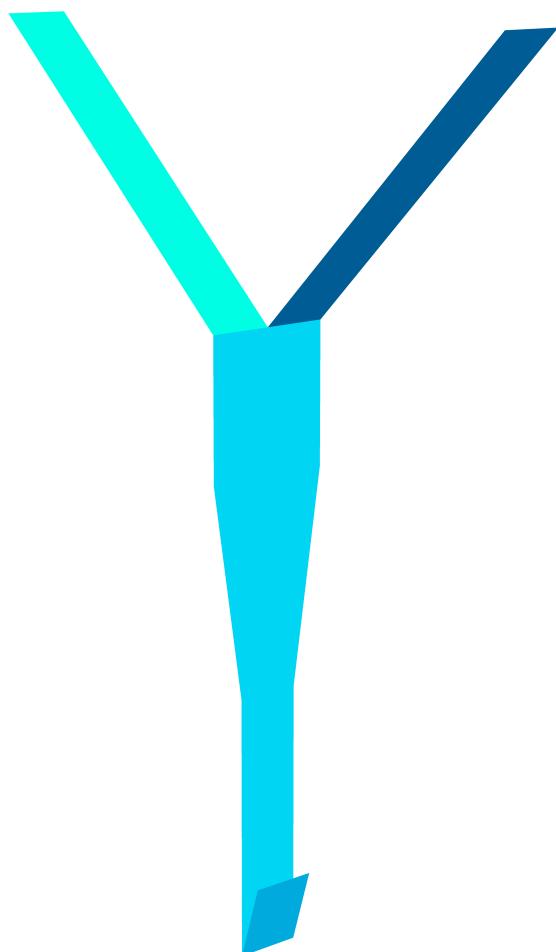


RAZISKOVALNA NALOGA

# **FIZKIALNI MODEL LETA PAPIRNIH LETAL**

**FIZIKA**

**2021**



## **ZAHVALA**

Zahvaljujem se mentorju prof. dr. Danielu Svenšku za napotke, ki so mi bili v pomoč pri raziskovalni nalogi. Somentorici Marjani Benedik se zahvaljujem za napotke pri oblikovanem urejanju raziskovalne naloge. Zahvaljujem se tudi moji družini, ki me je podpirala in mi pomagala.

# KAZALO

<b>1</b>	<b>POVZETEK</b>	<b>6</b>
<b>2</b>	<b>UVOD</b>	<b>7</b>
<b>3</b>	<b>TEORETIČNE OSNOVE SIMULACIJE LETAL</b>	<b>8</b>
3.1	RAZDELITEV LETALA NA TRIKOTNIKE . . . . .	8
3.2	MASA TRIKOTNIKA IN PLOSKV LETALA . . . . .	11
3.3	TEŽIŠČE TRIKOTNIKA IN PLOSKV LETALA . . . . .	11
3.4	VZTRAJNOSTNI MOMENT TRIKOTNIKA IN PLOSKVE LETALA . . . . .	12
3.5	SILA ZRAKA NA TRIKOTNIK IN PLOSKV LETALA . . . . .	14
3.6	NAVOR ZRAKA NA TRIKOTNIK . . . . .	16
3.7	GIBANJE ENE PLOSKVE . . . . .	17
3.8	SILE VZMETI MED PLOSKVAMI . . . . .	21
3.9	NAVORI PREGIBOV MED PLOSKVAMI . . . . .	22
3.10	GIBANJE LETALA . . . . .	24
3.11	POPRAVEK Z RAVNANJEM PLOSKEV . . . . .	27
3.12	POPRAVEK Z OHRANITVIJO ENERGIJE . . . . .	27
3.13	POPRAVEK Z DUŠENJEM VZMETI . . . . .	29
3.14	SNEMALNI KOT KAMERE . . . . .	31
<b>4</b>	<b>SLEDENJE LETALOM IZ POSNETKA</b>	<b>32</b>
<b>5</b>	<b>PIRAMIDNO LETALO</b>	<b>36</b>
<b>6</b>	<b>EKSPEIMENTALNI DEL</b>	<b>39</b>
6.1	REZULTATI ZA PIRAMIDNA LETALA . . . . .	42
6.2	REZULTATI ZA HELIKOPTER LETALO . . . . .	44
6.3	REZULTATI ZA W LETALO . . . . .	46
<b>7</b>	<b>RAZPRAVA IN ZAKLJUČEK</b>	<b>47</b>
<b>8</b>	<b>VIRI IN LITERATURA</b>	<b>48</b>
<b>9</b>	<b>PRILOGE</b>	<b>49</b>
9.1	SIMULACIJA GIBANJA ENE PLOSKVE . . . . .	49
9.2	SIMULACIJA GIBANJA PAPIRNEGA LETALA . . . . .	50
9.3	ff-MPEG SKRIPTA ZA IZVOZ VIDEO-POSNETKA IZ SLIK . . . . .	51
9.4	PROGRAM ZA SLEDENJE PIKAM . . . . .	52
9.5	PROGRAMI ZA IZRIS MREŽ LETAL . . . . .	53



# KAZALO SLIK IN GRAFIK

1	obračanje ploskve iz 2D v 3D . . . . .	8
2	triangulacija0 . . . . .	9
3	triangulacija1 . . . . .	10
4	trikotnik . . . . .	11
5	trikotnik na koščke . . . . .	12
6	Fu . . . . .	14
7	T-ploskva . . . . .	19
8	slika Mp . . . . .	23
9	poševno nihalo . . . . .	25
10	zavorna sila vzmeti . . . . .	29
11	kot kamere . . . . .	31
12	RGB vektorji . . . . .	32
13	sledenje gibanju v prostoru . . . . .	34
14	piramidni . . . . .	36
15	sile na piramidno letalo . . . . .	37
16	izrezana letala . . . . .	39
17	mehanizem za spuščanje papirnih letal . . . . .	40
18	kamera in merjenje delta Y . . . . .	41
19	Vmax meritve piramidni . . . . .	42
20	$z(t)$ helikopterja . . . . .	44
21	W letalo . . . . .	46

# 1 POVZETEK

V nalogi obravnavamo gibanje enoslojnih papirnih letal iz ravnih ploskev. Za gibanje preprostega primera piramidnega letala je izpeljana analitična rešitev. Za gibanje splošnega enoslojnega papirnega letala iz ravnih ploskev pa je sprogramirana fizikalna simulacija. Sprogramirano je tudi sledenje gibanju papirnih letal. To je kasneje uporabljeno pri analizi posnetkov leta papirnih letal. Na koncu raziskovalne naloge je eksperimentalno preverjeno, kako dobro deluje model. Vsi programi so napisani v jeziku Wolfram Mathematica.

## KLJUČNE BESEDE

Papirna letala, sila curka, vztrajnostni moment, navor pregibov, simulacija, Wolfram Mathematica, video analiza, piramidno letalo, helikopter letalo, W letalo

## KEY WORDS

Paper planes, stream force, moment of inertia, bending torque, simulation, Wolfram Mathematica, video analysis, pyramid plane, helicopter plane, W plane

## 2 UVOD

Vedno sem se rad ukvarjal s papirnimi letali, saj so se mi zdela zabavna in fizikalno zanimiva. Na prvi pogled se zdijo papirna letala nekaj enostavnega, vendar pa so fizikalno zelo zapletena za obravnavo. Šele letos sem znal dovolj fizike, matematike in programiranja, da sem se lahko lotil fizikalnega opisa zelo preprostih papirnih letal.

Moj prvi cilj je bil splošna simulacija vsakega možnega papirnega letala, s pravim pretkom zraka. Naloga se je izkazala za neizvedljivo s srednješolskim znanjem. Zato sem se omejil na papirna letala iz enega sloja papirja, z ravnimi ploskvami in približkom mirujočega zraka okoli letala. S tem približkom lahko nekatera letala opišemo presenetljivo dobro, drugih pa ne moremo. Cilji, ki sem si jih zastavil in jih tudi izpolnil so bili

- Simulacija papirnih letal z zgornjimi predpostavkami
- Sprogramirano sledenje gibanju papirnih letal iz posnetka
- Eksperimentalno preverjanje modela

Programiral sem v jeziku Wolfram Mathematica, saj ima ta že ugrajenih veliko matematičnih funkcij, funkcij za delo s seznamimi, simbolično računanje, izris grafik, skratka vse, kar sem za to raziskovalno nalogu potreboval. Za enostavnejše programiranje sem uporabil tudi aplikacijo Touch Portal in makro gumbe na miški, v katere sem shranil ukaze (funkcije) in njihove pogoste kombinacije. Tudi vse skice v raziskovalni nalogi so narejene v Mathematici. Za pisanje raziskovalne naloge sem uporabil program LaTeX.

Za simulacijo je bilo potrebnih več korakov. Najprej sem ugotovil algoritem, ki papirno letalo razdeli na trikotnike. Nato sem izpeljal enačbe za maso, težišče in vztrajostni moment posamezne ploskve papirnega letala ter silo in navor zraka nanjo. Nato sem izpeljal še enačbe za medsebojne sile in navore med ploskvami. Vse te elemente sem potem povezal v simulacijo. Za boljše delovanje simulacije, sem raziskoval tudi tri različne popravke.

Za eksperimentalni del sem izpeljal enačbe za sledenje gibanju papirnega letala iz posnetka, tudi prostorskemu, ki pa v praksi deluje le pri zelo dobrih pogojih. Nato sem v eksperimentalnem delu na primeru treh papirnih letal ugotavljal kako dobro deluje simulacija.

Naloga je izvirna, saj so vsi obstoječi opisi papirnih letal, ki se jih je dalo najti le kvalitativni in ne fizikalni.

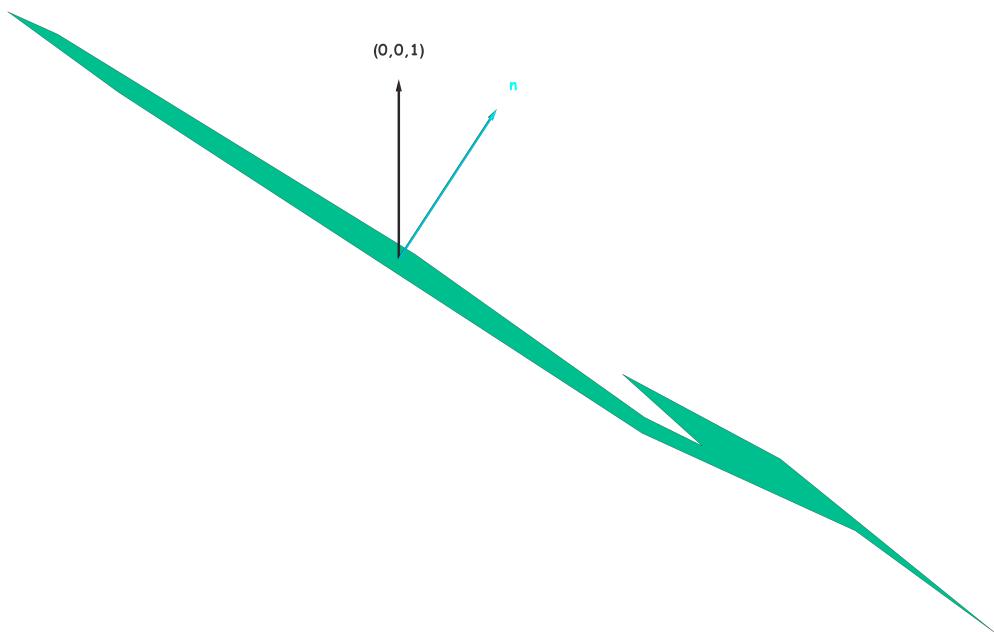
Z raziskovanjem bom nadaljeval, saj se da simulacijo izboljšati še na veliko načinov. Nekaj jih je naštetih v zaključku.

### 3 TEORETIČNE OSNOVE SIMULACIJE LETAL

#### 3.1 RAZDELITEV LETALA NA TRIKOTNIKE

Ukvarjali se bomo s papirnimi letali, ki imajo ravna krila. Temu najbolj ustreza letala iz trdega papirja, ki letijo pri nizki hitrosti. Če je letalo sestavljeno iz samih ravnih ploskev, lahko vsako od teh razdelimo na trikotnike. Razdelitev nam bo v pomoč pri nadaljnji obravnavi letala.

Treba je ugotoviti algoritem, ki bo vsako ploskev razdelil na trikotnike. Ploskve bomo v simulaciji podali kot seznam zaporednih oglišč v prostoru. Lažje bi bilo, če bi bila ploskev podana v  $2D$ , zato moramo najprej spremeniti koordinate oglišč ploskev v  $2D$ . To lahko naredimo tako, da zasučemo ploskev v vodoravni položaj, z množenjem vektorjev leg oglišč ploskev z ustrezno rotacijsko matriko. To je matrika transformacije, ki zavrti normalni vektor ploskeve  $\hat{n}$  v vektor  $(0, 0, 1)$  (glej sliko).

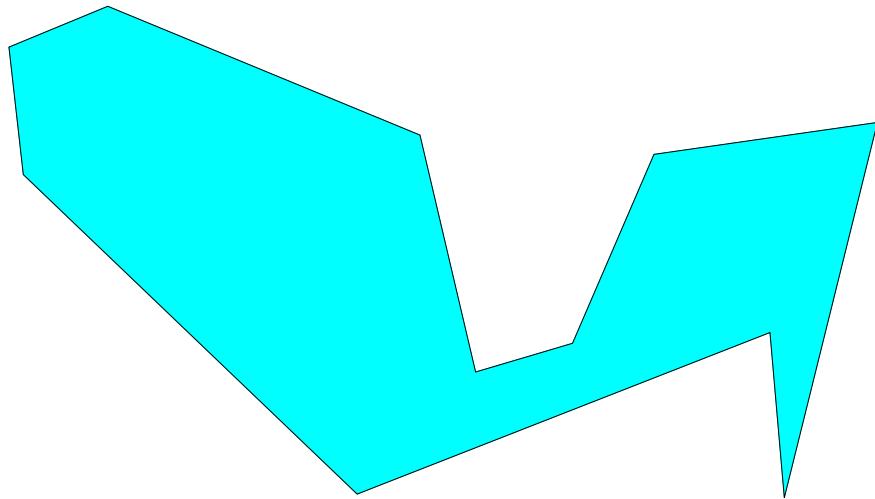


Slika 1: obračanje ploskev iz  $2D$  v  $3D$

Wolfram Mathematica, v kateri so narejene simulacije, ima že vgrajeno funkcijo `RotationMatrix[u,v]`, ki z numeričnimi metodami poišče rotacijsko matriko transformacije, ki zavrti vektor  $u$  tako, da kaže v smeri vektorja  $v$ .

Po množenju oglišč z omenjeno matriko problem razdelitve večkotnika na trikotnike postane  $2D$ .

