

Računalna aplikacija za trodimenzijski prikaz brodskog vijka

Rosandić, Ante

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:235:457305>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-08**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Ante Rosandić

Zagreb, 2017.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor:

Prof. dr. sc. Nastia Degiuli, dipl. ing.

Student:

Ante Rosandić

Zagreb, 2017.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se profesorici Degiuli na iskazanom povjerenju, podršci tokom studija i izrade rada. Zahvalio bih se i Ivanu Muniću, dipl.ing. koji mi je uvelike pomogao sa svojim stručnim savjetima tijekom izrade rada. Također se zahvaljujem prijateljima, kolegama, obitelji i djevojci.

Ante Rosandić



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za završne i diplomske ispite studija brodogradnje



Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

ZAVRŠNI ZADATAK

Student: **Ante Rosandić** Mat. br.: 0035193272

Naslov rada na hrvatskom jeziku: **RAČUNALNA APLIKACIJA ZA TRODIMENZIJSKI PRIKAZ BRODSKOG VIJKA**

Naslov rada na engleskom jeziku: **COMPUTER APPLICATION FOR THREE-DIMENSIONAL VISUALISATION OF A SHIP PROPELLER**

Opis zadatka:

Računalni trodimenzijski prikaz postaje standard u projektiranju broda. Često zamjenjuje izradu fizičkih modela i pomaže pri projektiranju dijelova broda koji nisu prikladno opisani tradicionalnim, linijskim metodama, te pri izračunu potrebne količine materijala za dijelove broda, i omogućuje lakši i izravniji pregled konačnog proizvoda. Brodski vijak je jedan od najkompleksnije zakrivljenih dijelova broda jer je rotacijsko tijelo zakrivljeno u više smjerova. Stoga je najčešće potreban trodimenzijski prikaz vijaka kako bi se dobio zorni uvid u konačni izgled. To je moguće ili izradom modela-makete ili računalnim alatima (aplikacijama). Izrada jedne ovakve aplikacija pomogla bi u nastavi i dala trodimenzijski uvid u izgled više različitih serija brodskih vijaka, a omogućila bi i pripremu ulaznih podataka za više metoda izrade modela vijaka poput lijevanja u kalup ili CNC izradu.

U radu je potrebno:

- Prikupiti podatke o geometrijskim karakteristikama više serija brodskih vijaka kako bi se što kvalitetnije odredila struktura ulaznih podataka za računalnu aplikaciju.
- Analizirati mogućnosti različitih dostupnih računalnih modela prikaza ploha te odabrati najprikladniji za trodimenzijski prikaz krila vijaka.
- Izraditi računalnu aplikaciju za standardni prikaz brodskog vijaka.
- Izraditi računalnu aplikaciju za trodimenzijski prikaz brodskog vijaka.
- Ulazni dio aplikacije napraviti prilagodljivim različitim serijama vijaka i na taj način omogućiti jednostavno dodavanje novih serija vijaka.
- Grafički dio aplikacije prilagoditi odabranom računalnom modelu prikaza plohe kako bi ploha bila što jednostavnije, a i dalje točno opisana.
- Izraditi korisničke upute za aplikaciju.

Zadatak zadan:
30. studenog 2016.


Rok predaje rada:
1. rok: 24. veljače 2017.
2. rok (izvanredni): 28. lipnja 2017.
3. rok: 22. rujna 2017.

Predviđeni datum obrane:
1. rok: 27.2. - 03.03. 2017.
2. rok (izvanredni): 30. 06. 2017.
3. rok: 25.9. - 29. 09. 2017.

Zadatak zadao:


Prof. dr. sc. Nastija Degiuli

Predsjednica Povjerenstva:


Prof. dr. sc. Nastija Degiuli

SADRŽAJ

POPIS SLIKA	II
POPIS TABLICA.....	III
POPIS OZNAKA	IV
SAŽETAK	V
SUMMARY	VI
1. Brodski propulzori	1
1.1 Općenito o propulzorima	1
1.2 Vrste brodskih propulzora.....	1
1.3 Podjela brodskih propulzora.....	2
1.4 Vijak s konstantnim usponom	3
1.5 Geometrija vijka	4
1.6 Uspon vijka	6
1.7 Dijelovi vijka.....	7
2. Nacrt broskog vijka	9
3. Serije brodskih vijaka.....	12
3.1.1 Wageningen B serija	12
3.1.2 Japanska AU serija.....	13
3.1.3 Gawn-ova serija	13
3.1.4 KCA serija.....	14
3.1.5 Lindgren (MA) serija	14
3.1.6 Newton-Rader serija	15
4. Izrada aplikacije za trodimenzijski prikaz broskog vijka.....	17
4.1 NURBS krivulje	17
4.2 NURBS plohe.....	19
4.3 Rhinoceros 5.....	19
4.4 Ulazni podaci za crtanje vijka	20
5. Zaključak.....	29
Literatura.....	30
PRILOZI	31

POPIS SLIKA

Slika 1.	Podjela brodskih propulzora [1].....	2
Slika 2.	Vijak s konstantnim usponom.....	4
Slika 3.	Referentne krivulje krila [1]	4
Slika 4.	Oblik segmentnih profila na različitim radijusima ($0,50R$ i na $0,750R$)	5
Slika 5.	Presjek profila na radijusu r [3]	5
Slika 6.	Nastajanje vijčanice [1]	6
Slika 7.	Trokut uspona vijčanice [1]	6
Slika 8.	Detaljan prikaz vijka [2]	8
Slika 9.	Zaobljeni vrh ulaznog i izlaznog brida [1]	9
Slika 10.	Nacrt brodskog vijka	11
Slika 11.	Geometrija profila Wageningen B serije [3]	13
Slika 12.	Krivulje krila <i>Gawn</i> -ove serije [3]	14
Slika 13.	Oblik presjeka profila Newton-Rader serije vijaka.....	15
Slika 14.	Kontrolni poligon, i pomak kontrolnih točaka [4]	18
Slika 15.	Vanjske krivulje krila	22
Slika 16.	Izračunata geometrija profila	22
Slika 17.	Točka rotacije.....	23
Slika 18.	Rotacija koordinatnog sustava u ravnini [8].....	23
Slika 19.	Zakretanje točaka profila i prebacivanje iz 2D u 3D	25
Slika 20.	Spajanje zakrenutih profila i lijepljenje na cilindar	25
Slika 21.	Spojene točke ulaznog i izlaznog brida nagnutog profila koji je zalijepljen na cilindar	26
Slika 22.	Ploha generirana kao mreža krivulja.....	27
Slika 23.	Nacrtana produljena ploha koja ulazi u glavinu	28
Slika 24.	Prikaz vijka u 3D.....	28

POPIS TABLICA

Tablica 1. Dijelovi vijka	7
Tablica 2. Omjeri dimenzija trokrilnog vijka po radiusima [3]	10
Tablica 3. Karakteristike serija u ovisnosti o raznim parametrima [3]	16
Tablica 4. Omjeri dimenzija više krilnog vijka po radiusima [3]	20
Tablica 5. Vrijednosti V1 koje se koriste za računanje ordinata točaka [3]	21
Tablica 6. Vrijednosti V2 koje se koriste za računanje ordinata točaka [3]	21

POPIS OZNAKA

Oznaka	Opis	Mjerna jedinica
D	Promjer vijka	m
P	Uspón vijka	m
A_E/A_0	Omjer površine krila i površine diska vijka	-
Z	Broj krila vijka	-
P/D	Omjer uspona i promjera vijka	-
α	Nagib krila vijka	rad
ψ	Srpolikost krila vijka	rad
φ	Kut uspona vijčanice	rad
c	Duljina tetive profila	m
a	Duljina tetive od ravnine vijka do izlaznog brida	m
b	Duljina tetive od točke najveće debljine do ulaznog brida	m
Ar	Bezdimenzijski koeficijenti za računanje omjera t/D	-
Br	Bezdimenzijski koeficijenti za računanje omjera t/D	-
β	Kut rotacije profila vijka	rad
t	Debljina profila vijka	m
r/R	Omjer visine presjeka i radijusa krila vijka	-
θ	Kut savijanja profila vijka	rad

SAŽETAK

Ideja ovog rada bila je olakšati studentima i inženjerima trodimenzijski prikaz najkompleksnijeg dijela broda, a to je brodski vijak. Dana je općenita podjela brodskih propulzora, značajke brodskih vijaka s konstantnim usponom i geometrija brodskog vijka Wageningen B serije. U izradi programskog koda za trodimenzijski prikaz brodskog vijka korištena je poznata geometrija Wageningen B serije, NURBS krivulje i programski paket Rhinoceros 5, točnije RhinoScript koji koristi Microsoft-ov Visual Basic for Applications programski jezik. U radu su priložene korisničke upute kako bi se olakšalo korištenje programskog koda. Upotrebom programskog koda moguće je napraviti dvodimenzijski nacrt kao i trodimenzijski plošni prikaz brodskog vijka. Programski kod je napravljen tako da je, uz poznavanje geometrije, moguće prikazati i brodski vijak bilo koje druge serije.

Ključne riječi: brodski vijak, trodimenzijski prikaz, Wageningen B serija, Rhinoceros 5, RhinoScript

SUMMARY

The main idea of this paper was to facilitate three dimensional visualization of the ship's most complex part, marine propeller for students and engineers. General classification of marine propulsors, the characteristics of constant pitch propellers and the geometry of Wageningen B series propellers are given. The well known Wageningen B series geometry, NURBS curves and Rhinoceros 5 programme package, more precisely RhinoScript which uses Microsoft Visual Basic for Applications programming language, were used to create source code for three dimensional visualization of marine propeller. User's manual is attached within the paper to simplify the use of the source code. By using the source code it is possible to create two dimensional drawing as well as the three dimensional surface visualization. The source code was created in order to be able to present marine propeller of any other series with known geometry.

Key words: marine propeller, three dimensional visualization, Wageningen B series, Rhinoceros 5, RhinoScript

1. Brodski propulzori

1.1 Općenito o propulzorima

Propulzor je tehnička naprava koja je ugrađena na plovilo/vozilo i djeluje prikladno usmjerenom silom kako bi se započelo gibanje plovila/vozila. Propulzori se najčešće nalaze u tekućem mediju kroz koji ostvaruju gibanje (npr. silu poriva). Mediji koji su uobičajeni za rad propulzora su voda, zrak, vakuum.

Propulzori za ostvarivanje svoga primarnog zadatka što je gibanje kroz medij koriste energiju raznih oblika. Najčešće je to toplinska energija dobivena iz goriva (diesel, bunker C, kerozin) pomoću toplinskih strojeva, ali može biti dobivena i iz drugih izvora kao što su elektromotori punjeni fotonaponskim ćelijama, nuklearni pogon...

Imajući u vidu da se propulzori ne koriste samo kao pogon broda, već i na raznim objektima kao npr. avioni, svemirske letjelice, dronovi i drugo, dalje u ovom radu kada se spomene propulzor odnosit će se isključivo na brodski propulzor [1].

1.2 Vrste brodskih propulzora

Propulzore dijelimo prema mediju u kojemu se nalaze, primarnim zadaćama tj. eksploataciji, razini uronjenosti, konstrukcijskoj složenosti, izgledu, načinu izrade te načinu na koji se stvara sila poriva.

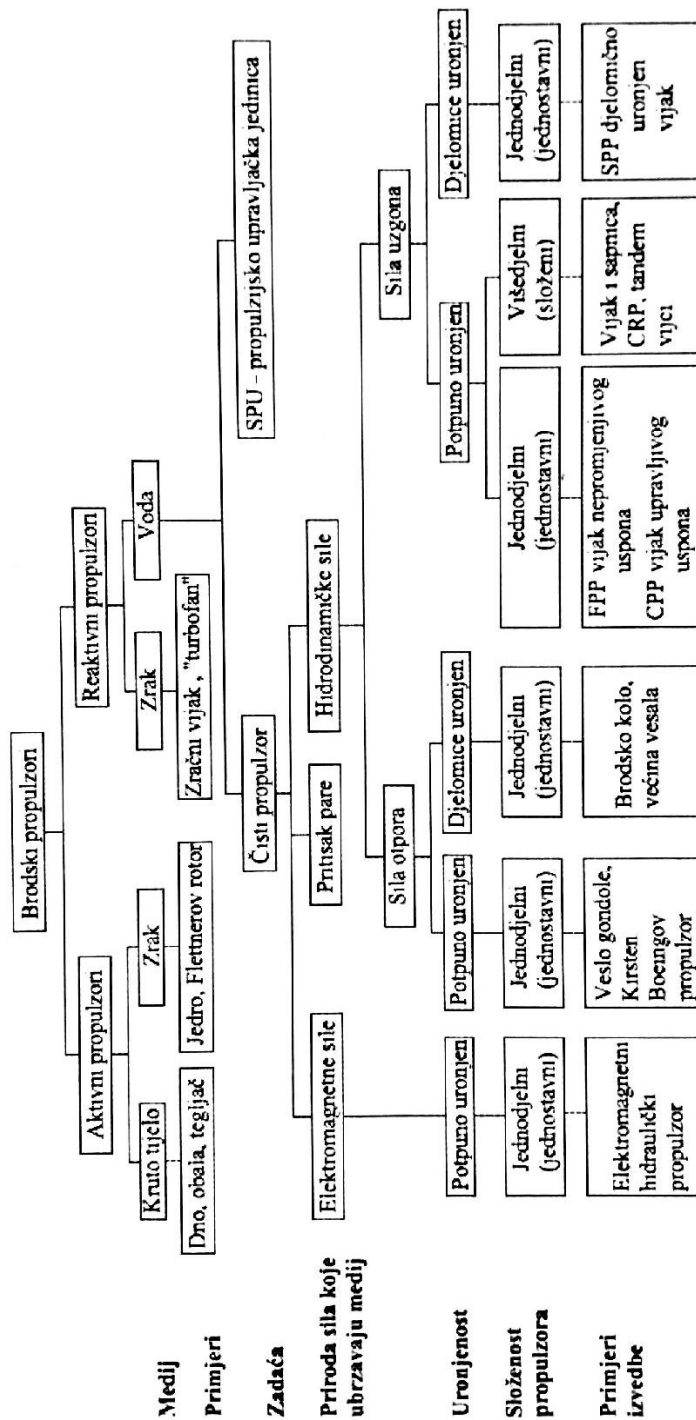
Postoje dva načina stvaranja sile porive. Prvi način je sila otpora koja je ujedno i najstariji način pokretanja broda (veslo, jedro). Drugi način je da se na vodu djeluje silom uzgona.

S obzirom na navedeno, brodski propulzori koji se najviše koriste u praksi su [2]:

- vijci s konstantnim usponom (eng. *fixed pitch propeller* FPP),
- vijci s promjenjivim usponom (eng. *controllable pitch propeller* CPP),
- vijci u sapnici (eng. *ducted propellers*),
- pod – propulzori (eng. *thrusters-pod drives*),
- cikloidni ili *Voith-Schneiderovi* propeleri,
- vodomlazni propulzori (eng. *water jet*).

1.3 Podjela brodskih propulzora

Podjela brodskih propulzora prikazana je na Slici 1.



Slika 1. Podjela brodskih propulzora [1]

1.4 Vijak s konstantnim usponom

Brodski vijak s konstantnim usponom je tip vijka koji je najrasprostranjeniji, Slika 2. Sastoji se od glavine kroz koju se prostire osovina za koju je poželjno da bude vodoravna i paralelna uzdužnoj simetrali broda, i krila kojih uobičajeno ima 2 – 9. Krila se za glavinu drže svojim korijenom koji mora biti izveden s postupnim zadebljanjem tako da se opterećenje ravnomjerno raspodjeli a strujanje bude što manje poremećeno. Vijci s konstantnim usponom spadaju pod propulzore koji stvaraju silu poriva stvarajući uzgon na krilnim tijelima (krila vijka).

Presjek krila u smjeru strujanja ima karakterističan oblik koji se naziva hidrodinamički, aerodinamički, strujni ili uzgonski profil. Na takvim krilima vrlo lako dolazi do nastanka kavitacije te je iz tog razloga potrebno vijak projektirati za svaki brod posebno. Iznos poriva i brzine broda mijena se s promjenom broja okretaja vijka.

Gledamo li vijak s krme prema pramcu, a brod napreduje pramcem i vijak se okreće u smjeru kazaljke na satu, kažemo da je vijak desnokretan. Vijak može biti monolitan, ako je lijevan u jednom dijelu (glavina i krila skupa) ili sastavljeni ako su krila izrađena svako za sebe te potom spojena s glavinom.

Neke od prednosti ovakvog propelera su:

- visoki stupanj djelovanja,
- jednostavan prijenos snage od glavnog pogonskog stroja do vijka,
- povećanje gaza bolje djeluje na rad vijka,
- mogu se ugraditi i na brodove s malim gazom (potrebna tunelska krma).

Nedostatci vijka s konstantnim usponom su:

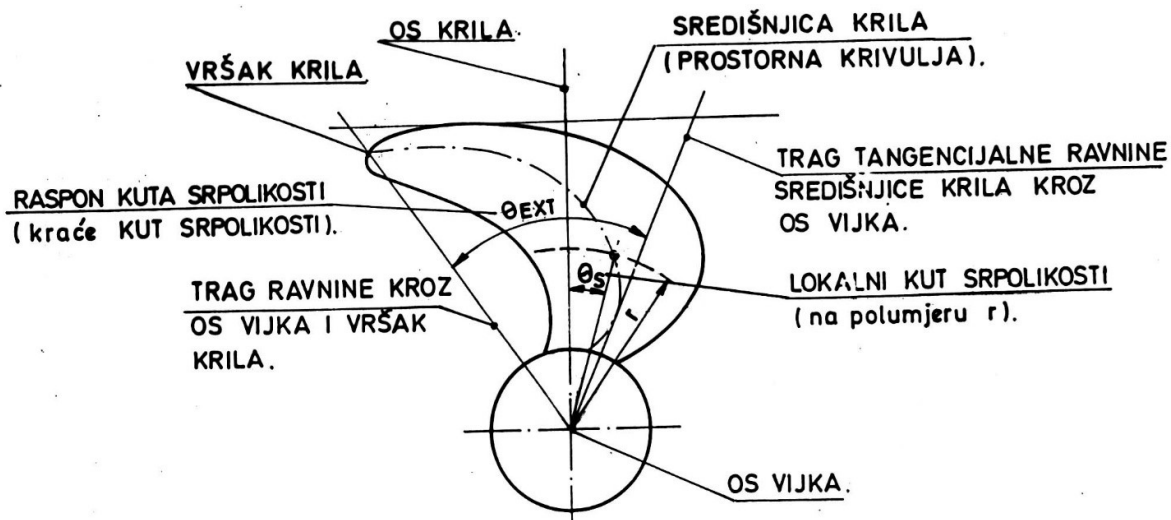
- vibracije uslijed rada glavnog stroja,
- vibracije nastale zbog kavitacije,
- grba otpora – glavni stroj se nalazi u preopterećenju i dolazi do nepotpunog izgaranja goriva zbog nepovoljnog područja radne točke,
- mala iskoristivost kod brzih brodova pa se najviše koristi za deplasmanske brodove,
- plovidba krmom- potreban prekretni stroj ili prekretni reduktor.



