Izvajalec študije:

Inštitut za biologijo

MORSKA BIOLOŠKA

POSTAJA

Fornače 41, Piran 6330

Geofizikalno-ekološki pristop k disperziji odplak piranskega izpusta

Uporabnik/sofinancer:Javno podjetje

OKOLJE Piran d.o.o., Fornače 33, Piran 6330

Nosilec naloge: doc.dr. Vlado Malačič

Sodelavci: dr. Aleksander Vukovič

Študija se navaja kot poročilo o znanstveno-raziskovalni nalogi:

Malačič V., 1998. Geofizikalno-ekološki pristop k disperziji odplak piranskega izpusta. Fazno poročilo 2, Nacionalni inštitut za biologijo, Morska biološka postaja Piran, 43 str.

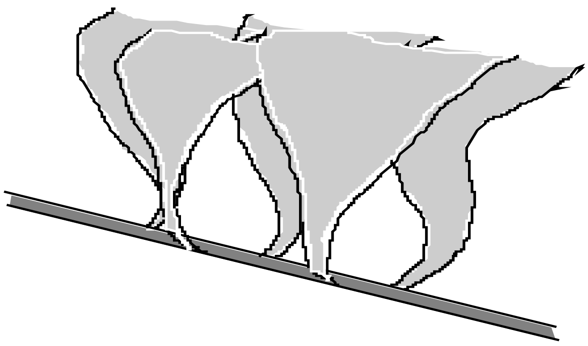
V Piranu, 23. novembra 1998

KAZALO

1. NUMERIČNI MODEL ZAČETNE DILUCIJE

1.1 Uvod v numerični model

V prvem faznem poročilu (Malačič, 1997) so bile opisane zahteve za difuzor odplak in opravljen je bil hidravlični izračun za difuzor novejšega izpusta, ki je bil na čelu modificiran z betonskim zamaškom z odprtino premera 10 cm. V tem poročilu bo opisan naslednji korak pri oceni delovanja difuzorja. Z numeričnim modelom bomo simulirali prvo fazo potovanja odplak iz difuzorja. Pozornost bo usmerjena predvsem na višino dviga in na faktor redčenja odplak. Ta faza se zaključi s formiranjem začetnega madeža odpadne vode. To je tudi edina faza, na katero lahko vplivamo s konstrukcijo difuzorja, hkrati se v njej odvija najizrazitejše redčenje odplak.

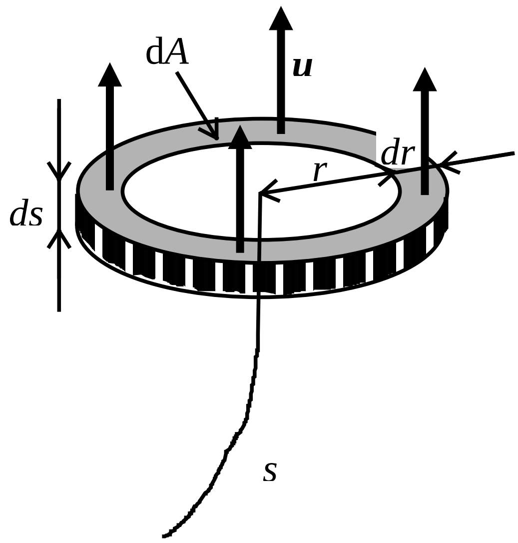


1.2 Osnovne enačbe modela

Za praktične namene so pri oceni redčenja in dviganja odplak še najbolj v rabi inženirske metode, ki bolj ali manj temeljijo na grafično-tabelaričnem deduciranju stopnje redčenja in višine dviga (oz. dolžine "iztočnega jezika") iz nomogramov (npr. Kolar, 1983; Quetin in De Rouville, 1986). Pri tem je potrebno kot vhodne podatke navesti tako podatke, ki so posledica konstrukcije difuzorja (npr. začetna razlika med gostoto okolne tekočine in gostoto efluenta, iztočna hitrost in dimenzije odprtine), kot tudi podatke o okolju (npr. vertikalni gradient gostote morske vode). Iz teh količin se nato tvorijo brezdimenzijska števila, ki imajo bistven vpliv na potek procesa (npr. interno Froudeovo število). Rezultat je nato grafično odčitan.

Ocenili smo, da bo za namene naše študije bolje razviti numerični model na osnovi splošno veljavnih ohranitvenih zakonov, pri čemer pa se naslonimo na spoznanja, dobljena z opazovanjem procesov v laboratoriju ali v naravi, ki so opisana v ustrezni literaturi. Za ta korak smo se odločili zato, da bomo z numeričnim modelom kasneje opazovali začetno fazo redčenja tudi pri kompliciranem vertikalnem profilu gostote v sezonskem obdobju, ko lahko razlikujemo več gostotnih plasti v vodnem stolpu, med katerimi so nekatere (pridnena in površinska mejna plast) gostotno homogene zaradi intenzivnega vertikalnega mešanja, druge pa izredno stratificirane. Takšno strukturo je težko nadomestiti s preprosto linearno vertikalno spremembo gostote vodnega stolpa. Poleg tega smo želeli s simulacijo dobiti podrobnejši vpogled v samo dinamiko začetne faze. Pri konstruiranju numeričnega modela smo se naslonili na delo, ki ga je naredil R. E. Featherstone (1984), katero smo dopolnili s spoznanji opisanimi v delu Fischer in drugi (1979). Model bo simuliral širjenje in redčenje vzgonskega curka, ki izhaja iz ene odprtine (krožnega preseka) v stratificirano mirujoče morje, ki je pogosto tudi mnogo manj turbulentno od curka odplak.

Najprej označimo osnovne količine, ki so integrirane po preseku iztočnega vzgonskega curka (buoyant jet). Naj je dQ (=dV/dt = dAds/dt = u dA) pretok volumna dV (=dAds) tekočine (oz. specifični pretok mase), ki teče s hitrostjo $\boldsymbol{u}$ skozi osnosimetrično valjasto cev polmera r, majhne višine ds in ploščine osnovne ploskve dA (Sl. 1). Cev naj je pravokotna na trajektorijo curka na oddaljenosti s od odprtine difuzorja, merjene vzdolž trajektorije. Podobno definirajmo še pretok mase d$\Psi = \rho$dQ (= ρ dAds /dt = $\rho\, u$ dA) in pretok gibalne količine ρdM = dG/dt (= dm u/dt = ρ dAds u/dt = ρ dAu^2), kjer je dG gibalna količina valjaste cevi curka, ρ pa gostota curka. V enačbah pa bosta nastopala specifični pretok gibalne količine dM in specifični pretok pretok vzgona dB. Slednji nastopa v pretoku vzgona ρdB = $g(\rho_a$ - $\rho)$ dAds/dt = $(\rho_a$ - $\rho)$ dA u , kjer je ρ_a gostota okolne tekočine, g pa težni pospešek. Definirajmo še vzgonsko silo na enoto višine valjaste cevi ρf = dF/ds (= (ρ_a -ρ) g dA), kjer je F sila vzgona na tekočino v omenjeni cevki.



