

Numerički model tsunamija

Denk, Luka

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, University of Split, Faculty of science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:166:592790>

Rights / Prava: [Attribution-NonCommercial-NoDerivatives 4.0 International/Imenovanje-Nekomercijalno-Bez prerada 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2024-11-30**

Repository / Repozitorij:

[Repository of Faculty of Science](#)



UNIVERSITY OF SPLIT



Sveučilište u Splitu
Prirodoslovno – matematički fakultet

NUMERIČKI MODEL TSUNAMIJA

Završni rad

Luka Denk

Split, rujan 2018.

Temeljna dokumentacijska kartica

Sveučilište u Splitu
Prirodoslovno – matematički fakultet
Odjel za fiziku
Ruđera Boškovića 33, 21000 Split, Hrvatska

Završni rad

Numerički model tsunamija

Luka Denk

Sveučilišni preddiplomski studij Fizika

Sažetak:

Ponašanje tsunamija, valova iznimno dugačke valne duljine, blisko je povezano sa dubinom oceana. Cilj ovog rada je numeričkim metodama analizirati to ponašanje koristeći dobro poznatu valnu jednadžbu i samo osnovne pretpostavke, da je valna duljina mnogo puta veća od dubine oceana te da poznajemo dno iznad kojega se val kreće. Umjesto pretpostavljenih vrijednosti mogli bi se koristiti i podaci dobiveni stvarnim mjerenjima. Glavno ponašanje koje želimo simulirati je rast amplitude vala sa smanjenjem dubine. Provjerom rezultata dobivenih numeričkim rješavanjem valne jednadžbe i onih iz eksperimentalnih opažanja dolazimo do zaključka da jednostavni model tsunamija prikazan u ovom radu demonstrira ponašanje realnih valova velikih valnih duljina na otvorenom moru i plitkim vodama.

- Ključne riječi:** valna jednadžba, numerički model, tsunami
- Rad sadrži:** 17 stranica, 5 slika, 7 literaturnih navoda. Izvornik je na hrvatskom jeziku
- Mentor:** doc. dr. sc. Petar Stipanović
- Ocjenjivači:** doc. dr. sc. Petar Stipanović
prof. dr. sc. Mile Dželalija
Hrvoje Vrcan, mag. phys.
- Rad prihvaćen:** 25.09.2018.

Rad je pohranjen u knjižnici Prirodoslovno – matematičkog fakulteta, Sveučilišta u Splitu.

Basic documentation card

University of Split
Faculty of Science
Department of Physics
Ruđera Boškovića 33, 21000 Split, Croatia

Bachelor thesis

Numerical model of tsunami

Luka Denk

University undergraduate study programme Physics

Abstract:

Behaviour of tsunamis, waves of extraordinarily long wavelengths, is closely related to the depth of the ocean. The goal of this thesis is to analyse the behaviour of tsunamis using only the wave equation and the basic assumptions that the wavelength of a tsunami is a lot a lot bigger than the depth of the body of water the tsunami is traveling in. Empirically gathered data could be used instead of the assumptions. The main behaviour we are looking to simulate is the growth of amplitude with the decrease of depth. By comparing the results of calculations in this thesis with empirical observations we can conclude that the simple model of a tsunami presented in this thesis demonstrates the expected behaviour of real waves of long wavelengths on the open sea and in the shallow water.

Keywords: wave equation, numerical model, tsunami

Thesis consists of: 17 pages, 5 figures, 7 reference. Original language: Croatian

Supervisor: Assist. Prof. Dr. Petar Stipanović

Reviewers: Assist. Prof. Dr. Petar Stipanović
Prof. Mile Dželalija
Hrvoje Vrcan, MSc. Phys.

Thesis accepted: September 25. 2018.

Thesis is deposited in the library of the Faculty of Science, University of Split.

Sadržaj

1	Uvod	1
1.1	Tsunami	1
2	Analitičko rješenje valne jednačbe.....	3
2.1.1	Varijabilna brzina	4
3	Numeričko rješavanje valne jednačbe	5
3.1	Jednostavni model s konstantnom brzinom	5
3.2	Varijabilna brzina u numeričkom modelu	6
3.3	Model tsunamija	7
4	Rezultati	9
5	Zaključak	12
6	Literatura.....	13
A	Tipični kod za tsunami-model.....	14
B	Tipični primjer gnuplot-skripte.....	17

1 Uvod

Mnogobrojni problemi u fizici mogu se opisati parcijalnim diferencijalnim jednačbama. Jedna od njih je valna jednačba [1]

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{v^2} \frac{\partial^2 u}{\partial t^2}. \quad (1.1)$$

Navedeni oblik predstavlja (1+1)-dimenzionalnu (jedna dimenzija prostora i jedna vremena) valnu jednačbu čije je rješenje funkcija $u = u(x, t)$, dok za (2+1)-dimenzionalnu valnu jednačbu rješenje ima oblik $u = u(x, y, t)$.

Valnu jednačbu s više prostornih dimenzija možemo primjerice zapisati [2]

$$\nabla^2 u = \frac{1}{v^2} \frac{\partial^2 u}{\partial t^2} \quad (1.2)$$

gdje je $\nabla^2 = \sum_i \frac{\partial^2}{\partial x_i^2}$ laplasijan zapisan po i -tim Kartezijevim koordinatama, dok je v konstanta koja predstavlja brzinu kretanja vala u danom mediju, npr. za slučaj elektromagnetskih valova u vakuumu $v = c$, brzina odgovara brzini svjetlosti u vakuumu.

Matematički gledano, valna jednačba je linearna hiperbolična parcijalna diferencijalna jednačba drugog reda kojom u fizici opisuje velik broj pojava, širenje zvuka, kretanje elektromagnetskih valova, valove na vodi, širenje poremećaja na nategnutoj žici instrumenta... Kako se valna jednačba pojavljuje u brojnim područjima fizike, korisno je poznavati njezino ponašanje i rješenja.

Ovaj rad će se prvenstveno fokusirati na numeričko rješavanje valne jednačbe, no dati će i općenito analitičko rješenje. Uvest će se jednostavan model vala sa pripadajućim početnim i rubnim uvjetima te će se naknadno proširiti sa potrebnim opažanjima, pretpostavkama i ograničenjima potrebnim za modeliranje posebnih valova na vodi - tsunamija.

1.1 Tsunami

Za razliku od valova uzrokovanih vjetrom ili gravitacijom (plimni valovi), uzrok stvaranja tsunamija je nagli pomak (en. *displacement*) ogromnog volumena vode, bilo to zbog podvodne eksplozije, pada meteora u veliku površinu vode, seizmičke aktivnosti, odloma glečera ili nečeg drugoga [3], [4], [5], [6].