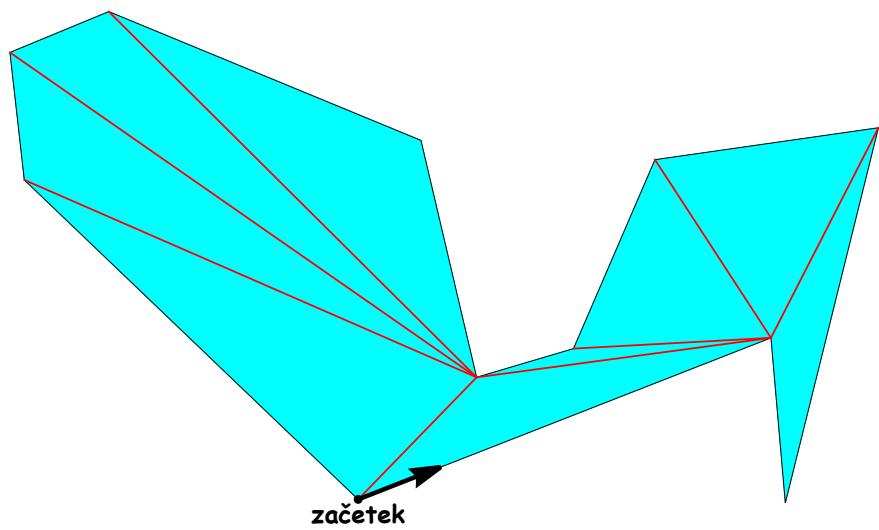
Če moramo razdeliti na trikotnike naslednji večkotnik in to poskusimo narediti s svinčnikom na papir, je naloga precej očitna.



Slika 2: triangulacija0

Če razmislimo, kako smo se odločali, kje bomo narisali črto, dobimo preprost algoritem. Zamislimo si, da stojimo v enem oglišču. V nasprotni smeri urinega kazalca gremo po eno oglišče naprej od oglišča do oglišča. Pri vsakem ugotovimo, ali prejšnje, to in naslednje oglišče tvorijo trikotnik, ki ga lahko odrežemo od večkotnika. Če v nekem oglišču zavijemo v desno, tega trikotnika ne moremo odrezati, saj je to "vdolbina" v večkotnik. Če zavijemo v levo, pa moramo še preveriti, ali morebitna povezava seka katero izmed ostalih stranic. Če je ne, lahko odrežemo ta trikotnik. S tem smo dobili nov večkotnik. Z dobljenim večkotnikom proces ponavljamo, dokler nam ne ostane trikotnik. Za vsak dobljeni trikotnik na koncu še ugotovimo indekse njegovih oglišč, med oglišči začetnega večkotnika in jih zberemo v seznam.

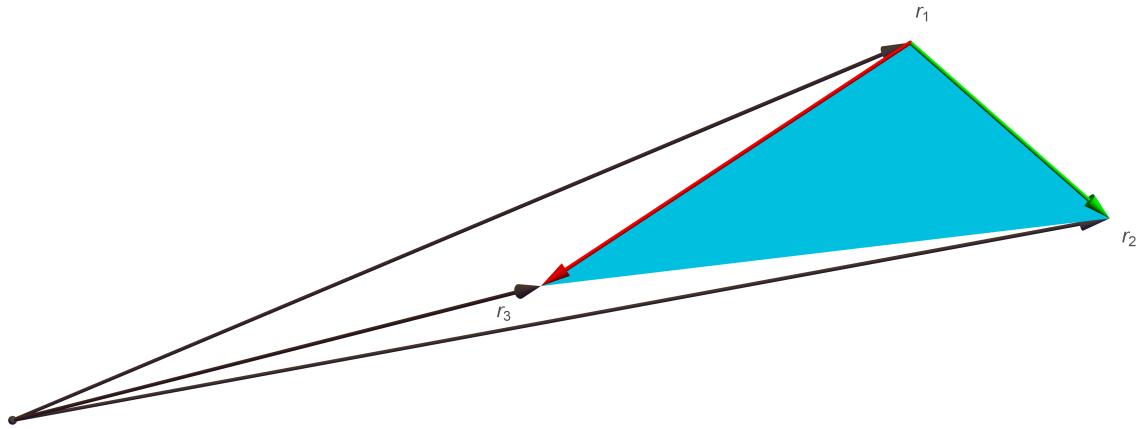
Naslednja slika prikazuje razdelitev prejšnjega večkotnika na trikotnike.



Slika 3: triangulacija1

## 3.2 MASA TRIKOTNIKA IN PLOSKEV LETALA

Trikotnik naj bo podan z oglišči  $r_1$ ,  $r_2$  in  $r_3$  (glej sliko).



Slika 4: trikotnik

Zeleni vektor lahko izrazimo kot  $r_2 - r_1$ , rdečega pa kot  $r_3 - r_1$ .

Recimo, da je letalo narejeno iz papirja s ploščinsko gostoto  $\rho_p$ . Ta je natančno izmerjena in podana za vsak papir. Masa posamezne ploskve letala je vsota mas trikotnikov:

$$\sum_i \rho_p S_i = \sum_i \rho_p \frac{|(r_{2i} - r_{1i}) \times (r_{3i} - r_{1i})|}{2} = \frac{\rho_p}{2} \sum_i |(r_{2i} - r_{1i}) \times (r_{3i} - r_{1i})|$$

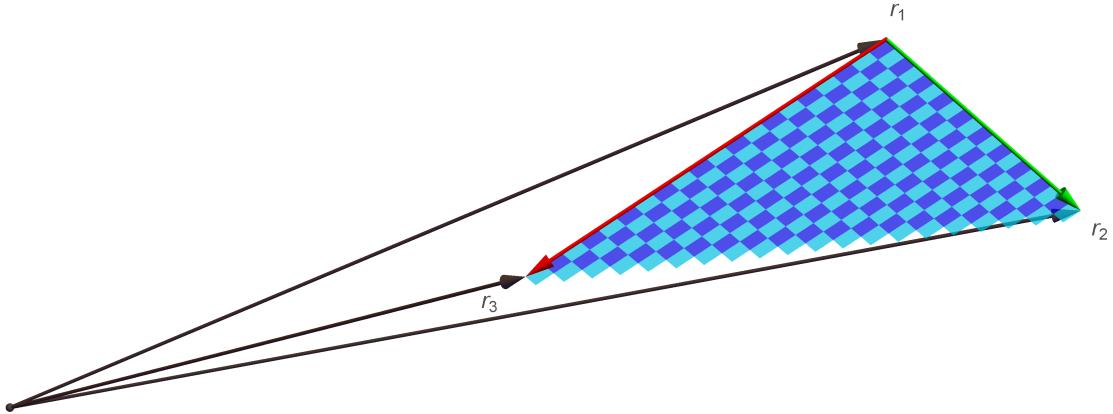
## 3.3 TEŽIŠČE TRIKOTNIKA IN PLOSKEV LETALA

Za izračun težišča uporabimo znano dejstvo, da je težišče trikotnika na povprečju njegovih oglišč  $r_{*i} = \frac{r_{1i} + r_{2i} + r_{3i}}{3}$ . Težišče celotne ploskve je zato:

$$r_* = \frac{\sum_i m_i r_{*i}}{\sum_i m_i} = \frac{\sum_i |(r_{2i} - r_{1i}) \times (r_{3i} - r_{1i})| (r_{1i} + r_{2i} + r_{3i})}{3 \sum_i |(r_{2i} - r_{1i}) \times (r_{3i} - r_{1i})|}$$

### 3.4 VZTRAJNOSTNI MOMENT TRIKOTNIKA IN PLOSKVE LETALA

Trikotnik lahko razdelimo na neskončno majhne koščke (paralelograme), kot kaže slika.



Slika 5: trikotnik na koščke

Recimo, da smo razdelili trikotnik na koščke z mrežo, sestavljenou iz  $\frac{1}{dC_1}$  vrstic, vzporednih rdečemu vektorju in  $\frac{1}{dC_2}$  zelenemu vektorju.

Ker gresta  $dC_1$  in  $dC_2$  proti nič, lahko lego koščka glede na  $r_*$  opišemo z vektorjem:

$$\Delta \mathbf{r}(C_1, C_2) = \mathbf{r}_1 + C_1(\mathbf{r}_2 - \mathbf{r}_1) + C_2(\mathbf{r}_3 - \mathbf{r}_1) - \mathbf{r}_*$$

Kjer je:

- $\mathbf{r}_*$  pozicijski vektor težišča ploskve, katere del je ta trikotnik
- $C_1$  delež zelenega vektorja
- $C_2$  delež rdečega vektorja

Če si v  $2D$  predstavljamo, da je zeleni vektor  $\hat{i}$ , rdeči pa  $\hat{j}$ , se lahko hitro prepričamo, da za točke v trikotniku velja  $C_1 + C_2 \leq 1$ .

Zdaj pa ugotovimo maso  $dm$  enega koščka. Postavimo se v  $2D$ . Ploščina kvadratka, ki ga oklepata  $\hat{i}$  in  $\hat{j}$  je 1. Ploščina paralelograma, ki ga v prostoru oklepata zeleni in rdeči vektor je torej  $|(\mathbf{r}_{2i} - \mathbf{r}_{1i}) \times (\mathbf{r}_{3i} - \mathbf{r}_{1i})|$  krat večja od tiste v  $2D$ . Ploščina koščka bi bila v  $2D$  enaka  $dC_1 dC_2$ . Zato velja:

$$dm = \rho_p |(\mathbf{r}_{2i} - \mathbf{r}_{1i}) \times (\mathbf{r}_{3i} - \mathbf{r}_{1i})| dC_1 dC_2$$

S tem lahko izrazimo vztrajnostni moment  $dJ$  enega koščka:

$$dJ = \begin{bmatrix} (\Delta \mathbf{r} \cdot \hat{y})^2 + (\Delta \mathbf{r} \cdot \hat{z})^2 & -(\Delta \mathbf{r} \cdot \hat{x})(\Delta \mathbf{r} \cdot \hat{y}) & -(\Delta \mathbf{r} \cdot \hat{x})(\Delta \mathbf{r} \cdot \hat{z}) \\ -(\Delta \mathbf{r} \cdot \hat{x})(\Delta \mathbf{r} \cdot \hat{y}) & (\Delta \mathbf{r} \cdot \hat{x})^2 + (\Delta \mathbf{r} \cdot \hat{z})^2 & -(\Delta \mathbf{r} \cdot \hat{y})(\Delta \mathbf{r} \cdot \hat{z}) \\ -(\Delta \mathbf{r} \cdot \hat{x})(\Delta \mathbf{r} \cdot \hat{z}) & -(\Delta \mathbf{r} \cdot \hat{y})(\Delta \mathbf{r} \cdot \hat{z}) & (\Delta \mathbf{r} \cdot \hat{x})^2 + (\Delta \mathbf{r} \cdot \hat{y})^2 \end{bmatrix} dm$$

Vztrajnostni moment trikotnika je vsota vztrajnostnih momentov vseh neskončno majhnih koščkov, na katere smo ga razdelili, torej integral:

$$J_i = \int dJ = \int_0^1 \int_0^{1-C_1} \begin{bmatrix} & & \\ & . & \\ & & . \end{bmatrix} dm = \\ = \rho_p |(\mathbf{r}_{2i} - \mathbf{r}_{1i}) \times (\mathbf{r}_{3i} - \mathbf{r}_{1i})| \int_0^1 \int_0^{1-C_1} \begin{bmatrix} & & \\ & . & \\ & & . \end{bmatrix} dC_2 dC_1$$

Vztrajnostni moment ploskve pa je vsota vztrajnostnih momentov trikotnikov, ki jo sestavljajo

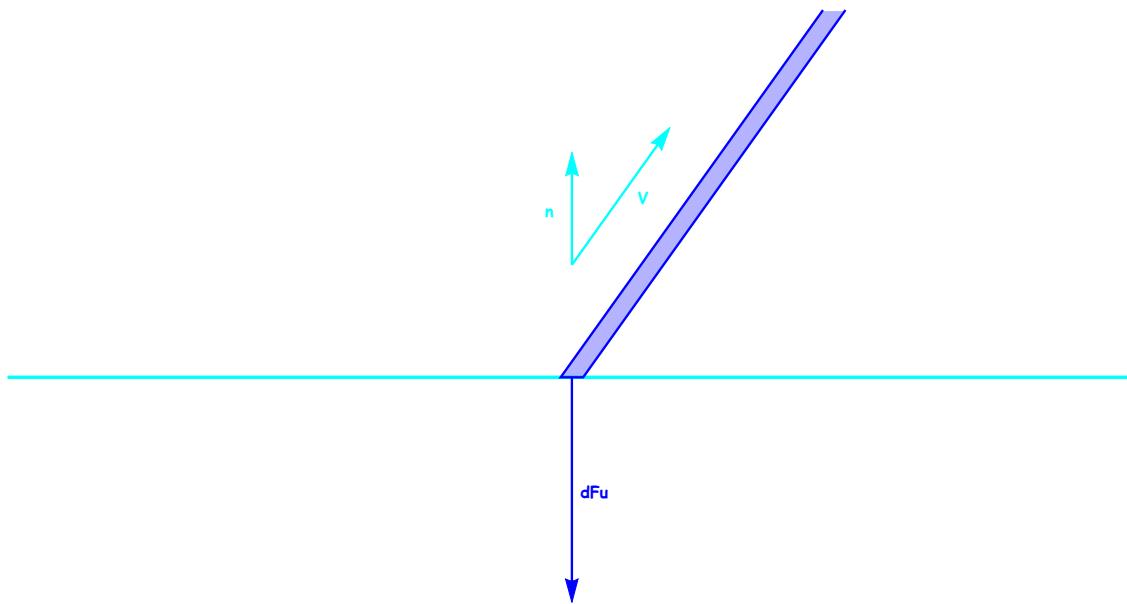
$$J = \sum_i J_i = \rho_p \sum_i |(\mathbf{r}_{2i} - \mathbf{r}_{1i}) \times (\mathbf{r}_{3i} - \mathbf{r}_{1i})| \int_0^1 \int_0^{1-C_1} \begin{bmatrix} & & \\ & . & \\ & & . \end{bmatrix} dC_2 dC_1$$

Ko to računalnik integrira in poenostavi, dobimo zelo dolgo enačbo. To najdete v prilogi.

### 3.5 SILA ZRAKA NA TRIKOTNIK IN PLOSKEV LETALA

Za izračun sile in navora zraka na trikotno ploskev moramo, kot smo to naredili že pri izračunu vztrajnognega momenta, integrirati prispevke po celi ploskvi. Da lahko izračunamo te prispevke, moramo poznati tlak in gradient hitrosti zraka glede na ploskev, v vsaki točki na obeh straneh ploskve. Da bi to dvoje poznali, bi morali narediti simulacijo pretoka zraka okoli letala. Takšne simulacije se dela z numeričnim reševanjem Navier-Stoksovih enačb. To je zelo zapleteno in presega gimnazjsko znanje, poleg tega pa so take simulacije tudi zelo počasne na računalniku. Zato sem v raziskovalni nalogi vzel velik približek, da zrak okoli letala miruje in lahko silo in navor zraka izračunamo kot silo in navor curka zraka. Približek v veliko situacijah zadošča, saj je viskoznost zraka relativno majhna.