Slika 2. Vijak s konstantnim usponom

1.5 Geometrija vijka

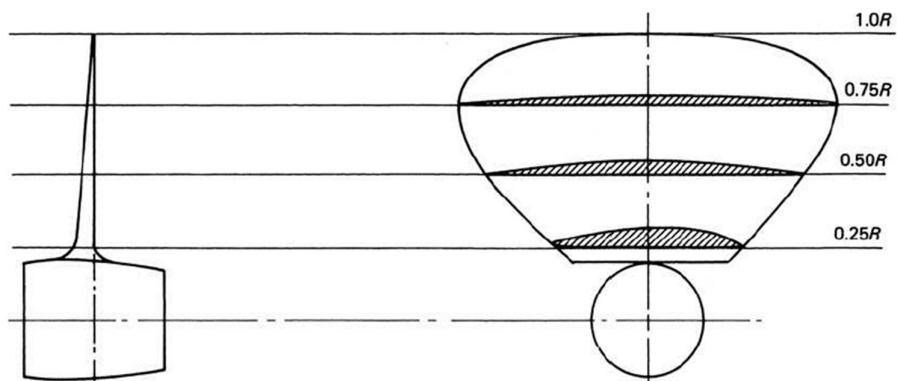
Krilo vijka definirano je oko osi koja je okomita na os osovine koju nazivamo os krila (eng. *propeller reference line*), Slika 3. Aeroprofil (strujni) profili koji zajedno čine krilo propelera su definirani kao presjek cilindra koji ih opisuje (cilindrični presjek), a čija je uzdužna os kolinearna s osi osovine. Zbog toga se izraz „kružni profil“ (presjek krila ili kraće profil) često koristi kao pojam kod opisivanja geometrije propelera. Za opisivanje geometrije potrebno je definirati koordinatni sustav, a ishodište tog koordinatnog sustava je u središtu vijka.



Slika 3. Referentne krivulje krila [1]

Prije pojave strujnih profila, najviše su se pojavljivali segmenti profila sa zaobljenim vrhom. Danas se takvi profila koriste kod serija vijaka za brže brodove kao što je Gawn-ova serija i

Newton-Rader-ova serija. Na Slici 4. se vidi da su to profili oblika kružnog odsječka sa zaobljenim vrhovima. Također se vidi da os glavine prolazi kroz središte vijka.

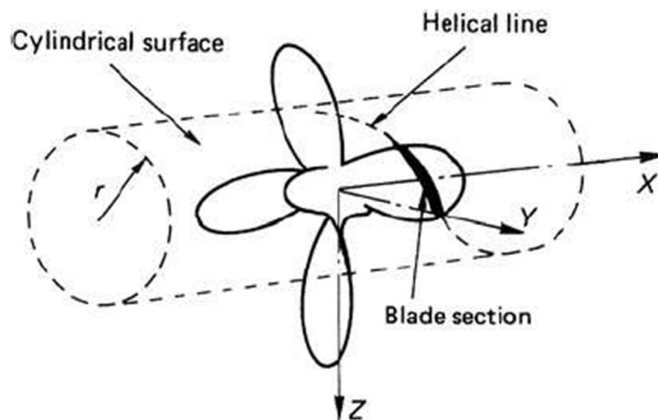


Slika 4. Oblik segmentnih profila na različitim radijusima (0,50R i na 0,750R)

Najvažniji parametri geometrije vijka su:

- promjer D ,
- uspon P ,
- omjer površine krila i površine diska A_E / A_0 ,
- broj krila Z ,
- omjer uspona i promjera P / D ,
- nagib krila α ,
- srpolikost, ψ .

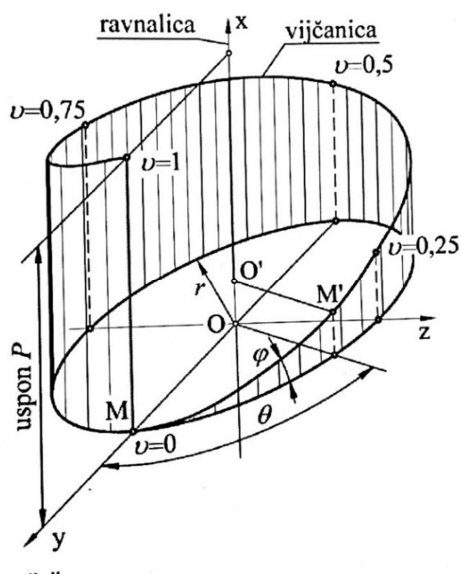
Presjek profila na radijusu r je prikazan na Slici 5. gdje je *cylindrical surface* valjak radijusa r , koji siječe krilo vijka radijusa R , *helical line* je linija zavojnice, a *blade section* je presjek krila vijka s cilindrom radijusa r .



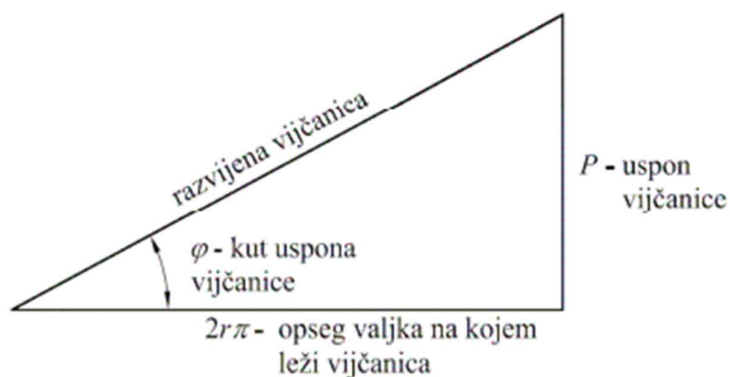
Slika 5. Presjek profila na radijusu r [3]

1.6 Uspon vijka

Uspon ili korak vijka (eng. *pitch*) P , definira se kao teorijski aksijalni pomak vijka prema naprijed pri jednom okretaju vijka ($2\pi r$), slika 6. Shvatljivo je da je uspon zapravo razlika između neke početne točke, nazovimo je P_0 , koja se nalazi na površini cilindra radijusa „ r “ i pomiče se duž cilindra po zavojnici, sve dok vijak ne napravi jedan puni okret (360°) i ta točka se nalazi u novom položaju P_1 koja je okomita na točku P_0 . Za točku P_0 možemo reći da je ortogonalna projekcija točke P_1 [1].



Slika 6. Nastajanje vijčanice [1]



Slika 7. Trokut uspona vijčanice [1]

Na Slici 7., [1], vidljivo je da je kut uspona vijčanice φ , određen usponom koji je vijčanica napravila i opsegom valjka na kojem ona leži te slijedi:

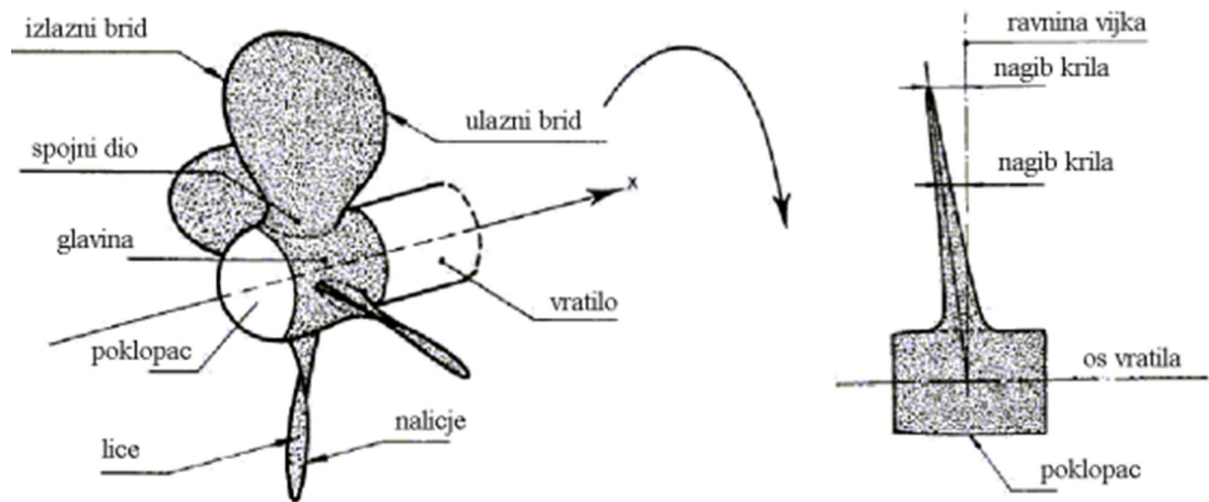
$$\varphi = \arctg^{-1} \frac{P}{2\pi r} \quad (1)$$

1.7 Dijelovi vijka

Da bi bolje razumjeli kako radi vijak, i kako ga najbolje prikazati, potrebno je detaljno opisati svaki njegov dio. Detaljan opis dijelova vijka dan je u Tablici 1, a detaljan prikaz vijka dan je na Slici 8, [2].

Tablica 1. Dijelovi vijka

Ulazni brid	dio krila koji „prvi“ udara u vodu, možemo zamisliti kao da je siječe, a povezuje lice i naličje krila (eng. <i>leading edge</i>)
Izlazni brid	dio krila koji prati ulazni brid i također spaja lice i naličje (eng. <i>trailing edge</i>)
Spojni dio	ili korijen, dio gdje se spajaju glavina i krilo, potrebno je da zadebljanje bude postepeno izvedeno (eng. <i>blade root</i>)
Glavina	dio vijka koji je pričvršćen za završetak osovine i preko glavine se okreće vijak (eng. <i>hub</i>)
Poklopac	zatvara glavinu, mora biti strujno dobro oblikovan kako bi strujanje što homogenije dolazilo na vijak (eng. <i>cap</i>)
Vratilo	osovina dolazi iz pogonskog stroja i ima svoj smjer vrtnje i broj okretaja (eng. <i>shaft</i>)
Lice	strana vijka koja se vidi kada gledamo iz krme prema pramcu, tlačna strana (eng. <i>pressure side, face</i>)
Naličje	strana vijka koja se ne vidi kada gledamo s krme, nalazi se na strani osovine vijka, podtlačna strana (eng. <i>suction side, back</i>)
Ravnina vijka	ravnina okomita na osovinu vijka i sjecište tetive na presjecištu sa središnjicom osovine (eng. <i>propeller plane</i>)
Vrh krila	najudaljenija točka od središta glavine, spaja ulazni i izlazni brid (eng. <i>blade tip</i>)

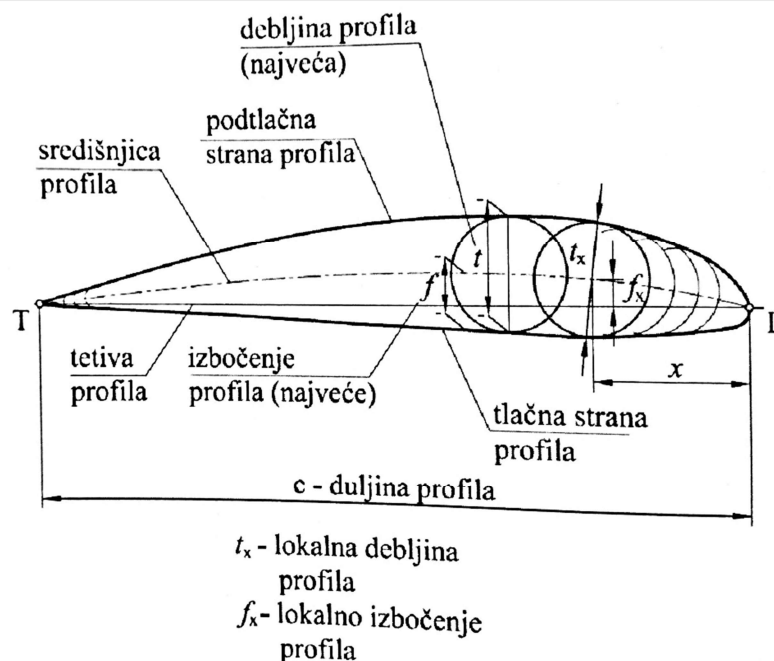


Slika 8. Detaljan prikaz vijka [2]

2. Nacrt broskog vijka

Nacrti nam općenito služe kako bi si mogli predočiti ono što radimo. Kod tijela koje je rotacijsko, a još k tome zakrivljeno u više smjerova, nacrt je upravo ono što nam olakšava crtanje i predodžbu.

Približavanje geometrije krila će nam biti utoliko lakše ako odustanemo od stvarnih strujnih profila, i zanemarimo zaobljenja na vrhu, Slika 9.



Slika 9. Zaobljeni vrh ulaznog i izlaznog brida [1]

U pravilu, nacrt broskog vijka sadrži i rašireno krilo vijka. Rašireno krilo vijka je u suštini deformirana slika stvarnog krila.

Nacrt nam daje informaciju o izgledu (obliku) u paru s nekom brojčanom vrijednosti (dimenzija). Izrada tehničke dokumentacije je zahtijevala da ti nacrti budu precizni, točni, i lako čitljivi, što je projektantima zadavalo puno problema. Ručno crtanje nacrti danas je u potpunosti zamijenila upotreba raznih računalnih alata namijenjenih za crtanje/projektiranje, a oni su uvelike ubrzali izradu i pospješili same nacрте u odnosu na prijašnju upotrebu letvica i krivuljara. U ovom radu se koristio programski paket Rhinoceros.

Tumačenje nacrti može biti problematično, a pogotovo stvara problem ljudima koji nemaju iskustva u čitanju nacrti. Zbog toga je nužno i poželjno da se zahtjevne nacрте, a nacrt broskog vijka to svakako jest, prikaže u trodimenzijskoj projekciji (3D). Nedostatak takvih

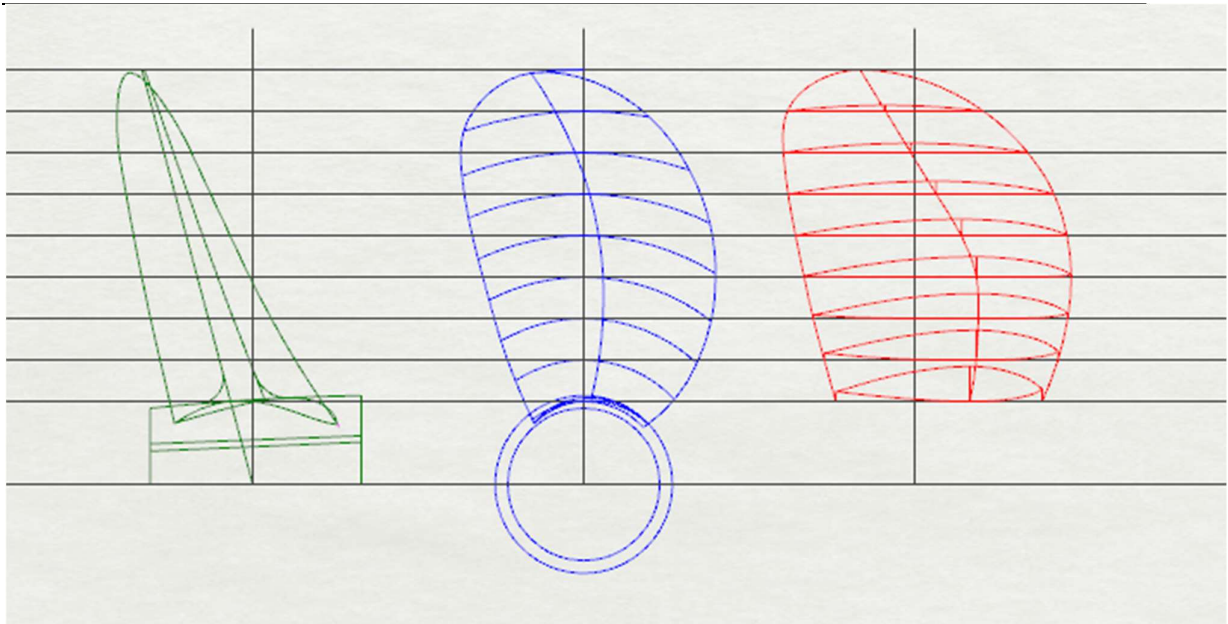
alata, koji bi bili fleksibilni pri projiciranju broskog vijka u trodimenzijskom prikazu (prvenstveno se misli na mijenjanje ulaznih parametara npr. D , P/D , Z , A_E/A_0 i neki drugi) potaknuo nas je da kroz ovaj rad omogućimo i olakšamo prvenstveno studentima prikaz vijka u naravi.

Tablica 2. Omjeri dimenzija trokrilnog vijka po radijusima [3]

Dimensions for three-bladed propellers

r/R	$\frac{c}{D} \cdot \frac{Z}{A_E/A_0}$	a/c	b/c	$t/D = A_r - B_r Z$	
				A_r	B_r
0.2	1.633	0.616	0.350	0.0526	0.0040
0.3	1.832	0.611	0.350	0.0464	0.0035
0.4	2.000	0.599	0.350	0.0402	0.0030
0.5	2.120	0.583	0.355	0.0340	0.0025
0.6	2.186	0.558	0.389	0.0278	0.0020
0.7	2.168	0.526	0.442	0.0216	0.0015
0.8	2.127	0.481	0.478	0.0154	0.0010
0.9	1.657	0.400	0.500	0.0092	0.0005
1.0	0.000	0.000	0.000	0.0030	0.0000

U Tablici 2. dani su omjeri dimenzija trokrilnog vijka po radijusima, omjer r/R , je zadani presjek krila na određenom radijusu r u odnosu na radijus cijelog krila R , vidi Sliku 5., a/c i b/c su omjeri duljina gdje je c duljina tetive, a je duljina između izlaznog brida od ravnine vijka, b je duljina između ulaznog brida do mjesta najveće debljine profila, D promjer vijka, Z broj krila vijka, A_E/A_0 omjer površine krila i diska. Bezdimenzijski koeficijenti za određivanje najveće debljine profila su A_r i B_r . Uz ove vrijednosti iz Tablice 2., jednadžbi 4,5 i koeficijenta $V1$ i $V2$ računamo koordinate svih točaka profila vijka. Nacrt broskog vijka dan je na Slici 10.