Od valova, koje svakodnevno vidamo na vodi, razlikuju se time [5] što imaju iznimno dugačke valne duljine (stotine kilometara), vrlo dugačke periode (od par minuta pa sve do par sati) i niske amplitude na otvorenom moru (manje od 1 metra) te ih je upravo zbog toga vrlo teško opaziti.

Primicanjem obali, njihova se brzina smanjuje zbog sve pliće dna, no kako je frekvencija izvora, a time i tok energije isti, amplituda tsunamija raste do nekoliko desetaka metara u procesu koji se zove nagomilavanje valova (eng. *wave shoaling*).

Riječ tsunami dolazi od japanskih riječi „tsu“, val, i „nami“, luka. Ime dolazi od japanskih ribara koji bi se vratili u luku poharanu valovima koje nisu primijetili na pučini, pa su zaključili da se ti valovi stvaraju u lukama.

Tsunamiji imaju dobro evidentiranu povijest svoje razorne moći. 11. ožujka 2011. potres magnitude 9.0 sa epicentrom 400 km od Tokija prouzročio je tsunami koji je odnio 20896 života. Nadalje, 26. 12. 2004. katastrofalno razorit tsunami pogodio je obale diljem Indijskog oceana i prouzročio procijenjenih 230 000 žrtava.

Brzina plitkih/površinskih valova velikih valnih duljina dana je sa

$$v = \sqrt{gH} \quad (1.3)$$

gdje g je akceleracije sile teže, H je prosječna dubina. Naziv plitki/površinski odnosi se na omjer valne duljine λ i dubine H . Kada je taj omjer $\lambda/H > 20$, tada vrijedi gornja formula. Iz jednadžbe je jasno vidljivo da će brzina tsunamija biti mala u plitkim morima. Zanimljivo je primijetiti da ako u formulu uvrstimo prosječnu dubinu Tihog oceana, 4280 m, za brzinu tsunamija dobijemo $v \cong 740$ m/s.

2 Analitičko rješenje valne jednadžbe

Valna jednadžba (1.1) poznata je još od 18. stoljeća. S obzirom da se pojavljuje u mnogim fizikalnim pojavama, njezinom je rješavanju dato mnogo pažnje. Općenito, rješenje nema fizikalnog smisla samo po sebi, on dolazi kod uvođenja početnih i rubnih uvjeta.

Promotrimo sljedeće početne i rubne uvjete

$$\begin{aligned} c^2(u_{xx} + u_{yy}) &= u_{tt} & x, y \in (0, L), t > 0 \\ u(x, y, 0) &= f(x, y) & x, y \in (0, L) \\ u(0, 0, t) &= u(L, L, t) = 0 & t > 0 \\ \frac{\partial u}{\partial t} \Big|_{t=0} &= g(x, y) & x, y \in (0, L). \end{aligned}$$

Oznaka varijable u indeksu funkcije označava derivaciju po toj varijabli $u_{xx} = \frac{\partial^2 u}{\partial x^2}$. Separiramo li prostorne i vremenske koordinate pretpostavljajući rješenje u obliku

$$u(x, y, t) = F(x, y)G(t)$$

dobivamo

$$\frac{G_{tt}}{c^2 G} = \frac{1}{F} (F_{xx} + F_{yy}) = -v^2.$$

Lijeva i desna strana su neovisne jedna o drugoj pa moraju biti konstantne, čime smo dobili dvije diferencijalne jednadžbe. Pretpostavimo sličan separirani oblik rješenja za $F(x, y)$,

$$F(x, y) = H(x)Q(y).$$

Nadalje imamo

$$\frac{H_{xx}}{H} = \frac{-1}{Q} (Q_{yy} + Qv^2) = -\kappa^2.$$

I u ovom slučaju lijeva i desna strana su neovisne, prema tome jednake konstanti $-\kappa^2$, iz čega imamo još dvije diferencijalne jednadžbe

$$H_{xx} + \kappa^2 H = 0, \quad Q_{yy} + \rho^2 Q = 0, \quad \rho^2 + \kappa^2 = v^2,$$

Rješenja za $H(x)$ i $Q(y)$ dana sa trigonometrijskim funkcijama [1]

$$H(x) = A \cos(\kappa x) + B \sin(\kappa x), \quad Q(y) = C \cos(\rho y) + D \sin(\rho y).$$

Uz rubne uvjete $H(0) = H(L) = Q(0) = Q(L) = 0$ slijedi [2]

$$F_{mn}(x, y) = \sin\left(\frac{m\pi x}{L}\right) \sin\left(\frac{n\pi y}{L}\right).$$

$\rho^2 = v^2 - \kappa^2$ i $\lambda = cv$ daju spektar vrijednosti $\lambda = c \sqrt{\kappa^2 + \rho^2}$, odnosno

$$\lambda_{mn} = c\pi/L \sqrt{m^2 + n^2}.$$

Rješenje za G je

$$G_{mn}(t) = B_{mn} \cos(\lambda_{mn}t) + D_{mn} \sin(\lambda_{mn}t)$$

Koeficijenti B_{mn} i D_{mn} utvrđuju se iz Fourierovih koeficijenata, uz početne uvjete

$$u(x, y, 0) = f(x, y) \quad \text{i} \quad \frac{\partial u}{\partial t} \Big|_{t=0} = g(x, y) \quad [2]$$

$$B_{mn} = \frac{2}{L} \int_0^L \int_0^L dx dy f(x, y) \sin\left(\frac{m\pi x}{L}\right) \sin\left(\frac{n\pi y}{L}\right)$$

$$D_{mn} = \frac{2}{L} \int_0^L \int_0^L dx dy g(x, y) \sin\left(\frac{m\pi x}{L}\right) \sin\left(\frac{n\pi y}{L}\right)$$

Valna je funkcija tada dana sa

$$u(x, y, t) = \sum_{m,n=1}^{\infty} u_{mn}(x, y, t) = \sum_{m,n=1}^{\infty} F_{mn}(x, y) G_{mn}(t).$$

2.1.1 Varijabilna brzina

U prethodnom izvodu, brzina vala c , uzeta je kao konstanta. U našem realnom slučaju tsunamija, brzina ovisi o dubini oceana [1]

$$v(x, y) = \sqrt{gH(x, y)} \quad (2.1)$$

S ovakvom definicijom brzine vala, izračun valne funkcije postaje vrlo kompliciran i numeričke metode pružaju dobar način pronalaska rješenja.