Recimo, da se točka na ploskvi premika s hitrostjo  $\mathbf{V}$ . Potem po približku mirujočega zraka, zrak na ploskev v tej točki vpada s hitrostjo  $-\mathbf{V}$  (glej spodnjo sliko).



Slika 6:  $F_u$

Uporabili bomo enačbo za silo curka

$$F_c = \rho S V^2$$

V našem primeru bo to

$$dF_{zi} = \rho (\mathbf{V} \cdot \hat{n})^2 dS$$

Sila curka  $d\mathbf{F}_{zi}$  je vektor, ki je usmerjen vzporedno vektorju  $\hat{n}$ , stran od  $\mathbf{V}$ . To z enačbo lahko zapišemo takole:

$$d\mathbf{F}_{zi} = \begin{cases} \mathbf{V} \cdot \hat{n} \leq 0, 1 \\ \mathbf{V} \cdot \hat{n} > 0, -1 \end{cases} \quad dF_{zi}\hat{n} = \begin{cases} \mathbf{V} \cdot \hat{n} \leq 0, 1 \\ \mathbf{V} \cdot \hat{n} > 0, -1 \end{cases} \rho (\mathbf{V} \cdot \hat{n})^2 \hat{n} dS$$

Hitrost  $\mathbf{V}$  gibanja točke na trikotni ploskvi pa je  $\mathbf{V} = \mathbf{V}_* + \omega \times \Delta\mathbf{r}$ , kjer  $\Delta\mathbf{r}$  izrazimo tako, kot prej pri izračunu vztrajnostnega momenta trikotnika

$$\Delta\mathbf{r} = C_1(\mathbf{r}_2 - \mathbf{r}_1) + C_2(\mathbf{r}_3 - \mathbf{r}_1) + \mathbf{r}_1 - \mathbf{r}_*$$

Tudi celotno silo upora na trikotnik  $\mathbf{F}_{zi}$  izračunamo, kot smo vztrajnostni moment trikotnika

$$\begin{aligned} \mathbf{F}_{zi} &= \int d\mathbf{F}_{zi} = \int_0^1 \int_0^{1-C_1} \begin{cases} \mathbf{V} \cdot \hat{n} \leq 0, 1 \\ \mathbf{V} \cdot \hat{n} > 0, -1 \end{cases} \rho (\mathbf{V} \cdot \hat{n})^2 \hat{n} dS = \\ &= \rho |(\mathbf{r}_{2i} - \mathbf{r}_{1i}) \times (\mathbf{r}_{3i} - \mathbf{r}_{1i})| \hat{n} \\ &\quad \int_0^1 \int_0^{1-C_1} \begin{cases} \mathbf{V}(C_1, C_2) \cdot \hat{n} \leq 0, 1 \\ \mathbf{V}(C_1, C_2) \cdot \hat{n} > 0, -1 \end{cases} (\mathbf{V}(C_1, C_2) \cdot \hat{n})^2 dC_2 dC_1 \end{aligned}$$

Za ta integral ne poznamo analitične rešitve, lahko pa ga računalnik izračuna numerično. Ker je v simulaciji velikokrat izračunana sila upora  $\mathbf{F}_{zi}$  na trikotnike, to simulacijo zelo upočasnuje. Lahko pa vzamemo približek, pri katerem zanemarimo vrtenje trikotnika. To simulacijo pospeši in da rezultate, ki se v preizkušenih primerih ujemajo na tri mesta natančno. Če vzamemo ta približek, je hitrost  $(C_1, C_2) = \mathbf{V}_*$  povsod na trikotniku. Zato bo izraz za  $\mathbf{F}_{zi}$  tak, kot za  $d\mathbf{F}_{zi}$

$$\mathbf{F}_{zi} = \begin{cases} \mathbf{V}_* \cdot \hat{n} \leq 0, 1 \\ \mathbf{V}_* \cdot \hat{n} > 0, -1 \end{cases} \rho (\mathbf{V}_* \cdot \hat{n})^2 S \hat{n}$$

Kot že prej, je tudi sila zraka na ploskev enaka vsoti sil zraka na trikotnike, ki jo sestavljajo.

$$\mathbf{F}_z = \sum_i \mathbf{F}_{zi}$$

### 3.6 NAVOR ZRAKA NA TRIKOTNIK

Navor zraka na trikotnik  $M_{zi}$  bomo izračunali na enak način, kot silo zraka  $F_{zi}$  na trikotnik

$$M_{zi} = \int dM_{zi} = \int \Delta r \times dF_{zi} =$$

$$= \rho |(\mathbf{r}_{2i} - \mathbf{r}_{1i}) \times (\mathbf{r}_{3i} - \mathbf{r}_{1i})|$$

$$\int_0^1 \int_0^{1-C_1} \left( \begin{cases} \mathbf{V}(C_1, C_2) \cdot \hat{n} \leq 0, 1 & (\mathbf{V}(C_1, C_2) \cdot \hat{n})^2 \Delta r(C_1, C_2) \times \hat{n} \\ \mathbf{V}(C_1, C_2) \cdot \hat{n} > 0, -1 \end{cases} \right) dC_2 dC_1$$

Tudi ta integral ni analitično rešljiv, ampak pri temu nisem našel dovolj dobrega približka. Zato bom uporabil numerično integriranje. Da to simulacije ne bo preveč upočasnilo, bo program na novo izračunal  $M_{zi}$  manjkrat na sekundo.

Kot že prej, je tudi navor zraka na ploskev enak vsoti navorov zraka na trikotnike, ki jo sestavljajo.

$$\mathbf{M}_z = \sum_i M_{zi}$$

### 3.7 GIBANJE ENE PLOSKVE

Gibanje ploskve lahko razdelimo na vrtenje in postopno gibanje. Na ploskev delujeta teža in sila zraka. Ta povzroči tudi navor.

Postopno gibanje ploskve lahko preprosto opišemo z 2. Newtonovim zakonom

$$\ddot{\mathbf{r}}_* = \mathbf{g} + \frac{\mathbf{F}_z}{m}$$

Izraza za  $m$  in  $\mathbf{F}_{zi}$  najdete v prejšnjih poglavijih.

Opisati vrtenje je bolj zapleteno. Zamislimo si, da hkrati s ploskvijo vrtimo tudi začetne bazne vektorje. V koordinatnem sistemu zavrtencih vektorjev imajo oglišča ploskve vedno iste koordinate, če postavimo izhodišče v težišče ploskve. Posledično je v tem koordinatnem sistemu konstanten tudi vztrajnostni moment  $J = J_0$ . Poimenujmo ta koordinatni sistem koordinatni sistem ploskve.

Zavrtene začetne bazne vektorje lahko zberemo v stolpce matrike

$$T = \begin{bmatrix} \hat{i}_1 & \hat{j}_1 & \hat{k}_1 \\ \hat{i}_2 & \hat{j}_2 & \hat{k}_2 \\ \hat{i}_3 & \hat{j}_3 & \hat{k}_3 \end{bmatrix}$$

$T$  je tudi matrika transformacije, ki naš koordinatni sistem spremeni v ploskvinega. Vemo pa tudi, da je vztrajnostni moment  $J$  matrika transformacije, ki vektor  $\omega$  spremeni v  $\Gamma$ .  $J_0^{-1}$  lahko pretvorimo v naš koordinatni sistem tako:

$$J^{-1} = TJ_0^{-1}T^{-1}$$

Dokaz za to je v viru [1]

Kotno hitrost plaskve  $\omega$  lahko zato izrazimo z njeno vrtilno količino  $\Gamma$

$$\omega = TJ_0^{-1}T^{-1}\Gamma$$

Oglišča ploskve  $\mathbf{r}_i$ , bazni vektorji koordinatnega sistema ploskve  $\hat{i}, \hat{j}, \hat{k}$  in normalni vektor ploskve  $\hat{n}$  se v času  $dt$  zavrtijo za kot  $\omega dt$  okoli  $\omega$ . To dosežemo z množenjem z ustrezno rotacijsko matriko. Tudi za rotacijsko matriko transformacije vrtenja za kot  $\phi$  okoli vektorja  $w$ , ima Wolfram Mathematica ugrajeno funkcijo `RotationMatrix[\phi,w]`. To bomo označili z  $RM(\phi, w)$ . S tem lahko zapišemo naslednji izraz

$$\hat{i}(t + dt) = RM(\omega(t) dt, \omega(t)) \cdot \hat{i}(t)$$

Tak izraz bo prišel tudi pri  $\hat{j}$ ,  $\hat{k}$  in  $\hat{n}$ .

Pri koordinatah oglšč je pomembno tudi postopno gibanje. Zato pred množenjem z  $RM$  premaknemo ploskev tako, da je težišče ploskve v koordinatnem izhodišču in po vrtenju nazaj, na koncu pa dodamo premik težišča.

$$\mathbf{r}_i(t+dt) = RM(\omega(t) dt, \omega(t)) \cdot (\mathbf{r}_i(t) - \mathbf{r}_*(t)) + \mathbf{r}_*(t) + \mathbf{V}_*(t) dt$$

V zadnjih dveh enačbah  $\omega$  izrazimo iz  $\Gamma$  po enačbi pred temo.  $\Gamma$  v simulaciji dobimo tako, da ga na začetku izračunamo kot

$$\Gamma_0 = J_0 \cdot \omega_0$$

Potem ob vsakem časovnem koraku vrtilni količini prištejemo sunek navora.

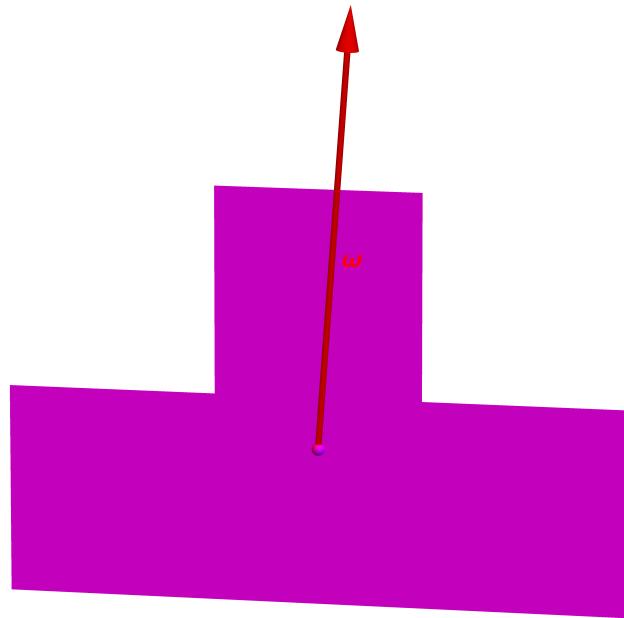
$$\Gamma(t+dt) = \Gamma(t) + \mathbf{M}_z(t) dt \iff \dot{\Gamma} = \mathbf{M}_z$$

Zdaj lahko povzamemo potek simulacije tako:

- Podamo začetne podatke:
  - Koordinate oglšč ploskve  $\mathbf{r}_i$
  - Hitrost težišča ploskve  $\mathbf{V}_*$
  - Kotna hitrost ploskve  $\omega$
  - Ploščinska gostota papirja  $\rho_p$
  - Gostota zraka  $\rho$
  - Gravitacijski pospešek  $\mathbf{g}$
  - Časovni korak v simulaciji  $dt$
- Program enkrat:
  - Razdeli ploskev na trikotnike
  - Izračuna ploščino  $S$  ploskve
  - Izračuna maso  $m$  ploskve
  - Izračuna težišče  $\mathbf{r}_*$  ploskve
  - Izračuna vztrajnostni moment  $J_0$  ploskve
  - Izračuna začetno vrtilno količino  $\Gamma_0$  ploskve

- Program ponavlja korake:
  - Zavrti in premakne oglšča  $r_i$  ter zavrti bazne vektorje  $\hat{i}, \hat{j}, \hat{k}$  koordinatnega sistema ploskve in normalni vektor ploskve  $\hat{n}$
  - Premakne težišče  $r_*$
  - Izračuna silo zraka  $F_z$
  - Izračuna navor zraka  $M_z$
  - Izračuna pospešek težišča ploskve  $\ddot{r}_*$
  - Izračuna novo hitrost težišča ploskve  $V_*$
  - Izračuna novo vrtilno količino  $\Gamma$
  - Izračuna novo kotno hitrost  $\omega$
  - Vsako  $1/60s$  v simulaciji izvozi sliko
  - Gre za  $dt$  naprej

Simulacijo lahko preizkusimo na preprostem primeru. Zamislimo si ploskev, oblikovano kot T-kocka v Tetrisu. Recimo, da je v breztežnostnem prostoru in, da na začetku kotna hitrost  $\omega$  kaže malo stran od simetrijske osi ploskve (glej sliko).



Slika 7: T-ploskva

V viru [2] je lepo razloženo, da v takem primeru pride do gibanja po učinku teniškega loparja. Če simulacija deluje pravilno, se mora to zgoditi tudi v simulaciji. Ko poženemo simulacijo, se zgodi naslednje: če je razmerje  $\rho_p/\rho$  majhno, se ploskev hitro zustavlja in

gre proti vodoravn legi. Da se v tej legi ploskev stabilneje vrti, lahko intuitivno pojasnimo tako: če je kotna hitrost približno pravokotna na ploskev, je navor zraka majhen, če pa se ploskev malo nagne, jo navor zraka potiska nazaj v to lego. To je le za predstavo, ni pa dokaz. Če je razmerje  $\rho_p/\rho$  zelo veliko, se dobro vidi učinek teniškega loparja. Pomembno je le, da je časovni korak  $dt$  v simulaciji dovolj majhen, sicer se rezultat hitro oddaljuje od pravilnega. Simulacijo bi se dalo izboljšati tako, da bi uporabljala RK-metodo višjega, npr 4. reda. Potem bi bila simulacija z enakim časovnim korakom  $dt$  natančnejša.

### 3.8 SILE VZMETI MED PLOSKVAMI

Papirno letalo je sestavljeno iz več med seboj povezanih ploskev. Te med seboj druga na drugo delujejo s silami in navori. Ni očitno, kako bi lahko dobili sile med ploskvami. Nekaj podobnega je simulacija dvojnega fizičnega nihala, brez približka za majhne kote [4]. Že pri navadnem dvojnem nihalu se stvar silno grdo zaplete. Pri papirnem letalu je tak problem v treh dimenzijah, poleg tega pa mora simulacija delovati za poljubno število poljubnih in poljubno povezanih ploskev. Brezupno bi bilo problem reševati na tak način. Zato si izmislimo bolj enostaven način.

Predstavljammo si, da so ploskve med seboj v stikajočih se ogliščih povezane z namišljenimi vzmetmi, brez mase, osnovne dolžine 0. To seveda ni resnično in ni pravilno, ampak, ko gre prožnostni koeficient vzmeti  $k$  proti  $\infty$  (v praksi veliki vrednosti), to v limiti pravilno opisuje gibanje ploskev papirnega letala. To je zato, ker gre v limiti razmanknost ploskev proti 0. Če to ne bi bilo res, bi imeli protislovje z ohranitvijo energije. To je zato, ker je prožnostna energija vzmeti, za katero velja Hookov zakon, če gre  $k$  proti  $\infty$  enaka

$$\lim_{k \rightarrow \infty} W_{pr} = \lim_{k \rightarrow \infty} \frac{k \Delta l^2}{2}$$

Vidimo, da bi bil zgornji izraz enak  $\infty$ , če  $\Delta l$  ne bi šel proti 0.