Slika 10. Nacrt brodskog vijka

3. Serije brodskih vijaka

Kroz povijest pojavio se veliki broj standardnih serija vijaka testiranih u raznim bazenima diljem svijeta. Prikazati sve bilo bi prekomplikirano i opširno da bi se obradilo u ovom radu. Zato će se ovdje spomenuti i analizirati samo neki od njih, tj. oni koji su se nekako najviše istaknuli i koji se još koriste (izrađuju).

Glavni cilj je sistematsko predstavljanje serija kako bi se projektantima prikazalo koji vijak je najpogodniji za potrebne nekog broda, koji to čimbenici i kako utječu na rad propelera te sveukupne performanse broda.

3.1.1 Wageningen B serija

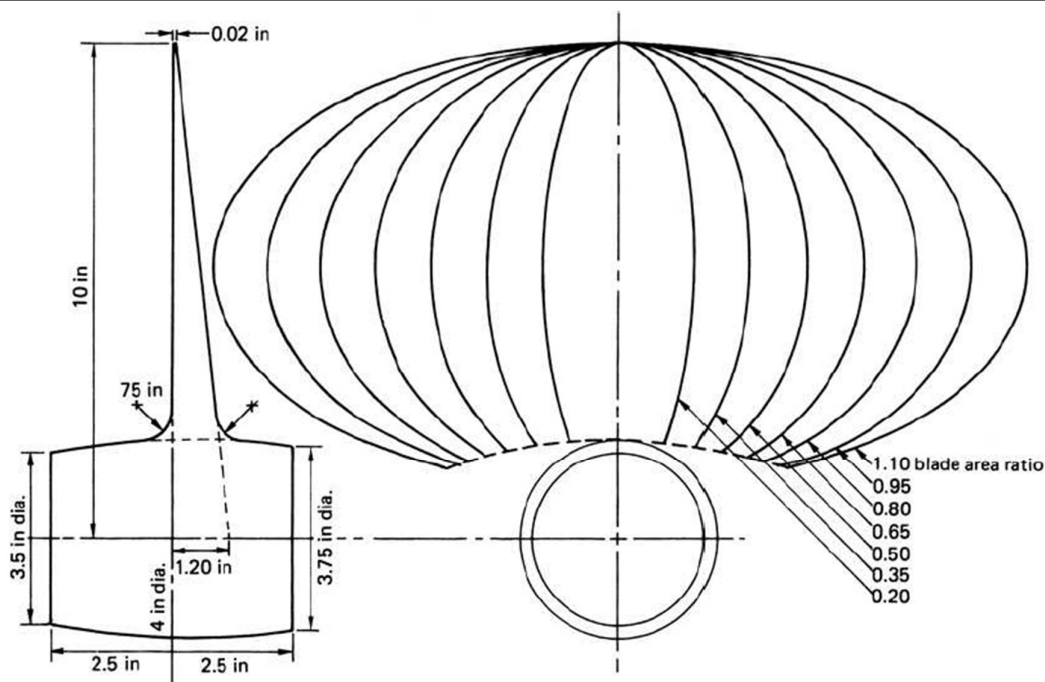
Wageningen B serija je vrlo vjerojatno najraširenija, najispitivanija i najkorištenija serija vijka. Serija je nastala u 40-im godinama 20. stoljeća. Kroz godine istraživanja vijka ove serije, dobili smo sveobuhvatni model vijka s konstantnim usponom. Analiziranjem svojstava vijka zaključilo se da dolazi do odstupanja u odnosu na nacrtane dijagrame i tu pogrešku su pripisali skaliranju različitih ispitivanja modela. To je rezultiralo novim procjenama serije u kojim su razlike u ispitivanju odmah uzete u obzir, a to je prvi objavio u svojem radu Van Lammeren [3].

Proširenje serije u smislu broja krila i omjera površina A_E/A_0 , rezultiralo je time da imamo 20 različitih kombinacija broja krila i omjera površina.

Oznaka ove serije općenito glasi BZ.y, gdje B označava tip serije, Z broj krila, a y – omjer površina A_E/A_0 . Uspon krila se kreće u rasponu od 0,6 – 1,4.

Wageningen B serija je serija široke namjene, konstantnog uspona a široku primjenu je ostvarila kao serija koja se najviše ispitivala i projektirala. Malo drugačija izvedba Wageningen B serije, je označena kao BB serija. BB serija je predstavljena kada se primijetilo da B serija ima duljinu tetive pri vrhu koja nije u potpunosti bila prihvaćena u praksi. Prema tome BB serija je imala redefiniranu konturu sa širim vrhom profila za razliku od B serije. BB serija unatoč tome nije naišla na širu primjenu.

Na Slici 11, [3], prikazana je geometrija profila Wageningen B serije gdje je *position of genetator line* pozicija ravnine vijka, *position of maximum thickness* pozicija najveće debljine profila, TE izlazni vrh, LE ulazni vrh, MT točka gdje je najveća debljina profila, tj. pozicija najveće debljine profila, a DI točka pozicije ravnine vijka.



Slika 12. Krivulje krila Gawn-ove serije [3]

3.1.4 KCA serija

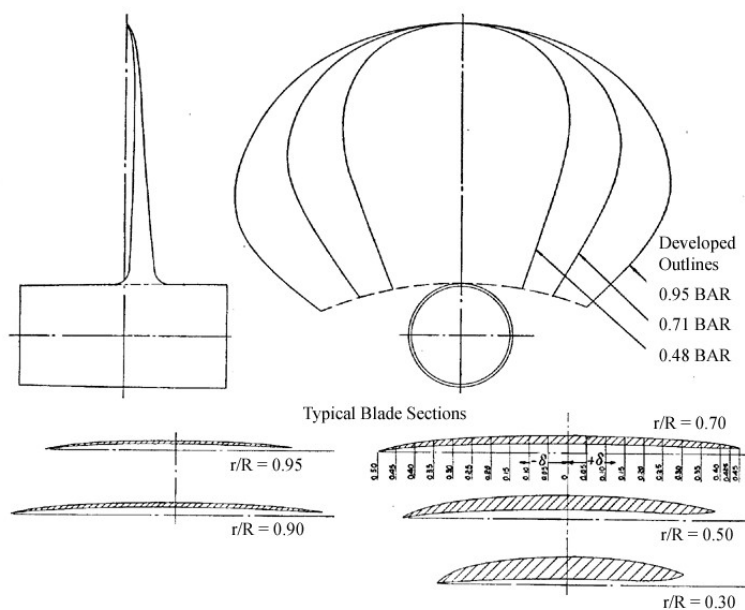
KCA serija za koju se ponekad govori Gawn – Burrill serija je po mnogočemu dodatni dio Gawn-ove serije. KCA serija uspoređuje 30 trokrilnih vijaka, promjera 406 mm obuhvaćajući raspon uspona od 0,6 – 2,0 i omjer površina od 0,5 – 1,1. Uočljivo je da ovakvi vijci mogu u dobrom dijelu pokriti vijke iz Gawn-ove serije zbog gornjih granica P/D i A_E/A_0 . Ovi vijci su ispitani u kavitacijskom tunelu na Sveučilištu u Newcastleu. Zbog toga je ova serija ispitana i na kavitaciju tj. s različitim kavitacijskim brojevima.

3.1.5 Lindgren (MA) serija

Lindgren serija je obuhvatila ispitivanja na trokrilnim i peterkrilnim vijcima, obuhvaća raspon omjera uspona i promjera P/D od 1 – 1,45 i raspon omjera površina od 0,75 – 1,2. Iz Tablice 3., vidljivo je da je u ovoj seriji ispitano ukupno 32 vijka. Svi vijci su konstantnog uspona s modificiranim eliptičnim oblikom krila. Promjer vijka ove serije je 250 mm i manji je od svih dosadašnjih, a promjer na koji mu dolazi glavina iznosi 0,2 D . Debljina krila varira ovisno o broju krila. Ova serija je ispitana i u bazenu za ispitivanja i u kavitacijskom tunelu i zbog toga imamo opsežne podatke o ispitivanju ove serije.

3.1.6 Newton-Rader serija

Newton-Rader je relativno mala serija, Tablica 3., koja je sadržavala 12 trokrilnih propelera za potrebe plovila za visokom brzinom. Značajke ove serije su da je raspon uspona 1,05 - 2,08 a omjer površina 0,5 – 1 zajedno s nelinearnom raspodjelom debljina profila po radijusu. Početni model ove serije je projektiran za posebne brodove s promjerom od 254 mm. Značajka početnog modela je bila da ima konstantan uspon od 1,2 i konstantan omjer površina 0,75. Zbog pojave kavitacije morao se modificirati oblik krila. Serija je ispitana za 9 različitih kavitacijskih brojeva. U Tablici 3. Dane su karakteristike serija u ovisnosti o raznim parametrima [3].



Slika 13. Oblik presjeka profila Newton-Rader serije vijaka

Tablica 3. Karakteristike serija u ovisnosti o raznim parametrima [3]

Series	Number of propellers in series	Range of parameters			$D(mm)$	r_h/R	Cavitation data available	Notes
		Z	A_E/A_O	P/D				
Wageningen B-series	$\simeq 120$	2–7	0.3–1.05	0.6–1.4	250	0.169	No	Four-bladed propeller has non-constant pitch dist
Au-series	34	4–7	0.4–0.758	0.5–1.2	250	0.180	No	
Gawn-series	37	3	0.2–1.1	0.4–2.0	508	0.200	No	
KCA-series	$\simeq 30$	3	0.5–1.25	0.6–2.0	406	0.200	Yes	
Ma-series	32	3 and 5	0.75–1.20	1.0–1.45	250	0.190	Yes	
Newton–Rader series	12	3	0.5–1.0	1.05–2.08	254	0.167	Yes	
KCD-series	24	3–6 (mainly 4)	0.587 Principal 0.44–0.8	0.6–1.6	406	0.200	Yes	Propellers not geosyms
Meridian series	20	6	0.45–1.05	0.4–1.2	305	0.185	Yes	Propellers not geosyms

4. Izrada aplikacije za trodimenzijski prikaz broskog vijka

4.1 NURBS krivulje

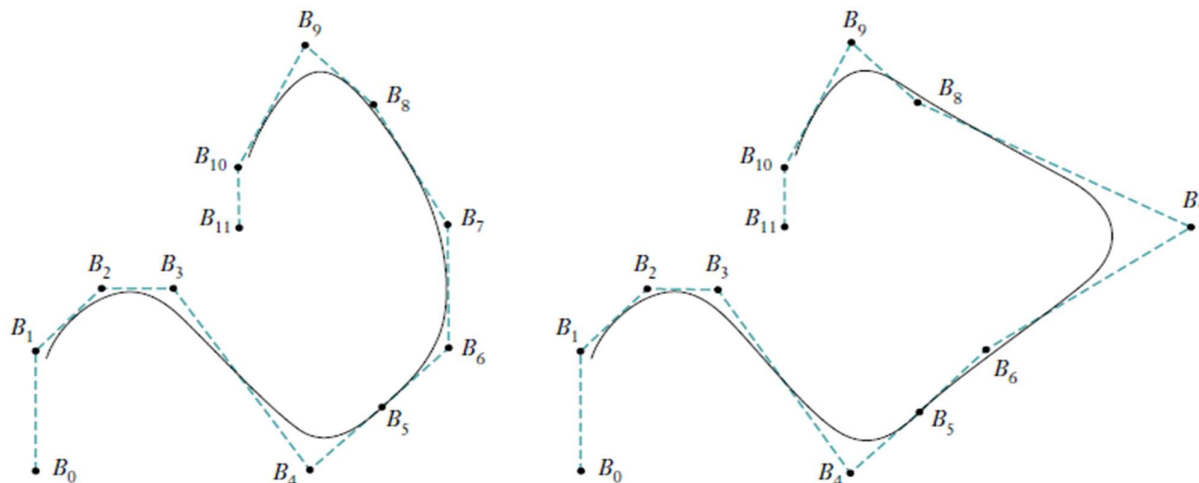
NURBS (Non-uniforma rational B-spline) je matematički model koji se uobičajeno koristi u računalnoj grafici za generiranje raznih krivulja i ploha. Velika fleksibilnost i preciznost za manipuliranje matematički opisanim krivuljama i plohama (modelima) je ono što ga čini poželjnim za korištenje. Često se koristi u raznim CAD, CAM, CAE alatima te također ima veliku primjenu u 3D modeliranju. NURBS-ove plohe su funkcija mapiranja dva parametra na plohu u trodimenzijskom prostoru. Oblik površine je određen s kontrolnim točkama.

Jedna od glavnih prednosti NURBS krivulje je mogućnost lakog manipuliranja s glatkoćom krivulje. NURBS model krivulja nam omogućava definirati krivulju bez oštih rubova i vrpčastih završetaka. Glatka krivulja nema naglih prijelaza, oštih rubova i brze promjene brzine ili smjera. Zamislimo li strelicu koja pokazuje smjer kojim bi hipotetski djelić krivulje putovao duž krivulje, matematički to možemo opisati kao smjer strelice koja odgovara tangenti na krivulju koja je definirana kao derivacija funkcije.

Na Slici 14. se jasno vidi razlika između kontinuirane, glatke i diskontinuirane funkcije koja ima oštar prijelaz (u točki spajanja možemo imati dvije različite tangente).

Sada kada znamo kako NURBS krivulja funkcionira, možemo je s razumijevanjem definirati. Nazovemo li našu funkciju $C(t)$, gdje je t parametar ovisan o vremenu, procjenjivanje ove funkcije za različite t -ove, dobit ćemo parove točaka koje možemo iskoristiti za grafički prikaz krivulje. Kada smo definirali lijevi dio jednakosti, moramo to napraviti i za desni dio jednakosti.

Jedna od ključnih karakteristika NURBS krivulja je ta da je njihov oblik određen pozicijama parova točaka koje se još nazivaju kontrolne točke (eng. *control points*). Kontrolne točke su uobičajeno spojene linijom kako bi se potvrdila njihova povezanost s krivuljom. Niz takvih spojenih točaka se još naziva i kontrolni poligon (eng. *control polygon*).



Slika 14. Kontrolni poligon, i pomak kontrolnih točaka [4]

Na Slici 14. vide se dvije iste krivulje, ali desna krivulja ima točku B_7 malo pomaknutu u desno. Uočimo da se oblik krivulje nije generalno promijenio nego samo lokalno u blizini točke B_7 . To je vrlo poželjno svojstvo koje nam omogućava promjenu oblika krivulje na lokalnoj razini pomakom pojedinih kontrolnih točaka bez utjecaja na cjelokupni izgled krivulje. Svaka kontrolna točka utječe na dio krivulje u blizini točke, i ima mali ili gotovo nikakav utjecaj na dio krivulje koji nije u blizini te točke.

Jedan od načina razmišljanja je da se uzme u obzir koliko utjecaja na kontrolne točke duž putanje ima pomak naše male čestice u svakom vremenskom trenutku. U bilo kojem trenutku t , pozicija čestice će biti težinski prosjek svih kontrolnih točaka, pri čemu točke bliže čestici nose veću „težinu“ od točaka dalje od nje. To možemo intuitivno matematički izraziti kao:

$$C(t) = \frac{\sum_{i=0}^n N_{i,p}(t) \cdot w_i \cdot P_i}{\sum_{i=0}^n N_{i,p}(t) \cdot w_i} \quad (2)$$

Drugim riječima, kako bi znali točnu poziciju putujuće čestice u određenom trenutku t , moramo zbrojiti pozicije svih kontrolnih točaka P_i i varirati iznos doprinosa kontroliranja svake kontrolne točke u vremenu w_i .

Funkcija $N_{i,p}(t)$ označava koliko kontrolna točka P_i jako utječe na krivulju u trenutku t i nazivamo je temeljna funkcija. Vrijednost temeljne funkcije je realan broj.

4.2 NURBS plohe

NURBS ploha se dobije kao matrični umnožak dviju NURBS krivulja i zadana je s dva nezavisna parametra u i v . Neuniformna racionalna B-splajn ploha nastaje poopćenjem B-splajn ploha. NURBS ploha reda (p, q) je u stvari omjer dvaju splajnova, a definirana je izrazom :

$$S(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) \cdot N_{j,q}(v) \cdot w_{i,j} \cdot P_{i,j}}{\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) \cdot N_{j,q}(v) \cdot w_{i,j}} \quad (3)$$

gdje su $N_{i,p}$ i $N_{j,q}$ bazne funkcije B-splajna, $P_{i,j}$ su kontrolne točke, a težina $w_{i,j}$ kontrolnih točaka $P_{i,j}$ je zadnja ordinata homogene točke $P_{i,j}^w$ [5].

4.3 Rhinoceros 5

Rhinoceros je alat koji se koristi za stvaranje, uređivanje, analiziranje, dokumentiranje, izradu animacija, lako manipuliranje s krivuljama, plohami ili tijelima... Nema ograničenje na kompleksnost rada, stupanj složenosti i drugo, nego samo na hardverske mogućnosti računala. I pri najzamršenijim zadacima pokazao je odličnu funkcionalnost i efikasnost.

Jedna od glavnih značajki je izuzetno brz 3D prikaz, neograničene veličine, mogućnost raznoraznih pogleda iz svakakvih kutova. Program ima mogućnost doprogramiranja novih alata (*SDK – software development kit*) i koristi *openNURBS* zapis ploha koji je otvorenog tipa. Mogućnost korištenja raznih programskih jezika kao što su Python, Visual Basic for Applications, Grasshopper čini ga pristupačnim svi korisnicima. Korisničko sučelje je krajnje jednostavno i pregledno te su sve opcije razvrstane po karticama.

Neograničena mogućnost *undo* i *redo* tipke te opcija *multiple undo* neke su od stavki koje pojednostavljuju korištenje Rhinoceros-a. Međutim, pozitivnih značajki zbog kojih smo odabrali korištenje ovog programa ima puno više, a to su npr.: vrlo jednostavno uključivanje/isključivanje kontrolnih točaka te pomicanje istih, zadavanje i promjena *layera* krivulja, automatsko dovršavanje naredbi u *command line*-u, zaključavanje/otključavanje *layera*, pokazivanje/skrivanje *layera* te laka mogućnost izrade raznih plug in-ova.

Rhinoscript je alat za kodiranje (pisanje skripti, skriptiranje) temeljen na Microsoft-ovom Visual Basic for Applications programskom jeziku. Pomoću RhinoScripte vrlo lako možemo zamijeniti ručno crtanje automatskim generiranjem krivulja, ploha ili tijela. Gotovo sve funkcije koje Rhinoceros posjeduje, dostupne su i Rhinoscript-i. U slučaju da ne postoji

naredba u Rhinocerosu, slobodni smo je sami generirati što je vrlo zanimljivo u odnosu na većinu drugih programa za crtanje/modeliranje [6].