3 Numeričko rješavanje valne jednadžbe

3.1 Jednostavni model s konstantnom brzinom

Radi jednostavnosti, prvo ćemo promotriti kako prilagoditi $(2 + 1)$ -dimenzionalnu valnu jednadžbu

$$u_{xx} + u_{yy} = u_{tt} \quad x, y \in (0,1), t > 0 \quad (3.1)$$

za numeričko rješavanje gdje su x i y dani relativno u odnosu na duljinu L te je uzeta jedinična brzina. Kako je brzina konstantna Pri tom pretpostavljamo da su nam zadani sljedeći početni i rubni uvjeti

$$\begin{aligned} u(x, y, 0) &= g(x, y) & x, y \in (0,1) \\ u(0,0,t) = u(1,1,t) &= 0 & t > 0 \\ \left. \frac{\partial u}{\partial t} \right|_{t=0} &= 0 & x, y \in (0,1) \end{aligned}$$

Diskretizacija vremena i prostora

$$\begin{aligned} t &= l\Delta t & l > 0 \\ x &= i\Delta x & 0 \leq i \leq n_x \\ y &= j\Delta y & 0 \leq j \leq n_y \end{aligned}$$

gdje su i, j, l cijeli brojevi, omogućava nam traženje rješenja $u_{i,j}^l = u(i\Delta x, j\Delta y, l\Delta t)$ u konačno mnogo točaka prostora, odnosno računalno rješavanje problema. Odaberemo $n_x = n_y = N_{xy}$ i $\Delta x = \Delta y = h$ radi jednostavnosti. Tada parcijalne derivacije aproksimiramo izrazima

$$\begin{aligned} u_{tt} &= \frac{u_{i,j}^{l+1} - 2u_{i,j}^l + u_{i,j}^{l-1}}{\Delta t^2} \\ u_{xx} &= \frac{u_{i+1,j}^l - 2u_{i,j}^l + u_{i-1,j}^l}{h^2} \\ u_{yy} &= \frac{u_{i,j+1}^l - 2u_{i,j}^l + u_{i,j-1}^l}{h^2} \end{aligned}$$

Gornji indeks predstavlja vrijeme, donji indeksi redom, x i y položaji. Uvrštavanjem u valnu jednadžbu (3.1) dobijemo sljedeći izraz [2]

$$u_{i,j}^{l+1} = 2u_{i,j}^l - u_{i,j}^{l-1} + \frac{\Delta t^2}{h^2} (u_{i+1,j}^l - 4u_{i,j}^l + u_{i-1,j}^l + u_{i,j+1}^l + u_{i,j-1}^l). \quad (3.2)$$

Ovaj nam izraz predstavlja eksplicitnu shemu računanja $u_{i,j}^{l+1}$ pod pretpostavkom da su nam poznati svi u^l i u^{l-1} .

Uzmimo u obzir i početni uvjet $\frac{\partial u}{\partial t}\Big|_{t=0} = 0$ čiji diskretizirani oblik glasi

$$u_t = \frac{u_{i,j}^1 - u_{i,j}^{-1}}{2\Delta t} = 0$$

što implicira $u_{i,j}^1 = u_{i,j}^{-1}$ i uvršteno u (3.2) omogućava procjenu stanja prije početnog trenutka

$$u_{i,j}^{-1} = u_{i,j}^0 + \frac{\Delta t^2}{2h^2} (u_{i+1,j}^0 - 4u_{i,j}^0 + u_{i-1,j}^0 + u_{i,j+1}^0 + u_{i,j-1}^0). \quad (3.3)$$

3.2 Varijabilna brzina u numeričkom modelu

Promotrimo sada složeniji oblik valne jednadžbe, gdje kvadrat brzine $\lambda(x,y)$ ovisi o prostornim koordinatama,

$$\nabla(\lambda(x,y)\nabla u) = \frac{\partial^2 u}{\partial t^2}. \quad (3.4)$$

U slučaju kada je λ konstantna gornja se jednadžba svodi običnu valnu jednadžbu. Ako pak nije konstantna, vrijedi

$$\nabla(\lambda(x,y)\nabla u) = \frac{\partial}{\partial x} \left(\frac{\lambda(x,y)\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\lambda(x,y)\partial u}{\partial y} \right)$$

ili diskretno [2]

$$\frac{\partial}{\partial x} \left(\frac{\lambda(x,y)\partial u}{\partial x} \right) \approx \frac{1}{\Delta x} \left(\lambda_{i+1/2,j} \left[\frac{u_{i+1,j}^l - u_{i,j}^l}{\Delta x} \right] - \lambda_{i-1/2,j} \left[\frac{u_{i,j}^l - u_{i-1,j}^l}{\Delta x} \right] \right) \quad (3.5)$$

$$\frac{\partial}{\partial y} \left(\frac{\lambda(x,y)\partial u}{\partial y} \right) \approx \frac{1}{\Delta y} \left(\lambda_{i,j+1/2} \left[\frac{u_{i,j+1}^l - u_{i,j}^l}{\Delta y} \right] - \lambda_{i,j-1/2} \left[\frac{u_{i,j}^l - u_{i,j-1}^l}{\Delta y} \right] \right) \quad (3.6)$$

Kombiniranjem ovih izraza neposredno i istim postupkom kao u prethodnom slučaju dobije se eksplicitna shema za računanje $u_{i,j}^{l+1}, \forall l > 0$.

Međutim, osim što moramo voditi računa da nam diskretizirani prostor bude dovoljno gust da vjerno oslikava ponašanje kontinuuma, također moramo voditi računa o odnosima prostornih i vremenskih koraka. Naime, stabilnosti ovog algoritma dana je uvjetom [2]

$$\Delta t \leq \frac{1}{\sqrt{\max \lambda(x,y)}} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)^{-\frac{1}{2}}. \quad (3.7)$$

3.3 Model tsunamija

Raniji izvod bio je općenit i može se primijeniti na sve valove. Sada ćemo uvesti ponašanje tsunamija kroz funkciju $\lambda = g H(x, y)$ koja predstavlja kvadrat brzine te kroz početne i rubne uvjete.