Rezultat, da gre razmik med ploskvami proti 0, zagotavlja, da se ploskve držijo skupaj tudi ob predpostavki, da so ploskve povezane z opisanimi vzmetmi.

Sile in navore vzmeti na ploskve izračunamo po Hookovem zakonu

$$\mathbf{F}_V = \sum_i \mathbf{F}_{Vi} = \sum_i k \delta \mathbf{r}_i = k \sum_i (\mathbf{r}_{i2} - \mathbf{r}_{i1})$$

$$\mathbf{M}_V = \sum_i \mathbf{M}_{Vi} = \sum_i \Delta \mathbf{r}_i \times \delta \mathbf{r}_i k = k \sum_i (\mathbf{r}_{i1} - \mathbf{r}_*) \times (\mathbf{r}_{i2} - \mathbf{r}_{i1})$$

Kjer je:

- $\delta \mathbf{r}_i$  vektor med koncem na prvi in koncem na drugi ploskvi  $i$ -te vzmeti
- $\mathbf{r}_{i1}$  konec  $i$  te vzmeti na prvi ploskvi
- $\mathbf{r}_{i2}$  konec  $i$  te vzmeti na drugi ploskvi

### 3.9 NAVORI PREGIBOV MED PLOSKVAMI

V modelu manjkajo še navori pregibov  $M_p$ . Ti so odvisni od velikega števila dejavnikov:

- Dolžina pregiba (premo sorazmerje)
- Vrsta papirja
- Osnovni kot v pregibu
- Sprememba kota v pregibu
- "Razmajanost"pregiba
- Drugi dejavniki

Navori v pregibih so premajhni, da bi jih lahko natančno izmerili in prišli do kakih ugotovitev. Zato bomo za model poenostavljeno predpostavili, da je navor v pregibu sorazmeren z dolžino pregiba in je poljubna funkcija spremembe kota med ploskvama v pregibu. Največkrat vzamemo kar linearni model s sučnim koeficientom  $D$ . Naj bo  $\mathbf{u}_p$  vektor od enega do drugega skupnega oglišča med stikajočima ploskvama. Da se pri programiranju ne bi preveč zapletlo, je narejeno samo za primere, pri katerih se sosednji ploskvi stikata v dveh ogliščih. Definirajmo funkcijo  $\varphi(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$ . Ta funkcija naj pove kot, za katerega moramo zavrteti vektor  $\mathbf{u}_1$ , okoli vektorja  $\mathbf{u}_3$ , po pravilu desne roke, da dobimo vektor  $\mathbf{u}_2$ .

$$\varphi(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3) = \begin{cases} (\mathbf{u}_1 \times \mathbf{u}_2) \cdot \mathbf{u}_3 \geq 0, \phi(\mathbf{u}_1, \mathbf{u}_2) \\ (\mathbf{u}_1 \times \mathbf{u}_2) \cdot \mathbf{u}_3 < 0, 2\pi - \phi(\mathbf{u}_1, \mathbf{u}_2) \end{cases}$$

Kjer je  $\phi(\mathbf{u}_1, \mathbf{u}_2)$  manjši kot med vektorjema  $\mathbf{u}_1$  in  $\mathbf{u}_2$ . Za spremembo kota med ploskvama papirnega letala pa vzamemo razliko od osnovnega kota med ploskvama in omejimo spremembo kota med  $-\pi$  in  $\pi$ . Predznak bo pomemben za smer navora med ploskvama.:

$$\Delta\varphi' = \varphi(\hat{\mathbf{n}}_1, \hat{\mathbf{n}}_2, \mathbf{u}_p, t = t_1) - (\hat{\mathbf{n}}_1, \hat{\mathbf{n}}_2, \mathbf{u}_p, t = 0)$$

$$\Delta\varphi = \begin{cases} \Delta\varphi' < -\pi, \Delta\varphi' + 2\pi \\ -\pi \leq \Delta\varphi' \leq \pi, \Delta\varphi' \\ \Delta\varphi' > \pi, \Delta\varphi' - 2\pi \end{cases}$$

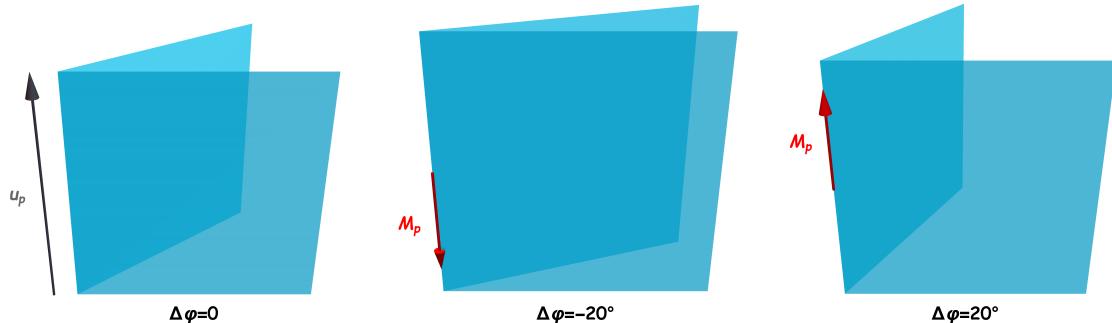
Iz že prej napisanega sledi izraz za navor pregiba na ploskev  $M_p$

$$M_p = u_p f(\Delta\varphi) \hat{u}_p = f(\Delta\varphi) \mathbf{u}_p$$

In linear model za  $f(\Delta\varphi) = D\Delta\varphi$

$$M_p = D\Delta\varphi \mathbf{u}_p$$

Naslednja slika prikazuje navor pregiba  $M_p$  na sprednjo ploskev.



Slika 8: slika  $M_p$

### 3.10 GIBANJE LETALA

Gibanje papirnega letala bomo obravnavali tako, da bomo računali, kako se giblje vsaka ploskev posebej. Zato lahko uporabimo večino razmislekov in izpeljav, ki smo jih naredili za gibanje ene same ploskve.

V enačbi za postopno gibanje težišča ploskve moramo upoštevati še sile vzmeli  $\mathbf{F}_v$  na ploskev

$$\ddot{\mathbf{r}}_* = \mathbf{g} + \frac{\mathbf{F}_z + \mathbf{F}_v}{m}$$

Prav tako je treba v enačbi za vrtilno količino upoštevati še navore vzmeli  $\mathbf{M}_v$  in navore pregibov  $\mathbf{M}_p$  na ploskev

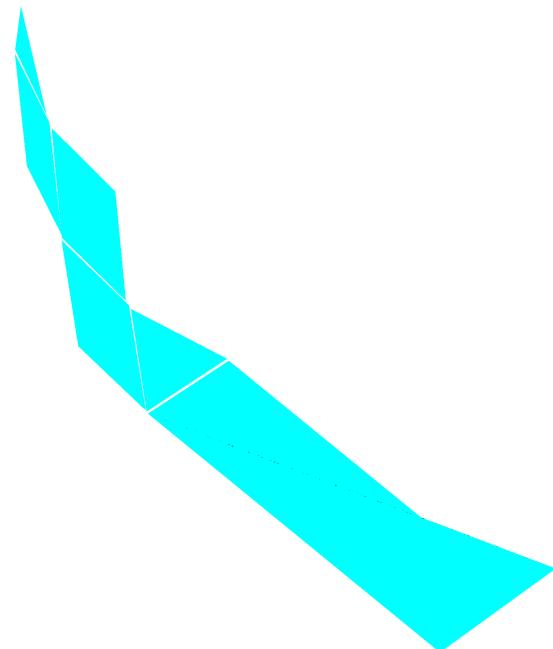
$$\dot{\Gamma} = \mathbf{M}_z + \mathbf{M}_v + \mathbf{M}_p$$

Vse ostale enačbe ostanejo nespremenjene. Tudi v simulaciji moramo dodati nekaj reči. Potek simulacije gibanja papirnega letala izgleda tako:

- Vnesemo začetne podatke:
  - Koordinate oglišč ploskev  $\mathbf{r}_i$
  - Hitrosti težišč ploskev  $\mathbf{V}_*$
  - Kotne hitrosti ploskev  $\omega$
  - Prožnostni koeficient vzmeli  $k$
  - Sučni koeficient pregibov  $D$  ali druga funkcija  $f(\Delta\varphi)$
  - Ploščinska gostota papirja  $\rho_p$
  - Gostota zraka  $\rho$
  - Gravitacijski pospešek  $\mathbf{g}$
  - Časovni korak v simulaciji  $dt$
- Program enkrat:
  - Razdeli ploskve na trikotnike
  - Poišče indekse oglišč in ploskev, ki so med seboj povezane
  - Izračuna ploščine  $S$  ploskev
  - Izračuna mase  $m$  ploskev
  - Izračuna težišča  $\mathbf{r}_*$  ploskev
  - Izračuna vztrajnostne momente  $J_0$  ploskev
  - Izračuna začetne vrtilne količine  $\Gamma_0$  ploskev

- Program ponavlja korake:
  - Zavrti in premakne oglišča  $r_i$  ter zavrti bazne vektorje  $\hat{i}, \hat{j}, \hat{k}$  koordinatnih sistemov ploskev in normalne vektorje ploskev  $\hat{n}$
  - Premakne težišča  $r_*$
  - Izračuna sile zraka  $F_z$
  - Izračuna navore zraka  $M_z$
  - Izračuna sile vzmeti  $F_v$
  - Izračuna navore vzmeti  $M_v$
  - Izračuna navore pregibov  $M_p$
  - Izračuna pospeške težišč ploskev  $\ddot{r}_*$
  - Izračuna nove hitrosti težišč ploskev  $V_*$
  - Izračuna nove vrtilne količine  $\Gamma$
  - Izračuna nove kotne hitrosti  $\omega$
  - Vsako  $1/60s$  v simulaciji izvozi sliko
  - Gre za  $dt$  naprej

Simulacijo bomo najprej preizkusili na preprostem primeru. Imamo 6 ploskev, ki so vezane v nekakšno postrani šesterno fizično nihalo (glej sliko).



Slika 9: poševno nihalo

Pri tem je zgornji rob vrtljivo vpet. Za to je v program vključeno, da lahko na začetku vnesemo tudi katera oglišča bodo vpeta na svoje začetne položaje. Sile na vpeta oglišča računamo tako, kot sile med ploskvami. Vpeti rob bi lahko uporabili tudi, če bi nas zanimalo, kako se zaradi teže spremenijo koti med ploskvami letala pred spustom. Še ena možnost uporabe bi bila za simulirati, kako se spreminjajo koti med ploskvami ob metu papirnega letala. Pri zadnjem bi se pritrdišča premikala po krivulji meta, nato pa izginila.

Če poženemo simulacijo, vidimo, da se ob dovolj majhnem časovnem koraku  $dt$  prejšnje "nihalo" giblje tako, kot bi pričakovali. Če pa je  $dt$  prevelik, gibanje ploskev hitro podivja in razdalje med njimi grejo proti neskončno. Do tega verjetno pride, ker se oglišča vsakič premaknejo malo predaleč. Če na primer začnejo s hitrostjo  $V(t)$  in se premaknejo za  $V(t) dt$ ,  $V$  pa se manjša, je povprečna hitrost malo manjša kot  $V(t)$ . Vzmet je tako vsakič malo bolj raztegnjena in ploskve hitro grejo neskončno narazen. O tem lahko premislimo na najlažjem primeru (navadno vzmetno nihalo). Pri navadnem vzmetnem nihalu je število potrebnih korakov na sekundo, da simulacija ne divergira, razmerno  $\sqrt{k}$  (fiksno število korakov na nihaj  $t_0 = 2\pi\sqrt{\frac{m}{k}}$ ). Če preizkusimo različne vrednosti  $k$ , vidimo, da se potrebno število korakov na sekundo res veča približno razmerno  $\sqrt{k}$ .

Težava je v tem, da za boljšo natančnost simulacije, potrebujemo večjo vrednost prožnostnega koeficiente  $k$ . Če pa izberemo večji  $k$ , moramo tudi zmanjšati časovni korak v simulaciji in jo s tem upočasnimo. En način, kako bi se dalo izboljšati simulacijo, da bi zadoščal že večji časovni korak  $dt$ , bi bil z uvedbo RK-metode višjega reda. To v simulacijo ni vključeno, so pa v naslednjih poglavjih opisani še trije različni načini popravkov.

### 3.11 POPRAVEK Z RAVNANJEM PLOSKEV

Ta popravek bi se ponovil le na vsakih nekaj časovnih korakov, zato simulacije praktično ne bi upočasnili. Popravek deluje tako: najprej poiščemo najmanj razmaknjeni oglišči, ki bi se mogli stikati. Nato premaknemo eno ploskev tako, da se tidve oglišči stikata. Potem ploskev zavrtimo da se stikata še drugi dve skupni oglišči. To ponovimo za vse ploskve. Kotne hitrosti ploskev in hitrosti težišč popravimo tako, da za razliko kotnih hitrosti sosednjih ploskev vzamemo samo komponento, vzporedno pregibu.

Ta popravek je le možna ideja, saj v simulaciji ni dokončan. Izkaže se, da se programiranje tega popravka kljub temu, da zveni enostavno, zelo zaplete.

