksplicitno
u = zeros(1,N); u(1) = 1;    % granicni uslov
for i = 2:N
    u(i) = u(i-1)*(1-dx*u(i-1));
end
figure, plot(x,u,'-',x,1./(x+1),'-.','linewidth',2);
```

Neinearna jednačina, implicitni metod

Malo drugačijim zapisom:

$$-u_i + (1 + \Delta x \cdot u_{i+1})u_{i+1} = 0.$$

Sistem algebarskih jednačina opet se može zapisati matrično:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 + \Delta x \cdot u_2 & 0 & 0 \\ 0 & -1 & 1 + \Delta x \cdot u_3 & 0 \\ 0 & 0 & -1 & 1 + \Delta x \cdot u_4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Primećujemo da je sistem znatno komplikovaniji i da matrica koeficijenata zavisi od promenljivih!
Ovakav sistem rešava se iterativno, uz pretpostavku početnog rešenja $[u_{g,1}\ u_{g,2}\ u_{g,3}\ u_{g,4}]'$.

Nelinearna jednačina, implicitni metod

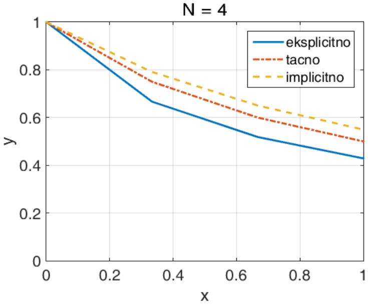
Kod u Matlab-u:

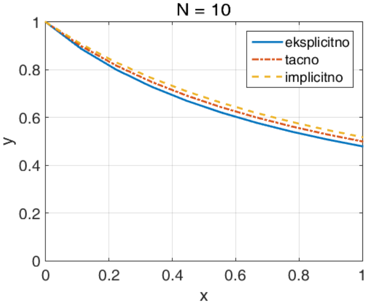
```
ug = rand(N,1); eps = 0.001*e; i = 1;
A = spdiags([-e (1+dx*ug).*e],[-1 0],N,N);
A(1,1) = 1;  % korekcija, zbog granicnog uslova
u2 = A\B;
while sum(abs(ug-u2)>eps)
    i = i+1;
    ug = u2;
    A = spdiags([-e (1+dx*ug).*e],[-1 0],N,N);
    A(1,1) = 1;  % korekcija, zbog granicnog uslova
    u2 = A\B;
end
```

Nelinearna jednačina, analiza i poredjenje rezultata

- Naći analitičko rešenje.
- Uporediti rezultate, oceniti grešku korišćenih metoda. Šta se primećuje pri poredjenju eksplicitnog i implicitnog metoda? Koliko iteracija je potrebno sprovesti pri implicitnom rešavanju problema?
- Proveriti kvalitet proračunske mreže (sprovesti studiju konvergencije mreže). Šta se dobija za različit (veći) broj tačaka proračunskog domena N?

Nelinearna jednačina, analiza i poredjenje rezultata





Nelinearna jednačina, $m = 3$

Rešiti diferencijalnu jednačinu:

$$\frac{\mathrm{d}u}{\mathrm{d}x} + u^3 = 0,\ 0 \leq x \leq 1,\ u(0) = 1,$$

razmatranim metodama.

Šta se dogadja sa brojem potrebnih iteracija za istu željenu tačnost sa povećanjem stepena člana u^m?

Diferencijalna jednačina višeg reda

Rešiti diferencijalnu jednačinu:

$$\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} + y = x^3,\ 0 \leq x \leq \pi,\ y(0) = 1,\ y'(0) = -7$$

razmatranim metodama.
Obratiti pažnju na potreban broj dopunskih uslova ...

Korisno:

```
syms y(x)
eqn = diff(y,x,2) + y == x^3;
ySol(x) = dsolve(eqn)
```

Numerička integracija

Često je nemoguće analitički rešiti integral f-je $f(x)$:

$$\int_a^b f(x)dx,$$

pa se tada pristupa numeričkoj integraciji:

$$\int_a^b f(x)dx \approx \sum_a^b f(x_i)\Delta x.$$

Najprostiji metod je trapezno pravilo, ali se često koristi i Simpsonovo pravilo.
Postojeće f-je u Matlab-u su **integral**, **trapz**, **polyint** ... (više se može naći pod *Numerical Integration and Differentiation*).

Trapezno pravilo

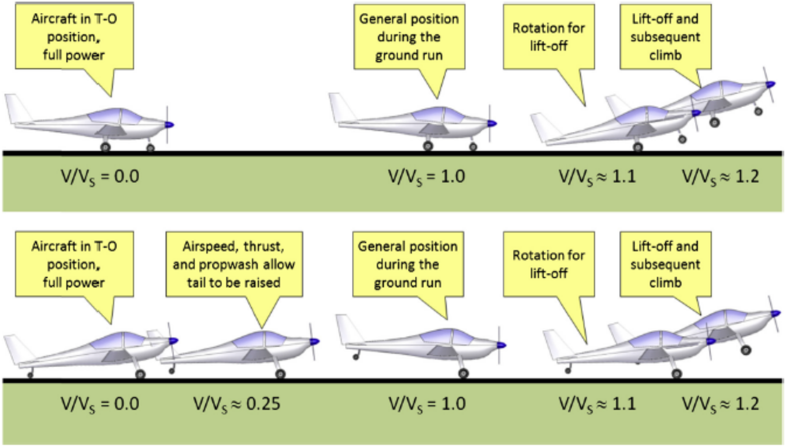
Osnovna pretpostavka za interval $x_i \leq x \leq x_{i+1}$ je:

$$\int_{x_i}^{x_{i+1}} f(x)dx \approx \frac{\Delta x}{2}\left(f_i + f_{i+1}\right),$$

gde je $\Delta x = x_{i+1} - x_i$.
Odnosno, geometrijski pretpostavljamo da se fja f može aproksimirati pravom linijom na posmatranom intervalu. Razmatranje možemo usložniti ukoliko fju f aproksimiramo parabolom (Simpsonovo pravilo).

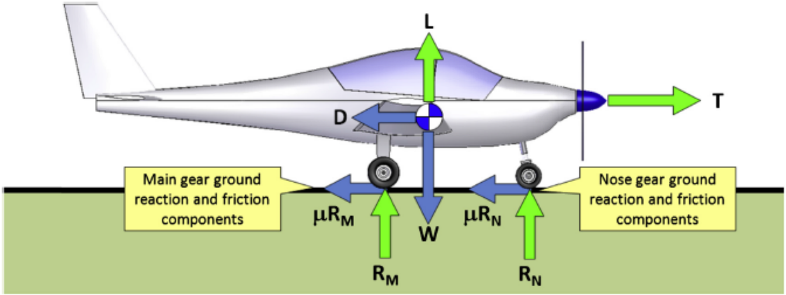
Zalet aviona

Zalet, kao početna faza poletanja, predstavlja rastojanje S_G koje letelica predje od tačke mirovanja do odvajanja od tla.



Zalet aviona

Sile koje deluju na letelicu u zaletu (tokom kretanja po tlu):



$$
\begin{aligned}
ma_x = m\ddot{x} &= T - D - \mu(R_N + R_M) \\
ma_y = 0 &= W - L - (R_N + R_M)
\end{aligned}
$$

Zalet aviona

Zamenom

$$L = \frac{\rho V^2}{2} S C_L,\ D = \frac{\rho V^2}{2} S C_D = \frac{\rho V^2}{2} S \left(C_{D_o} + k C_L^2 \right),$$

u prethodne izraze i znajući da je

$$a_x = \frac{dV_x}{dt} \Rightarrow dt = \frac{dV_x}{a_x},\ V_x = \frac{dx}{dt} \Rightarrow dx = V_x dt = \frac{V_x dV_x}{a},$$

može se izvesti potrebni izraz:

$$S_G = \int_0^{t_p} V dt = \int_0^{V_p} \frac{V dV}{\frac{T}{m} - \mu g - \frac{\rho V^2}{2m} S \left(C_D - \mu C_L \right)}.$$

Gornja granica integrala, brzina poletanja V_p, uglavnom iznosi $V_p \in [1.1 V_{\min}, 1.2 V_{\min}]$, gde je $V_{\min}$ brzina sloma uzgona u poletanju (izvučena mehanizacija).

Zadatak 2 – Zalet (kao jedna od faza poletanja)

Laki lovac bombarder, mase $m = 5100$ kg i površine krila 16.5 m^2 ima mlazni motor koji razvija statički potisak od $T = 2240$ daN. Polara aviona u slučaju izvučenog stajnog trapa i izvučenih zakrilaca na $\delta = 20^\circ$ može se predstaviti izrazom:

$$C_D = C_{D_o} + \kappa C_L^2 = 0.023 + 0.09 C_L^2.$$

Maksimalni koeficijent uzgona sa zakrilcima na $\delta = 20^\circ$ i uticajem zemlje iznosi $C_{L_{\max}} = 1.08$. Odrediti dužinu S_G i vreme zaleta t_p u slučaju poletanja sa suvog, nabijenog, travnatog aerodroma pod predpostavkom da pilot u zaletu održava napadni ugao α_{opt} kome odgovara optimalna finesa $(C_L/C_D)_{\max}$ u poletanju. Koeficijent trenja zemljišta je $\mu = 0.05$.

Zadatak 2 – Zalet (kao jedna od faza poletanja)

Potrebno je numerički rešiti integral (izračunati sumu):

$$S_G = \int_0^{V_p} \frac{VdV}{\frac{T}{m} - \mu g - \frac{\rho V^2}{2m} S\left(C_D - \mu C_L\right)}.$$

Većina promenljivih je poznata, ali je još potrebno izračunati:

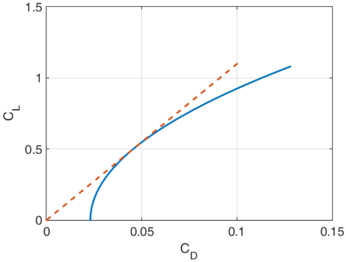
$$V_{\min} = \sqrt{\frac{2mg}{\rho S C_{L_{\max}}}},\ V_p \approx 1.1 V_{\min}.$$

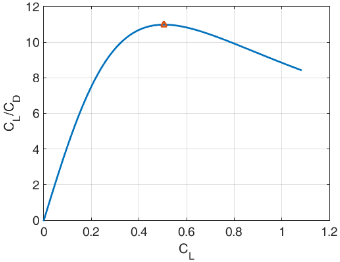
Potrebni aerodinamički koeficijenti (uzgona i otpora) mogu se naći iz uslova $(C_L/C_D)_{\max}$:

$$\frac{d(C_L/C_D)}{dC_L} = 0 \Rightarrow 1 \cdot (C_{D_o} + \kappa C_L^2) - C_L \cdot 2\kappa C_L = 0 \Rightarrow C_L = \sqrt{\frac{C_{D_o}}{\kappa}}.$$

Zadatak 2 – Zalet (kao jedna od faza poletanja)

Prethodno pomenute aerodinamičke koeficijente moguće je skicirati:





Optimalne vrednosti finese ostvaruju se pri

$$C_L = \sqrt{\frac{C_{D_o}}{\kappa}}, \; C_D = 2C_{D_o}.$$

Zadatak 2 – Kod