Pretpostavimo da su valne duljine mnogo veće od dubine, tj. $\tilde{\lambda}/H > 20$, što nas vodi na formulu površinskog vala (2.1). Nećemo analizirati neko određeno stvarno morsko dno pa ćemo zanemariti zakrivljenost Zemljine površine i Coriolisov efekt jer puno složeniji proračun [7] ne bi značajno doprinio osnovnom cilju ovog rada, tj. demonstraciji porasta amplitude vala. Obalu ćemo predočiti sa kvadratnom mrežom, a za rubne uvjete obale uzeti

$$\frac{\partial u}{\partial n} = \nabla u \cdot \mathbf{n} = 0 \quad (3.8)$$

$\partial u / \partial n$ je derivacija u odnosu normale na granicu. U ovom modelu granicu ćemo zbog jednostavnosti postaviti paralelno s y-osi.

Početne uvjete ćemo postaviti kao nagli pomak velikog volumena vode, uzrokovan potresom. Pretpostavit ćemo da se proces dogodi brzo u odnosu na period valova, što nam omogućuje da izdizanje morskog dna aproksimirano početnim oblikom površine vode kojeg zadamo sa funkcijom koja ima Gaussov oblik

$$\begin{aligned} u(x, y, 0) &= H(x, y) & x, y &\in (0, L) \\ \partial u / \partial t |_{t=0} &= 0 & x, y &\in (0, L) \\ H(x, y) &= A_0 \exp\left(-\left[\frac{x - x_c}{\sigma_x}\right]^2 - \left[\frac{y - y_c}{\sigma_y}\right]^2\right). \end{aligned}$$

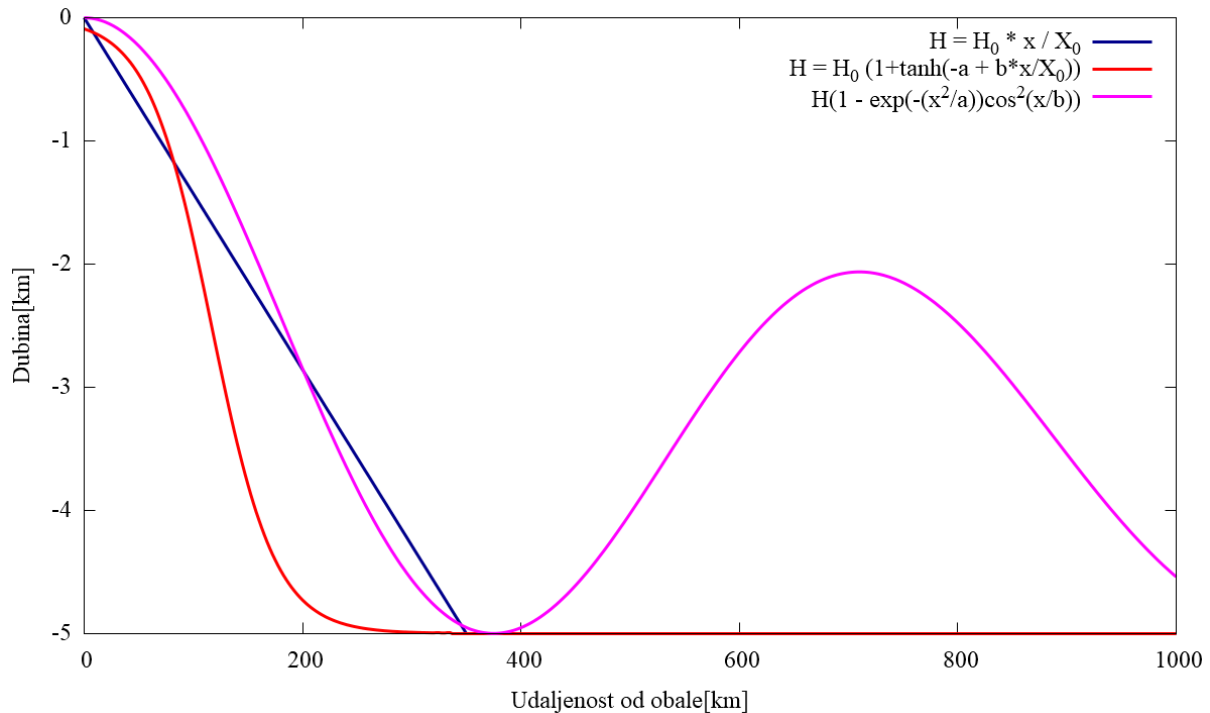
A_0 je početna amplituda površine vode i iznosi tipično 1 – 2 metra. Varijable x_c i y_c su koordinate epicentra, dok su σ_x i σ_y standardne devijacije u gausijanu površine na koju je utjecao potres. Tipične su vrijednosti [2] $\sigma_x = 50$ km, $\sigma_y = 200$ km. Alternativno se svi rubovi mogu tretirati prema (3.8) te pri prolasku vala treba spriječiti refleksiju učvršćivanjem ruba.

Za model morskog dna uzimamo maksimalnu dubinu $H_0 = 5000$ m i nekoliko različitih oblika morskog dna:

- 1) linearni nagib $f_1(x) \sim x$ do dubine H_0 , odnosno udaljenosti X_0 od obale;
- 2) $f_2(x) \sim \tanh\left(a_1 + a_2 \frac{x}{X_0}\right) / a_3$;
- 3) $f_3(x) \sim 1 - \exp(-(x/a_4)^2) \cos^2(x * a_4)$;

gdje su a_i neke konstante. U odabranim primjerima pretpostavljena je ovisnost dubine samo o x-koordinati, odnosno da su izobate paralelne sa y-osi. Pri diskretizaciji ponašanja $\lambda = g H(x, y) = g H(x)$, $H_0 = 5000$ m, potrebno nam je $(2n+1)$ točaka zbog (3.5) i (3.6), dok je

X_0 je koordinata (udaljenost od obale) nakon koje se dubina smanjuje kako idemo prema obali u 1. i 2. slučaju. 3. slučaj, koji se sastoji od eksponencijalnog i oscilatornog člana bi mogao predstavljati morsko dno na kojem se pojavljuje podvodni greben. Usporedba morskog dna za spomenuta 3 slučaja dana je na slici 1, a korišteni parametri a_i mogu se pročitati iz primjerka koda danog u dodatku A.