Sl. 1. Skica geometrije elementa vzgonskega curka.

Vse količine skupaj integrirajmo po ceveh različnih polmerov (površin) in enake dolžine d*s*. Tako dobimo pretoke volumna *Q*, gibalne količine *M* in vzgona *B* ter vzgonsko silo na efluent, ki teče skozi rezino vzgonskega curka višine d*s*, ki je simetrično porazdeljena okoli trajektorije, ki rezino prebada in je pravokotna na smer širjenja efluenta:

$$Q = \int_A u dA; \quad \Psi = \int_A \rho u dA; \quad M = \int_A u^2 dA;$$
$$B = g\int_A (\Delta\rho/\rho_0) u dA; \quad f = g\int_A (\Delta\rho/\rho_0) dA, \tag{1.1}$$

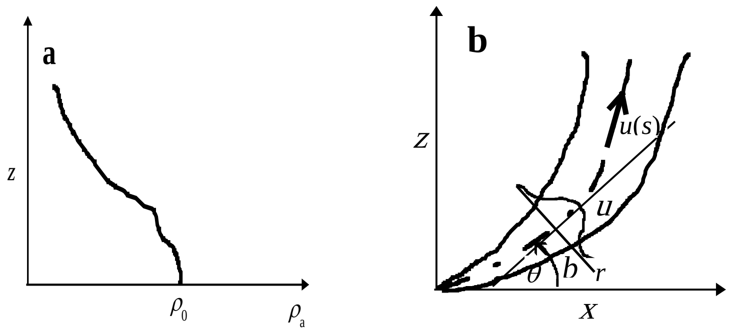
pri čemer je *A* presek vzgonskega curka, *u* velikost hitrosti ***u*** v curku, za katero predpostavimo, da je osno simetrično porazdeljena (odvisna od oddaljenosti *r* od osi curka) in da je usmerjena vzdolž osi širjenja curka. Gostotna razlika $\Delta\rho = \rho_a - \rho$, kjer je ρ_a gostota okolišnje tekočine, ki je odvisna od višine *z* nad difuzorjem, ρ pa gostota efluenta v curku. Naj je gostota okolice pri iztočni odprtini difuzorja $\rho_a(0) = \rho_0$ (Sl. 2). Pretok gibalne količine *M* se vzdolž trajektorije curka spreminja predvsem na račun kvadrata hitrosti u^2, medtem ko se gostota v curku ρ pri spremembi višine za pribl. 20 m zagotovo spremeni za manj kot 3 %. Zato smo $\Delta\rho/\rho$ v izrazih za *B* in *f* v (1.1) smiselno aproksimirali z $\Delta\rho/\rho_0$. Za *x*-komponento gibalne količine v horizontalni smeri in *z*-komponento gibalne količine v vertikalni smeri pa velja ohranitveni zakon:

$$\frac{\mathrm{d}}{\mathrm{d}s}\left(M\cos\theta\right) = 0; \quad \frac{\mathrm{d}}{\mathrm{d}s}\left(M\sin\theta\right) = f, \tag{1.2}$$

kjer je θ kot med tangento na trajektorijo in x-osjo (Sl. 2). Zapisati moramo še proces, ki bistveno vpliva na redčenje vzgonskega curka. Gre za t.im. vnos ali zajem ('entrainment') okolne tekočine v curek, s čimer se spremeni pretok volumna vzdolž trajektorije curka:

$$\frac{\mathrm{d}Q}{\mathrm{d}s} = E = 2\pi b\alpha\, u(s). \tag{1.3}$$

Spremembo pretoka na enoto dolžine curka E smo zapisali kot sorazmerno produktu med obsegom curka pri polmeru b, pri katerem pade hitrost curka na 1/e vrednosti na osi curka, in hitrostjo $u(s)$ na osi curka (Fischer in drugi, 1979). Sorazmernostni koeficient vnosa α je eksperimentalno (numerično) določen, o njem bo še tekla beseda.

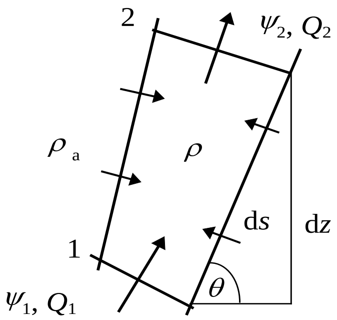


Sl. 2.a) Skica vertikalnega profila gostote vodnega stolpa ρ_a, ki ima pri difuzorju (z = 0) vrednost ρ_0. b) Skica vzgonskega curka, ki izhaja iz odprtine v izhodišču koordinatnega sistema. Polmer curka, pri katerem upade hitrost (Gaussov profil) na vrednost 1/e, je enak b.

Iz Sl. 3 je razvidno, da se vzdolž dolžine ds curka spremeni pretok volumna za dQ = Eds, medtem ko se spremeni pretok mase za dψ = $E\rho_a$ds. Za Q in ψ upoštevajmo izraza (1.1) pa dobimo zvezi

$$\frac{\mathrm{d}\left(\int \rho_0 u dA\right)}{\mathrm{d}s} = E\rho_0; \quad \frac{\mathrm{d}\left(\int \rho u dA\right)}{\mathrm{d}s} = E\rho_a, \tag{1.4}$$

od tod pa sledi



Sl. 3. Skica majhnega dela vzgonskega curka dolžine ds in gostote ρ, vzdolž katerega se zaradi turbulentnega vnosa spremeni volumski pretok med presekoma 1 in 2 za Eds ter masni pretok za $E\rho_a$ds.

$$\frac{\mathrm{d}}{\mathrm{d}s}\left[\int(\rho-\rho_0)u dA\right]=(\rho_a-\rho_0)E\,. \tag{1.5}$$

Na levi strani (1.5) zapišimo ρ - ρ_0 = ρ_a - ρ_0 + ρ - ρ_a = ρ_a - ρ_0 - $\Delta\rho$, ponovno upoštevajmo (1.1) za Q in B. Zavedamo se, da ρ_0 ni funkcija poti s vzdolž trajektorije. Leva stran (1.5) tako postane: Qdρ_a /ds - (ρ_0/g)dB/ds + (ρ_a - ρ_0)dQ/ds. Upoštevajmo, da se pretok volumna na poti ds spremeni: dQ/ds = E. Tako se (1.5) poenostavi:

$$\frac{\mathrm{d}B}{\mathrm{d}s}=\frac{g}{\rho_0}\frac{\mathrm{d}\rho_a}{\mathrm{d}s}Q\,. \tag{1.6}$$