### 3.12 POPRAVEK Z OHRANITVIJO ENERGIJE

Ta popravek temelji na tem, da sproti računamo energije in delo zraka. Nato hitrosti težišč  $\mathbf{V}_*$  in kotne hitrosti  $\omega$  pomnižimo s takim faktorjem  $\alpha$ , da velja energijski zakon. Če v enačbo zberemo prisotne oblike energije in delo dobimo:

$$W_{p1} + W_{pr1} + W_{M1} + W_{k1} + W_{kr1} = W_{p0} + W_{k0} + W_{kr0} - A \iff \\ W_{k1} + W_{kr1} = W_{p0} + W_{k0} + W_{kr0} - W_{p1} - W_{pr1} - W_{M1} - A$$

Kjer je:

- $W_{p0}$  začetna potencialna energija
- $W_{k0}$  začetna kinetična energija
- $W_{kr0}$  začetna rotacijska kinetična energija
- $A$  delo, ki ga opravi zrak
- $W_{p1}$  potencialna energija
- $W_{pr1}$  prožnostna energija vzmeti
- $W_{M1}$  kotni ekvivalent prožnostne energije, v pregibih zaradi navora
- $W_{k1}$  kinetična energija
- $W_{kr1}$  rotacijska kinetična energija

Na levi strani prejšnje enačbe imamo  $W_{k1} + W_{kr1}$ , kar je na desni strani izraženo z drugimi energijami. Recimo, da bi v simulaciji dobili preveliko hitrost ali pa kotno hitrost. Potem bi bila številska vrednost, ki bi jo dobili z vstavljanjem podatkov, na levi strani večja kot na desni. Recimo, da vzamemo podatke na desni strani za prave. Da bi enačba držala, bi morali zmanjšati levo stran. To lahko naredimo tako, da hitrosti in kotne hitrosti pomnožimo s faktorjem  $\alpha$ . Ker tako hitrost, kot tudi kotna hitrost v izrazu  $W_{k1} + W_{kr1}$  nastopata, kot kvadrata, lahko iz izraza izpostavimo  $\alpha^2$ . Levo stran enačbe zamenjamo z  $\alpha^2 (W_{k1} + W_{kr1})$  in dobimo

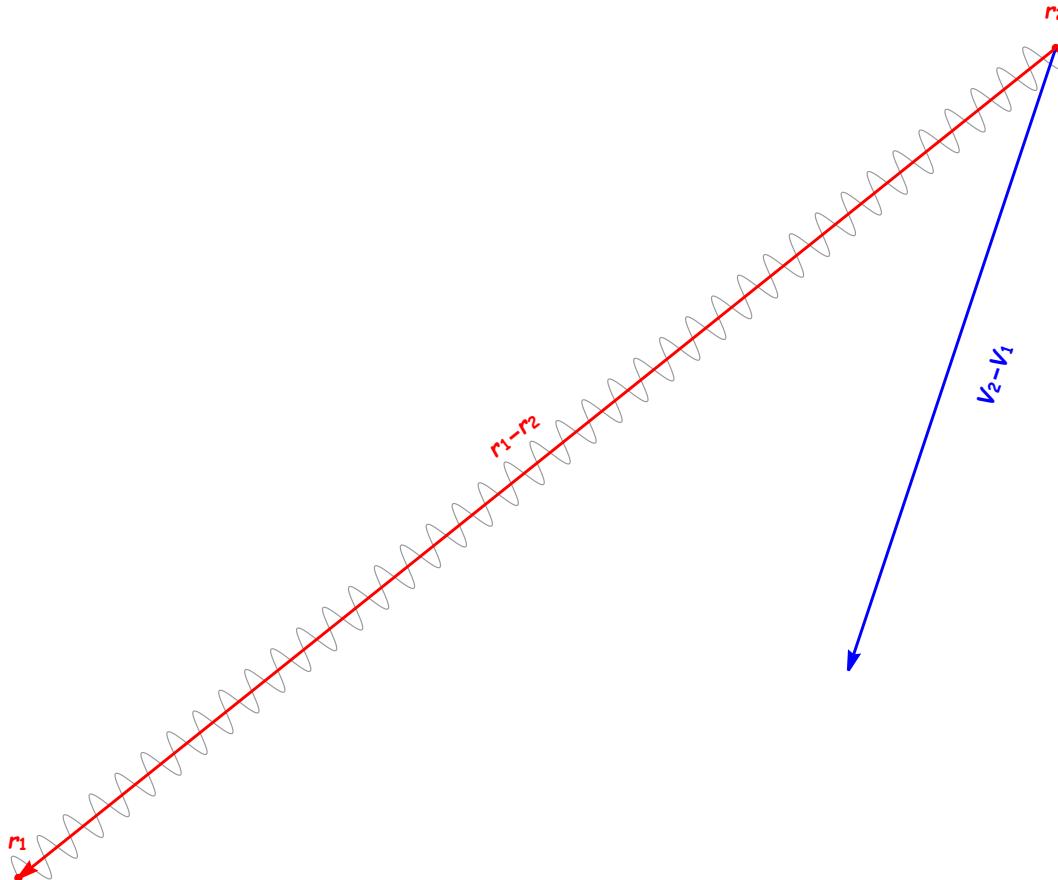
$$\begin{aligned}\alpha^2 (W_{k1} + W_{kr1}) &= W_{p0} + W_{k0} + W_{kr0} - W_{p1} - W_{pr1} - W_{M1} - A \iff \\ \alpha &= \sqrt{\frac{W_{p0} + W_{k0} + W_{kr0} - W_{p1} - W_{pr1} - W_{M1} - A}{W_{k1} + W_{kr1}}}\end{aligned}$$

S tem smo dobili izraz za faktor  $\alpha$ , s katerim pomnožimo hitrosti in kotne hitrosti, da velja energijski zakon. Izrazi za posamezne od teh energij so splošno znani in so našteti spodaj

- $W_p = \sum mgh = g \sum m\mathbf{r}_* \cdot (0, 0, 1)$
- $W_k = \frac{1}{2} \sum mV_*^2$
- $W_{kr} = \frac{1}{2} \sum \omega \cdot J \cdot \omega$
- $W_{pr} = \frac{k}{2} \sum \delta r^2$
- $W_M = u_p \sum \int_0^{\Delta\varphi_1} f(\Delta\varphi) d\Delta\varphi \implies \frac{D}{2} \sum \Delta\varphi^2$
- $dA = -dt \sum \mathbf{F}_z \cdot \mathbf{V}_* + \mathbf{M}_z \cdot \omega$

### 3.13 POPRAVEK Z DUŠENJEM VZMETI

Pri temu popravku si zamislimo, da vzmeti ustvarjajo zavorno silo, razmerno hitrosti stiskanja oz. raztezanja vzmeti. Recimo, da je eno krajišče vzmeti  $\mathbf{r}_1$ , drugo  $\mathbf{r}_2$ , hitrost prvega krajišča  $\mathbf{V}_1$  in drugega  $\mathbf{V}_2$  (glej sliko)



Slika 10: zavorna sila vzmeti

Če se postavimo v koordinatni sistem prvega krajišča, je hitrost drugega krajišča  $\mathbf{V}_2 - \mathbf{V}_1$  (modri vektor na sliki). Hitrost krčenja vzmeti je enaka komponenti  $\mathbf{V}_2 - \mathbf{V}_1$ , ki kaže v smeri vzmeti

$$V_{kr} = (\mathbf{V}_2 - \mathbf{V}_1) \cdot \frac{\mathbf{r}_1 - \mathbf{r}_2}{|\mathbf{r}_1 - \mathbf{r}_2|}$$

Zavorna sila dušenja vzmeti je zato enaka

$$\mathbf{F}_D = k_D \left( (\mathbf{V}_2 - \mathbf{V}_1) \cdot \frac{\mathbf{r}_1 - \mathbf{r}_2}{|\mathbf{r}_1 - \mathbf{r}_2|} \right) \frac{\mathbf{r}_1 - \mathbf{r}_2}{|\mathbf{r}_1 - \mathbf{r}_2|} = \frac{k_D (\mathbf{V}_2 - \mathbf{V}_1) \cdot (\mathbf{r}_2 - \mathbf{r}_1)}{|\mathbf{r}_2 - \mathbf{r}_1|^2} (\mathbf{r}_2 - \mathbf{r}_1)$$

Celotna sila vzmeti  $\mathbf{F}_V$ , če vključimo dušenje je potem enaka

$$\mathbf{F}_V = \sum_i \left( k + \frac{k_D (\mathbf{V}_{i2} - \mathbf{V}_{i1}) \cdot (\mathbf{r}_{i2} - \mathbf{r}_{i1})}{|\mathbf{r}_{i2} - \mathbf{r}_{i1}|^2} \right) (\mathbf{r}_{i2} - \mathbf{r}_{i1})$$

To isto naredimo še za navor vzmeti  $\mathbf{M}_V$

$$\mathbf{M}_V = \sum_i \left( k + \frac{k_D (\mathbf{V}_{i2} - \mathbf{V}_{i1}) \cdot (\mathbf{r}_{i2} - \mathbf{r}_{i1})}{|\mathbf{r}_{i2} - \mathbf{r}_{i1}|^2} \right) (\mathbf{r}_{i1} - \mathbf{r}_*) \times (\mathbf{r}_{i2} - \mathbf{r}_{i1})$$

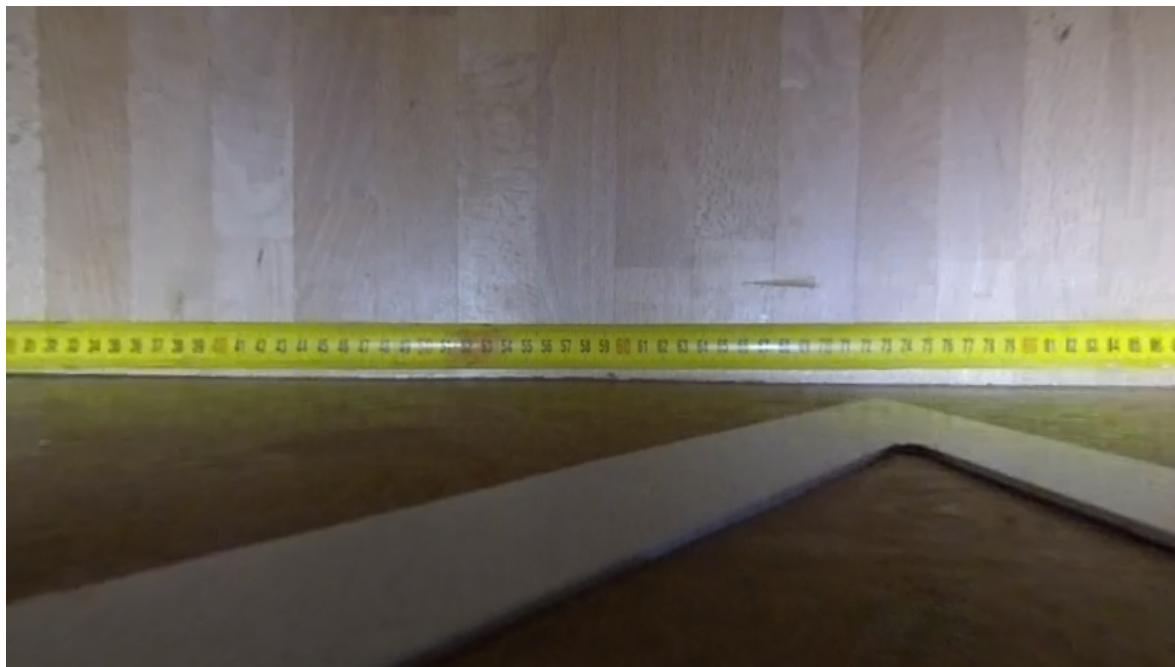
$\mathbf{V}_{i1}$  in  $\mathbf{V}_{i2}$  pa poznamo že od prej

$$\mathbf{V}_{i1} = \mathbf{V}_{*i1} + \omega_1 \times (\mathbf{r}_{i1} - \mathbf{r}_{*i1})$$

$$\mathbf{V}_{i2} = \mathbf{V}_{*i2} + \omega_2 \times (\mathbf{r}_{i2} - \mathbf{r}_{*i2})$$

### 3.14 SNEMALNI KOT KAMERE

Za snemanje papirnih letal je bila uporabljena kamera, ki posname 240 slik v sekundi. Posnetek je v osnovi ukrivljen, zato v programu Premiere pro nastavimo odpravljanje ukrivljenosti posnetka in izvozimo slike png. Z odpravljanjem ukrivljenosti izgubimo nekaj robov posnetka. Po odpravljanju ukrivljenosti je ločljivost posnetka  $848 \times 480$  pikslov. Naslednji korak je, da določimo zorni kot posnetka. Najprej postavimo kamero vzporedno merilnemu traku. To najlažje naredimo tako, da oboje postavimo na meje med ploščami parketa, ki so si vzporedne. Na isto črto, kjer je kamera postavimo še laserski merilnik razdalje. Da sta na natančno isti črti naredimo tako, da ju postavimo za isto ravnilo (glej sliko).



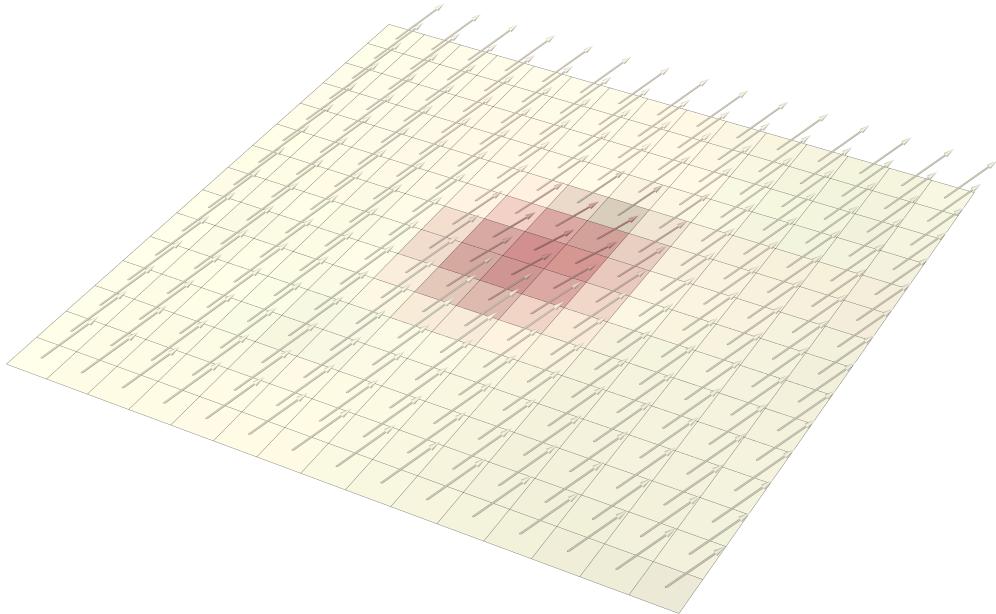
Slika 11: kot kamere

Nato z laserskim mmerilnikom izmerimo razdaljo do merilnega traku  $\Delta Y_0 = 39,1\text{cm}$ . Potem iz merilnega traku odčitamo številke na robovih posnetka in dobimo  $\Delta X_0 = 57\text{cm}$ . Zdaj lahko izračunamo snemalni kot kamere

$$\theta_x = 2 \arctan \left( \frac{\frac{\Delta X_0}{2}}{\Delta Y_0} \right) = 1.26$$

## 4 SLEDENJE LETALOM IZ POSNETKA

Posnetek letala je sestavljen iz slik, slike pa iz mreže pikslov. Barva vsakega piksla je podana z vektorjem  $\text{RGB} = (R, G, B)$ , kjer so komponente deleži moči  $P$  svetjenja rdeče, modre in zelene luči v pikslu (glej sliko).