Slika 1: Grafička usporedba dubine mora za 3 različita oblika morskog dna: 1) linearni rast dubine do neke udaljenosti X_0 nakon čega je konstanta, 2) oblik tangensa hiperbolnog do neke udaljenosti X_0 i 3) simulacija podvodnog grebena

Iterativni postupak određivanja rješenja na osnovu zadanih početnih uvjeta možemo sažeti pseudokodom:

```

diskretizacija prostora (i,j) i vremena t
koordinatni pomak h, vremenski korak dt
alfa = dt * dt / (h * h)
diskretizacija kvadrata brzine lambda za svakih h/2
u1, u2, u3 = elongacije morske razine prethodno, sadašnjem i sljedećem t
u2 prema početnim uvjetima
u1 prema (3.3)
for (t = 1; t <= Nt; t++) {
  for (i = 0; i <= Nxy; i++) {
    for (j = 0; j <= Nxy; j++) {
      rjesenja na rubu
      sprjecavanje nezelenih refleksija na rubu
      rjesenja unutar podrucja
      if (i != 0 && i != Nxy && j != 0 && j != Nxy)
        u3[i][j] = 2 * u2[i][j] - u1[i][j] + alfa *
          ( lambda[2*i+1] * (u2[i+1][j] - u2[i][j])
            - lambda[2*i-1] * (u2[i][j] - u2[i-1][j])
            + lambda[2 * i] * (u2[i][j+1] - u2[i][j])
            - lambda[2 * i] * (u2[i][j] - u2[i][j-1]));
      ispis rjesenja prema želji
      za svaki i, j: u1[i][j] = u2[i][j]; u2[i][j] = u3[i][j];
    }
  }
}

```

4 Rezultati

Prilikom prezentacije konačnih rezultata bitno je obratiti pozornost na diskretizaciju prostora i uvjet stabilnosti (3.7)

$$\Delta t \leq \frac{1}{\sqrt{\max \lambda(x, y)}} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)^{-\frac{1}{2}}.$$

Odabrane su sljedeće vrijednosti

$$L = 1\,000\,000 \text{ m}$$

$$H_0 = 5\,000 \text{ m}$$

$$\Delta x = \Delta y = h$$

$$h = L/N_{xy}.$$

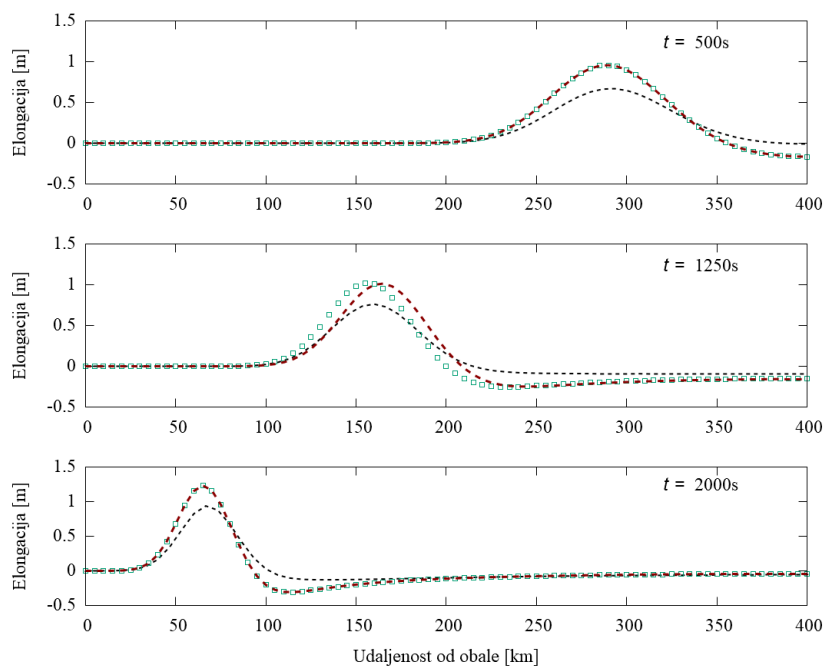
Ako raspišemo (3.7) dobijemo

$$\Delta t \leq \frac{1}{\sqrt{gH_0}} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta x^2} \right)^{-\frac{1}{2}} = \frac{1}{\sqrt{gH_0}} \left(\frac{2}{h^2} \right)^{-\frac{1}{2}} = \frac{h}{\sqrt{2gH_0}} = \frac{L}{N_{xy}\sqrt{2gH_0}}.$$

U slučaju $N_{xy} = 200$, za kojeg su dobiveni rezultati na slikama 3 – 5 vrijedi