To pa je enačba za spremembo pretoka vzgona vzdolž vzgonskega curka, ki skupaj z (1.2) in (1.3) tvori sistem enačb za pretoke Q, M in B ter naklonski kot θ. Sedaj pa predpostavimo, da se hitrost u lahko zapiše kot produkt amplitude $u(s)$ in brezdimenzijske funkcije ω, ki podaja radialno odvisnost hitrosti $u(s, r) = u(s)$ $\omega(r/b(s))$, kjer je kvocient $r/b(s)$ funkcija tako r kot s. Podobno zapišemo gostotno razliko (ρ_a - ρ) = $\Delta\rho$ = $\Delta\rho(s)\omega(r/\lambda b)$, kjer pa je $1 < \lambda < 2$. Ker je predpostavljena funkcija za prečno odvisnost ω v obeh primerih ista, pogoj $\lambda > 1$ pomeni, da bo prečni profil gostote (gostotne razlike) širši od profila hitrosti, kar je bilo eksperimentalno ugotovljeno (Fischer in drugi, 1979). Končno predpostavimo, da je prečni profil Gaussov, $\omega(r/\lambda b)$ = $e^{-(r/\lambda b)^2}$, pa zapišemo

$$u(s,r)=u(s)e^{-r^2/b^2};\quad \Delta\rho(s,r)=\Delta\rho(s)e^{-r^2/\lambda^2 b^2}\,. \tag{1.7}$$

Nastavka (1.7) vstavimo v izraze za Q, M in B v (1.1), kjer za dA vzamemo ploščino kolobarja $2\pi r$dr in integriramo po polmeru od 0 do ∞. Tako dobimo

$$Q = \pi u(s) b^2(s); \quad M = \frac{\pi u^2(s) b^2(s)}{2}; \quad B = \frac{\pi g \lambda^2 \Delta\rho(s) u(s) b^2(s)}{\rho_0 \left(1 + \lambda^2\right)} \tag{1.8}$$

in za silo f

$$f = \frac{\pi \lambda^2 g \Delta\rho(s)}{\rho_0} b^2(s). \tag{1.9}$$

Integralne količine Q, M in B v (1.8) vstavimo skupaj z (1.9) v sistem enačb (1.2), (1.3) in (1.6) in dobimo:

$$\frac{\mathrm{d}\left(ub^2\right)}{\mathrm{d}s} = 2\alpha bu, \tag{1.10}$$

$$\frac{\mathrm{d}\left(u^2 b^2 \cos\theta\right)}{\mathrm{d}s} = 0, \tag{1.11}$$

$$\frac{\mathrm{d}\left(u^2 b^2 \sin\theta\right)}{\mathrm{d}s} = \frac{2 g \Delta\rho}{\rho_0} \lambda^2 b^2, \tag{1.12}$$

$$\frac{\mathrm{d}\left(\Delta\rho\, ub^2\right)}{\mathrm{d}s} = \frac{\left(1 + \lambda^2\right)}{\lambda^2} ub^2 \frac{\mathrm{d}\rho_a}{\mathrm{d}s}. \tag{1.13}$$

Sistem enačb (1.10) - (1.13) moramo prevesti v sistem enačb za razvoj spremenljivk u, b, θ in $\Delta\rho$ (=ρ_a - ρ, kjer je ρ gostota na osi curka), ki so funkcije poti s, ki jo opravi element tekočine na osi curka med dviganjem v stratificirano tekočino. Zato izraz v oklepaju (1.11) odvajamo po s kot produkt $u\cos\theta$ ter ub^2, kjer za slednjega uporabimo zvezo (1.10). Tako dobimo

$$\frac{\mathrm{d}\theta}{\mathrm{d}s} = \frac{\cos\theta}{u\sin\theta} \frac{\mathrm{d}u}{\mathrm{d}s} + \frac{2\alpha\cos\theta}{b\sin\theta}. \tag{1.14}$$

Podobno naredimo z izrazom v oklepaju zveze (1.12)

$$ub^2\left[\sin\theta \frac{\mathrm{d}u}{\mathrm{d}s} + u\cos\theta \frac{\mathrm{d}\theta}{\mathrm{d}s}\right] + 2bu^2\alpha\sin\theta = \frac{2g\Delta\rho}{\rho_0}\lambda^2 b^2. \tag{1.15}$$

V (1.15) vstavimo $\mathrm{d}\theta/\mathrm{d}s$ iz (1.14), množimo s $\sin\theta/ub^2$ in dobimo prvo iskano enačbo

$$\frac{\mathrm{d}u}{\mathrm{d}s} = \frac{2g\lambda^2\Delta\rho}{\rho_0 u}\sin\theta - \frac{2\alpha u}{b}. \tag{1.16}$$

Sedaj pa v (1.10) nadomestimo $\mathrm{d}u/\mathrm{d}s$ z desno stranjo (1.16) in dobimo drugo enačbo

$$\frac{\mathrm{d}b}{\mathrm{d}s} = 2\alpha - \frac{g\lambda^2 \Delta\rho b}{\rho_0 u^2}\sin\theta . \qquad (1.17)$$

Nato še enkrat izkoristimo du/ds iz (1.16) in vstavimo v (1.14). Tako pridobimo tretjo enačbo

$$\frac{\mathrm{d}\theta}{\mathrm{d}s} = \frac{2g\lambda^2 \Delta\rho}{\rho_0 u^2}\cos\theta . \qquad (1.18)$$

Poslednjo enačbo za $\Delta\rho$ izpeljemo iz (1.13), kjer oklepaj na levi odvajamo kot produkt $\Delta\rho$ in ub^2. Za odvod d(ub^2)/ds uporabimo (1.10). Dodatno upoštevajmo, da gostota okolne tekočine ni neposredno funkcija opravljene poti s, ki jo opravi delec vzgonskega curka, ampak le višine z. Ker velja ds sinθ = dz (Sl. 3), nadomestimo dρ_a/ds na desni strani (1.13) s sinθ dρ_a/dz. Četrta enačba tako postane

$$\frac{\mathrm{d}\Delta\rho}{\mathrm{d}s} = \frac{\left(1+\lambda^2\right)}{\lambda^2}\frac{d\rho_a}{dz}\sin\theta - \frac{2\alpha\Delta\rho}{b} . \qquad (1.19)$$

1.3 Spremljajoče enačbe, parametri in začetni pogoji modela

S štirimi enačbami (1.16)-(1.19) smo zapisali razvoj štirih spremenljivk med dviganjem efluenta. Te so: hitrost u na osi curka, polmer b, pri katerem pade hitrost na 1/e od tiste v sredini, naklonski kot θ in gostotna razlika $\Delta\rho$ med gostoto na osi curka in gostoto morske vode na enaki višini, kot se nahaja središče namišljene rezine tekočine. Ta se nahaja na mestu curka do katerega je prispel majhen element tekočine na osi curka s trenutno hitrostjo u, ki je opravil pot s od odprtine. Tanka rezina tekočine, za katero smo zapisali pretoke volumna, gibalne količine in vzgona, ima debelino (višino) ds, njena os pa oklepa kot θ z vodoravno osjo x. Dodajmo še enačbi za koordinate središča vzgonskega elementa (x,z):

$$\frac{\mathrm{d}x}{\mathrm{d}s} = \cos\theta; \quad \frac{\mathrm{d}z}{\mathrm{d}s} = \sin\theta . \qquad (1.20)$$

Pri tem se zavedamo, da sta koordinati x in z izpeljani iz odvisne spremenljivke θ in neodvisne spremenljivke s. Zato sta enačbi (1.20) le pridruženi k sistemu štirih enačb (1.16)-(1.19).