4.4 Ulazni podaci za crtanje vijka

U kodu je trenutno moguće crtanje Wageningen B i BB serije vijaka, ali je omogućeno jednostavno dodavanje bilo koje druge serije vijaka. Potrebni su samo: promjer vijka, broj krila vijka, nagib krila vijka, raspodjela uspona vijka po polumjeru i tablični zapis točaka profila na različitim polumjerima.

Za crtanje vijka u trodimenzijskom prikazu, potrebna nam je topologija presjeka krila prema Slici 11. Točke na krivulji presjeka profila računamo prema sljedećim izrazima:

$$Y_{face} = V_1(t_{max} - t_{l.e.}) \quad Y_{back} = (V_1 + V_2)(t_{max} - t_{l.e.}) + t_{l.e.} \quad \text{za } P \leq 0 \quad (4)$$

$$Y_{face} = V_1(t_{max} - t_{l.e.}) \quad Y_{back} = (V_1 + V_2)(t_{max} - t_{l.e.}) + t_{l.e.} \quad \text{za } P \geq 0 \quad (5)$$

Zbog pojednostavljenja koda i izbjegavanja komplicirane geometrije debljine izlaznih krajeva vijka $t_{t.e.}$ i $t_{l.e.}$ su zanemarene. Nadogradnja postojećeg koda zasigurno bi u sebi sadržavala debljine na krajevima profila. U Tablici 4., prikazani su omjeri dimenzija više krilnog vijka po radijusima [3].

Tablica 4. Omjeri dimenzija više krilnog vijka po radijusima [3]

Dimensions of four-, five-, six- and seven-bladed propellers						
r/R	$\frac{c}{D}$	$\frac{Z}{A_E/A_O}$	a/c	b/c	$t/D = A_r - B_r Z$	
					A_r	B_r
0.2	1.662		0.617	0.350	0.0526	0.0040
0.3	1.882		0.613	0.350	0.0464	0.0035
0.4	2.050		0.601	0.351	0.0402	0.0030
0.5	2.152		0.586	0.355	0.0340	0.0025
0.6	2.187		0.561	0.389	0.0278	0.0020
0.7	2.144		0.524	0.443	0.0216	0.0015
0.8	1.970		0.463	0.479	0.0154	0.0010
0.9	1.582		0.351	0.500	0.0092	0.0005
1.0	0.000		0.000	0.000	0.0030	0.0000

Tablica 5. Vrijednosti V_1 koje se koriste za računanje ordinata točaka [3]

Values of V_1 for use in the equations

r/R	P	-1.0	-0.95	-0.9	-0.8	-0.7	-0.6	-0.5	-0.4	-0.2	0
0.7-1.0	0	0	0	0	0	0	0	0	0	0	0
0.6	0	0	0	0	0	0	0	0	0	0	0
0.5	0.0522	0.0420	0.0330	0.0190	0.0100	0.0040	0.0012	0	0	0	0
0.4	0.1467	0.1200	0.0972	0.0630	0.0395	0.0214	0.0116	0.0044	0	0	0
0.3	0.2306	0.2040	0.1790	0.1333	0.0943	0.0623	0.0376	0.0202	0.0033	0	0
0.25	0.2598	0.2372	0.2115	0.1651	0.1246	0.0899	0.0579	0.0350	0.0084	0	0
0.2	0.2826	0.2630	0.2400	0.1967	0.1570	0.1207	0.0880	0.0592	0.0172	0	0
0.15	0.3000	0.2824	0.2650	0.2300	0.1950	0.1610	0.1280	0.0955	0.0365	0	0

r/R	P	+1.0	+0.95	+0.9	+0.85	+0.8	+0.7	+0.6	+0.5	+0.4	+0.2	0
0.7-1.0	0	0	0	0	0	0	0	0	0	0	0	0
0.6	0.0382	0.0169	0.0067	0.0022	0.0006	0	0	0	0	0	0	0
0.5	0.1278	0.0778	0.0500	0.0328	0.0211	0.0085	0.0034	0.0008	0	0	0	0
0.4	0.2181	0.1467	0.1088	0.0833	0.0637	0.0357	0.0189	0.0090	0.0033	0	0	0
0.3	0.2923	0.2186	0.1760	0.1445	0.1191	0.0790	0.0503	0.0300	0.0148	0.0027	0	0
0.25	0.3256	0.2513	0.2068	0.1747	0.1465	0.1008	0.0669	0.0417	0.0224	0.0031	0	0
0.2	0.3560	0.2821	0.2353	0.2000	0.1685	0.1180	0.0804	0.0520	0.0304	0.0049	0	0
0.15	0.3860	0.3150	0.2642	0.2230	0.1870	0.1320	0.0920	0.0615	0.0384	0.0096	0	0

Tablica 6. Vrijednosti V_2 koje se koriste za računanje ordinata točaka [3]

Values of V_2 for use in the equations

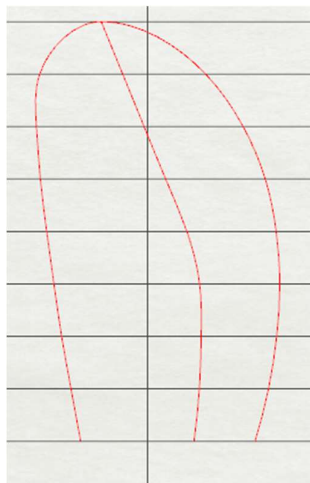
r/R	P	-1.0	-0.95	-0.9	-0.8	-0.7	-0.6	-0.5	-0.4	-0.2	0
0.9-1.0	0	0.0975	0.19	0.36	0.51	0.64	0.75	0.84	0.96	1	1
0.85	0	0.0975	0.19	0.36	0.51	0.64	0.75	0.84	0.96	1	1
0.8	0	0.0975	0.19	0.36	0.51	0.64	0.75	0.84	0.96	1	1
0.7	0	0.0975	0.19	0.36	0.51	0.64	0.75	0.84	0.96	1	1
0.6	0	0.0965	0.1885	0.3585	0.5110	0.6415	0.7530	0.8426	0.9613	1	1
0.5	0	0.0950	0.1865	0.3569	0.5140	0.6439	0.7580	0.8456	0.9639	1	1
0.4	0	0.0905	0.1810	0.3500	0.5040	0.6353	0.7525	0.8415	0.9645	1	1
0.3	0	0.0800	0.1670	0.3360	0.4885	0.6195	0.7335	0.8265	0.9583	1	1
0.25	0	0.0725	0.1567	0.3228	0.4740	0.6050	0.7184	0.8139	0.9519	1	1
0.2	0	0.0640	0.1455	0.3060	0.4535	0.5842	0.6995	0.7984	0.9446	1	1
0.15	0	0.0540	0.1325	0.2870	0.4280	0.5585	0.6770	0.7805	0.9360	1	1

r/R	P	+1.0	+0.95	+0.9	+0.85	+0.8	+0.7	+0.6	+0.5	+0.4	+0.2	0
0.9-1.0	0	0.0975	0.1900	0.2775	0.3600	0.51	0.6400	0.75	0.8400	0.9600	1	1
0.85	0	0.1000	0.1950	0.2830	0.3660	0.5160	0.6455	0.7550	0.8450	0.9615	1	1
0.8	0	0.1050	0.2028	0.2925	0.3765	0.5265	0.6545	0.7635	0.8520	0.9635	1	1
0.7	0	0.1240	0.2337	0.3300	0.4140	0.5615	0.6840	0.7850	0.8660	0.9675	1	1
0.6	0	0.1485	0.2720	0.3775	0.4620	0.6060	0.7200	0.8090	0.8790	0.9690	1	1
0.5	0	0.1750	0.3056	0.4135	0.5039	0.6430	0.7478	0.8275	0.8880	0.9710	1	1
0.4	0	0.1935	0.3235	0.4335	0.5220	0.6590	0.7593	0.8345	0.8933	0.9725	1	1
0.3	0	0.1890	0.3197	0.4265	0.5130	0.6505	0.7520	0.8315	0.8020	0.9750	1	1
0.25	0	0.1758	0.3042	0.4108	0.4982	0.6359	0.7415	0.8259	0.8899	0.9751	1	1
0.2	0	0.1560	0.2840	0.3905	0.4777	0.6190	0.7277	0.8170	0.8875	0.9750	1	1
0.15	0	0.1300	0.2600	0.3665	0.4520	0.5995	0.7105	0.8055	0.8825	0.9760	1	1

Tablica 5., i Tablica 6., prikazuje koeficijente V_1 i V_2 , te uz pomoć tih koeficijenata računaju se ordinate profila, tj. računa se ordinatu lica i naličja za određenu poziciju P .

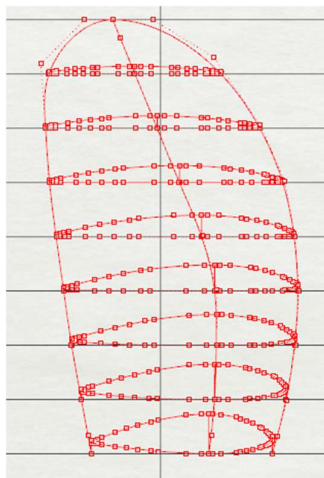
P iz Tablica 5. i 6. je bezdimenzijski koeficijent te se pomoću njega mogu izračunati ordinate točaka na različitim mjestima duž presjeka. Mjeri se od točke maksimalne debljine do ulaznog brida ($P = 1$) te od ravnine vijka do izlaznog brida ($P = -1$).

Nakon što je definirana raspodjela ordinata lica i naličja i definirani ulazni parametri vijka (D , Z , P/D , A_E/A_0) može se izračunati duljinu tetive, udaljenost od izlaznog brida do mjesta najveće debljine i na drugu stranu do ulaznog brida. Nakon toga računaju se koordinate krajnjih točaka po svim radijusima kao i najveća debljina profila na određenom radijusu. Na Slici 15., vidi se prikaz razvijene površine krila vijka s ucrtanim pozicijama maksimalnih debljinama po radijusima (0,2-1R).



Slika 15. Vanjske krivulje krila

Ulazne i izlazne krivulje napravljene su pomoću rubnih točaka profila, po 8 točaka za svaku krivulju te jedna zajednička točka (vrh krila). Zatim smo po izrazima za određivanje ordinate izračunali točke svakog profila po radijusima i spojili ih NURBS krivuljom, Slika 16.



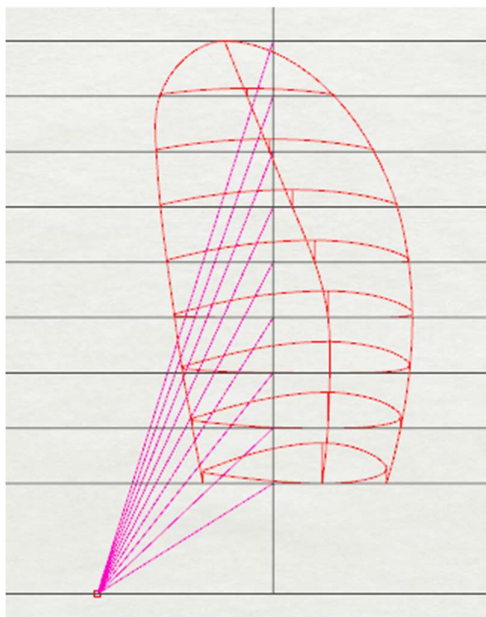
Slika 16. Izračunata geometrija profila

Kut rotacije svakog profila $\beta(i)$, računa se prema sljedećoj jednadžbi:

$$\beta(i) = \arctg^{-1}\left(\frac{P}{2\pi R(i)}\right) \quad (6)$$

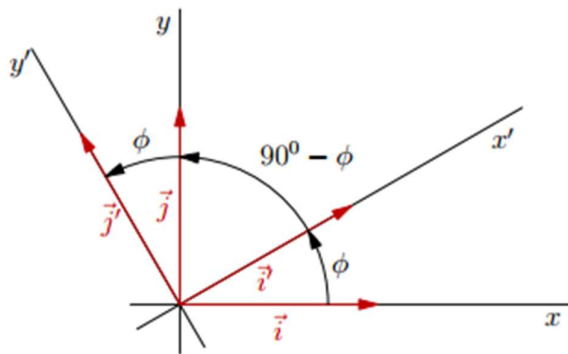
Točka rotacije profila je određena dvjema veličinama, prva je apscisa koja iznosi $\frac{P}{2\pi}$, a druga veličina je dužina koja prolazi iz ravnine vijka na nekoj visini $R(i)$ kroz točku rotacije.

Kut između te dvije veličine je kut rotacije $\beta(i)$, gdje je $\frac{P}{2\pi}$ apscisa točke rotacije, a $R(i)$ njena ordinata. To se vidi na Slici 17., na kojoj je označena točka rotacije.



Slika 17. Točka rotacije

Nakon što se izračunaju sve koordinate točaka projekcije vijka u 2D, moraju se prebaciti u 3D i napraviti još jednu transformaciju kako bi se profil vijka savio te lijepo zalijepio na glavinu. To se napravilo s klasičnim jednadžbama za transformaciju koordinatnog sustava.



Slika 18. Rotacija koordinatnog sustava u ravnini [8]

Indeksi koordinata točaka nakon svake transformacije su:

- 0 – za crtanje profila,
- 1 – nakon rotacije,
- 2 – nagib,
- 3 – prebacivanje iz 2D u 3D (zadavanje treće dimenzije),
- 4 – savinuti profili.

Kako bi vijak u 3D bio točno i precizno određen, potrebno je egzaktno izračunati koordinate svih njegovih točaka profila po svim presjecima.

Određivanje koordinata zarotiranih točaka profila računaju se prema jednadžbama:

$$x_1 = x_0 \cos \beta - y_0 \sin \beta \quad (7)$$

$$y_1 = x_0 \sin \beta + y_0 \cos \beta \quad (8)$$

Na taj način napravljena je rotacija točaka, koje se i dalje nalaze u ravnini ali su zakrenute. Slijedeća transformacija je dodavanje nagiba krilu, a nagib je dodan na slijedeći način:

$$y_2 = y_1 + \tan(\alpha) \cdot R(i) \quad (9)$$

U jednadžbi 9 vidljiv je član $\tan(\alpha) \cdot R(i)$, to je pomak zbog nagiba vijka karakterističan za Wageningen B seriju dok je kut $\alpha = 15^\circ$.

Nakon rotacije i nagiba krila, slijedi prebacivanje svih točaka iz ravninskog crteža u prostorni crtež, a to je napravljeno na način da se svakoj točki doda njezina z-komponenta, Slika 19., kako slijedi:

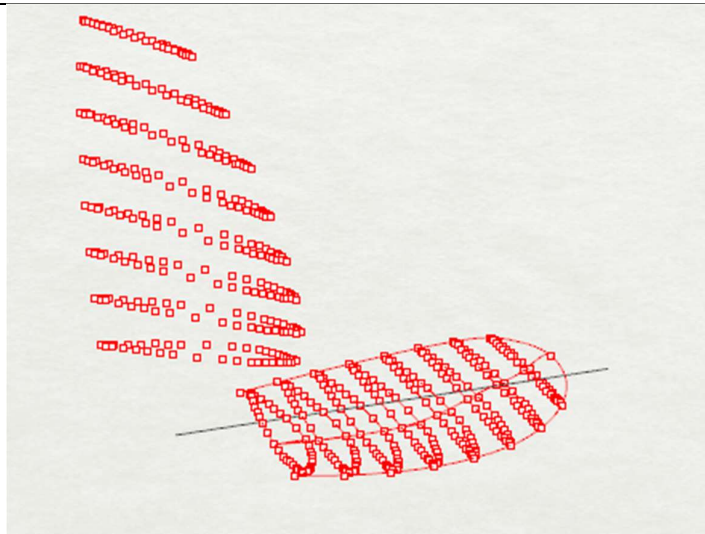
$$z_3 = R(i) \quad (10)$$

Nakon toga potrebno je izračunati kut savijanja profila vijka, a to je napravljeno uz pomoć jednostavnih trigonometrijskih funkcija i pravila o sličnosti trokuta. Sada se s dobivenim kutom savijanja, tj. njegovim sinusnim i kosinusnim komponentama, izračunaju koordinate savinutih točaka, Slika 20., kako slijedi:

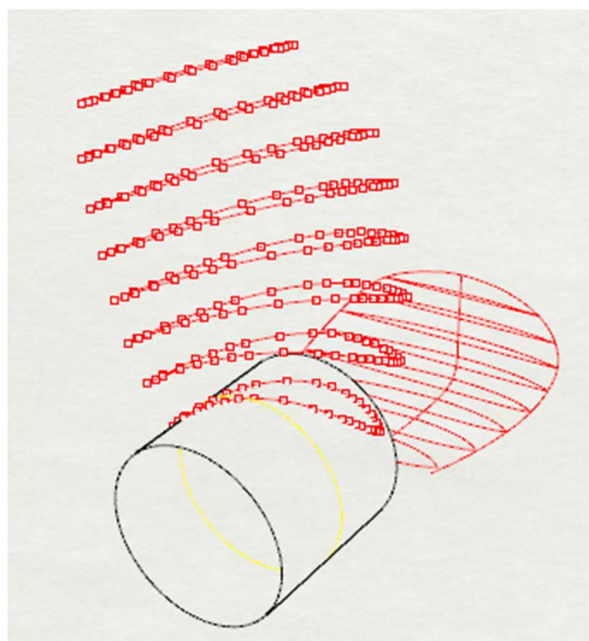
$$\theta = \frac{x_3}{R(i)} \quad (11)$$

$$x_4 = \sin \theta \cdot R(i) \quad (12)$$

$$z_4 = \cos \theta \cdot R(i) \quad (13)$$

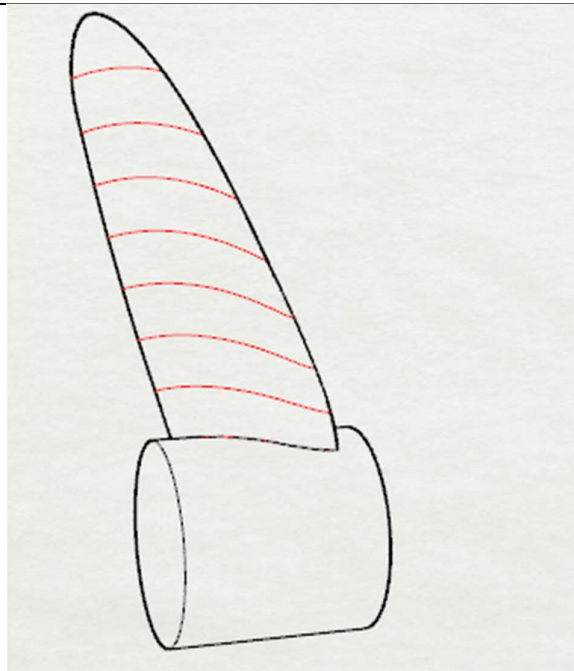


Slika 19. Zakretanje točaka profila i prebacivanje iz 2D u 3D



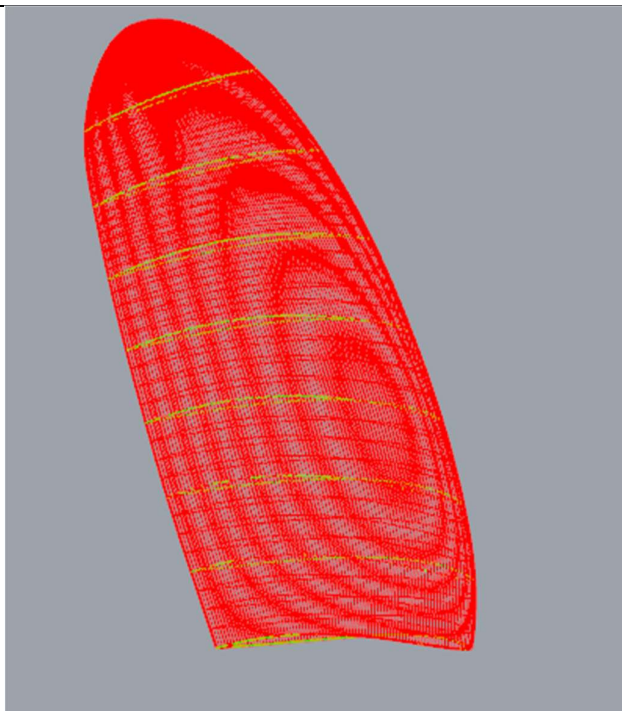
Slika 20. Spajanje zakrenutih profila i lijepljenje na cilindar

Zatim se crtaju ulazne i izlazne krivulje krila. Spajanje točaka profila izvršeno je pomoću NURBS krivulje. U ovom radu se radilo s polinomima 3. stupnja. Na Slici 21., vidi se kako izgleda nagnuto krilo vijka koje je zalijepljeno na glavinu.