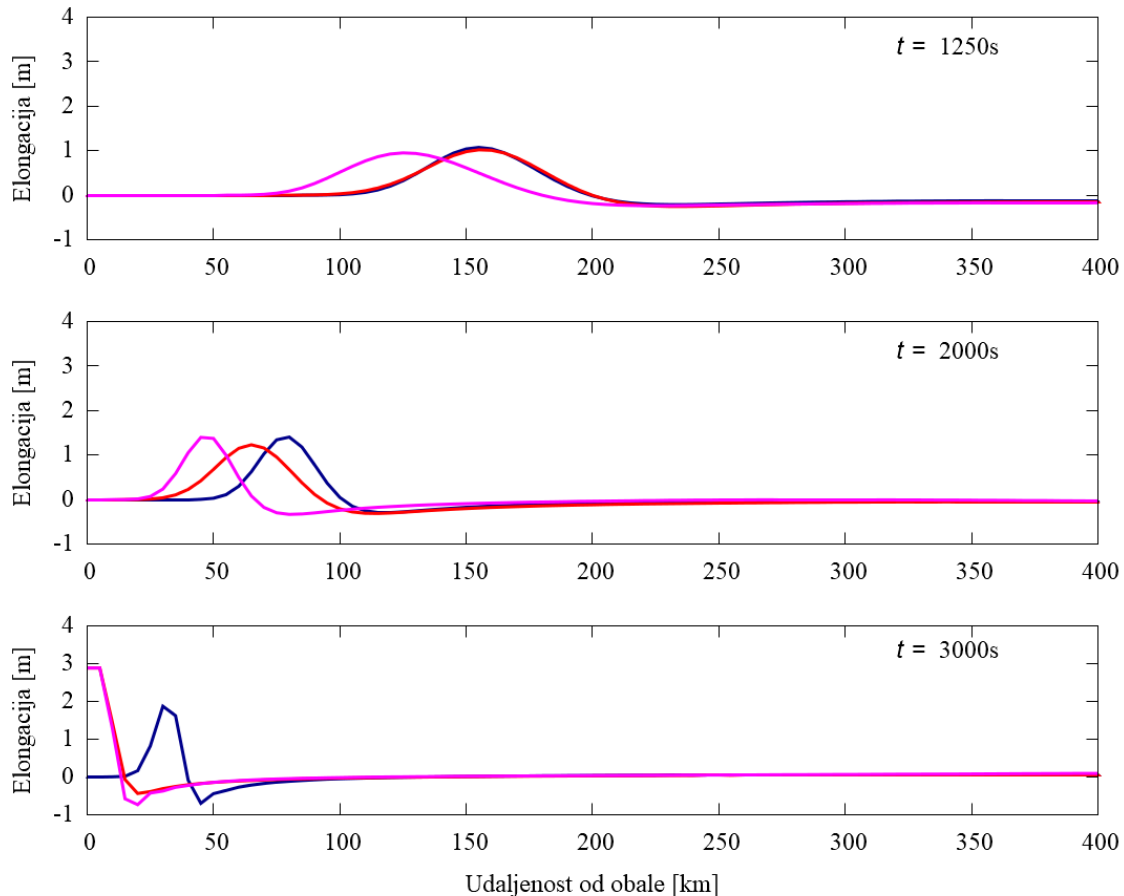
$$\Delta t \leq 15.9637714.$$

Slika 2 prikazuje vremensku evoluciju vala u slučaju dubine 1) za različite vrijednosti diskretizacije prostora korištene pri računanju. Dvije linije, koje se podudaraju, imaju finiji prostorni korak ($N_{xy}=200$) od crne točkaste ($N_{xy}=150$). Iako linije koje se podudaraju imaju različit Δt (0.2 s za crvenu i 0.4 s za zelenu), oba zadovoljavaju relaciju (3.7), dopušten im je finiji h , a time i preciznije računanje.



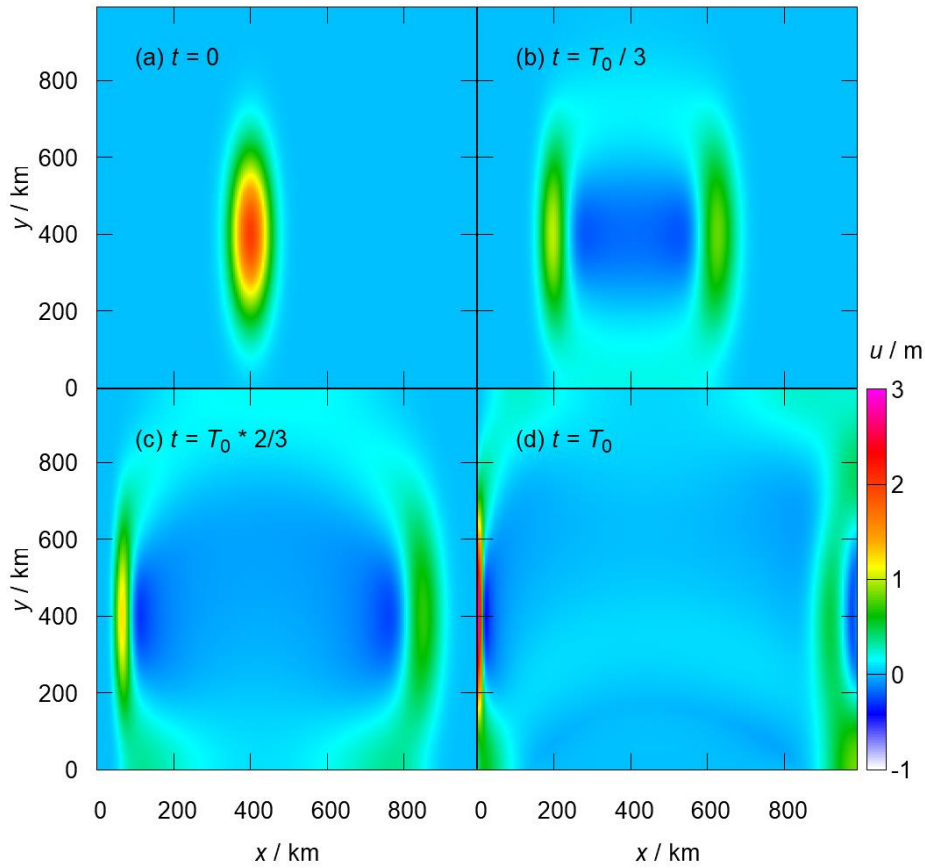
Slika 2: Razina mora prikazana je za različite prostorne korake h i vremenske korake t .

Rezultati promjene razine morske površine dobiveni za 3 slučaja spomenuta u prethodnom poglavlju prikazani su na slici 3. Rezultati se za 3 različita tipa morskog dna očekivano razlikuju i to u smislu da je val koji putuje iznad morskog dna oblika 2) najbrži, dok je val u slučaju 3) najsporiji. Sva tri vala pokazuju očekivan rast amplitude sa smanjenjem dubine.

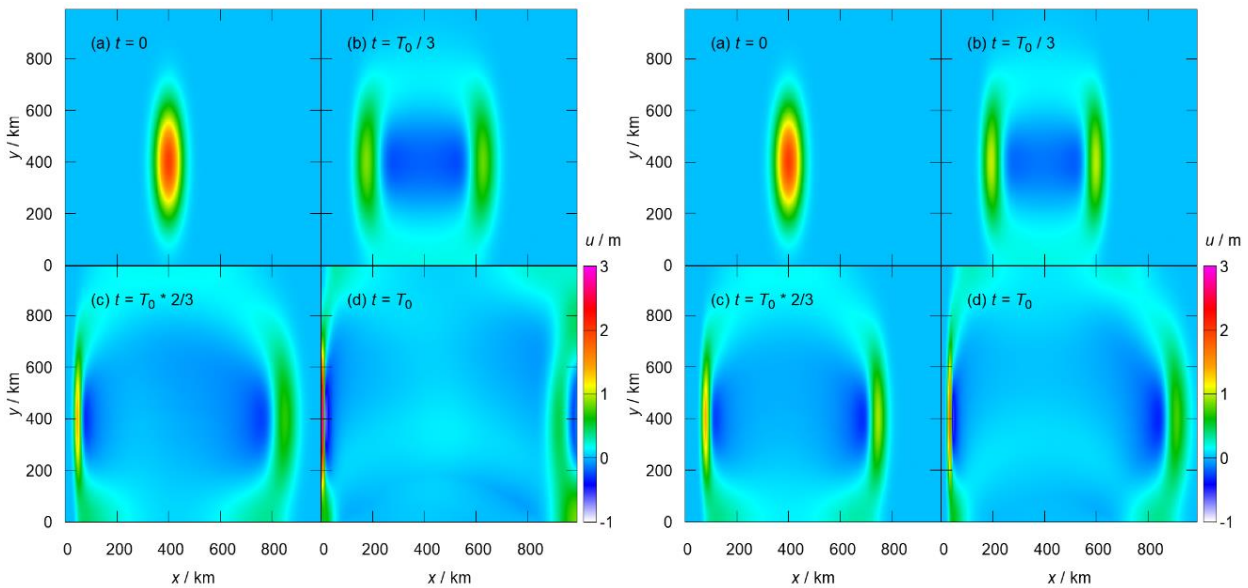


Slika 3: Prikazuje valove koji putuju u lijevo u 3 različita vremenska trenutka. Redom s lijeva na desno su valovi 2), 1) i 3). U 3) prosječna je dubina bila najmanja što se slaže sa izrazom (1.3) za brzinu širenja vala