```
% proracun zaleta lakog lovca bombardera
clear all, clc

% ulazni podaci
m = 5100;       % masa, [kg]
g = 9.81;       % ubrzanje zemljine teze, [m/s^2]
rho = 1.225;    % gustina, [kg/m^3]
S = 16.5;       % povrsina krila, [m^2]
T = 22400;      % potisak, [N]
Clmax = 1.08;   % max koeficijent uzgona
mu = 0.05;      % koeficijent trenja piste

% proracun
Vmin = sqrt(2*m*g/rho/S/Clmax);
Vp = 1.1*Vmin;  % brzina u trenutku odvajanja od tla

% aerodinamicki koeficijenti
Cdo = 0.023; kappa = 0.09;
CL = sqrt(Cdo/kappa); CD = 2*Cdo;
CL_CD = CL/CD;

% zalet
V = linspace(0, Vp, 101);   % promena brzine, [m/s]
dV = V(2) - V(1);           % prirastaj brzine, [m/s]
fV = V./(T/m - mu*g - rho*V.^2/2/m*S*(CD - mu*CL));     % fV = f(V)
fVsr = (fV(1:end-1) + fV(2:end))/2; % srednja vrednost
Sg = sum(fVsr*dV);          % zalet kao suma
disp('Zalet je [m]')
disp(Sg)
```

Zadatak 2 – Zalet (kao jedna od faza poletanja)

Problem je moguće rešiti i *Metodom konačnih razlika* . . .
Nezavisna promenljiva je vreme t, a proračun se vrši dok brzina ne dostigne vrednost V_p.
Unutar ciklične petlje potrebno je sprovesti sledeće proračune:

$$
\begin{aligned}
a_{i+1} &= \frac{T}{m} - \mu g - \frac{\rho V_i^2}{2m} S \left(C_D - \mu C_L \right), \\
V_{i+1} &= V_i + a_{i+1} dt, \\
s_{i+1} &= s_i + V_{i+1} dt.
\end{aligned}
$$

u svakoj iteraciji.

Zadatak 2 – Kod (2. način)