Slika 21. Spojene točke ulaznog i izlaznog brida nagnutog profila koji je zalijepljen na cilindar
Kako bi vrh krila imao zadovoljavajuću glatkoću trebalo je izračunati krajnju točku obje krivulje (ulazni i izlazni brid) tako da se koordinate tangente poklapaju po iznosu, a razlikuju po predznaku.

Nakon nacrtanih krivulja (4 krivulje po presjeku), krivulje lica i naličja su se spojile (cijelo lice i cijelo naličje) kako bi se kasnije iz tih spojenih krivulja mogla generirati ploha. Nakon generiranja, donji kraj plohe produžen je prema osi vrtnje vijka za slučaj da je promjer glavine manji od $0,2D$. Ploha se crta kao mreža krivulja (gustoća se može odabrati) koje se međusobno presijecaju. Ovako se dobije glatka i točna ploha, koja ne odstupa od nijedne ulazne krivulje, umjerenog broja kontrolnih točaka kao što je prikazano na Slici. 22. i koju možemo lako upotrijebiti u radu s drugim aplikacijama za proračun karakteristika vijka ili za izradu vijka.

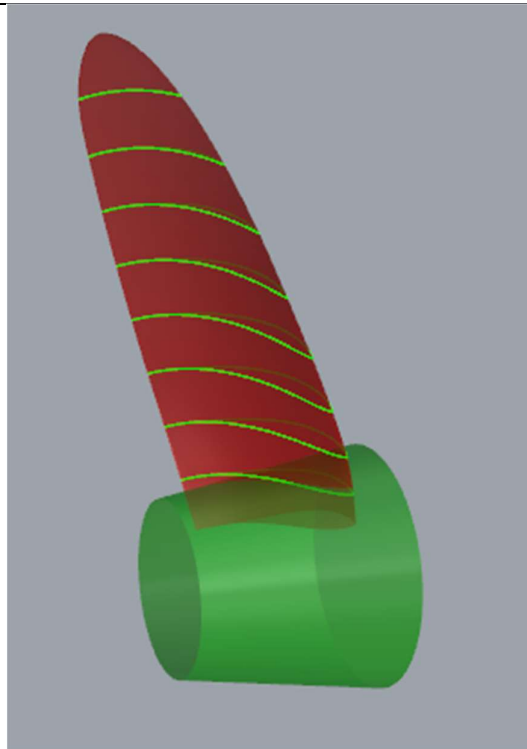


Slika 22. Ploha generirana kao mreža krivulja

Nakon produljenja plohe, potrebno je generirati glavinu vijka. Za generiranje glavine mora se zadati ravnina crtanja, a glavina je zapravo krnji stožac. Ravnina crtanja glavine definira se zajedno s ulaznim podacima kao i druge dimenzije glavine (duljina prema licu i naličju te promjeri na strani lica i naličja). Zatim se ploha krila spaja s glavinom na način da višak krila „odrežemo“. Na Slici 23., vidi se prodor krila u glavinu prije rezanja viška krila.

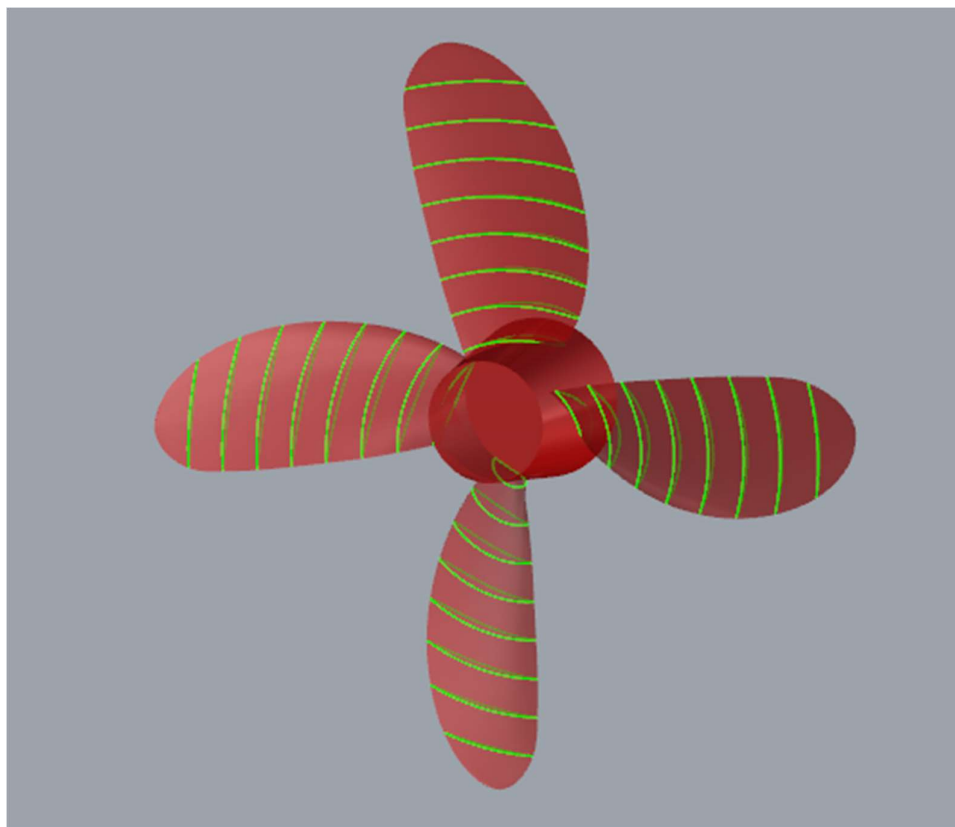
Nakon spajanja krila i glavine sve je zakrenuto za $Z-1$ puta kako bi se dobio vijak sa zadanim brojem krila. S obzirom da vijak može imati proizvoljnu glavinu (promjer i duljina) i da promjer glavine ne mora biti jednak $0,2D$ potrebno je ograničiti minimalne dimenzije glavine kako bi se kasnije uspjelo točno spojiti krilo s glavinom (u slučaju da je duljina glavine manja od duljine krila, krilo se ne bi zalijepilo na glavinu).

Na kraju svega označene su sve krivulje presjeka profila (8 presjeka), i postavljeni su u drugu boju kako bi presjeci bili vidljivi u finalnoj projekciji.



Slika 23. Nacrtna produljena ploha koja ulazi u glavinu

Na Slici 24., prikazan je vijak u tri dimenzije.



Slika 24. Prikaz vijka u 3D

5. Zaključak

U ovom završnom radu analiziran je proces izrade dvodimenzijskog nacrtu i trodimenzijske projekcije broskog vijka. Programski kod je izrađen za vijak Wageningen B serije, koja je najkorištenija i najispitivanija serija vijaka za primjenu na deplasmanskim brodovima. Geometrija profila krila predstavlja ulazne podatke u programski kod i postupno se geometrijskim transformacijama dobiju 3D krivulje za generiranje ploha krila. Korišten je alat Rhinoceros 5, koji koristi OpenNURBS matematičku osnovu i uz mogućnost pisanja skripte, RhinoScript napravljen je kod za automatsko crtanje vijka s unaprijed određenom rutinom računanja koordinata njegovih točaka i spajanja istih. Funkcionalnost njegovih naredbi je iskorištena za lakše manipuliranje NURBS krivuljama uz zadovoljavajuće rezultate. Dobivene plohe vijka su dovoljno točne i jednostavne te ih je moguće lako prenijeti u druge oblike digitalnih zapisa. U programski kod je moguće jednostavno dodati i neku drugu seriju vijaka uz uvjete navedene u radu. Upotreba ovog programskog koda olakšat će studentima razumijevanje nacrtu vijka i prikaza u tri dimenzije.

Literatura

- [1] Sambolek, M., Propulzija broda, Brodarski institut, Zagreb, 2012.
- [2] Radan, D., Uvod u hidrodinamiku broda, Sveučilište u Dubrovniku, Dubrovnik, 2004.
- [3] Carlton, J., Marine Propellers and Propulsion, Elsevier Ltd., 2012.
- [4] <https://www.scribd.com/document/154183304/NURB-Curves-a-Guide-for-the-Uninitiated>
- [5] <https://www.fsb.unizg.hr/geometrija.broda/>
- [6] <https://www.rhino3d.com/>
- [7] Ević, D., Hidrodinamički i tehnološki aspekti kape broskog vijka, Završni rad, FSB, Sveučilište u Zagrebu, Zagreb, 2013.
- [8] http://www.phy.pmf.unizg.hr/~mbacani/Fizika/OF1/04_analiticka.pdf
- [9] <http://www.scam-marine.hr/upload/Propeller%20Terminology%201%20hrvatski.pdf>
- [10] <https://wiki.mcneel.com/developer/rhinoscript>

PRILOZI

- I. Skripta za crtanje vijka
- II. Upute za korištenje skripte za prikaz broskog vijka

PRILOG 1

Skripta za crtanje vijka

```

0 Option Explicit
1 'Script version Završni rad, 20.2.2017.
2 Call Crtanje_vijka()
3
4 Sub Crtanje_vijka()
5
6     Rhino.UnitSystem 2
7     Rhino.UnitAbsoluteTolerance 0.001
8     Rhino.MaximizeRestoreView "Perspective"
9
10    'definiranje glavnih parametara
11    Dim AEA0,D,PdivD,Z,Seriya,vrh,projekcija
12    Dim a(8),b(8),c(8),t(8)
13    Dim R(8)
14    Dim hte(8),hle(8),pte(7),ple(7)
15    Dim tte(8),tle(8)
16    Dim kotA1(8),kotB1(8),kotA2(8),kotB2(8), kotD1(8), kotD2(8)
17    Dim razmak,zapomak
18    Dim blnredraw,Ra,Rf,La,Lf
19
20
21    'blnRedraw = Rhino.EnableRedraw(False)
22
23    'pozivanje funkcija za crtanje vijka
24    Call ulaz(AEA0, D, PdivD, Z, Serija, R, Ra, Rf, La, Lf, Projekcija)
25    razmak = 0.8 * R(8)
26    Call abc(AEA0, D, Z, a, b, c, t, R, vrh, Serija, projekcija)
27    Call koord(R, razmak, projekcija)
28    Call profili(a, b, c, t, R, hte, hle, tte, tle, pte, ple, D, PdivD, vrh,
29    Z, Ra, Rf, La, Lf, projekcija)
30    Call kote(a, c, R, hte, hle, tte, hle, pte, ple, vrh, PdivD, kotA1, kotB1
31    , kotA2, kotB2, kotD1, kotD2, projekcija)
32    Call lukovi(R, kotA1, kotB1, kotA2, kotB2, kotD1, kotD2, projekcija)
33
34
35    'pomak 2d projekcija
36    If Projekcija = 1 Then
37
38        zapomak = Rhino.ObjectsByLayer("NVnacrt")
39        Rhino.MoveObjects zapomak, array(0, 0, 0), array(-razmak, 0, 0)
40
41        zapomak = Rhino.ObjectsByLayer("NVbokocrt")
42        Rhino.MoveObjects zapomak, array(0, 0, 0), array(-2 * razmak, 0, 0)
43        Rhino.CurrentLayer "NVpomocne"
44        zapomak = Rhino.ObjectsByType(1)
45        Rhino.ObjectLayer zapomak, "NVkote"
46
47    End If
48
49    Rhino.ZoomExtents
50
51    'blnRedraw = Rhino.EnableRedraw(True)
52
53 End Sub
54
55 ' unos ulaznih podataka vijka
56 Sub ulaz(AEA0, D, PdivD, Z, Serija, R, Ra, Rf, La, Lf, Projekcija)
57

```

```

53     Dim dane,i
54     dane = "n"
55     Do While dane <> "d"
56         Rhino.Print "Unos ulaznih podataka"
57         D = Rhino.GetReal("Promjer vijka u mm")
58         AEA0 = Rhino.GetReal("AE/A0 [0.3-1.05]")
59         PdivD = Rhino.GetReal("P/D [0.6-1.4]")
60         Z = Rhino.GetReal("Broj krila vijka [2-7]")
61         Ra = Rhino.GetReal("Radijus baze glavine na strani lica, ne manje od
10% D, u mm")
62         Rf = Rhino.GetReal("Radijus nalicja glavine u mm")
63         La = Rhino.getReal("Izlazna duljina glavine , ne manje od 10% D ,u mm
")
64         La = -La
65         Lf = rhino.getREal("Ulazna duljina glavine u mm")
66         Serija = Rhino.GetReal("Serija vijka (1 = WB serija, 2 = WBB serija
)")
67         Projekcija = Rhino.GetReal("Zelite li 2d (1) ili 3d (2) projekciju ?
")
68         Rhino.Print "D = " & D & " mm, AE/A0 = " & AEA0 & ", P/D = " & PdivD
& ", Z = " & Z & ", Serija = WB "
69         Rhino.Print "La = " & -La & "mm, Lf = " & Lf & "mm,Ra = " & Ra & "mm,
Rf = " & Rf & "mm"
70         dane = Rhino.GetString("U redu? (d/n)", "d")
71
72     Loop
73     ' racunanje polumjera po visini
74     For i = 0 To 8
75
76         R(i) = D / 2 * (i / 10 + 0.2)
77
78     Next
79
80 End Sub
81
82 Sub abc(AEA0, D, Z, a, b, c, t, R, vrh, serija, projekcija)
83
84     'izračun gabarita profila vijka Wageningen B i Wageningen BB serije
85
86     Dim i,ulazni(8),izlazni(8),sirine(8)
87     Dim chord,adivc,bdivc,Ar,Br
88     Rhino.AddLayer "NVrazvijena", RGB(255, 0, 0)
89     Rhino.CurrentLayer "NVrazvijena"
90
91     Select Case serija
92         Case 1
93             If Z = 2 Or Z = 3 Then
94
95                 chord = Array(1.633, 1.832, 2.000, 2.120, 2.186, 2.168, 2.127
, 1.657, 0.000)
96                 adivc = Array(0.616, 0.611, 0.599, 0.583, 0.558, 0.526, 0.481
, 0.400, 0.000)
97                 bdivc = Array(0.350, 0.350, 0.350, 0.355, 0.389, 0.442, 0.478
, 0.500, 0.000)
98
99             Else
100
101

```

```

101         chord = Array(1.662, 1.882, 2.050, 2.152, 2.187, 2.144, 1.970
, 1.582, 0.000)
102         adivc = Array(0.617, 0.613, 0.601, 0.586, 0.561, 0.524, 0.463
, 0.351, 0.000)
103         bdivc = Array(0.350, 0.350, 0.351, 0.355, 0.389, 0.443, 0.479
, 0.500, 0.000)
104
105         End If
106
107     Case 2
108         chord = Array(1.600, 1.832, 2.023, 2.163, 2.243, 2.247, 2.132, 1.798
, 0.000)
109         adivc = Array(0.581, 0.584, 0.580, 0.570, 0.552, 0.524, 0.480, 0.402
, 0.000)
110         bdivc = Array(0.350, 0.350, 0.351, 0.355, 0.389, 0.443, 0.486, 0.500
, 0.000)
111
112     Case Else
113         chord = Array(1.662, 1.882, 2.050, 2.152, 2.187, 2.144, 1.970, 1.582
, 0.000)
114         adivc = Array(0.617, 0.613, 0.601, 0.586, 0.561, 0.524, 0.463, 0.351
, 0.000)
115         bdivc = Array(0.350, 0.350, 0.351, 0.355, 0.389, 0.443, 0.479, 0.500
, 0.000)
116
117     End Select
118
119     Ar = Array(0.0526, 0.0464, 0.0402, 0.0340, 0.0278, 0.0216, 0.0154, 0.0092
, 0.0030)
120     Br = Array(0.0040, 0.0035, 0.0030, 0.0025, 0.0020, 0.0015, 0.0010, 0.0005
, 0.0000)
121     'tablice iz J.Carlton: Marine Propellers
122     For i=0 To 8
123
124         c(i) = (D * AEA0 / Z) * chord(i)
125         a(i) = adivc(i) * c(i)
126         b(i) = bdivc(i) * c(i)
127         t(i) = (Ar(i) - Z * Br(i)) * D
128
129     Next
130     'racunanje glavnih dimenzija profila
131     For i=0 To 7
132
133         ulazni(i) = Array(a(i), R(i), 0)
134         izlazni(i) = Array(a(i) - c(i), R(i), 0)
135         sirine(i) = Array(a(i) - b(i), R(i), 0)
136
137     Next
138     'spremanje rubnih tocaka profila vijka
139     vrh = 2 * (a(7) - b(7)) - a(6) + b(6)
140     ulazni(8) = Array(vrh, R(8), 0)
141     izlazni(8) = ulazni(8)
142     sirine(8) = ulazni(8)
143     'crtanje krivulja ulaznog i izlaznog brida te sirine profila
144     If Projekcija = 1 Then
145
146         Rhino.AddInterpCurve ulazni, 3, 1,, Array(-1, 0, 0)
147