Slike 4 i 5 prikazuju pogled na xy ravninu (na morsku razinu odozgo) u različitim vremenskim trenucima, mjerenima u vremenu T_0 potrebnom da val dođe do obale. Može se vidjeti kako se val širi prema obali, a time i usporava (kraće valne duljine). Slika 4 prikazuje slučaj 1), dok slika 5 prikazuje: na lijevoj strani slučaj 2), a na desnoj 3). Iako su svima zadani isti početni uvjeti, brzina kretanja nije jednaka što se pripisuje obliku dna, koji je jedina razlika između slučajeva.



Slika 4: Vremenska evolucija za dubinu 1). Val putuje ulijevo na slici. Jasno se vidi smanjenje valne duljine prilikom približavanja obali (lijevo), a time i brzine.



Slika 5: Vremenska evolucija vala 2) (lijevo) i vala 3) (desno)

5 Zaključak

Numeričko rješavanje valne jednadžbe pruža prilično dobru aproksimaciju analitičkom rješenju. Počevši sa zapisom derivacije u konačno malom koraku, izveden je algoritam za eksplicitno pronalaženje stanja sustava uz zadane početne i rubne uvjete. Nakon toga je valna jednadžba prilagođena slučaju kada brzina vala nije stalna. Izraz koji određuje brzinu povezan je sa dubinom vode te je uz tu promjenu ponovno izveden prilagođeni algoritam rješavanja.

Kako dubina nije konstanta u prirodi, ovaj rad je uzeo u obzir 3 različita oblika morskog dna i na temelju je simulirano ponašanje vala koje je u skladu s očekivanjima: brzina je manja u plićim vodama, dolazi do brzog porasta amplitude kako se val približava kopnu. Jasno je vidljiv i pad brzine kada val prelazi preko grebena u slučaju 3). Iako se iz grafa ne vidi najjasnije, uvidom u numerička rješenja najvišu amplitudu pri kontaktu sa kopnom ima val 3) koji prilazi obali najmanjim nagibom morskog dna.

Pokazano je također da točnost simulacije ovisi o diskretnosti prostora. Dok je za vrlo velike valne duljine, prostorni korak od $h = 1$ km sasvim dobra aproksimacija, kako val dolazi u pliće vode, to se javlja potreba za boljom diskretizacijom prostora. Vremenski korak Δt ima gornju granicu danu sa izrazom (3.7) i nema utjecaj na točnost rješenja koliki ima h dok god je u granici stabilnosti.

Nedostaci ovog modela su u tome što točnost prestaje kada val dođe do obale, koja se ponaša kao čvrsti kraj od kojega se val odbija. Precizniji modeli bi morali uzeti u obzir kako se ponaša masa vode nakon što dođe na kopno. Također, oblik morskog dna u ovome modelu zadan je proizvoljnim funkcijama i ne predočuje stvarno morsko dno. Ova bi se metoda mogla poopćiti korištenjem stvarnih mjerenja dubine interpolacijom podataka iz izobata.

6 Literatura

- [1] Jr F. S. Crawford, *Waves, Berkeley Physics course*, McGraw-Hill book company, SAD, New York 1968.
- [2] Morten Hjorth-Jensen, *Computational Physics, Lecture Notes Fall 2015*, University of Oslo 2015.
- [3] Karin Lehnardt: *50 Interesting Facts about Tsunami*, URL:<https://www.factretriever.com/tsunami-facts> (25.09.2018.).
- [4] Wikipedia, *Tsunami*, URL:<https://en.wikipedia.org/wiki/Tsunami> (25.09.2018.).
- [5] University of Washington, *The Physics of Tsunamis*, URL:<https://earthweb.ess.washington.edu/tsunami/general/physics/characteristics.html> (25.09.2018.).
- [6] R. Nave, *HyperPhysics - Tsunami*, URL:<http://hyperphysics.phy-astr.gsu.edu/hbase/Waves/tsunami.html> (25.09.2018.).
- [7] James T. Kirby, Fengyan Shi, Babak Tehranirad, Jeffrey C. Harris, Stephan T. Grilli, *Dispersive tsunami waves in the ocean: Model equations and sensitivity to dispersion and Coriolis effects*, *Ocean Modelling*, 62, 39 (2013).

A Tipični kod za tsunami-model

```

// sve su velicine dane u SI jedinicama
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <stdbool.h>

#define dt    0.2    // vremenski t korak
#define Nt   50000   // broj koraka u t
#define pisi_t 500   // ispis rjesnja nakon svkih toliko koraka

#define L    1000.0E+3 // kvadratna morska površina L*L
#define Nxy  200     // diskretizacija prostora u Nxy*Nxy tocaka
#define H0   5.0E+3   // max dubina
#define X0   350.0E+3 // udaljenost nakon koje je dubina konstantna

#define A0   2.0     // amplituda izvora (maksimum gausijana)
#define Xc   400.0E+3 // x koordinata izvora vala
#define Yc   400.0E+3 // y koordinata izvora vala
#define sigx 50.0E+3 // standarda devijacija gausijana u x-smjeru
#define sigy 200.0E+3 // standarda devijacija gausijana u y-smjeru
#define g     9.81   // akceleracija slobodnog pada

// razni modeli morskog dna (dubine vode)
// u glavnom kodu potrebno pozvati odgovarajucu funkciju
double dubina1(double x) {
    if (x>X0) return H0;
    else return H0 * x / X0;
}
double dubina2(double x) {
    if (x>X0) return H0;
    else return H0 * (1 + tanh(-2 + 6 * x / X0)) / 2;
}
double dubina3(double x) {
    double a, b;
    a = 1.0E+6;
    b = 1.5E+6;
    return H0 * (1 - exp(-pow(x / a, 2)) * pow(cos(x * 2 * 3.14159265359 / b), 2));
}

int main() {
    FILE *dat;
    FILE *dat2;
    int i, j, t; // brojacji prostornih i vremenskih koraka
    bool pisi = false; // omogucuje upis u datoteku
    double max, alfa, h;

    h = L / Nxy; // prostorni korak
    alfa = dt*dt / (h*h); // konstanta
    max = 0.0; // za provjeru i odredivanje osi u gnuplotu
    dat = fopen("tanh_dt_5.0_Nxy_150.dat", "w"); /// ///PAZI NA IME DATOTEKE/////
    dat2 = fopen("xz_tanh_dt_5.0_Nxy_150.dat", "w");
    printf(" %e < %e ?\n", dt, sqrt(h*h / (2.0*H0*g)));

    // lambda = kvadrat brzine vala
    double lambda[2 * Nxy + 1];
    for (i = 0; i < (2 * Nxy + 1); i++)
        lambda[i] = g*dubina1(i*h / 2.); // ///PAZI NA FUNKCIJU DUBINE///