Poleg koordinate središča rezine efluenta pa želimo sproti opazovati tudi faktor redčenja odplak. Naj je dm_c masa tiste snovi v elementu, ki ni bila vnešena iz okolice, temveč je iztekla v času dt iz odprtine. Tedaj skozi zgornjo ploskev rezine efluenta odteče ravno toliko mase snovi (dm_c), kolikor jo priteče v rezino skozi spodnjo ploskev. Takšno sklepanje je na mestu za tiste snovi v efluentu, katerih koncentracija v okolnem morju je mnogo manjša. Podobno kot pri izpeljavi za pretoke volumna, mase in vzgona, tudi tukaj najprej zapišemo pretok mase snovi skozi valjasto cev $\mathrm{d}\psi_c = \mathrm{d}m_c/\mathrm{d}t = 2\pi C r \mathrm{d}r \mathrm{d}s/\mathrm{d}t = 2\pi C u \mathrm{d}r$, kjer je $C = \mathrm{d}m_c/\mathrm{d}V$ koncentracija snovi. Seveda ponovno predpostavimo Gaussov profil hitrosti u kot funkcijo r z osrednjo vrednostjo $u(s)$, za koncentracijo C pa predpostavimo enako odvisnost od r in s, kot smo jo za $\Delta\rho$, oboje najdemo v (1.7). Z integracijo po vseh ceveh polmerov od 0 do ∞ dobimo masni pretok polutanta skozi namišljeno rezino, ki je pravokotna na os izliva:

$$\Psi_c = \pi \frac{\lambda^2 C(s) u(s) b^2(s)}{1+\lambda^2}. \tag{1.21}$$

Ker pa je pot s, ki jo opravi element tekočine na osi vzgonskega curka, poljubna, se vzdolž dviganja ohranja produkt

$$C(s) u(s) b^2(s) = C_0^* u_0 b_0^2, \tag{1.22}$$

kjer $C_0^* = C(s_0)$ ni začetna koncentracija polutanta neposredno pri iztoku, pač pa koncentracija malo dlje od odprtine, na razdalji $s_0 \cong 6{,}2\ D$, kjer je D premer odprtine (Featherstone, 1984). Na tej oddaljenosti od odprtine je že prisoten t.im. režim 'vzpostavljenega toka' (ZEF - 'zone of established flow). Od odprtine do območja ZEF ($s < 6\ D$) pa je t.im. področje 'vzpostavljanja toka' (ZFE - 'zone of flow establishment'), v katerem je hitrost sredice curka praktično neodvisna od oddaljenosti od odprtine (Fischer in drugi, 1979). Turbulenca v resnici doseže ravnovesje na malo večji oddaljenosti od odprtine (pribl. 10 D), vendar pa ne bo storjena velika napaka, če bomo v modelu pričeli opazovati razvoj vzgonskega curka na oddaljenosti 6,2 D.

Preostane nam, da količine, ki v (1.22) nastopajo šele na razdalji s_0 od odprtine, povežemo s količinami neposredno pri odprtini. Še prej ocenimo Reynoldsovo število Re pri odprtini. Ker je $Re = u_0 D/\nu$, kjer je $\nu \cong 10^{-6}$ m^2/s kinematična viskoznost, premer odprtine D = 0,1 m, bo Re med 0,5 - 2,5 10^5 pri začetnih hitrostih iztekanja u_0 med 0,5 in 2,5 m/s. To pomeni, da je iztekanje zagotovo turbulentno, zato lahko mirno privzamemo, da je koncentracija polutanta pri iztoku iz odprtine praktično enaka po vsem preseku odprtine. Zato velja za pretok polutanta pri odprtini:

$$\psi_{\mathrm{c}} = \frac{\mathrm{d}m_{\mathrm{c}}}{\mathrm{d}V}\frac{\mathrm{d}V}{\mathrm{d}t} = C_0 Q_0 = C_0 \frac{\pi\, u_0 D^2}{4}\,, \tag{1.23}$$

kjer smo že upoštevali, da je hitrost sredice curka pri odprtini (znotraj območja vzpostavljanja toka -ZFE) praktično enaka tisti na oddaljenosti s_0 od odprtine: $u(s=0) = u(s_0) = u_0$. Znotraj območja ZFE zajemanje okolne vode še ne pride do izraza, zato se v tem območju ohranja (specifični) pretok gibalne količine, ki ga opredelimo s količinami pri odprtini: $M = M_0 = \pi\, u_0^2 D^2/4 = M(s_0)$. Na oddaljenosti s_0 od odprtine že velja območje vzpostavljenega toka in zato lahko za $M(s_0)$ uporabimo izraz (1.8), kjer s nadomestimo z s_0. Ponovno upoštevajmo, da je $u_0 = u(s_0)$ in dobimo polmer b_0 pri $s = s_0$ izražen s premerom odprtine:

$$b_0 = D\big/\sqrt{2}\,. \tag{1.24}$$

Pri oddaljenosti $s_0 = 6{,}2\ D$ sta profila hitrosti in koncentracije določena po (1.7) in je zato masni pretok polutanta ψ_{c} določen z (1.21), kjer pa je $s = s_0$. Premer $D = 0{,}1$ m je enak za vse odprtine na starem in novem difuzorju. Ker se na poti od odprtine do s_0 nič polutanta ne izgubi, niti pridobi, lahko enačimo pretok (1.21) s tistim pri odprtini po (1.23) in upoštevamo začetni polmer curka po (1.24). Tako dobimo

$$C_0^* = \frac{C_0\left(1+\lambda^2\right)}{2\lambda^2}\,, \tag{1.25}$$

kjer je $C_0^* = C(s_0)$, $C_0 = C(0)$, λ pa je konstanta, ki jo bomo določili s kalibracijo modela (Fischer in drugi, 1979: $\lambda = 1{,}2$; Featherstone, 1984: $\lambda = 1{,}16$). V izraz za ohranitev pretoka mase polutanta (1.22) vstavimo b_0 iz (1.24) in C_0^* iz (1.25) in zapišemo izraz za redčenje *sredice* vzgonskega curka podobno kot sta to storila Fan in Brooks (1966)

$$S(s) = \frac{C_0}{C(s)} = \frac{4\lambda^2 u(s) b^2(s)}{\left(1+\lambda^2\right) u_0 D_0^2}\,, \tag{1.26}$$

vendar pa v števcu nastopa koncentracija pri odprtini C_0 (enaka koncentraciji v cevi izpusta) namesto koncentracije C_0^* pri začetku območja vzpostavljenega toka curka. Izraz (1.26) velja za poljuben polutant v odplaki, ki ga praktično ni v okolni tekočini. Faktor, ki dodatno nastopa v (1.26) je $2\lambda^2/(1+\lambda^2) = 1{,}15$ (oz. 1,18), če je $\lambda = 1{,}16$ (oz. 1,2), kar pomeni dodatno povečanje faktorja dilucije za 15 - 18 %.