```

```

147         Rhino.AddInterpCurve izlazni, 3, 1,, Array(1, 0, 0)
148         Rhino.AddInterpCurve sirine, 3, 1
149
150     End If
151
152 End Sub
153
154 Sub koord(R, razmak, projekcija) ' crtanje koordinatnog sustava i pomoćnih linija
155
156     If Projekcija = 2 Then
157         Exit Sub
158     End If
159
160     Dim i,j,linstart, linend
161
162     Rhino.AddLayer "NVpomocne", RGB(64, 64, 64)
163     Rhino.CurrentLayer "NVpomocne"
164
165     linstart = Array(0, 0, 0)
166     linend = Array(0, R(8) * 1.1, 0)
167     Rhino.Addline linstart, linend
168     linstart = Array(-1.5 * R(8) * 2, 0, 0)
169     linend = Array(R(8), 0, 0)
170     Rhino.Addline linstart, linend
171
172     For i=0 To 8
173
174         linstart = Array(-1.5 * R(8) * 2, R(i), 0)
175         linend = Array(R(8), R(i), 0)
176         Rhino.Addline linstart, linend
177
178     Next
179
180     linstart = Array(-razmak, 0, 0)
181     linend = Array(-razmak, R(8) * 1.1, 0)
182     Rhino.Addline linstart, linend
183     linstart = Array(-2 * razmak, 0, 0)
184     linend = Array(-2 * razmak, R(8) * 1.1, 0)
185     Rhino.Addline linstart, linend
186
187 End Sub
188
189 Sub profili(a, b, c, t, R, hte, hle, tte, tle, pte, ple, D, PdivD, vrh, Z, Ra
, Rf, La, Lf, projekcija)
190     'crtanje profila WB i WBB serije i razvijene površine krila
191     Dim beta(8), P,apscisatr,fi1(9),fi2(10),fi3(9),fi4(10),fiv,kut,alfa 'kut
rotacije,kut savijanja,uspon,apscisarotacije
192     Dim i,j,linstart, linend,k,pb,pa 'brojila i polja distribucije
193     Dim v1a(8),v1b(8),v2a(8),v2b(8) 'pomocne varijable za racunanje koordinata
194
195     Dim fte(9),fle(10),bte(9),ble(10) 'polje za koordinate profila "f"-face,
"b"-back, "t"-trailing "l"-leading, "e"-edge
196     Dim crvle,crvfle,crvble,crvte,crvfte,crvbte,crvmax ' varijable za crtanje
krivulja u 2d
197     Dim distri, dane, zaobljenje , razmak 'pozivi funkcija, da/ne pitalica

```

```

197     Dim x(8),y(8),zz(8),sve(8),x1(8),y1(8),z1(8),svel(8),rub,x2,y2,z2,vrhx,vrhy
, vrhz 'koordinate tocaka
198     Dim glavina,rot(7),propeler,ravnina 'varijable za crtanje glavine
199     Dim duljinap,temp,temp2 'pomocne varijable
200     Dim krivuljeuk 'varijable za crtanje svih krila
201     Dim ploha,ploha1,plohe,spojenifte(7),spojenibte(7),spojenolice,spojenonalicje
,krivulja(7),krivuljal(7),spline,spline1 'plohe i krivulje
202
203
204     duljinap = 10 'duljina pravca na vrhu
205     razmak = 0
206     P = PdivD * D 'uspon
207     apscisatr = P / ( 2 * Rhino.PI ) 'apscisa rotacije
208     sve(0) = Array(0, 0, 0)
209     beta(8) = Rhino.ATan2(apscisatr, R(8))
210     alfa = -15 * rhino.PI / 180 'kut nagiba krila
211
212     Rhino.CurrentLayer "NVrazvijena"
213     'distribucija apscisa
214     pa = Array(-1.00, -0.95, -0.90, -0.80, -0.70, -0.60, -0.50, -0.40, -0.20
, 0.00)
215     pb = Array(+1.00, +0.95, +0.90, +0.85, +0.80, +0.70, +0.60, +0.50, +0.40
, +0.20, 0.00)
216
217     'parametri distribucije ordinata lica i naličja
218     vla(8) = Array(0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000
, 0.0000, 0)
219     vla(7) = Array(0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000
, 0.0000, 0)
220     vla(6) = Array(0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000
, 0.0000, 0)
221     vla(5) = Array(0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000
, 0.0000, 0)
222     vla(4) = Array(0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000
, 0.0000, 0)
223     vla(3) = Array(0.0522, 0.0420, 0.0330, 0.0190, 0.0100, 0.0040, 0.0012, 0.0000
, 0.0000, 0)
224     vla(2) = Array(0.1467, 0.1200, 0.0972, 0.0630, 0.0395, 0.0214, 0.0116, 0.0044
, 0.0000, 0)
225     vla(1) = Array(0.2306, 0.2040, 0.1790, 0.1333, 0.0943, 0.0623, 0.0376, 0.0202
, 0.0033, 0)
226     vla(0) = Array(0.2826, 0.2630, 0.2400, 0.1967, 0.1570, 0.1207, 0.0880, 0.0592
, 0.0172, 0)
227
228     vlb(8) = Array(0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000
, 0.0000, 0)
229     vlb(7) = Array(0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000
, 0.0000, 0)
230     vlb(6) = Array(0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000
, 0.0000, 0)
231     vlb(5) = Array(0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000
, 0.0000, 0)
232     vlb(4) = Array(0.0382, 0.0169, 0.0067, 0.0022, 0.0006, 0.0000, 0.0000, 0.0000
, 0.0000, 0)
233     vlb(3) = Array(0.1278, 0.0778, 0.0500, 0.0328, 0.0211, 0.0085, 0.0034, 0.0008
, 0.0000, 0)
234

```



```

234     v1b(2) = Array(0.2181, 0.1467, 0.1088, 0.0833, 0.0637, 0.0357, 0.0189, 0.0090
, 0.0033, 0.0000, 0)
235     v1b(1) = Array(0.2923, 0.2186, 0.1760, 0.1445, 0.1191, 0.0790, 0.0503, 0.0300
, 0.0148, 0.0027, 0)
236     v1b(0) = Array(0.3560, 0.2821, 0.2353, 0.2000, 0.1685, 0.1180, 0.0804, 0.0520
, 0.0304, 0.0049, 0)
237
238     v2a(8) = Array(0.0000, 0.0975, 0.1900, 0.3600, 0.5100, 0.6400, 0.7500, 0.8400
, 0.9600, 1)
239     v2a(7) = Array(0.0000, 0.0975, 0.1900, 0.3600, 0.5100, 0.6400, 0.7500, 0.8400
, 0.9600, 1)
240     v2a(6) = Array(0.0000, 0.0975, 0.1900, 0.3600, 0.5100, 0.6400, 0.7500, 0.8400
, 0.9600, 1)
241     v2a(5) = Array(0.0000, 0.0975, 0.1900, 0.3600, 0.5100, 0.6400, 0.7500, 0.8400
, 0.9600, 1)
242     v2a(4) = Array(0.0000, 0.0965, 0.1885, 0.3585, 0.5110, 0.6415, 0.7530, 0.8426
, 0.9613, 1)
243     v2a(3) = Array(0.0000, 0.0950, 0.1865, 0.3569, 0.5140, 0.6439, 0.7580, 0.8456
, 0.9639, 1)
244     v2a(2) = Array(0.0000, 0.0905, 0.1810, 0.3500, 0.5040, 0.6353, 0.7525, 0.8415
, 0.9645, 1)
245     v2a(1) = Array(0.0000, 0.0800, 0.1670, 0.3360, 0.4885, 0.6195, 0.7335, 0.8265
, 0.9583, 1)
246     v2a(0) = Array(0.0000, 0.0640, 0.1455, 0.3060, 0.4535, 0.5842, 0.6995, 0.7984
, 0.9446, 1)
247
248     v2b(8) = Array(0.0000, 0.0975, 0.1900, 0.2775, 0.3600, 0.5100, 0.6400, 0.7500
, 0.8400, 0.9600, 1)
249     v2b(7) = Array(0.0000, 0.0975, 0.1900, 0.2775, 0.3600, 0.5100, 0.6400, 0.7500
, 0.8400, 0.9600, 1)
250     v2b(6) = Array(0.0000, 0.1050, 0.2028, 0.2925, 0.3765, 0.5265, 0.6545, 0.7635
, 0.8520, 0.9635, 1)
251     v2b(5) = Array(0.0000, 0.1240, 0.2337, 0.3300, 0.4140, 0.5615, 0.6840, 0.7850
, 0.8660, 0.9675, 1)
252     v2b(4) = Array(0.0000, 0.1485, 0.2720, 0.3775, 0.4620, 0.6060, 0.7200, 0.8090
, 0.8790, 0.9690, 1)
253     v2b(3) = Array(0.0000, 0.1750, 0.3056, 0.4135, 0.5039, 0.6430, 0.7478, 0.8275
, 0.8880, 0.9710, 1)
254     v2b(2) = Array(0.0000, 0.1935, 0.3235, 0.4335, 0.5220, 0.6590, 0.7593, 0.8345
, 0.8933, 0.9725, 1)
255     v2b(1) = Array(0.0000, 0.1890, 0.3197, 0.4265, 0.5130, 0.6505, 0.7520, 0.8315
, 0.8920, 0.9750, 1)
256     v2b(0) = Array(0.0000, 0.1560, 0.2840, 0.3905, 0.4777, 0.6190, 0.7277, 0.8170
, 0.8875, 0.9750, 1)
257
258     Call LeTe(t, tte, tle, distri, zaobljenje, projekcija)
259
260     For i=0 To 7
261
262         beta(i) = Rhino.ATan2(apscisatr, R(i)) ' kut uspona
263
264         For j=0 To 10
265
266             If j < 10 Then
267                 fte(j) = Array((c(i) - b(i)) * pa(j) + a(i) - b(i), R(i) + (
t(i) - tte(i)) * vla(i)(j), 0) ' racunanje koordinata tocaka na licu izlaznog
brida
268

```

```

268
269         bte(j) = Array((c(i) - b(i)) * pa(j) + a(i) - b(i), R(i) + (
racunanje koordinata
tocaka na naličju izlaznog brida
270
271         End If
272         fle(j) = Array((b(i)) * pb(j) + a(i) - b(i), R(i) + (t(i) - tle(
i)) * vlb(i)(j), 0) ' racunanje koordinata tocaka na licu ulaznog brida
273
274         ble(j) = Array((b(i)) * pb(j) + a(i) - b(i), R(i) + (t(i) - tle(
i)) * (vlb(i)(j) + v2b(i)(j)) + tle(i), 0) ' racunanje koordinata tocaka na
naličju ulaznog brida
275
276     Next
277     hte(i) = (t(i) - tte(i)) * v1a(i)(0) + tte(i)
278     hle(i) = (t(i) - tle(i)) * v1b(i)(0) + tle(i)
279
280     If Projekcija = 1 Then
281
282         crvmax = Rhino.Addline(fle(10), ble(10)) ' maksimalna debljina profila
283
284         Rhino.Addline Array(fle(0)(0), R(i), fle(0)(2)), ble(0)
285         crvle = Rhino.Addline(fle(0), ble(0)) ' ulazna debljina profila
286         Rhino.Addline Array(fte(0)(0), R(i), fte(0)(2)), bte(0)
287         crvte = Rhino.Addline(fte(0), bte(0)) ' izlazna debljina profila
288
289         crvfte = Rhino.AddInterpCurve(fte, 3, 0,, Array(1, 0, 0)) 'lice
izlazni
290         crvfle = Rhino.AddInterpCurve(fle, 3, 0,, Array(-1, 0, 0)) 'lice
ulazni
291         crvbte = Rhino.AddInterpCurve(bte, 3, 0,, Array(1, 0, 0)) 'naličje
izlazni
292         crvble = Rhino.AddInterpCurve(ble, 3, 0,, Array(-1, 0, 0)) 'naličje
ulazni
293
294         Rhino.AddPoints fte 'crtanje tocaka fte
295         Rhino.AddPoints bte 'crtanje tocaka bte
296         Rhino.AddPoints fle 'crtanje tocaka fle
297         Rhino.AddPoints ble 'crtanje tocaka ble
298
299     End If
300
301     For j=0 To 10 ' petlja za racunanje koordinata tocaka profila vijka
nakon transformacije iz 2D u 3D
302         If j < 10 Then
303
304             fte(j)(0) = fte(j)(0) - 0
305             fte(j)(1) = fte(j)(1) - R(i)
306             temp2 = fte(j)(0)
307             fte(j)(0) = fte(j)(0) * cos(beta(i)) - fte(j)(1) * sin(beta(
i)) ' PRVA TRANSFORMACIJA
308             fte(j)(1) = temp2 * sin(beta(i)) + fte(j)(1) * cos(beta(i))
+ tan(alfa) * R(i)
309             fte(j)(2) = R(i)
310             fil(j) = fte(j)(0) / R(i) ' DRUGA TRANSFORMACIJA
311
312             fte(j)(0) = sin(fil(j)) * R(i)

```

```

310         fte(j)(2) = cos(fil(j)) * R(i)
311         x(i) = fte(0)(0) ' x koordinata fte-a za crtanje vanjske krivulje
312
313         y(i) = fte(0)(1) ' y koordinata fte-a za crtanje vanjske krivulje
314
315         zz(i) = fte(0)(2) ' z koordinata fte-a za crtanje vanjske krivulje
316
317         sve(i) = Array(x(i), y(i), zz(i)) ' polje tocaka vanjske krivulje
318         izlaznog brida
319         bte(j)(0) = bte(j)(0) - 0
320         bte(j)(1) = bte(j)(1) - R(i)
321         temp2 = bte(j)(0)
322         bte(j)(0) = bte(j)(0) * cos(beta(i)) - bte(j)(1) * sin(beta(
323 i)) ' PRVA TRANSFORMACIJA
324         bte(j)(1) = temp2 * sin(beta(i)) + bte(j)(1) * cos(beta(i))
325 + tan(alfa) * R(i)
326         bte(j)(2) = R(i)
327         fi3(j) = bte(j)(0) / R(i) ' DRUGA TRANSFORMACIJA
328         bte(j)(0) = sin(fi3(j)) * R(i)
329         bte(j)(2) = cos(fi3(j)) * R(i)
330
331         End If
332
333         fle(j)(0) = fle(j)(0) - 0
334         fle(j)(1) = fle(j)(1) - R(i)
335         temp2 = fle(j)(0)
336         fle(j)(0) = fle(j)(0) * cos(beta(i)) - fle(j)(1) * sin(beta(i))
337 ' PRVA TRANSFORMACIJA
338         fle(j)(1) = temp2 * sin(beta(i)) + fle(j)(1) * cos(beta(i)) + tan
339 (alfa) * R(i)
340         fle(j)(2) = R(i)
341         fi2(j) = fle(j)(0) / R(i) ' DRUGA TRANSFORMACIJA
342         fle(j)(0) = sin(fi2(j)) * R(i)
343         fle(j)(2) = cos(fi2(j)) * R(i)
344         ble(j)(0) = ble(j)(0) - 0
345         ble(j)(1) = ble(j)(1) - R(i)
346         temp2 = ble(j)(0)
347         ble(j)(0) = ble(j)(0) * cos(beta(i)) - ble(j)(1) * sin(beta(i))
348 ' PRVA TRANSFORMACIJA
349         ble(j)(1) = temp2 * sin(beta(i)) + ble(j)(1) * cos(beta(i)) + tan
350 (alfa) * R(i)
351         ble(j)(2) = R(i)
352         fi4(j) = ble(j)(0) / R(i) ' DRUGA TRANSFORMACIJA
353         ble(j)(0) = sin(fi4(j)) * R(i)
354         ble(j)(2) = cos(fi4(j)) * R(i)
355
356         x1(i) = fle(0)(0) ' x koordinata fle-a za crtanje vanjske krivulje
357
358         y1(i) = fle(0)(1) ' x koordinata fle-a za crtanje vanjske krivulje
359
360         z1(i) = fle(0)(2) ' x koordinata fle-a za crtanje vanjske krivulje
361
362         svel(i) = Array(x1(i), y1(i), z1(i)) ' polje tocaka vanjske krivulje
363         ulaznog brida
364
365         Next
366
367

```

```

353
354
355     If Projekcija = 2 Then
356
357         crvfte = Rhino.AddInterpCurve(fte, 3, 0) ' crtanje krivulje fte
- lice izlazni
358         crvbte = Rhino.AddInterpCurve(bte, 3, 0) ' crtanje krivulje bte
- naličje izlazni
359         crvfle = Rhino.AddInterpCurve(fle, 3, 0) ' crtanje krivulje fle
- lice ulazni
360         crvble = Rhino.AddInterpCurve(ble, 3, 0) ' crtanje krivulje ble
- 'naličje ulazni
361         krivulja(i) = Array(crvfte, crvfle) ' polje za spajanje lica
362         krivuljal(i) = Array(crvbte, crvble) ' polje za spajanje nalicja
363         spojenifte(i) = Rhino.JoinCurves(krivulja(i)) ' spajanje 2 krivulje
profila u jednu
364         spojenibte(i) = Rhino.JoinCurves(krivuljal(i)) ' spajanje 2 krivulje
profila u jednu
365         Rhino.DeleteObject(crvfte) ' brisanje pojedinačnih linija profila
366         Rhino.DeleteObject(crvbte) ' brisanje pojedinačnih linija profila
367         Rhino.DeleteObject(crvfle) ' brisanje pojedinačnih linija profila
368         Rhino.DeleteObject(crvble) ' brisanje pojedinačnih linija profila
369
370     End If
371
372     If i = 0 Then
373
374         If la > (fte(0)(1) - 0.025 * D / 2) Then
375
376             La = fte(0)(1) - 0.025 * D / 2
377             rhino.print"La je korigiran zbog krivog iznosa duljine"
378             rhino.print"Novi La iznosi " & -La
379         End If
380
381
382         If lf < (fle(0)(1) + 0.025 * D / 2) Then
383
384             Lf = fle(0)(1) + 0.025 * D / 2
385             rhino.print"Lf je korigiran zbog krivog iznosa duljine"
386             rhino.print"Novi Lf iznosi " & Lf
387
388         End If
389
390     End If
391
392     If zaobljenje And distri Then
393
394         Call zaob(crvle, crvfle, crvble, ple(i), projekcija)
395         Call zaob(crvte, crvfte, crvbte, pte(i), projekcija)
396         Rhino.DeleteObject crvle
397         Rhino.DeleteObject crvte
398
399