```

```

// u - elongacija morske razine
double u1[Nxy + 1][Nxy + 1]; // u prethodnom trenutku
double u2[Nxy + 1][Nxy + 1]; // u sadšnjem trenutku
double u3[Nxy + 1][Nxy + 1]; // u sljedećem trenutku
for (i = 0; i <= Nxy; i++)
    for (j = 0; j <= Nxy; j++) {
        u1[i][j] = 0.;
        u2[i][j] = 0.;
        u3[i][j] = 0.;
    }
// pocetni uvjeti, t=0
for (i = 0; i <= Nxy; i++) {
    for (j = 0; j <= Nxy; j++) {
        u2[i][j] = A0*exp(-pow((i*h-Xc)/sigx, 2) - pow((j*h-Yc)/sigy, 2));
        u1[i][j] = u2[i][j];
        fprintf(dat, "%10.2lf\t%10.2lf\t%2.5lf\n", h*i, h*j, u2[i][j]);
        if (j == 100) fprintf(dat2, "%10.2lf\t%2.5lf\n", h*i, u3[i][j]);
    }
    fprintf(dat, "\n");
}
fprintf(dat, "\n");
fprintf(dat2, "\n");
// procjena prije pocetnog trenutka
for (i = 1; i < Nxy; i++)
    for (j = 1; j < Nxy; j++)
        u1[i][j] = u2[i][j] + alfa / 2 *
            ( lambda[2 * i + 1] * (u2[i + 1][j] - u2[i][j])
              - lambda[2 * i - 1] * (u2[i][j] - u2[i - 1][j])
              + lambda[2 * i] * (u2[i][j + 1] - u2[i][j])
              - lambda[2 * i] * (u2[i][j] - u2[i][j - 1]));

// proracun u svim ostalim trenucima
for (t = 1; t <= Nt; t++) {
    if (t % pisi_t == 0) {
        printf("%.2lf%%\t%2.5lf\n", t*100. / Nt, max);
        pisi = true;
    }
    for (i = 0; i <= Nxy; i++) {
        for (j = 0; j <= Nxy; j++) {
            // rjesenja na rubu
            if (i == 0) u3[i][j] = u2[i + 1][j];
            else if (i == Nxy) u3[i][j] = u2[i - 1][j];
            else if (j == 0) u3[i][j] = u2[i][j + 1];
            else if (j == Nxy) u3[i][j] = u2[i][j - 2];
            // sprjecavanje nezelenih refleksija na rubu
            if (u3[i][j] < 0) u3[i][j] = 0;
            // rjesenja unutar podrucja
            if (i != 0 && i != Nxy && j != 0 && j != Nxy)
                u3[i][j] = 2 * u2[i][j] - u1[i][j] + alfa *
                    ( lambda[2*i+1] * (u2[i+1][j] - u2[i][j])
                      - lambda[2*i-1] * (u2[i][j] - u2[i - 1][j])
                      + lambda[2 * i] * (u2[i][j + 1] - u2[i][j])
                      - lambda[2 * i] * (u2[i][j] - u2[i][j - 1]));
            if (pisi) fprintf(dat, "%10.2lf\t%10.2lf\t%2.5lf\n", h*i,
                h*j, u3[i][j]);
            if (pisi && j == 100) fprintf(dat2, "%10.2lf\t%2.5lf\n",
                h*i, u3[i][j]);
            if (u3[i][j] > max) max = u3[i][j];
        }
        if (pisi) fprintf(dat, "\n");
    }
    if (pisi) fprintf(dat2, "\n");
}

```

```
    if (pisi) fprintf(dat, "\n");

    pisi = false;
    for (i = 0; i<=Nxy; i++) {
        for (j = 0; j<=Nxy; j++) {
            u1[i][j] = u2[i][j];
            u2[i][j] = u3[i][j];
        }
    }
}
printf("\n-----%1f-----", max); ///maksimalna amplituda
fclose(dat);
fclose(dat2);
return 1337;
}
```

B Tipični primjer gnuplot-skripte

```
unset multiplot
reset
set encoding utf8
set term pngcairo background "#ffffff" font "Arial,20pt" size 1050,1000
set output 'ecos.png'
set samples 10000
set bmargin 0.
set lmargin 0.
set rmargin 0.
set tmargin 0.

unset key
set size ratio -1
set xrange [0:990]
set yrange [0:990]
set xtics 200.0
set ytics 200.0
Ymin=-1.0

set pm3d map
set palette defined (0.0001 "white", 1.0 "blue", 2.0 "cyan", 2.8 "web-green", \
                    3.7 "yellow", 4.2 "orange", 5.8 "red", 7.0 "magenta")
set cbrange [Ymin:3.0]
set cbtics ("-1" -1.0, "0" 0.0, "1" 1.0, "2" 2.0, "3" 3.0) \
            offset graph -0.02, graph 0.00

set multiplot
set origin 0.035,0.07 # dolje lijevo
set size 0.58,0.58
unset colorbox
set format x
set format y
set xlabel "{/Arial-Italic x} / km"
set ylabel "{/Arial-Italic y} / km"
set label 11 "(c) {/Arial-Italic t = T}_0 * 2/3" front at graph 0.10, graph 0.87
splot [::][:][:] 'ecos.dat' i 220 u ($1/1000.):($2/1000.):($3) w pm3d

set origin 0.448,0.07 # dolje desno
set size 0.58,0.58
set colorbox
set format x
set format y ""
unset ylabel
set label 11 "(d) {/Arial-Italic t = T}_0" front at graph 0.10, graph 0.87
splot [::][:][:] 'ecos.dat' i 330 u ($1/1000.):($2/1000.):($3) w pm3d

# analogno za gore lijevo i gore desno .....

unset multiplot
unset output
reset
set terminal wxt
```