Povzemimo začetne vrednosti količin, ki neposredno nastopajo v sistemu enačb, katerega bomo reševali z numeričnim modelom:

$$s_0 = 6{,}2\ D, \tag{1.27}$$

$$b_0 = 2^{-1/2}\ D, \tag{1.28}$$

$$u_0 = 4Q_0/\pi D^2, \tag{1.29}$$

$$\theta_0 = \theta(s{=}0), \tag{1.30}$$

$$(\Delta\rho)_0 = (1+\lambda^2)(\rho_0 - (\rho)_0)/2\lambda^2. \tag{1.31}$$

V (1.30) smo upoštevali, da vzgonska sila v območju vzpostavljanja toka (ZFE) zagotovo še ne spremeni smeri širjenja curka in je ta na mestu $s = s_0$ enaka tisti pri odprtini, kjer je $s = 0$. Gostotno razliko $(\Delta\rho)_0$ pri $s = s_0$ v (1.31) smo izrazili z razliko gostot $(\rho_0 - (\rho)_0)$, kjer je $\rho_0 = \rho_a(s = 0)$, $(\rho)_0$ pa je začetna gostota odplak pri odprtini. Pri tem smo se naslonili na (1.25). Zaradi enostavnosti smo postavili:

$$(\rho)_0 = 1000{,}0\ \mathrm{kg/m^3}. \tag{1.32}$$

Poleg tega pa smo opredelili tudi začetni koordinati curka:

$$x_0 = s_0\cos\theta_0; \qquad z_0 = s_0\sin\theta_0, \tag{1.33}$$

ki ju sproti izračunavamo skupaj s faktorjem redčenja. Slednji je na začetku poti ($s = s_0$) po (1.26):

$$S_0 = 2\lambda^2/(1+\lambda^2), \tag{1.34}$$

kar je med 1,15 in 1,18, pač odvisno od λ.

Na koncu je potrebna še beseda o drugem parametru - parametru zajemanja okolne vode α, ki nastopa v (1.3). Fischer in drugi (1979) navajajo dve možnosti. Po eni strani je lahko α = konst. vzdolž dviganja elementa v curku. Tedaj razlikujemo režim curka ("jet"), v katerem je $\alpha = \alpha_j$, od režima izliva ("plume"), ko je $\alpha = \alpha_p$. Vrednost konstant je:

$$\alpha_j = 0{,}0535 \pm 0{,}0025; \qquad \alpha_p = 0{,}0833 \pm 0{,}0042. \tag{1.35}$$

Seveda je potrebno oceniti, v katerem režimu se curek nahaja. Pričakujemo, da se bo vzgonski curek v bližini odprtine, ko ima še veliko hitrost, obnašal predvsem kot curek, kjer je pomemben pretok gibalne količine. V tem območju se curek ne ukrivi in obrži začetni nagib θ_0. Dlje od odprtine pa se vede kot vzgonski curek ali vzgonski izliv, kjer igra glavno vlogo pretok vzgona. Zapišimo merilo za oddaljenost, pri kateri že prevladuje drugi (vzgonski) režim zaradi delovanja vzgonske sile (Fischer in drugi, 1979) in pri kateri se vzgonski curek zasuka navzgor, če že prej ni bil vertikalno usmerjen:

$$l_M = \frac{M_0^{3/4}}{B_0^{1/2}} = \left(\frac{\pi}{4}\right)^{1/4} \frac{u_0}{\sqrt{g\Delta\rho/(D\rho_0)}}, \tag{1.36}$$

kjer je $M_0 = u_0^2\ (\pi D^2)/4$ začetni pretok gibalne količine, $B_0 = Q_0 g\Delta\rho/\rho_0$ pa začetni pretok vzgona, v katerem nastopa začetni volumski pretok $Q_0 = u_0\ (\pi D^2)/4$. Na oddaljenostih $s > l_M$ od odprtine postane pretok vzgona pomembnejši od pretoka gibalne količine curka. Naredimo oceno območja vrednosti za l_M. V globini več kot 20 m (Mavrič V., 1979: globina difuzorja je 20,7 m) je vrednost gostote ρ_0 morske vode v Tržaškem zalivu prav gotovo višja od 1020 kg/m^3 in nižja od 1028 kg/m^3, kar izhaja iz sezonskih nihanj temperatur in nepredvidenih, relativno majhnih fluktuacij slanosti (Malačič, 1991). Za začetno gostoto odplak $(\rho)_0$ privzamemo po (1.32) 1000 kg/m^3 in tako ocenimo, da je $\Delta\rho/\rho_0$ med 0,02 in 0,03. Hitrost iztekanja u_0 ocenimo (Malačič, 1997), da je med 0,5 m/s do 2,5 m/s. Tako iz (1.36) sledi ocena

$$(l_M)_{min} = 0{,}3\ \text{m}; \qquad (l_M)_{max} = 1{,}7\ \text{m}. \tag{1.37}$$

V večini primerov dviganja vzgonskega curka, kjer je dvig okoli 10 m, torej prevladuje režim vzgonskega curka oz. izliva. Zato bomo v primeru, ko bi se naj α vzdolž dviganja vzgonskega elementa ne spreminjal, privzeli, da je $\alpha = \alpha_p$ po (1.35).

V modelu pa smo upoštevali tudi primer, ko α ni konstanten, ampak funkcija Richardsonovega (Ri) števila odprtine (Fischer in drugi, 1979):

$$Ri = \frac{QB^{1/2}}{M^{5/4}} = \sqrt{\frac{4\sqrt{2\pi}\lambda^2}{\left(1+\lambda^2\right)}\left(\frac{gb\Delta\rho}{\rho_0 u^2}\right)}. \tag{1.38}$$

Blizu odprtine difuzorja nastopata pretoka Q_0 in M_0. Vpeljimo značilno razdaljo l_Q (v neposredni bližini odprtine), znotraj katere prevladuje vpliv specifičnega pretoka volumna Q_0:

$$l_Q = \frac{Q_0}{M_0^{1/2}}. \tag{1.39}$$

Med razdaljama l_Q in l_M pa prevladuje specifični pretok gibalne količine M_0. Ri število odprtine je pri odprtini definirano kot: $Ri_0 = l_Q/l_M = Q_0 B_0^{1/2}/M_0^{5/4}$. Iz (1.39) sledi za krožno odprtino, da je $l_Q = \pi^{1/2}D/2 = 0{,}89D$, kar je manj kot 9 cm v našem primeru. Tako smo dobili, da je l_Q več desetkrat manjša od l_M. Desno stran (1.38) smo dobili z upoštevanjem pretokov Q, M in B po (1.8). Funkcija, ki podaja odvisnost α od Ri pa je

$$\alpha = \alpha_{\mathrm{j}} \exp\left[\ln\left(\frac{\alpha_{\mathrm{p}}}{\alpha_{\mathrm{j}}}\right)\left(\frac{Ri}{Ri_{\mathrm{p}}}\right)^2\right], \tag{1.40}$$

kjer je Ri_{p} = 0,557 konstanta, in sicer Richardsonovo število za popoln izliv, v katerem je pomemben le pretok vzgona, brez prehodnih območij v katerih sta pomembna pretok gibalne količine in pretok volumna.