```

```

399         End If
400
401     Next
402
403     vrhx = vrh * cos(beta(8)) ' PRVA transformacija najvise tocke krila
404     vrhy = vrh * sin(beta(8)) + tan(alfa) * R(8)
405     vrhz = R(8)
406     fiv = vrhx / R(8)
407     vrhx = R(8) * sin(fiv) ' DRUGA transformacija najvise tocke krila
408     vrhz = R(8) * cos(fiv)
409     sve(8) = Array(vrhx, vrhy, vrhz) ' dodavanje tocke vrha krila u polje za
crtanje vanjske krivulje
410     svel(8) = Array(vrhx, vrhy, vrhz) ' dodavanje tocke vrha krila u polje za
crtanje vanjske krivulje
411     Rhino.Prompt int((i + 1) / 8 * 100) & "%" 'postotak obavljenog
412
413     x2 = duljinap * cos(beta(8)) * cos(fiv) ' racunanje koordinata tocaka na
pravcu cilindra
414     y2 = duljinap * sin(beta(8)) ' racunanje koordinata tocaka na pravcu cilindra
415
416     z2 = -duljinap * cos(beta(8)) * sin(fiv)
416
417     If Projekcija = 2 Then
418
419         spline = Rhino.AddInterpCurve(sve, 3, 1,, Array(x2, y2, z2)) ' crtanje
vanjske krivulje izlaznog brida do vrha
420         splinel = Rhino.AddInterpCurve(svel, 3, 1,, Array(-x2, -y2, -z2)) '
crtanje vanjske krivulje ulaznog brida do vrha
421         spojenolice = Array(spojenifte(0)(0), spojenifte(1)(0), spojenifte(2)
(0), spojenifte(3)(0), spojenifte(4)(0), spojenifte(5)(0), spojenifte(6)(0)
, spline, splinel)
422         spojenonalicje = Array(spojenibte(0)(0), spojenibte(1)(0), spojenibte
(2)(0), spojenibte(3)(0), spojenibte(4)(0), spojenibte(5)(0), spojenibte(6)(
0), spline, splinel)
423         rub = Array(spline, splinel)
424         ploha = Rhino.AddNetworkSrf(spojenolice)
425         plohal = Rhino.Addnetworksrf(spojenonalicje)
426         Rhino.DeleteObject(spline) 'brisanje pojedinih vanjskih krivulja
427
         Rhino.DeleteObject(splinel) 'brisanje pojedinih vanjskih krivulja
428
         temp = Rhino.SurfaceClosestPoint(ploha, array(0, 0, d / 10))
429
         rhino.ExtendSurface ploha, temp, r(0) / 2
430         rhino.ExtendSurface plohal, temp, r(0) / 2
431         krivuljeuk = rhino.allobjects()
432         plohe = rhino.JoinSurfaces(Array(krivuljeuk(0), krivuljeuk(1)))
433
         Rhino.DeleteObject(ploha)
434         Rhino.DeleteObject(plohal)
435         ravnina = Array(Array(0.0, La, 0.0), Array(1.0, 0.0, 0.0), Array(0.0
, 0.0, 1.0))
436         glavina = Rhino.AddTruncatedCone(ravnina, Ra, -Lf + La, Rf, True)
437
         propeler = Rhino.BooleanUnion(Array(plohe, glavina))
438         krivuljeuk = rhino.allobjects()
439         kut = 360 / Z
440

```

```

440
441     End If
442     ' petlja za rotiranje krila
443     If Projekcija = 2 Then
444
445         For i = 1 To Z - 1
446             rot(i) = Rhino.RotateObjects(krivuljeuk, Array(0, 0, 0), kut * i
, Array(0, 1, 0), True)
447         Next
448
449     End If
450
451     If projekcija = 2 Then
452
453         krivuljeuk = Rhino.ObjectsByType(4)
454         Rhino.AddLayer "Presjek profila", RGB(0, 250, 0)
455         Rhino.CurrentLayer "Presjek profila"
456         Rhino.ObjectLayer krivuljeuk, "Presjek profila"
457
458     End If
459
460 End Sub
461
462 Sub kote(a, c, R, hte, hle, tte, tle, pte, ple, vrh, PdivD, kotA1, kotB1, kotA2
, kotB2, kotD1, kotD2, projekcija)
463
464     If projekcija = 2 Then
465         Exit Sub
466     End If
467
468     Dim i,j,linstart, linend
469     Dim x(7),y(7)
470     Dim koefa(5),koefb(5)
471     Dim xtemp,ytemp,koefatemp,koefbtemp
472     Dim lbc,duz,strduz,pta1,ptd1,pta2,ptd2
473     Rhino.AddLayer "NVkote", RGB(0, 255, 0)
474     Rhino.CurrentLayer "NVkote"
475     x(0) = -PdivD * R(8) / Rhino.Pi
476     y(0) = 0
477     For i=0 To 8
478         x(1) = 0
479         y(1) = R(i)
480         If i < 8 Then
481             'x(3) = a(i)
482             'y(3) = R(i) + hle(i)
483             'x(6) = a(i) - c(i)
484             'y(6) = R(i) + hte(i)
485         Else
486             x(3) = 0
487             y(3) = R(i)
488             x(6) = vrh
489             y(6) = R(i)
490         End If
491
492         Call p2t(x(0), y(0), x(1), y(1), koefa(0), koefb(0))
493         Call okt(koefa(0), koefb(0), x(1), y(1), koefa(1), koefb(1))
494         'Call sp(koefa(0), koefb(0), koefa(1), koefb(1), x(2), y(2))
495

```

```

495 'Call okt(koefa(1), koefb(1), x(3), y(3), koefa(2), koefb(2))
496 'Call okt(koefa(2), koefb(2), x(1), y(1), koefa(3), koefb(3))
497 'Call sp(koefa(2), koefb(2), koefa(3), koefb(3), x(4), y(4))
498 'Call okt(koefa(3), koefb(3), x(6), y(6), koefa(4), koefb(4))
499 'Call sp(koefa(3), koefb(3), koefa(4), koefb(4), x(5), y(5))
500 'Call okt(koefa(0), koefb(0), x(6), y(6), koefa(5), koefb(5))
501 'Call sp(koefa(0), koefb(0), koefa(5), koefb(5), x(7), y(7))
502 ' okomito-tangentna kota
503
504 If i < 8 Then
505     xtemp = a(i)
506     ytemp = R(i) + hle(i) - tle(i) / 2
507     koefbtemp = ytemp - koefa(1) * xtemp
508     koefbtemp = 2 * koefbtemp - koefb(1)
509     ytemp = R(i)
510     linstart = Array(0, koefbtemp, 0)
511     linend = Array((ytemp - koefbtemp) / koefa(1), ytemp, 0)
512     duz = rhino.AddLine(linstart, linend)
513     ptd1 = Rhino.CurveClosestObject(ple(i)(0), duz)
514     If Not (IsNull(ptd1)) Then
515         Rhino.addpoint ptd1(2)
516     End If
517     Rhino.DeleteObject duz
518     xtemp = a(i)
519     ytemp = R(i) + hle(i) - tle(i) / 2
520     koefbtemp = ytemp - koefa(0) * xtemp
521     koefbtemp = 2 * koefbtemp - koefb(0)
522     ytemp = R(i)
523     linstart = Array(0, koefbtemp, 0)
524     linend = Array((ytemp - koefbtemp) / koefa(0), ytemp, 0)
525     duz = rhino.AddLine(linstart, linend)
526     pta1 = Rhino.CurveClosestObject(ple(i)(0), duz)
527     If Not (IsNull(pta1)) Then
528         Rhino.addpoint pta1(2)
529     End If
530     Rhino.DeleteObject duz
531     xtemp = a(i) - c(i)
532     ytemp = R(i) + hte(i) - tte(i) / 2
533     koefbtemp = ytemp - koefa(0) * xtemp
534     koefbtemp = 2 * koefbtemp - koefb(0)
535     linstart = Array(0, koefbtemp, 0)
536     linend = Array((ytemp - koefbtemp) / koefa(0), ytemp, 0)
537     duz = rhino.AddLine(linstart, linend)
538     pta2 = Rhino.CurveClosestObject(pte(i)(0), duz)
539     If Not (IsNull(pta2)) Then
540         Rhino.addpoint pta2(2)
541     End If
542     Rhino.DeleteObject duz
543     xtemp = a(i) - c(i)
544     ytemp = R(i) + hte(i) - tte(i) / 2
545     koefbtemp = ytemp - koefa(1) * xtemp
546     koefbtemp = 2 * koefbtemp - koefb(1)
547     linstart = Array(0, koefbtemp, 0)
548     linend = Array((ytemp - koefbtemp) / koefa(1), ytemp, 0)
549     duz = rhino.AddLine(linstart, linend)
550     ptd2 = Rhino.CurveClosestObject(pte(i)(0), duz)
551     If Not (IsNull(ptd2)) Then
552         Rhino.addpoint ptd2(2)
553     End If
554 End If

```

```

552         End If
553         Rhino.DeleteObject duz
554         'Call p2t(x(0), y(0), x(1), y(1), koefa(0), koefb(0))
555         Call okt(koefa(0), koefb(0), ptd1(2)(0), ptd1(2)(1), koefa(1), koefb
(1))
556         Call sp(koefa(0), koefb(0), koefa(1), koefb(1), x(2), y(2))
557         Call okt(koefa(1), koefb(1), pta1(2)(0), pta1(2)(1), koefa(2), koefb
(2))
558         Call sp(koefa(1), koefb(1), koefa(2), koefb(2), x(3), y(3))
559         Call okt(koefa(2), koefb(2), x(1), y(1), koefa(3), koefb(3))
560         Call sp(koefa(2), koefb(2), koefa(3), koefb(3), x(4), y(4))
561         Call okt(koefa(3), koefb(3), pta2(2)(0), pta2(2)(1), koefa(4), koefb
(4))
562         Call sp(koefa(3), koefb(3), koefa(4), koefb(4), x(5), y(5))
563         Call okt(koefa(0), koefb(0), ptd2(2)(0), ptd2(2)(1), koefa(5), koefb
(5))
564         Call sp(koefa(4), koefb(4), koefa(5), koefb(5), x(6), y(6))
565         Call sp(koefa(0), koefb(0), koefa(5), koefb(5), x(7), y(7))
566         For j=0 To 6
567             linstart = Array(x(j), y(j), 0)
568             linend = Array(x(j + 1), y(j + 1), 0)
569             strduz = Rhino.Addline(linstart, linend)
570         Next
571         kotA1(i) = Sqr((x(2) - x(3)) ^ 2 + (y(2) - y(3)) ^ 2)
572         kotB1(i) = Sqr((x(3) - x(4)) ^ 2 + (y(3) - y(4)) ^ 2)
573         kotD1(i) = Sqr((x(2) - ptd1(2)(0)) ^ 2 + (y(2) - ptd1(2)(1)) ^ 2
)
574         kotA2(i) = Sqr((x(6) - x(7)) ^ 2 + (y(6) - y(7)) ^ 2)
575         kotB2(i) = Sqr((x(5) - x(6)) ^ 2 + (y(5) - y(6)) ^ 2)
576         kotD2(i) = Sqr((x(7) - ptd2(2)(0)) ^ 2 + (y(7) - ptd2(2)(1)) ^ 2
)
577     Else
578         Call sp(koefa(0), koefb(0), koefa(1), koefb(1), x(2), y(2))
579         Call okt(koefa(1), koefb(1), x(3), y(3), koefa(2), koefb(2))
580         Call okt(koefa(2), koefb(2), x(1), y(1), koefa(3), koefb(3))
581         Call sp(koefa(2), koefb(2), koefa(3), koefb(3), x(4), y(4))
582         Call okt(koefa(3), koefb(3), x(6), y(6), koefa(4), koefb(4))
583         Call sp(koefa(3), koefb(3), koefa(4), koefb(4), x(5), y(5))
584         Call okt(koefa(0), koefb(0), x(6), y(6), koefa(5), koefb(5))
585         Call sp(koefa(0), koefb(0), koefa(5), koefb(5), x(7), y(7))
586         For j=0 To 6
587             linstart = Array(x(j), y(j), 0)
588             linend = Array(x(j + 1), y(j + 1), 0)
589             strduz = Rhino.Addline(linstart, linend)
590         Next
591
592         kotA1(i) = Sqr((x(2) - x(3)) ^ 2 + (y(2) - y(3)) ^ 2)
593         kotB1(i) = Sqr((x(3) - x(4)) ^ 2 + (y(3) - y(4)) ^ 2)
594         kotD1(i) = kotA1(i)
595         kotA2(i) = Sqr((x(6) - x(7)) ^ 2 + (y(6) - y(7)) ^ 2)
596         kotB2(i) = Sqr((x(5) - x(6)) ^ 2 + (y(5) - y(6)) ^ 2)
597         kotD2(i) = kotA2(i)
598
599     End If
600
601 Next
602
603

```



```

603 End Sub
604
605 Sub lukovi(R, kotA1, kotB1, kotA2, kotB2, kotD1, kotD2, projekcija)
606   If projekcija = 2 Then
607     Exit Sub
608   End If
609   Dim i, linstart, linend, sredine
610   Dim le(8), te(8), mid(8)
611   Dim bok(16), tes(8), les(8)
612   Dim kut(2), at, bt
613   Dim fi
614
615   Rhino.AddLayer "NVnacrt", RGB(0, 0, 255)
616   Rhino.CurrentLayer "NVnacrt"
617   For i=0 To 8
618     kut(1) = kotA1(i) / R(i)
619     le(i) = array(sin(kut(1)) * R(i), cos(kut(1)) * R(i), 0)
620     linstart = array(0, R(i), 0)
621     linend = le(i)
622     Rhino.AddArcPtTanPt linstart, array(1, 0, 0), linend
623     kut(2) = kotA2(i) / R(i)
624     te(i) = array(-sin(kut(2)) * R(i), cos(kut(2)) * R(i), 0)
625     linstart = array(0, R(i), 0)
626     linend = te(i)
627     Rhino.AddArcPtTanPt linstart, array(-1, 0, 0), linend
628     kut(0) = (kut(1) - kut(2)) / 2
629     mid(i) = array(sin(kut(0)) * R(i), cos(kut(0)) * R(i), 0)
630     If i < 8 Then rhino.AddPoint le(i)
631     rhino.AddPoint te(i)
632   Next
633   le(8) = te(8)
634   mid(8) = te(8)
635   Call p2t(0, 0, te(8)(0), te(8)(1), at, bt)
636   at = -1 / at
637   Rhino.AddInterpCurve le, 3, 1,, Array(-1, -at, 0)
638   Rhino.AddInterpCurve te, 3, 1,, Array(1, at, 0)
639   sredine = Rhino.AddInterpCurve(mid, 3, 1)
640   For i=0 To 7
641     kut(1) = kotD1(i) / R(i)
642     le(i) = array(sin(kut(1)) * R(i), cos(kut(1)) * R(i), 0)
643     kut(2) = kotD2(i) / R(i)
644     te(i) = array(-sin(kut(2)) * R(i), cos(kut(2)) * R(i), 0)
645     rhino.AddPoint le(i)
646     rhino.AddPoint te(i)
647   Next
648   Rhino.AddInterpCurve le, 3, 1,, Array(-1, -at, 0)
649   Rhino.AddInterpCurve te, 3, 1,, Array(1, at, 0)
650   'linstart=Array(0,0,0)
651   'linend=Array(te(8)(0),te(8)(1),te(8)(2))
652   'Rhino.Addline linstart, linend
653
654   Rhino.AddLayer "NVbokocrt", RGB(0, 127, 0)
655   Rhino.CurrentLayer "NVbokocrt"
656
657   linstart = Array(0, 0, 0)
658   linend = Array(tan(rhino.ToRadians(-15)) * R(8), R(8), 0)
659   Rhino.Addline linstart, linend
660

```

```

660
661     For i=0 To 7
662         bok(i) = array(tan(rhino.ToRadians(-15)) * R(i) + kotB1(i), le(i)(1)
, 0)
663         bok(16 - i) = array(tan(rhino.ToRadians(-15)) * R(i) - kotB2(i), te(
i)(1), 0)
664
665     Next
666     bok(8) = array(tan(rhino.ToRadians(-15)) * R(8) - kotB2(8), te(8)(1), 0)

667     For i=0 To 16
668         rhino.AddPoint bok(i)
669     Next
670     Rhino.AddInterpCurve bok, 3
671 End Sub
672
673 Function top(aa, bb, xx, yy) 'udaljenost točke od pravca
674     Call okt(aa, bb, xx, yy, aa1, bb1)
675     Call sp(aa, bb, aa1, bb1, xx1, yy1)
676     top = Sqr((yy1 - yy) ^ 2 + (xx1 - xx) ^ 2)
677 End Function
678
679 Sub p2t(xx1, yy1, xx2, yy2, aa, bb) 'jednadžba pravca kroz dvije točke
680     aa = (yy2 - yy1) / (xx2 - xx1)
681     bb = (yy2 - yy1) / (xx2 - xx1) * (-xx1) + yy1
682 End Sub
683
684 Sub okt(aa1, bb1, xx, yy, aa2, bb2) 'okomiti pravac kroz točku
685     aa2 = -1 / aa1
686     bb2 = yy - aa2 * xx
687 End Sub
688
689 Sub sp(aa1, bb1, aa2, bb2, xx, yy) 'sjecište dvaju pravaca
690     xx = (bb2 - bb1) / (aa1 - aa2)
691     yy = aa1 * xx + bb1
692 End Sub
693
694 Sub zaob(str1, str2, str3, pe, projekcija) 'filetiranje rubova
695     If projekcija = 2 Then
696         Exit Sub
697     End If
698     Dim strFillet, arrCCX, arrTrim, dbl1
699     Dim strg, strd
700     Dim tol, skala
701     Dim blnRedraw
702
703     tol = 0.01
704     skala = 1 / tol
705     dbl1 = int(Rhino.CurveLength(str1) / 2)
706     arrCCX = 1
707     Do While skala > .9
708         Do While Not IsNull(arrCCX)
709             dbl1 = dbl1 + tol * skala
710             strFillet = Rhino.AddFilletCurve(str2, str3, dbl1)
711             If Not IsNull(strFillet) Then
712                 arrCCX = Rhino.CurveCurveIntersection(strFillet, str1, tol)
713                 Rhino.DeleteObject strFillet
714