Slika 12: RGB vektorji

Na vsako ploskev papirnega letala narišemo tri barvne pike. Program je napisan tako, da najprej na prvi sliki pritisnemo barvno piko. Od tod program dobi barvo pike  $\text{RGB}_0$ . Za sredino pike na sliki vzamemo kar povprečje koordinat pikslov te pike. Program mora najprej ugotoviti kateri piksli so del te pike. To naredimo tako, da za vsak piksel pregledamo, ali je njegova barva dovolj podobna barvi, izbrani na začetku. Zato si je treba izbrati smiselen kriterij za določanje, kdaj sta barvi dovolj podobni. Lahko bi vzeli velikost razlike barvnih vrednosti  $|\text{RGB}_1 - \text{RGB}_2|$ , ampak to ne bi delovalo dobro. Letalo se giblje, pri čemer se spreminja kot ploskve glede na kamero. Zaradi tega se spreminja tudi gostota svetlobnega toka, ki se od pike odbije proti kamери. To se na  $\text{RGB}$  kaže kot daljšanje ozira krajšanje vektorja  $\text{RGB}$ , ki pa še vedno kaže v isto smer. Za določitev če sta si barvi dovolj podobni, bi bil bolj smiselen kriterij: ali imata njuna  $\text{RGB}$  vektorja dovolj podobno smer. To pa lahko ugotovimo s skalarnim produktom smernih vektorjev. Če sta si barvi dovolj podobni, bo ta dovolj blizu 1.

$$\frac{\text{RGB}_1}{\text{RGB}_1} \cdot \frac{\text{RGB}_2}{\text{RGB}_2} > 1 - \epsilon$$

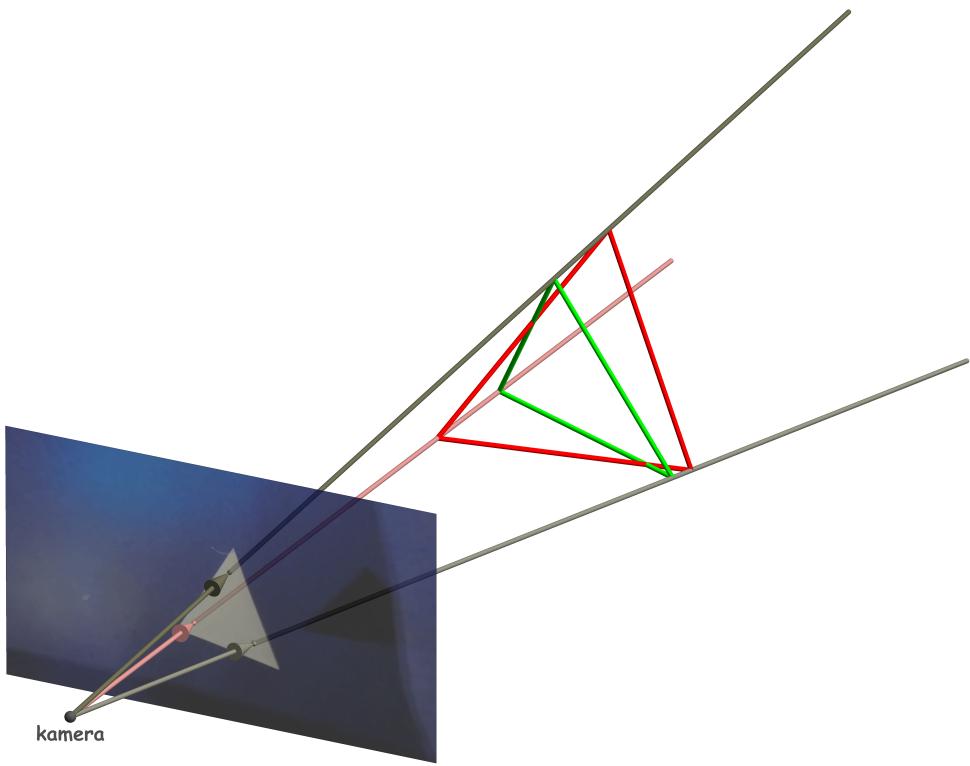
Če za  $\epsilon$  vzamemo premajhno vrednost, potem program ne bo našel vseh piksov barvne pike, saj se med seboj za malenkost razlikujejo. Če izberemo preveliko vrednost, pa bo program štel tudi piksle, ki ne spadajo v to piko. Izkaže se, da ponavadi dobro delujejo vrednosti  $\epsilon$  okoli 0,0005.

Če opisani algoritem za iskanje sredine pike sprogramiramo, se izkaže, da program deluje prveč počasi. Za vsako sliko potrebuje okoli 10s. Program je počasen, saj pregleduje vse piksle na sliki. Lahko ga pospešimo tako, da pregledamo samo piksle v okolici pike. Za to uporabimo funkcijo FindFit, vgrajeno v Mathematico. Ta poišče vrednosti parametrov v podani funkciji, pri katerih je vsota odstopanj točk najmanjsa, oziroma, se graf funkcije najbolje prilega točkam. To funkcijo uporabimo tako, da gibanje sredine pike na sliki razdelimo na navpično in vodoravno komponento. Za vsakega od teh s funkcijo FindFit poiščemo polinom čez zadnjih nekaj točk  $(t, x)$  oziroma  $(t, y)$ . Izkaže se, da dobro deluje polinom 5. stopnje skozi zadnjih 10 točk. S tem lahko ugibamo, kje približno bo sredina barvne pike na naslednji sliki.

$$(p_1(t + dt), p_2(t + dt))$$

Ko imamo začetno točko pregledovanja piksov, jo najprej zaokrožimo, da dobimo začetni piksel. Ta piksel je lahko že v barvni piki, lahko pa ni. Od tega piksla začnemo navzven v koncentričnih kvadratnih zankah piksov pregledovati piksle. Če bi kvadratna zanka slučajno segala čez rob slike, vzamemo pravokotno zanko, s problematičnim robom na robu slike. Tako iz začetnega piksla navzven skeniramo piksle, dokler ne trčimo v piksle prave barve. Od teh enega izberemo. Potem od tega izbranega piksla spet navzven pregledujemo piksle. To delamo, dokler je na trenutni zanki piksov vsaj 1 piksel prave barve. Ko na zanki ni več piksov prave barve, je celotna pika znotraj zanke, torej smo našli vse piksle pike.

Če na tak način na vsaki ploskvi sledimo trem barvnim pikam, lahko dobimo gibanje letala v prostoru. Poznati moramo začetne koordinate pik v simulaciji  $R_{0,1}$ ,  $R_{0,2}$ ,  $R_{0,3}$ . Te Točke tvorijo trikotnik. Izračunamo lahko dolžine stranic trikotnika  $|R_{0,3} - R_{0,2}|$ ,  $|R_{0,1} - R_{0,3}|$ ,  $|R_{0,2} - R_{0,1}|$ . Predstavljamo si lahko, da se slika kamere nahaja  $\Delta Y_0$  pred kamero in je široka  $\Delta X_0$ , kjer sta  $\Delta X_0$  in  $\Delta Y_0$  podatka, ki smo jih izmerili v poglavju snemalni kot kamere (glej sliko na naslednji strani).



Slika 13: sledenje gibanju v prostoru

Vektor do levega spodnjega vogala slike je potem enak

$$\left( -\frac{\Delta X_0}{2}, \Delta Y_0, -\frac{\Delta X_0 L_y}{2L_x} \right)$$

Kjer je  $L_x \times L_y$  ločljivost slike.

Ker je slika široka  $\Delta X_0$ , je širina piksla enaka  $\Delta X_0 / L_x$ . S tem lahko izrazimo lege sledenih barvnih pik na na sliki v 3D. Recimo, da je sredinski piksel sledene pike na sliki  $p_x$ -ti piksel od leve proti desni in  $p_y$ -ti piksel od dna proti vrhu. Potem bo lega tega piksla na sliki v prostoru enaka

$$\left( -\frac{\Delta X_0}{2} + p_x \frac{\Delta X_0}{L_x}, \Delta Y_0, -\frac{\Delta X_0 L_y}{2L_x} + p_y \frac{\Delta X_0}{L_x} \right)$$

Na zgornji sliki je so lege sredinskih piksov sledenih pik prikazane z vektorji. Recimo, da smo po prejšnji enačbi dobili, da so ti vektorji  $\mathbf{r}_1$ ,  $\mathbf{r}_2$  in  $\mathbf{r}_3$ .

Zdaj pa si zamislimo, kje v prostoru so dejanske ploskve letala in pike na njih. Pike gotovo ležijo nekje na premicah, narisanih na zgornji sliki. Vemo pa tudi, da tvorijo trikotnik z znanimi dolžinami stranic. To lahko zapišemo s sistemom enačb

$$|\mathbf{R}_{0,3} - \mathbf{R}_{0,2}| = |c_3 \mathbf{r}_3 - c_2 \mathbf{r}_2|$$

$$|\mathbf{R}_{0,1} - \mathbf{R}_{0,3}| = |c_1 \mathbf{r}_1 - c_3 \mathbf{r}_3|$$

$$|\mathbf{R}_{0,2} - \mathbf{R}_{0,1}| = |c_2 \mathbf{r}_2 - c_1 \mathbf{r}_1|$$

Iz zgornjega sistema dobimo parametre  $c_1$ ,  $c_2$  in  $c_3$  in s tem lege sledenih pik v prostoru  $c_1\mathbf{r}_1$ ,  $c_2\mathbf{r}_2$  in  $c_3\mathbf{r}_3$ . Dobimo 4 komplete rešitev. 2 od teh sta za kamero in nas zato ne zanimata, 2 pa sta pred njo. Na sliki sta označena z dvema trikotnikoma. To je tako kot znana optična iluzija sence lutke, za katero se ne da določiti v katero smer se vrti. Zato moramo v program ročno vnesti še katera rešitev je prava.

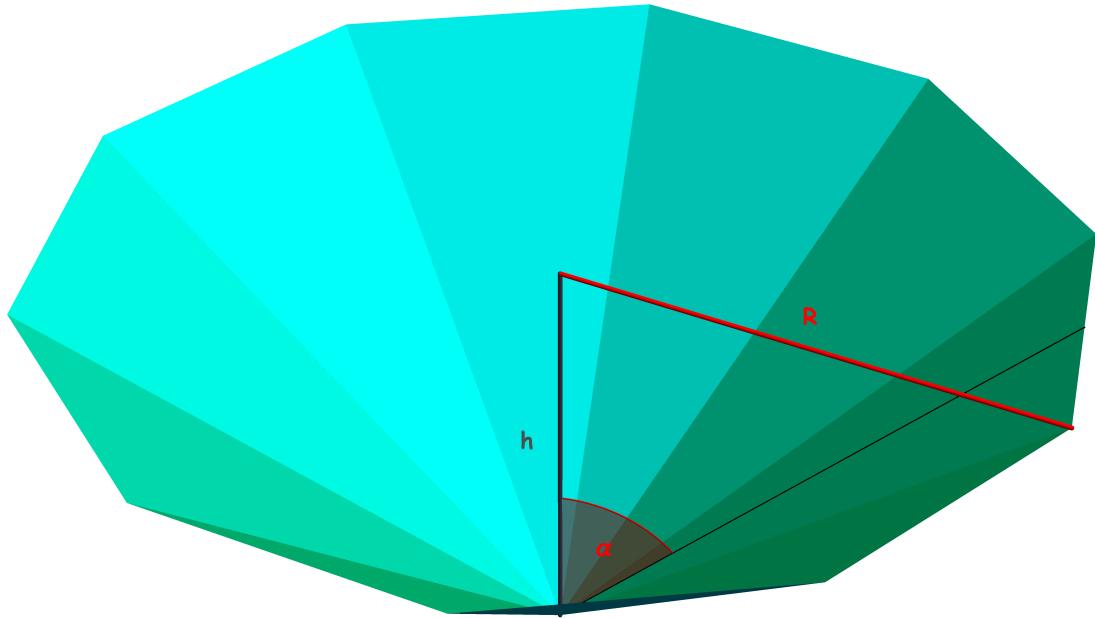
Na ta način lahko sledimo gibanju vsake ploskve papirnega letala. Metoda dobro deluje na preprostem primeru, ko je letalo snemano zelo od blizu in dobro osvetljeno. Sicer se izkaže, da je ločljivost kamere premajhna za to metodo in stvar ne deluje. Pri nižjem številu slik na sekundo in višji ločljivosti tudi ne deluje, saj kamera zajema svetlobo dlje časa in so pike zamazane.

Vseeno pa lahko sledimo celotnim letalom, kot točkastim telesom, če se ti ne gibajo v  $y$ -smeri. Za barvno piko vzamemo kar celotno letalo. Potem moramo poznati še  $\Delta Y$  razdaljo letala od kamere. To naredimo tako, da iz mesta spuščanja papirnega letala obesimo kocko. Nato laserski merilnik postavimo na mesto kamere in izmerimo razdaljo do kocke. Dobimo  $\Delta Y = 1.25m$ . Potem po Talesovem izreku dobimo  $(\Delta X, \Delta Z)$  koordinate papirnega letala.

$$(\Delta X, \Delta Z) = \frac{\Delta Y}{\Delta Y_0} \left( -\frac{\Delta X_0}{2} + p_x \frac{\Delta X_0}{L_x}, -\frac{\Delta X_0 L_y}{2L_x} + p_y \frac{\Delta X_0}{L_x} \right)$$

## 5 PIRAMIDNO LETALO

V tem poglavju bomo obravnavali primer papirnega letala, ki je dovolj enostavno, da lahko pod predpostavkami iz modela dobimo analitične rešitve za njegovo gibanje. Potem lahko tudi ugotovimo približni red natančnosti modela. Piramidno letalo definirajmo, kot stranske ploskve pravilne pokončne  $s$ -strane piramide (glej sliko).