```
% zalet, 2. nacin
N = 10001;                  % broj podela
t = linspace(0, 60, N);     % vreme, [s]
dt = t(2) - t(1);           % vremenski korak, [s]
% inicijalizacija potrebnih nizova
a = zeros(1, length(t));    % ubrzanje, [m/s^2]
V = zeros(1, length(t));    % brzina, [m/s]
s = zeros(1, length(t));    % predjeni put, [m]
% u pocetnom trenutku (i = 1) letelica miruje
i = 2;
while i<=N && V(i-1)<Vp
    a(i) = T/m - mu*g - rho*V(i-1)^2/2/m*S*(CD - mu*CL);
    V(i) = V(i-1) + a(i)*dt;
    s(i) = s(i-1) + V(i)*dt;
    i = i + 1;
end
Sg = s(i-1);
disp('Zalet je [m]')
disp(Sg)
tp = t(i-1);
disp('Vreme trajanja zaleta je [s]')
disp(tp)

figure
plot(s(1:i-1), V(1:i-1), 'linewidth', 2);
hold on, plot(s(1:i-1), Vp*ones(1,i-1), '--', 'linewidth', 2);
xlabel('s [m]'), ylabel('V [m/s]')
grid on, set(gca, 'fontsize', 14)
```

Sistem običnih diferencijalnih j-na

Sve prethodno rečeno o *Metodi konačnih razlika* može se primeniti i pri rešavanju sistema običnih diferencijalnih j-na (ODJ):

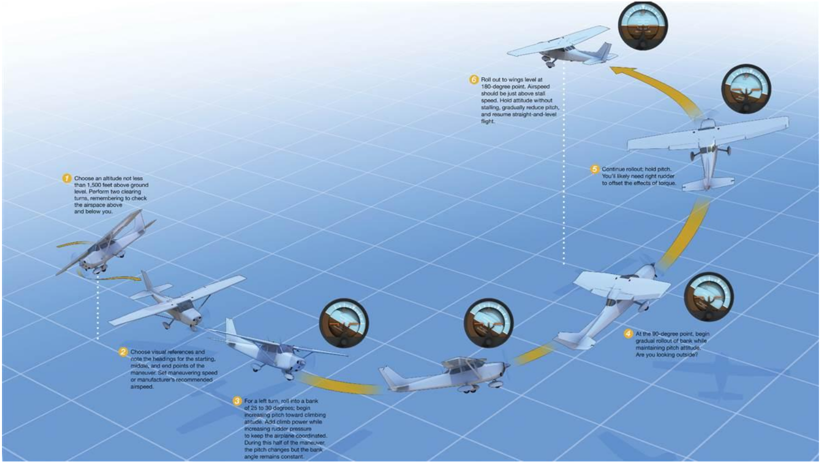
$$\begin{aligned}\frac{\mathrm{d}y_1}{\mathrm{d}x} &= f_1(x, y_1, y_2, \ldots, y_n)\\ \frac{\mathrm{d}y_2}{\mathrm{d}x} &= f_2(x, y_1, y_2, \ldots, y_n)\\ &\vdots\\ \frac{\mathrm{d}y_n}{\mathrm{d}x} &= f_n(x, y_1, y_2, \ldots, y_n).\end{aligned}$$

Iako postoje naprednije metode, u prvom trenutku može se koristiti i Ojlerova šema. Treba voditi računa o sprezanju promenljivih i redosledu rešavanja j-na.

Borbeni zaokret

- *Borbeni zaokret* je neustaljeni, prostorni, penjući (promena visine) zaokret s promenom pravca leta (za 180°) u što kraćem vremenu i uz što veće povećanje visine (i s izlaznom brzinom manjom za 30-40% od početne).
- Ovaj manevar se upotrebljava u vazdušnim borbama kada protivniku treba zaći za "rep" sa nadvišenjem (koje se potom, prevodjenjem aviona u poniranje, lako koristi za povećanje brzine).
- Povećanje visine ΔH je jedna od najvažnijih karakteristika borbenog zaokreta.
- Treba naglasiti da je u pitanju prostorni manevar, odnosno da je njegovo analitičko rešavanje komplikovano.
- Za formiranje fizičkog modela koristi se sistem ODJ (veličine 3).

Borbeni zaokret



Borbeni zaokret

J-ne koje opisuju borbeni zaokret (u polubrzinskom koordinatnom sistemu):

$$
\begin{aligned}
m\frac{\mathrm{d}V}{\mathrm{d}t} &= T\cos(\alpha-\alpha_s) - R_x - mg\sin\gamma \\
mV\frac{\mathrm{d}\gamma}{\mathrm{d}t} &= [R_z + T\sin(\alpha-\alpha_s)]\cos\phi - mg\cos\gamma \\
mV\cos\gamma\frac{\mathrm{d}\chi}{\mathrm{d}t} &= [R_z + T\sin(\alpha-\alpha_s)]\sin\phi
\end{aligned}
$$

gde su: ϕ ugao bočnog nagiba, χ ugao skretanja putanje tokom borbenog zaokreta i γ ugao penjanja aviona.

Zadatak 3 – Borbeni zaokret (sistem ODE)

Laki lovac bombarder, mase $m = 5100$ kg i površine krila 16.5 m^2 ima mlazni motor koji razvija potisak od $T = 2240$ daN. Polara "čistog" aviona može se predstaviti izrazom:

$$C_D = C_{D_o} + \kappa C_L^2 = 0.02 + 0.07 C_L^2.$$

Odrediti visinsku razliku $\Delta H = H_2 - H_1$, vreme trajanja t_z i izlaznu brzinu V_2 nakon borbenog zaokreta ukoliko avion u njega ulazi pri brzini $V_1 = 300$ m/s na visini $H_1 = 1000$ m. Može se pretpostaviti da pilot tokom borbenog zaokreta održava optimalni napadni ugao α_{opt}.

Razmotriti kako na performanse u zaokretu utiče ugao bočnog nagiba ϕ.

Zadatak 3 – Borbeni zaokret