2. NUMERIČNI PROGRAM

2.1 Namen programa

Namen numeričnega programa je reševanje sistema štirih navadnih diferencialnih enačb prvega reda (1.16)-(1.19) skupaj s spremljajočima enačbama (1.20) in (1.26). Na tem mestu jih ponovno zapišimo:

$$\frac{\mathrm{d}u}{\mathrm{d}s} = \frac{2g\lambda^2 \Delta\rho}{\rho_0 u}\sin\theta - \frac{2\alpha u}{b}, \tag{2.1}$$

$$\frac{\mathrm{d}b}{\mathrm{d}s} = 2\alpha - \frac{g\lambda^2 \Delta\rho b}{\rho_0 u^2}\sin\theta, \tag{2.2}$$

$$\frac{\mathrm{d}\theta}{\mathrm{d}s} = \frac{2g\lambda^2 \Delta\rho}{\rho_0 u^2}\cos\theta, \tag{2.3}$$

$$\frac{\mathrm{d}\Delta\rho}{\mathrm{d}s} = \frac{\left(1+\lambda^2\right)}{\lambda^2}\frac{d\rho_a}{dz}\sin\theta - \frac{2\alpha\Delta\rho}{b}, \tag{2.4}$$

pri čemer so začetne vrednosti spremenljivk (1.27)-(1.31). Dodani sta še spremljajoči enačbi (1.20) in (1.26) za koordinati trajektorije elementa vzgonskega curka in faktor redčenja:

$$\frac{\mathrm{d}x}{\mathrm{d}s} = \cos\theta; \quad \frac{\mathrm{d}z}{\mathrm{d}s} = \sin\theta, \tag{2.5}$$

$$S(s) = \frac{C_0}{C(s)} = \frac{4\lambda^2 u(s) b^2(s)}{\left(1+\lambda^2\right) u_0 D_0^2}. \tag{2.6}$$

Pri reševanju sistema enačb (2.1)-(2.4) skupaj z začetnimi pogoji (1.27)-(1.31) smo želeli izpolniti dva pogoja:

- Zadrževati lokalno napako (napaka numeričnega koraka pri pomiku Δs) vseh štirih odvisnih spremenljivk znotraj predpisane majhne vrednosti.
- Gradient gostote morske vode $\mathrm{d}\rho_a/\mathrm{d}z$ moramo poznati v poljubni višini nad odprtino. Zaradi prvega pogoja bo pomik Δs spremenljiv iz koraka v korak. Torej ne moremo vnaprej vedeti na kateri višini je potrebno poznati $\mathrm{d}\rho_a/\mathrm{d}z$. Problemu se izognemo z uporabo interpolacijske metode (zlepki ali "splines"), ki izračuna $\mathrm{d}\rho_a/\mathrm{d}z$ na poljubni višini z iz tabele vhodnih podatkov, ki je sestavljena iz parov vrednosti (z_j, $\rho_a[j]$).

Za izpolnitev prvega pogoja smo izbrali zanesljiv numeričen algoritem "Runge-Kutta" s prilagojenim korakom (Press in drugi, 1988). Program zadržuje relativno napako vseh štirih količin u, b, θ in $\Delta\rho$ v posameznem koraku pod majhno vrednostjo ε (= 10^{-10}). V resnici upošteva tudi odvode količin po s, saj je *željena* napaka za i-to količino izražena kot

$$\Delta_n[\mathrm{i}] = \varepsilon\left\{\left|y[\mathrm{i}]\right| + \left|\Delta s\frac{\mathrm{d}y}{\mathrm{d}s}[\mathrm{i}]\right|\right\};\quad \mathrm{i} = 1..4\,, \tag{2.7}$$

kjer je y[i] numerična vrednost ene od štirih količin. *Dejansko* napako metode Δ_s[i] pri metodi Runge-Kuta pa izračunamo tako, da za posamezno enačbo poiščemo rešitev $y(s+2\Delta s)$ na dva načina: Pri prvem načinu se pomaknemo za en korak $2\Delta s$. Pri drugem načinu pa se pomaknemo za dva koraka po Δs. Ker je metoda zapisana do petega reda natančno, razlika med obema rešitvama dá oceno za napako posamezne (i-te) količine, ki je sorazmerna z $(\Delta s)^5$. Ocena koraka Δs je opredeljena z maksimumom absolutnih vrednosti kvocienta med dejansko in željeno napako:

$$\Delta s_n = F\Delta s_s \max\left\{\left|\frac{\Delta_s}{\Delta_n}\right|^{-0,2}[\mathrm{i}]\right\}_{\mathrm{i}=1..4} \qquad \Delta_s \le \Delta_n \ldots\ldots \text{rast } \Delta s$$

$$\Delta s_n = F\Delta s_s \max\left\{\left|\frac{\Delta_s}{\Delta_n}\right|^{-0,25}[\mathrm{i}]\right\}_{\mathrm{i}=1..4} \qquad \Delta_s \le \Delta_n \ldots\ldots \text{upad } \Delta s\,, \tag{2.8}$$

kjer je faktor F = 0,9. Če je max $\{|\Delta_s/\Delta_n|\}_i > 1$, potem je napaka s korakom Δs_n večja od željene in bo korak zmanjšan (razmerje napak z eksponentom 0,25), v nasprotnem primeru pa je korak v redu: Tedaj Δs_n postane Δs_s , rezultate zapišemo in zračunamo novi (večji) korak Δs_n po (2.8), pri čemer bo tokrat upoštevano razmerje napak z eksponentom 0,2.