Slika 14: piramidni

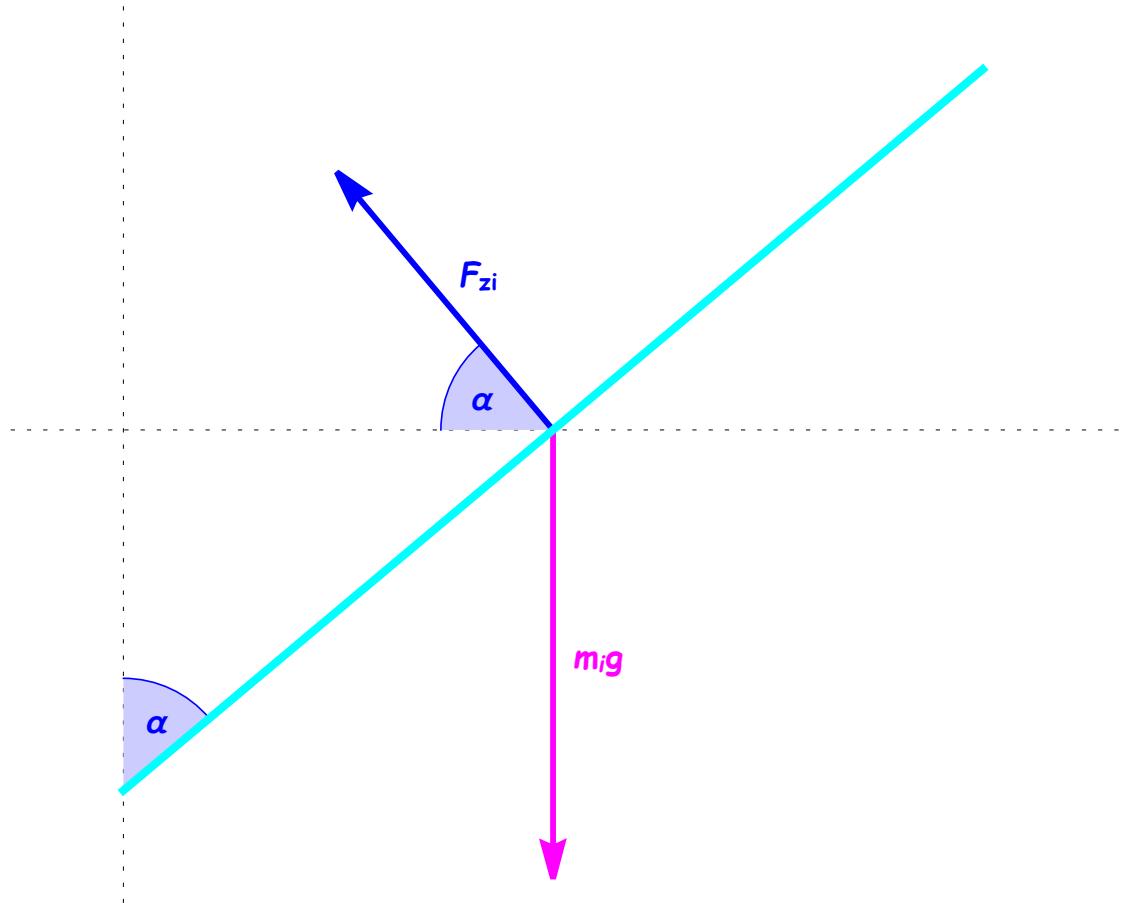
Najprej z  $R$  in  $h$  lahko izrazimo kot  $\alpha$ . Zgornje oglišče poševne črte na zgornji sliki je na sredini ene izmed stranic  $s$ -kotnika. Recimo, da je koordinatni sistem tak, da je  $s$ -kotnik na ravnini osi  $x$  in  $y$  in je eno oglišče  $R(1, 0, 0)$ . Naslednje oglišče je potem  $R(\cos(\frac{2\pi}{s}), \sin(\frac{2\pi}{s}), 0)$ . Povprečje teh dveh je na  $\frac{R}{2}(1 + \cos(\frac{2\pi}{s}), \sin(\frac{2\pi}{s}), 0)$ . S tem lahko izrazimo kot  $\alpha$

$$\alpha = \arctan \left( \frac{\frac{R}{2} \sqrt{(\cos(\frac{2\pi}{s}))^2 + (\sin(\frac{2\pi}{s}))^2}}{h} \right) = \arctan \left( \frac{R \cos(\frac{\pi}{s})}{h} \right)$$

Ker bomo piramidna letala izrezovali iz papirja, moramo za izris mrež piramidnih letal poznati še polmer očrtanega kroga mreži  $R_m$  in kote ob vrhu trikotnikov  $\varphi_m$ .  $R_m$  je po Pitagorovem izreku  $\sqrt{R^2 + h^2}$ .  $\varphi_p$  pa lahko izračunamo s skalarnim produktom

$$\varphi_p = \arccos \left( \frac{(R, 0, h) \cdot (R \cos(\frac{2\pi}{n}), R \sin(\frac{2\pi}{n}), h)}{R_m^2} \right) = \arccos \left( \frac{R^2 \cos(\frac{2\pi}{n}) + h^2}{R^2 + h^2} \right)$$

Zdaj lahko izpeljemo izraz za gibanje piramidnega papirnega letala z danim kotom  $\alpha$ . Spodnja slika prikazuje silo zraka  $F_{zi}$  in težo  $m_i g$  na eno izmed ploskev papirnega letala.



Slika 15: sile na piramidno letalo

Recimo, da bomo letalo spuščali samo obrnjeno z vrhom piramide navzdol. Potem je letalo simetrično in lahko vodoravne komponente  $F_{zi}$  zložimo v  $s$ -kotnik in je njihova vsota  $(0, 0, 0)$  oziroma se izničijo. Zato lahko za gibanje celotnega letala upoštevamo samo navpične komponente  $F_{zi}$ . Po 2. Newtonovem zakonu zato velja

$$\ddot{z} = g - \frac{F_z \sin(\alpha)}{m} = g - \frac{\rho S (V \sin(\alpha))^2 \sin(\alpha)}{\rho_p S} = g - \frac{\rho (\sin(\alpha))^3}{\rho_p} \dot{z}^2 = g - \beta \dot{z}^2$$

Če to enačbo vnesemo v Wolfram Mathematico, dobimo rešitev

$$z = \frac{c_2 + \ln(\cosh(\sqrt{g\beta}(t + c_1)))}{\beta},$$

$$c_1 = \beta x_0 - \ln\left(\sqrt{\frac{g}{g - \beta V_0^2}}\right); c_2 = \frac{\operatorname{arccosh}\left(\sqrt{\frac{g}{g - \beta V_0^2}}\right)}{\sqrt{g\beta}}$$

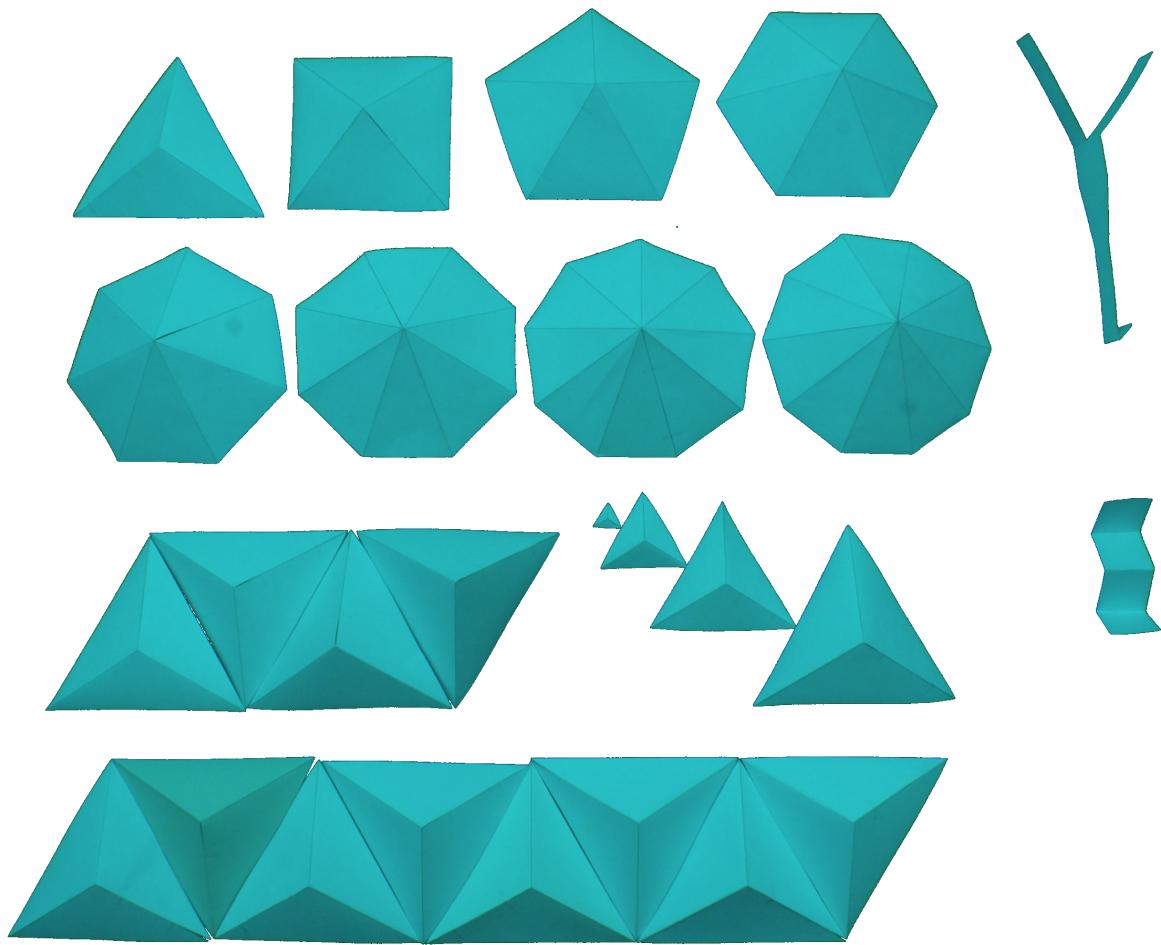
Lahko pa izračunamo tudi terminalno hitrost, če v prejšnjo diferencialno enačbo vstavimo  $\ddot{z} = 0$  ali pa izračunamo neskončno limito rešitve. Dobimo

$$V_{max} = \sqrt{\frac{g}{\beta}}$$

Iz predpostavk modela smo torej lahko izpeljali enačbe, iz katerih sledi, da je gibanje piramidnega letala pri stalnem zraku, z nespremenljivo gostoto odvisno samo od kota  $\alpha$  in ploščinske gostote papirja. Kasneje bom z eksperimentom preveril, ali to res drži in s tem preveril eno napoved modela.

## 6 EKSPERIMENTALNI DEL

V eksperimentalnem delu raziskovalne naloge so analizirane 3 vrste papirnih letal. Na spodnji sliki so prikazana ta letala, izdelana iz papirja.



Slika 16: izrezana letala

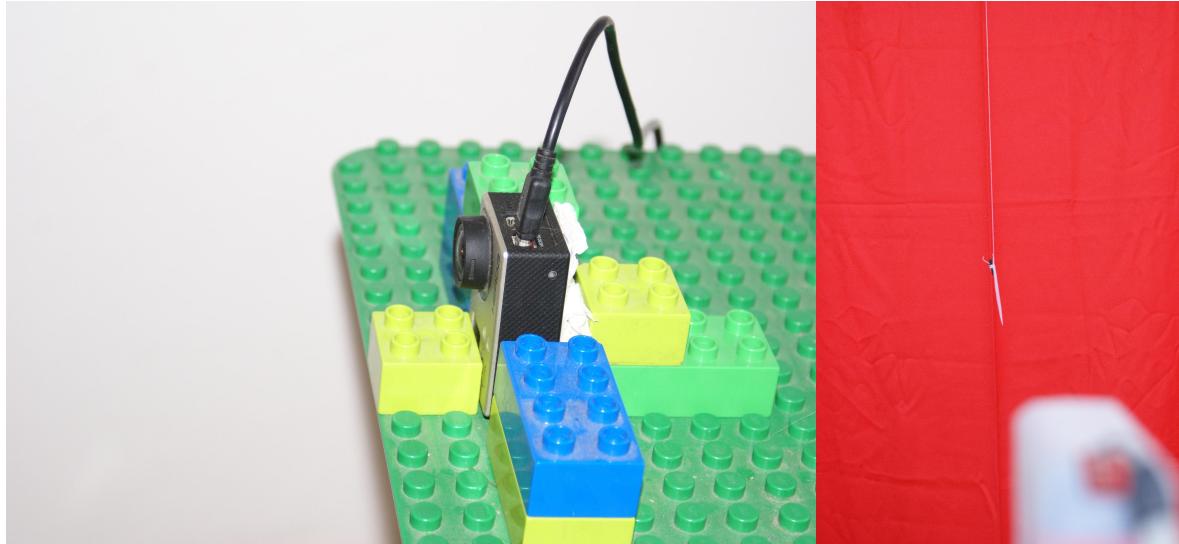
Desno zgoraj imamo helikoptersko letalo, ki je primer letala, katerega let model dobro opisuje. Pod njim je W letalo. To je primer letala, katerega gibanje model ne napove niti približno. Vse ostalo so Piramidna letala, pri katerih je preverjena natančnost modela in njegovih napovedi. V zgornjih dveh vrstah se letala razlikujejo samo po številu stranic  $s$ . V naslednji vrstici levo se razlikujejo samo po kotu  $\alpha$ . Desno v isti vrstici se razlikujejo samo po polmeru  $R$ . V spodnji vrstici se piramidna letala razlikujejo le po ploščinski gostoti papirja  $\rho_p$ , iz katerega so izdelana.

Let letal je bil nato posnet s kamero in analiziran. Da so letala spuščena iz iste točke, iz kock sestavimo mehanizem za spuščanje papirnih letal (glej sliko na naslednji strani).



Slika 17: mehanizem za spuščanje papirnih letal

Mehanizem sprožimo tako, da v levo povlečemo vrvico prožilca. Potem se zaradi navora elastike desni del mehanizma hitro umakne dol in letalo se začne gibati. Pod mesto spuščanja letala obesimo enobarvno ozadje. Poskrbeti moramo tudi za dobro osvetlitev. Osvetlitev z lučmi na izmenični tok za eksperiment ni dobra, saj kamera posname 240 slik v sekundi. Slike nastajajo dovolj malo časa, da je pri nekaterih svetlobni tok luči ves čas zajemanja majhen. Te slike so zato zrnate in neuporabne za obdelavo. Zato je eksperiment najlažje delati kar pri dnevni svetlobi. Zadnji pomemben korak priprave eksperimenta je poskrbeti, da je kamera usmerjena naravnost proti ravnini spuščanja papirnih letal. To najlažje naredimo z ravnanjem na črte parketa, ki so si vzporedne. Želimo pa tudi, da bo mesto spuščanja letala na sredini slike. To uredimo hkrati z merjenjem razdalje do mesta spuščanja  $\Delta Y$ . Razdaljo  $\Delta Y$  izmerimo tako, da iz mesta spuščanja obesimo kocko in z laserskim merilnikom, ki ga postavimo na mesto kamere izmerimo razdaljo do kocke (glej sliko).



Slika 18: kamera in merjenje delta Y

## 6.1 REZULTATI ZA PIRAMIDNA LETALA

Pri analiziranju posnetkov leta piramidnih letal se zgornjega dela leta ni dalo dobro analizirati zaradi prožilca in bele stene v ozadju. Zato se bomo osredotočili samo na terminalne hitrosti  $V_{max}$  teh letal. Za osnovo vzemimo piramidno letalo s podatki  $n = 3$ ,  $R = 7,0\text{cm}$ ,  $\alpha = 50^\circ$  in  $\rho_p = 75 \frac{\text{g}}{\text{m}^2}$ . Potem spremojmo po eno spremenljivko naenkrat (opisano pri sliki 16).

Ko posnamemo papirna letala in dobimo njihove koordinate, s funkcijo FindFit v Mathematici poiščemo strmino premic skozi zadnjih nekaj  $z(t)$  meritev. To je izmerjena terminalna hitrost letal. Tako izmerjene vrednosti terminalnih hitrosti so zbrane v spodnjih tabelah.