```
% proracun borbenog zaokreta lakog lovca bombardera
clear all, clc

% ulazni podaci
m = 5100;       % masa, [kg]
g = 9.81;       % ubrzanje zemljine teze, [m/s^2]
S = 16.5;       % povrsina krila, [m^2]
T = 22400;      % potisak, [N]
V1 = 300;       % brzina pri ulasku u borbeni zaokret, [m/s]
H1 = 1000;      % visina pri ulasku u borbeni zaokret, [m]
rho0 = 1.225;   % gustina na 0m, [kg/m^3]
% aerodinamicki koeficijenti
Czmax = 1.08;   % max koeficijent uzgona
Cxo = 0.020; kappa = 0.07;
Cz = sqrt(Cxo/kappa); Cx = 2*Cxo;
Cz_Cx = Cz/Cx;

% proracun
N = 10001;                  % broj podela
t = linspace(0, 20, N);     % vreme, [s]
dt = t(2) - t(1);           % vremenski korak, [s]
% inicijalizacija potrebnih nizova
V = zeros(1, length(t));    % brzina, [m/s]
V(1) = V1;
h = zeros(1, length(t));    % visina, [m]
h(1) = H1;
gamma = zeros(1, length(t));% ugao penjanja, [rad]
chi = zeros(1, length(t)); % ugao skretanja, [rad]
phi = pi/6;                 % bocni nagib, [rad]
nx = zeros(1, length(t)); % koeficijent tangencijalnog opterecenja
n = zeros(1, length(t)); % koeficijent normalnog opterecenja
% u pocetnom trenutku (i = 1) letelica ima brzinu V1 na visini H1
i = 2;
```

Zadatak 3 – Borbeni zaokret (nastavak)

```
while i<=N && chi(i-1)<pi
    % pomocne promenljive
    rho = rho0*(1-2.255906e-5*h(i-1))^4.265;
    Rx = 0.5*rho*V(i-1)^2*S*Cx;
    Rz = 0.5*rho*V(i-1)^2*S*Cz;
    % koeficijenti opterecenja
    nx(i) = (T*1 - Rx)/m/g;
    n(i) = (Rz + T*0)/m/g;
    % nepoznate velicine
    gamma(i) = gamma(i-1) + dt*(n(i)*cos(phi) - cos(gamma(i-1)))*g/V(i-1);
    chi(i) = chi(i-1) + dt*n(i)*sin(phi)*g/(V(i-1)*cos(gamma(i)));
    V(i) = V(i-1) + dt*(nx(i) - sin(gamma(i)))*g;
    h(i) = h(i-1) + dt*V(i)*sin(gamma(i));
    i = i + 1;
end
dh = h(i-1) - h(1);       % dh*2*g/V1^2
disp('Visinska razlika je [m]')
disp(dh)
tz = t(i-1);              % tz*g/V1
disp('Vreme trajanja zaokreta je [s]')
disp(tz)
V2 = V(i-1);
disp('Brzina na kraju zaokreta je [m/s]')
disp(V2)

figure
plot(t(1:i-1), V(1:i-1), 'linewidth', 2);
xlabel('t [s]'), ylabel('V [m/s]')
grid on, set(gca, 'fontsize', 14)
```