Za izvedbo drugega pogoja v programu smo zopet uporabili preverjene podprograme (Press in drugi, 1988) za zlepke, ki jih bomo navedli kot 'NUMERICAL RECIPES'. Interpolacijska metoda zlepkov temelji na predhodnem izračunu drugih odvodov $d^2\rho_a/dz^2$ v skoraj vseh višinah z_j, j = 2..N-1, kjer je N število danih parov (z_j, $\rho_a[j]$). Za prvo in poslednjo višino smiselno predpostavimo $d^2\rho_a/dz^2$ [j] = 0, kjer je j =1, N (naravni zlepki). V prvem in poslednjem segmentu se torej gostota linearno spreminja z višino. Predpostavka je smiselna, saj je tako pri dnu, kot na gladini morja prisotna mejna plast, kjer se gostota praktično nič ne spreminja. Poleg tega je korak višine Δz pri gladini in pri dnu običajno majhen (=0,1 m). Tabelo globin in gostot program na začetku prebere iz zunanje datoteke. Metoda kubnih zlepkov temelji na "lepljenju" kubnih polinomov. Z njimi interpoliramo gostoto ρ_a na poljubnih globinah z med tabeliranimi vrednostmi $\rho_a[j]$ na globinah z_j. Kubni polinomi za ρ_a med posameznimi segmenti višin [z_j, z_{j+1}] se ujemajo do drugega odvoda. Kadar je po uspešnem koraku Δs reševanja sistema diferencialnih enačb (2.1)-(2.4) znana nova višina ($z + \Delta z$), kjer je $\Delta z = \Delta s \sin \theta$, podprogram poišče segment [z_j, z_{j+1}], v katerem se ta višina nahaja. Nato s kubnim interpolacijskim polinomom za ρ_a izračuna tudi $d\rho_a/dz$, ki pa se znotraj segmenta spreminja kot kvadratna funkcija višine. Slednjo odvisnost smo dodali k standardni proceduri, ki so jo zapisali Press in drugi (1988).

Tako imamo poleg vrednosti štirih količin, ki so bile določene v prejšnjem koraku, določen tudi parameter okolne tekočine $d\rho_a/dz$ pri znani višini. Če smo se odločili, da se parameter vnosa okolne tekočine (morska voda) α spreminja med dviganjem odplak, potem α izračunamo po (1.40) še preden naredimo novi korak Δs. S tem imamo znane vse količine na desnih straneh sistema enačb (2.1)-(2.4) in lahko preidemo v izračun sprememb neznanih količin v naslednjem koraku.

2.2 Opis programa

Program *SPLINRUN* je napisan v programskem jeziku Pascal. V popisu podprogramov (procedur) bomo zapisali vhodne in izhodne spremenljivke tako, kot so imenovane v deklaraciji (glavi) podprograma in ne kot so (različno) imenovane pri klicanju. Nekatera značilna imena spremenljivk pri klicanju bodo navedena v oklepajih. Z oznako "odvisne" ali "neodvisne" spremenljivke v navednicah (npr. "z" in "ρ_a"), so mišljene tiste spremenljivke, ki so ali niso funkcije globine, prav gotovo pa niso funkcije neodvisne spremenljivke *s*, torej niso ena od štirih iskanih odvisnih spremenljivk. Nekatere od vhodnih spremenljivk so tudi izhodne spremenljivke podprogramov.

Podprogrami v *SPLINRUN* programu so naslednji:

- ReadData - prebere vhodne podatke. Globine (m) spremeni v višine nad difuzorjem.
  Vhodne spremenljivke:
  - *HeightArr* - niz globin (m) za katere je podana ρ_a
  - *GamArr* - niz podanih gostot ρ_a -1000,0 (kg/m^3)

  Izhodne spremenljivke:
  - *NumDat* - število podatkovnih parov (z_j, $\rho_a[j]$)
  - *HeightArr* - niz višin (m) nad difuzorjem
- HUNT - poišče segment višin [z_j, z_{j+1}], v katerem se znajde nova višina *z* (NUMERICAL RECIPES).
  Vhodne spremenljivke:
  - *Depth* - niz globin (neodvisnih količin)
  - *n* - število vrednosti v nizu *Depth*, ki bodo analizirane
  - *x* - vrednost argumenta, ki mora biti v segmentu [*Depth*[*jlo*], *Depth*[*jlo*+1]]

  Izhodne spremenljivke:
  - *jlo* - indeks, pri katerem je $Depth[jlo] \le x < Depth[jlo+1]$
- HUNTDRHO - podobno kot HUNT, le da po končanih izračunih poišče indeks *jlo* za interval v katerem se nahaja ena izmed štirih *izhodnih* spremenljivk. Zato je dodana *vhodna spremenljivka* z imenom *index* (vrednosti 1..4), ki pove za katero odvisno spremenljivko gre.
- SPLINE - izračuna tabelo drugih odvodov, ki so potrebni za zlepke (NUMERICAL RECIPES)
  Vhodne spremenljivke:
  - *x* - niz vrednosti "neodvisne" spremenljivke (*HeightArr*)
  - *y* - niz vrednosti "odvisne" spremenljivke ($GamArr[j]= \rho_a[j]$ - 1000,0)
  - *n* - število vrednosti parov (*x*, *y*), ki naj bi bile analizirane
  - *yp*1, *ypn* - začetna in končna vrednost drugega odvoda gostote morja:
    ($d^2\rho_a/dz^2\,[1] = d^2\rho_a/dz^2\,[n] = 0$)

  Izhodne spremenljivke:
  - *y2* - niz izračunanih vrednosti drugih odvodov $d^2\rho_a/dz^2\,[j]$, j = 2..n-1
- MakeRiAlpha - izračuna *Ri* število in parameter vnosa α.
  Vhodne spremenljivke:

y - niz trenutnih vrednosti štirih odvisnih spremenljivk

VaryAlpha - znak ("Y" ali "N"), ki pove, ali se naj α spreminja

Izhodne spremenljivke:

Ri - Richardsonovo število (po (1.38))

Alpha - vrednost parametra α

- SPLINDER - izračuna interpolirano vrednost, kot tudi prvi odvod s kubnim zlepkom, dopolnjen podprogram od NUMERICAL RECIPES.

  Vhodne spremenljivke:

  Xcut- vrednost "neodvisne" spremenljivke (višine *zzp*), pri kateri poiščemo vrednost odvisne spremenljivke

  xklo, *xkhi* - spodnja in zgornja vrednost intervala [*xklo*, *xkhi*], v katerem se nahaja *Xcut*

  yklo, *ykhi* - vrednosti "odvisnih" spremenljivk: *yklo* = *yklo*[*xklo*], *ykhi*=*ykhi*[*xkhi*]

  y2klo, *y2khi* - vrednosti drugih odvodov "odvisnih" spremenljivk: *y2klo* = *y2klo*[*xklo*], *y2khi*=*y2khi*[*xkhi*]

  Izhodne spremenljivke:

  yy - vrednost "odvisne" spremenljivke (npr. ρ_a)

  dydx - vrednost prvega odvoda (npr. $d\rho_a/dz$)

- DERIVS - tukaj so zapisane ključne enačbe (2.1)-(2.4) sistema. Predhodno je potrebno klicati podprograma SPLINDER in MakeRiAlpha.