$R [\text{cm}]$	$V_{max} [\frac{\text{m}}{\text{s}}]$
1,0	0,95
3,0	1,03
5,0	0,89
7,0	0,94

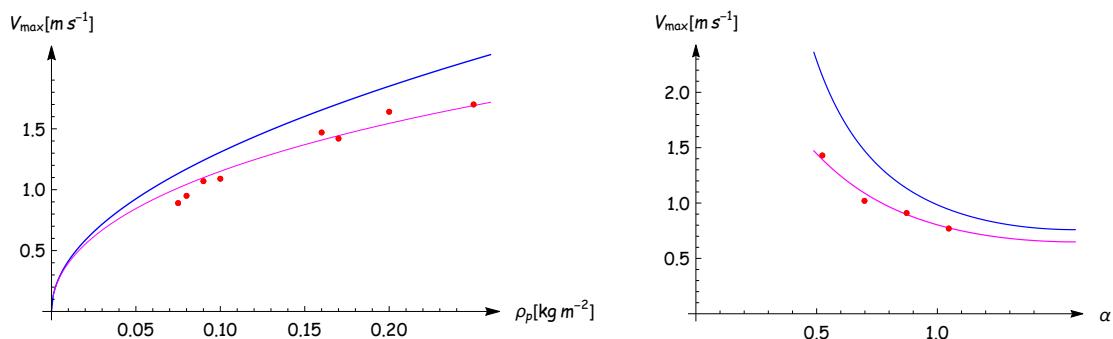
$n$	$V_{max} [\frac{\text{m}}{\text{s}}]$
3	0,90
4	1,02
5	1,03
6	0,97
7	0,96
8	0,95
9	1,03
10	1,03

$\rho_p [\frac{\text{g}}{\text{m}^2}]$	$V_{max} [\frac{\text{m}}{\text{s}}]$
75	0,89
80	0,95
90	1,07
100	1,09
160	1,47
170	1,42
200	1,64
250	1,70

$\alpha$	$V_{max} [\frac{\text{m}}{\text{s}}]$
$30^\circ$	1,43
$40^\circ$	1,02
$50^\circ$	0,91
$60^\circ$	0,77

Če primerjamo meritve piramidnega letala s podatki, ki smo jih vzeli za osnovo, lahko ocenimo, da so merske napake velikostnega reda nekaj stotink  $\frac{\text{m}}{\text{s}}$ . V okviru te natančnosti meritve potrjujejo napoved modela, da  $V_{max}$  ni odvisna od  $R$  in  $n$ .

Zdaj pa meritve primerjajmo še z modelom. Na naslednjih grafih so z rdečimi pikami prikazane meritve iz desnih dveh tabel. Modri krivulji predstavljata napoved modela



Slika 19:  $V_{max}$  meritve piramidni

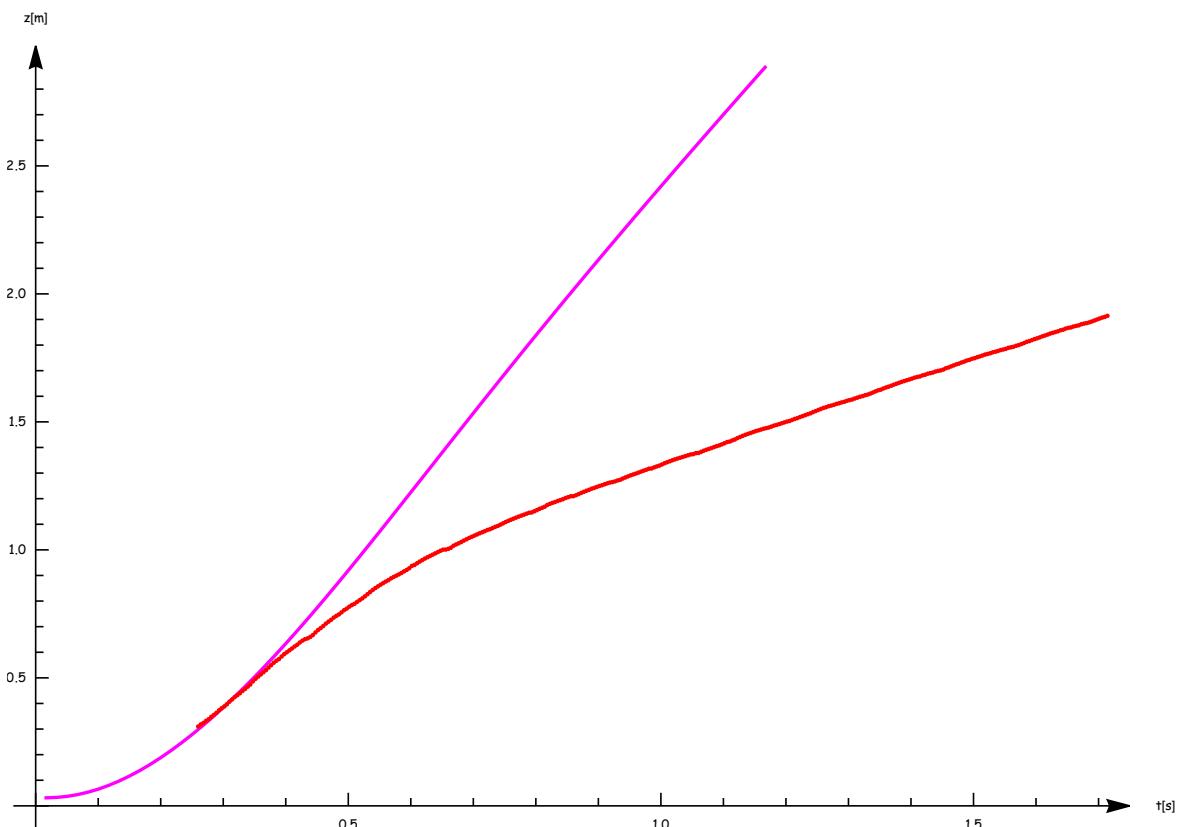
Na levem grafu lahko vidimo, da model za vse manjše kote  $\alpha$  napove vse bolj prevelike vrednosti. Odstopanje modela je reda  $0,2 \frac{m}{s}$ . Model temelji na približku stoječega zraka okoli letala in sile curka. V enačbo za terminalno hitrost dodamo še zavorni člen, linearen s hitrostjo. Ta predstavlja približek za viskozne sile. S FindFit poiščemo vrednost linearnega koeficiente in dobimo boljše prileganje, ki ga prikazujeta magenta črti. Opozimo lahko tudi, da je pri  $160 \frac{g}{m^2}$  papirju terminalna hitrost višja, kot pri naslednjem težjem papirju. Ker že vemo, da viskozne sile zraka niso zanemarljive, lahko to pojasnimo z dejstvom, da je bil  $160 \frac{g}{m^2}$  papir fotografski in manj hrapav od ostalih.

Preverimo lahko še lastnost, piramidnih latal, da se vedno obrnejo z vrhom navzdol. To dobimo tudi v simulaciji.

Simuliramo lahko tudi preprost primer ukrivljene ploskve, stožec. Tega dobimo v limiti  $\lim_{n \rightarrow \infty}$ . Ko vzamemo za  $n$  veliko število ploskev pa nastane težava. Ker je v spodnjem oglisču povezana vsaka ploskev z vsako drugo, računska kompleksnost narašča kot  $\mathcal{O}(n^2)$ . Ta problem lahko rešimo tako, da namesto trikotnikov vzamemo enakokrake trapeze, ki segajo skoraj do vrha piramide.

## 6.2 REZULTATI ZA HELIKOPTER LETALO

Za merjene helikopterje je bil uporabljen  $160 \frac{g}{m^2}$  papir, ki je dovolj čvrst, da ploskve med letom ostanejo ravne. Gibanje helikopter letala lahko opišemo tako: najprej začne padati s pospeškom  $g$ . Potem, ko dobi nekaj hitrosti, grejo krila zaradi sile zraka bolj skupaj. Hkrati se veča tudi kotna hitrost vrtenja tega letala okoli navpične osi. Krili gresta nato bolj narazen, zaradi povečane centripetalne sile nanju. Ker gresta bolj narazen, se padanje helikopterja upočasni. V limiti se hitrost in kotna hitrost letala ne spreminja več. Enak potek faz leta dobimo tudi v simulaciji. Na spodnjem grafu je prikazana višina helikopterja v odvisnosti od časa  $z(t)$ . Magenta črta je iz simulacije, rdeča pa predstavlja meritve.



Slika 20:  $z(t)$  helikopterja

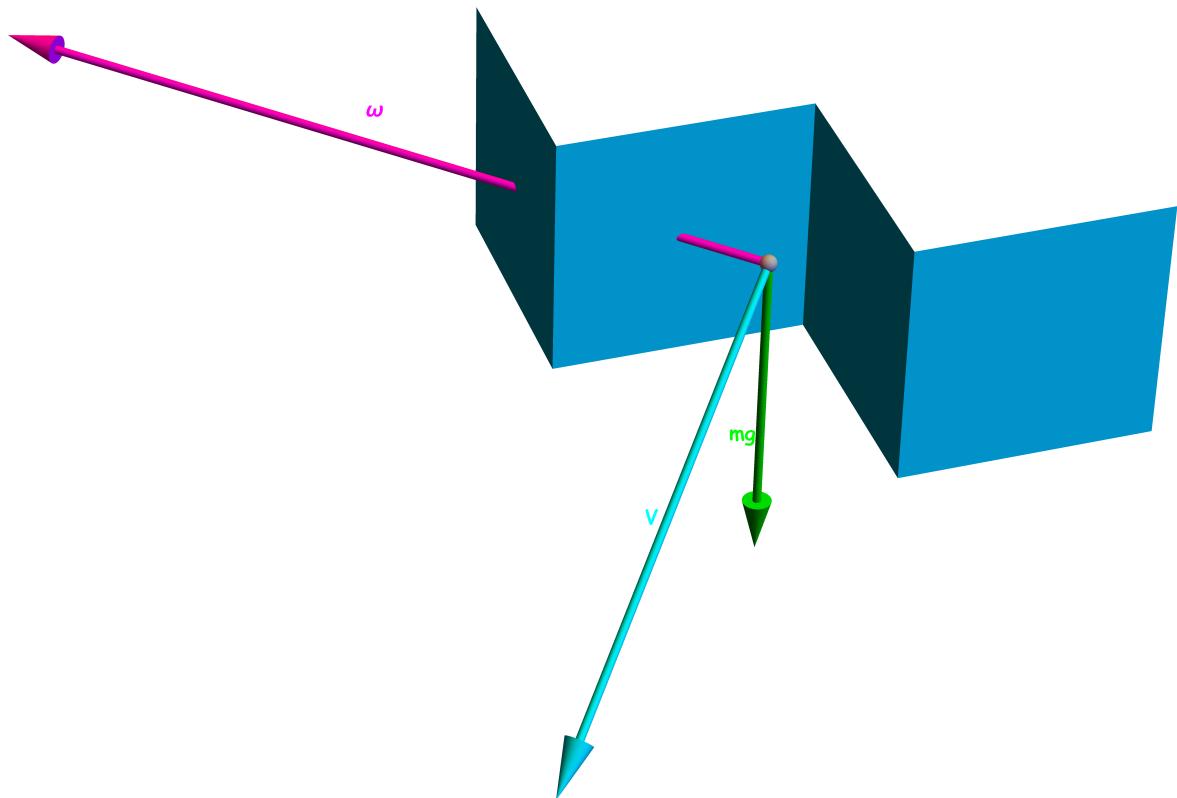
Opazimo lahko, da je simulacija na začetku precej natančna, saj so sile zraka še majhne in je gibanje precej blizu prostemu padu. V nadaljevanju se v simulaciji padanje upočasni, ampak veliko manj, kot v resnici.

Še ena lastnost helikopterja, ki jo dobimo tudi v simulaciji, je stabilnejši let, če je letalo spodaj obteženo. To v resnici naredimo tako, da spodno majhno ploskev večkrat zapognemo navzgor (5-krat konkretno na primeru testiranega helikopterja). Kot omenjeno, v simulaciji več plasti papirja ne deluje pravilno. Zato namesto tega samo pomnožimo

maso spodnje ploskve s 5.

### 6.3 REZULTATI ZA W LETALO

To se giblje na zelo zanimiv način. Neglede na to kako izpustimo to letalo, se po nekaj časa gibanje ustali tako, da se težišče letala giblje po premici, letalo pa se obrača s kotno hitrostjo v vodoravni ravnini in pravokotno na premico gibanja težišča (glej sliko).



Slika 21: W letalo

Letalo se v simulaciji ne začne obračati, temveč samo naravnost pada. Ker model ne deluje, lahko sklepamo, da je za opis gibanja tega letala ključno gibanje zraka okoli letala. V praksi se izkaže, da enako gibanje dobimo že pri podolgovatem papirju v obliki enojnega V in drugih sorodnih oblikah, a je pri W obliki v primerjav z ostalimi zelo hiter prehod na končno gibanje. W letalo je dober primer enostavnega papirnega letala, ki ga s tem modelom ne moremo opisati.

## 7 RAZPRAVA IN ZAKLJUČEK

Ko sem se lotil raziskovanja in si ustvaril pregled nad obsežnostjo primerov, ki so me zanimali, sem si moral postaviti izvedljive cilje. To so bili:

- Simulacija papirnih letal z zgornjimi predpostavkami
- Sprogramirano sledenje gibanju papirnih letal iz posnetka
- Eksperimentalno preverjanje modela

Cilje sem izpolnil. Naloga je izvirna in model je preverjen na treh vrstah papirnih letal. Naloga ponuja še mnogo možnosti za nadaljnje raziskovanje:

- Uvedba RK metode v simulaciji
- Uvedba različnih  $k$ -jev vzmeti (kjer so prisotne večje sile, večjih). Potreben bi bil algoritmom, ki bi ocenil kolikšen  $k$  rabimo med dvema ogliščema, da bo med njima odmik ves čas manjši od vnešene omejitve.
- Ugotoviti kako se navor pregibov spreminja s spremembo kota.
- Uporaba funkcije NDSolve za reševanje Navier Stoksovih enačb namesto približka s silo curka
- Simulacija za papirna letala iz več plasti papirja
- Simuliranje ukrivljanja papirja
- Izboljšanje obdelave videa
- GUI za origami pregibanje namesto ročne podaje koordinat oglišč ploskev letala

## 8 VIRI IN LITERATURA

- [1] 3Blue1Brown. *Change of basis - Essence of linear algebra, chapter 13*. Youtube. 2016. URL: <https://www.youtube.com/watch?v=P2LTAU01Tda>.
- [2] Stand-up Maths. *Ellipsoids and The Bizarre Behaviour of Rotating Bodies*. Youtube. 2020. URL: <https://www.youtube.com/watch?v=151LcwHOW7s>.
- [3] rubinhlndau. *8.5 ODE Algorithms*. Youtube. 2020. URL: [https://www.youtube.com/watch?v=Yr\\_ZSnu6XbY](https://www.youtube.com/watch?v=Yr_ZSnu6XbY).
- [4] The Coding Train. *Coding Challenge nr. 93: Double Pendulum*. Youtube. 2018. URL: [https://www.youtube.com/watch?v=uWzPe\\_S-RVE](https://www.youtube.com/watch?v=uWzPe_S-RVE).
- [5] WIRED. *Aerodynamics Explained by a World Record Paper Airplane Designer - Level Up - WIRED*. Youtube. 2020. URL: [https://www.youtube.com/watch?v=3KqjRPV9\\_PY](https://www.youtube.com/watch?v=3KqjRPV9_PY).

## **9 PRILOGE**

### **9.1 SIMULACIJA GIBANJA ENE PLOSKVE**

## **9.2 SIMULACIJA GIBANJA PAPIRNEGA LETALA**

### **9.3 ff-MPEG SKRIPTA ZA IZVOZ VIDEO-POSNETKA IZ SLIK**

## **9.4 PROGRAM ZA SLEDENJE PIKAM**

## **9.5 PROGRAMI ZA IZRIS MREŽ LETAL**

## **9.6 MREŽE LETAL**