```

```

714         End If
715     Loop
716     dbl1 = dbl1 - tol * skala
717     skala = skala / 10
718     arrccx = 1
719     Loop
720     strFillet = Rhino.AddFilletCurve(str2, str3, dbl1)
721     arrCCX = Rhino.CurveCurveIntersection(strFillet, str2, tol * 3)
722     arrtrim = Rhino.CurveDomain(str2)
723     strg = Rhino.TrimCurve(str2, Array(arrccx(0, 7), arrtrim(1)))
724     arrCCX = Rhino.CurveCurveIntersection(strFillet, str3, tol * 3)
725     arrtrim = Rhino.CurveDomain(str3)
726     strd = Rhino.TrimCurve(str3, Array(arrccx(0, 7), arrtrim(1)))
727     pe = rhino.joincurves(Array(strg, strFillet, strd), True)
728 End Sub
729
730 Sub LeTe(t, tte, tle, distri, zaobljenje, projekcija) 'distribucija debljina
    ulaznih i izlaznih bridova
731     Dim Distrib,i,dane
732     If projekcija = 2 Then
733         Exit Sub
734     End If
735     Distrib = 0
736     Distrib = Rhino.GetReal("Vrsta distribucije debljina ulaznih i izlaznih
    bridova")
737     distri = True
738     Select Case Distrib
739     Case 1
740         'prema
741         '
742         '
743         '
744         tte(7) = t(7) * 0.245
745         tte(6) = t(6) * 0.152
746         tte(5) = t(5) * 0.12
747         tte(4) = t(4) * 0.1
748         tte(3) = t(3) * 0.085
749         tte(2) = t(2) * 0.075
750         tte(1) = t(1) * 0.068
751         tte(0) = t(0) * 0.057
752
753         tle(8) = t(8) * 0
754         tle(7) = t(7) * 0.245
755         tle(6) = t(6) * 0.17
756         tle(5) = t(5) * 0.143
757         tle(4) = t(4) * 0.134
758         tle(3) = t(3) * 0.13
759         tle(2) = t(2) * 0.127
760         tle(1) = t(1) * 0.124
761         tle(0) = t(0) * 0.12
762     Case 2
763         'prema
764         'Parametric B - Wagenigen screw model.Verify compliance CFD computation
    With hydrodynamics plots.
765         'Andrzej Rachwalik
766         'Table 6
767         tte(7) = t(7) * 0.142969
768

```

```

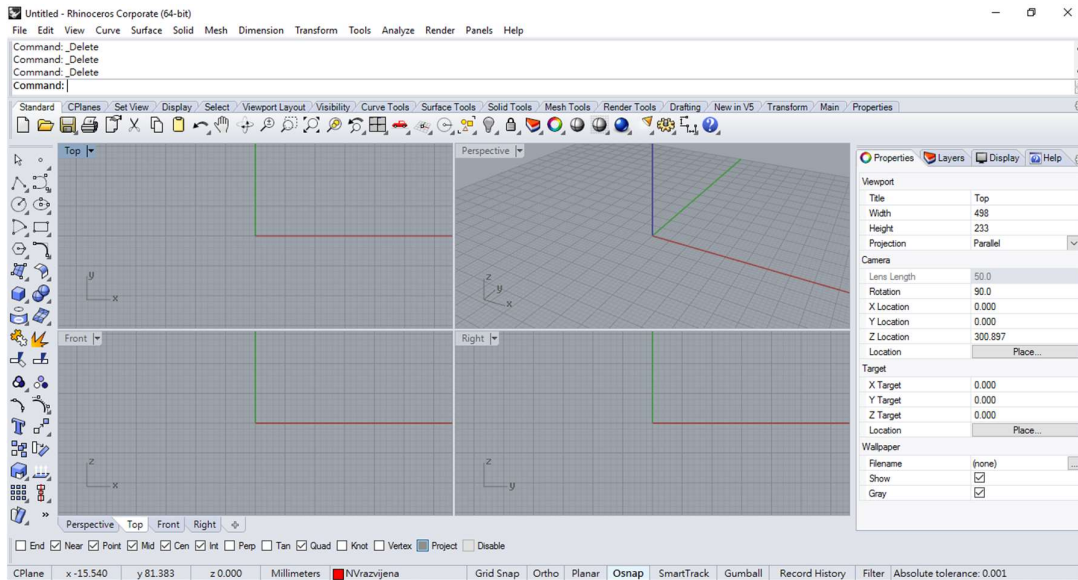
768         tte(6) = t(6) * 0.13
769         tte(5) = t(5) * 0.125
770         tte(4) = t(4) * 0.12
771         tte(3) = t(3) * 0.115
772         tte(2) = t(2) * 0.1
773         tte(1) = t(1) * 0.0775602
774         tte(0) = t(0) * 0.065562
775
776         tle(8) = t(8) * 0
777         tle(7) = t(7) * 0.142969
778         tle(6) = t(6) * 0.1335
779         tle(5) = t(5) * 0.121
780         tle(4) = t(4) * 0.0998
781         tle(3) = t(3) * 0.0994
782         tle(2) = t(2) * 0.0986
783         tle(1) = t(1) * 0.0968
784         tle(0) = t(0) * 0.1072
785     Case Else
786         'bez debljina ulaznih i izlaznih bridova
787         For i=0 To 8
788             tte(i) = 0
789             tle(i) = 0
790         Next
791
792         distri = False
793     End Select
794     dane = Rhino.GetString("Želite li zaobliti ulazne i izlazne bridove? (d/n)
795 ")
796     If dane = "d" Then
797         zaobljenje = True
798     Else
799         zaobljenje = False
800     End If
801 End Sub

```

PRILOG 2

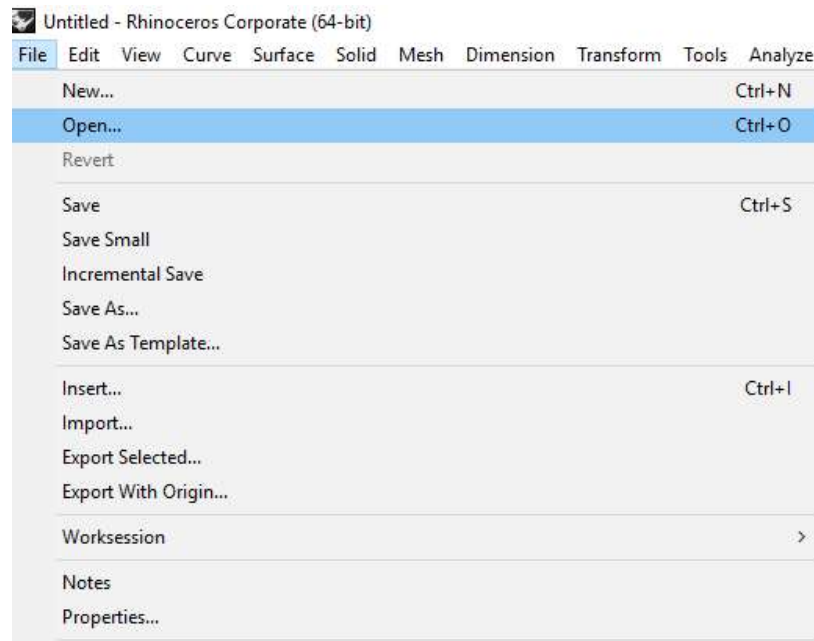
Upute za korištenje skripte za prikaz broskog vijka

1. Pokrenite program *Rhinoceros* koji mora biti instaliran na računalu na kojem želite pokrenuti aplikaciju za trodimenzijski prikaz brodskog vijka.



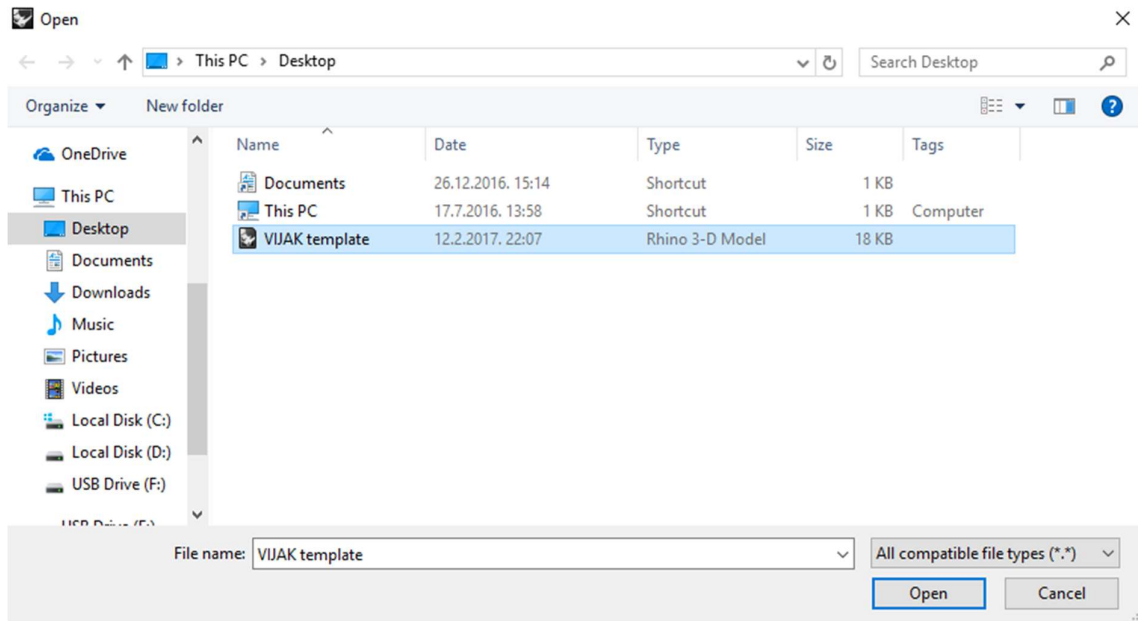
Slika 1. Početni zaslon programa *Rhinoceros*

2. Nakon što ste pokrenuli program *Rhinoceros*, kliknite na izbornik *file*, zatim na opciju *open*.



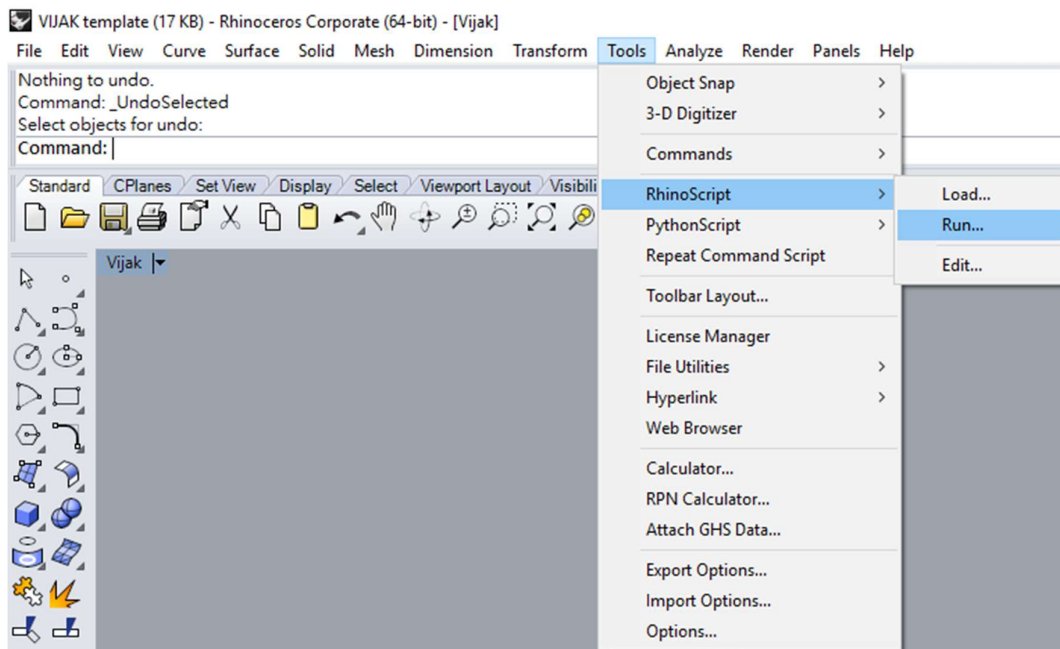
Slika 2. Upute za otvaranje predloška

3. Pojaviti će se prozor u kojemu ćete morati otvoriti predložak *Vijak template*. Nakon što ste ga pronašli otvorite predložak.



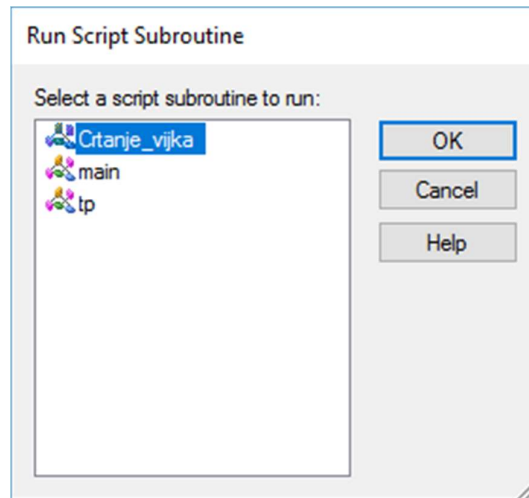
Slika3. Odabiranje predložka i otvaranje klikom na opciju *open*

4. Nakon otvaranja predložka *Vijak template*, odabirete izbornik *Tools*, zatim naredbu *RhinoScript*, zatim naredbu *Run...* kao na slici.



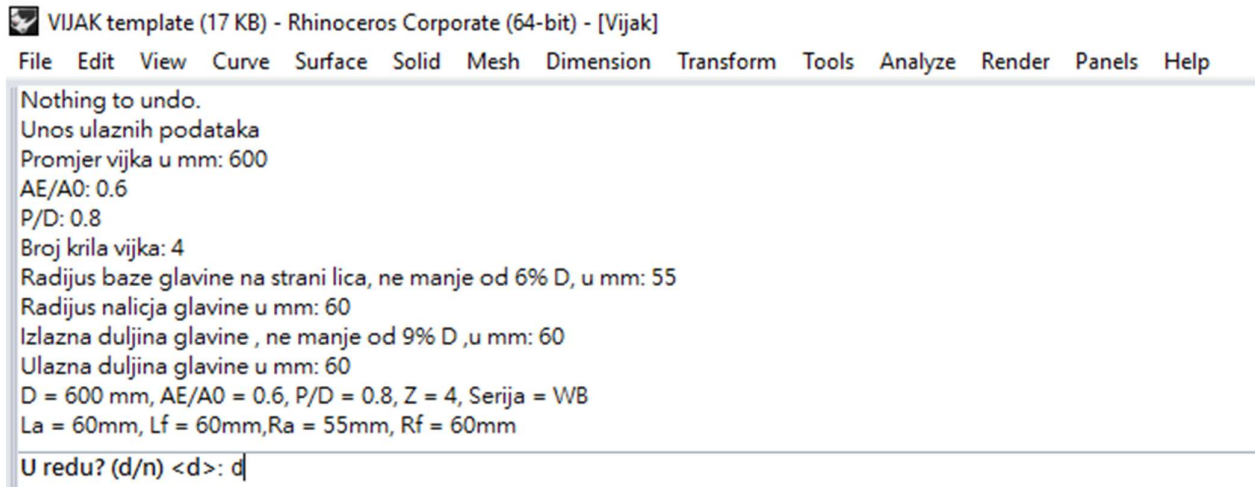
Slika 4. Pokretanje skipte unutar predložka

5. Odabir *subroutine* koja će pokrenuti izvršavanje skripte, tj crtanje vijka u 3D. Kliknete na *Crtanje_vijka* i zatim na *OK*.



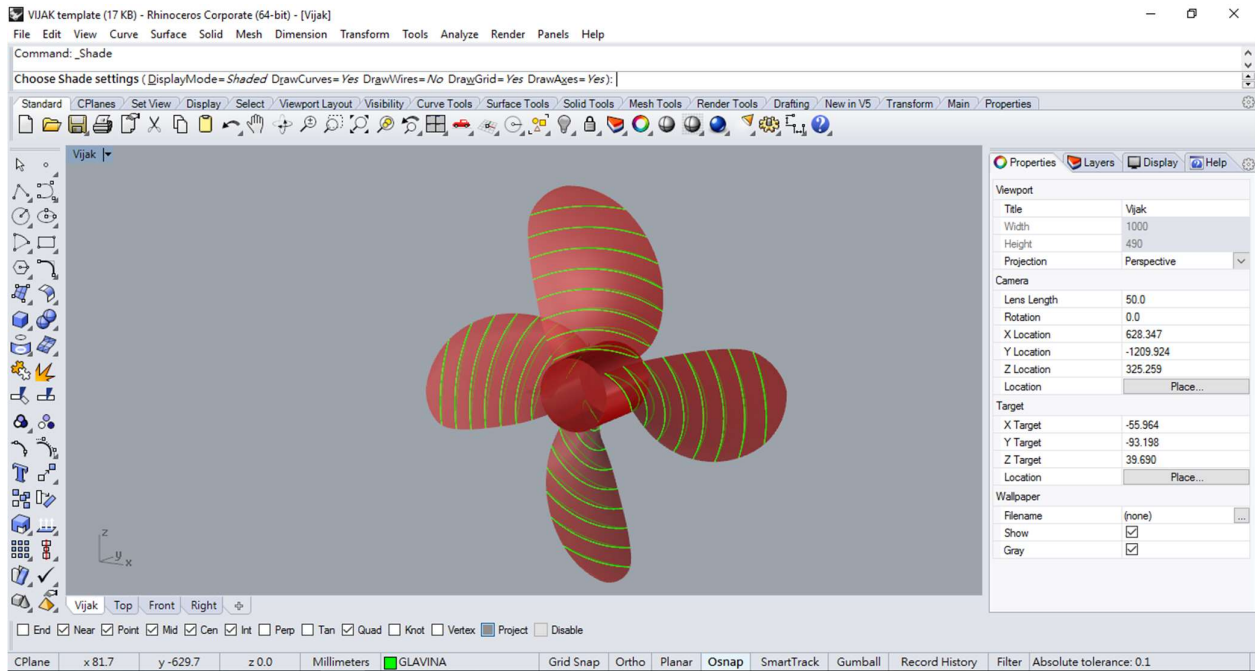
Slika5. Pokretanje skripte za crtanje vijka

6. U *command line* upisujete ulazne podatke po redoslijedu kako piše. Paziti na mjerne jedinice, sve duljine upisivati u milimetrima. Decimalne brojeve odvajati sa decimalnom točkom. Paziti da brojevi koje upisujete imaju fizikalni smisao. Nakon što ste upisali sve tražene podatke, provjerite je li sve uredno, ako je upišete „d“ i pritisnete tipku enter na tipkovnici, ako ste nešto krivo upisali onda upišete slovo „n“ i pritisnete tipku *enter* i ponovno upišete sve tražene vrijednosti. Nakon što ste provjerili ulazne vrijednosti, pritisnite slovo „d“ pa tipku *enter*. Tada će skripta nacrtati vijak sa dimenzijama koje ste upisali.



Slika 6. Unos ulaznih podataka

7. Prikaz broskog vijka u tri dimenzije. Skripta je automatski nacrtala vijak u ovisnosti o ulaznim podacima po geometriji Wageningenske B (WB) serije.



Slika 7. Prikaz vijka Wageningen B serije