  Vhodne spremenljivke:

  x - vrednost neodvisne spremenljivke (*s*)

  y - niz trenutnih vrednosti štirih spremenljivk

  Izhodne spremenljivke:

  dydx - niz štirih prvih odvodov sistema (2.1)-(2.4)

- RK4 - Runge-Kutta procedura četrtega reda (NUMERICAL RECIPES), izračuna nove vrednosti štirih odvisnih spremenljivk pri znanem koraku neodvisne spremenljivke. Klicana od RKQC.

  Vhodne spremenljivke:

  y - začetne (trenutne) vrednosti (štirih) odvisnih spremenljivk

  dydx - trenutnih vrednosti odvodov (štirih) spremenljivk

  n - število odvisnih spremenljivk (enačb; *n* = 4)

  x - vrednost neodvisne spremenljivke

  h - vrednost danega koraka (= Δs)

  Izhodne spremenljivke:

  yout - niz izhodnih vrednosti odvisnih spremenljivk

- RKQC - kliče RK4 in na osnovi velikosti napake metode določi korak Δs (NUMERICAL RECIPES)

  Vhodne spremenljivke:

  y - začetne (trenutne) vrednosti (štirih) odvisnih spremenljivk

  dydx - trenutnih vrednosti odvodov (štirih) spremenljivk

  n - število odvisnih spremenljivk (enačb; *n* = 4)

  x - vrednost neodvisne spremenljivke

  hþry - poskusna vrednost koraka (= Δs)

  eps - vrednost relativne napake ($\varepsilon = 10^{-10}$) po (2.7)

  yscal - niz štirih vrednosti, ki nastopajo na desni strani (2.7)

  Izhodne spremenljivke:

hdid - vrednost za Δs, ki se je obnesla

hnext - vrednost za Δs, ki bo upoštevana v naslednjem koraku

- FAECALI - kliče podprograme HUNT, SPLINDER, MakeRiAlpha, DERIVS in RKQC in vodi knjigovodstvo, beleži rezultate v nize glavnih spremenljivk *xp*[1..*NN*] (vrednosti neodvisne spremenljivke *s*), *yp*[1..4, 1..*NN*] (vrednosti štirih odvisnih spremenljivk), *xxp*[1..*NN*], (vrednosti *x*-koordinate) in *zzp*[1..*NN*] (vrednosti *z*-koordinate središča curka). NN je največje možno število izhodnih vrednosti (= 350). Podprogram zaustavi potek klicanja podprogramov, ko to ni več smiselno, oz. ko je $(x\text{-}x2)(x2\text{-}x1) \geq 0$ ali $((\theta < 0)$ in $(z > 0))$ ali $(z > zmax)$ .Tedaj shrani še poslednji rezultat.

  Vhodne spremenljivke:

  ystart - začetne vrednosti (štirih) odvisnih spremenljivk

  nvar - število odvisnih spremenljivk (=4)

  VaryAlpha - znak ("Y" ali "N"), ki pove, ali se naj α spreminja

  $x1$, $x2$ - začetna (= 6,2*D* = 0,62 m) in največja (=100 m) možna končna vrednost neodvisne spremenljivke *s*

  eps - vrednost relativne napake - po (2.7) ($\varepsilon = 10^{-10}$)

  *s*1 - začetna izbrana vrednost za Δs (= 0,001)

  smin - najmanjša možna vrednost za Δs (= 0,0)

  Izhodne spremenljivke:

  y, *dydx* - nove vrednosti štirih količin in njihovih odvodov po *s*

  nok - število uspešnih korakov

  nbad - število neuspešnih korakov metode (prevelik Δs)

- ANALYTIC - izračuna predvideno največje redčenje na osi vertikalnega vzgonskega curka in njegovo višino nad difuzorjem pri linearni rasti gostote z višino (Fischer in drugi, 1979, Lee in Neville-Jones, 1987) brez horizontalnega toka. Namenjena je za primerjavo numeričnih rezultatov z analitično - eksperimentalnimi rešitvami.

  Vhodne spremenljivke:

  diameter - premer odprtine na difuzorju

  dgamdz - srednja vrednost $d\rho_a/dz$ po vodnem stolpu

  *rho*0 - vrednost gostote ρ_0 morske vode pri difuzorju

  *u*0 - vrednost hitrosti curka pri odprtini

  Izhodne spremenljivke:

  ZZstra - vrednost največje višine (m) do katere se dvigne vzgonski curek

  SSstra - faktor redčenja na osi curka pri največji višini

- STRATIMIDDLE - izpis rezultatov na izhodne datoteke v primeru, da se vzgonski curek ne dvigne do gladine. Podprogram izračuna vrednosti za faktor redčenja za vsako vrednost izhodnih spremenljivk po (2.6). Iz dveh sosednjih zapisov rezultatov, pri katerih se spremeni predznak vrednosti spremenljivki $\Delta\rho$, izračuna z linearno interpolacijo u, b, θ in redčenje S za $\Delta\rho = 0$. Podprogram izračuna $<d\rho_a/dz>$ iz vrednosti za ρ_a pri dnu in pri gladini. Kliče podprograme HUNT, SPLINDER in ANALYTIC.

- STRATISURFACE - podobno kot podprogram STRATIMIDDLE, le za primer, ko vzgonski curek prileti na gladino. Zato podprogram namesto inetrpolacije za pogoj $\Delta\rho = 0$, ki ni dosežen, (linearno) interpolira rezultate na višino gladine (*DepthDif*) iz dveh sosednjih zapisov, ki imata vrednost višine okoli višine gladine. Ostala opravila, klici drugih podprogramov in izpisi so podobni kot pri podprogramu STRATIMIDDLE.

- **GLAVNI PROGRAM** (MAIN) - prebere podatke za ρ_a s klicem podprograma ReadData, kliče podprogram SPLINE za določitev drugih odvodov $d^2\rho_a/dz^2$ (koeficienti za kubne polinome), izračuna ρ_0 (kliče HUNT in SPLINDER), določi začetno vrednost neodvisne spremenljivke s_0 (SS0), največjo možno vrednost za s (Ssmax), začetne vrednosti spremenljivk u, b, θ in $\Delta\rho$, začetno vrednost za Δs (=s1), minimalno možno vrednost za Δs (=0) in začetni vrednosti za Ri in α. Pokliče podprogram FAECALI, iz zaporednih vrednosti za pot s in hitrost u izračuna niz vrednosti časa. Kliče podprogram HUNTDRHO, od koder dobi vrednost indeksa *tempklo*, $\Delta\rho$[*tempklo*] $\geq$ 0,0. Indeks *tempklo* primerja s številom uspešno opravljenih korakov *kount*, s čimer izvemo, ali se $\Delta\rho$ = 0 lahko pojavi znotraj vodnehga stolpa. Če temu je tako, potem pokliče podprogram STRATIMIDDLE za izpis rezultatov, sicer pa pokliče podprogram STRATISURFACE.

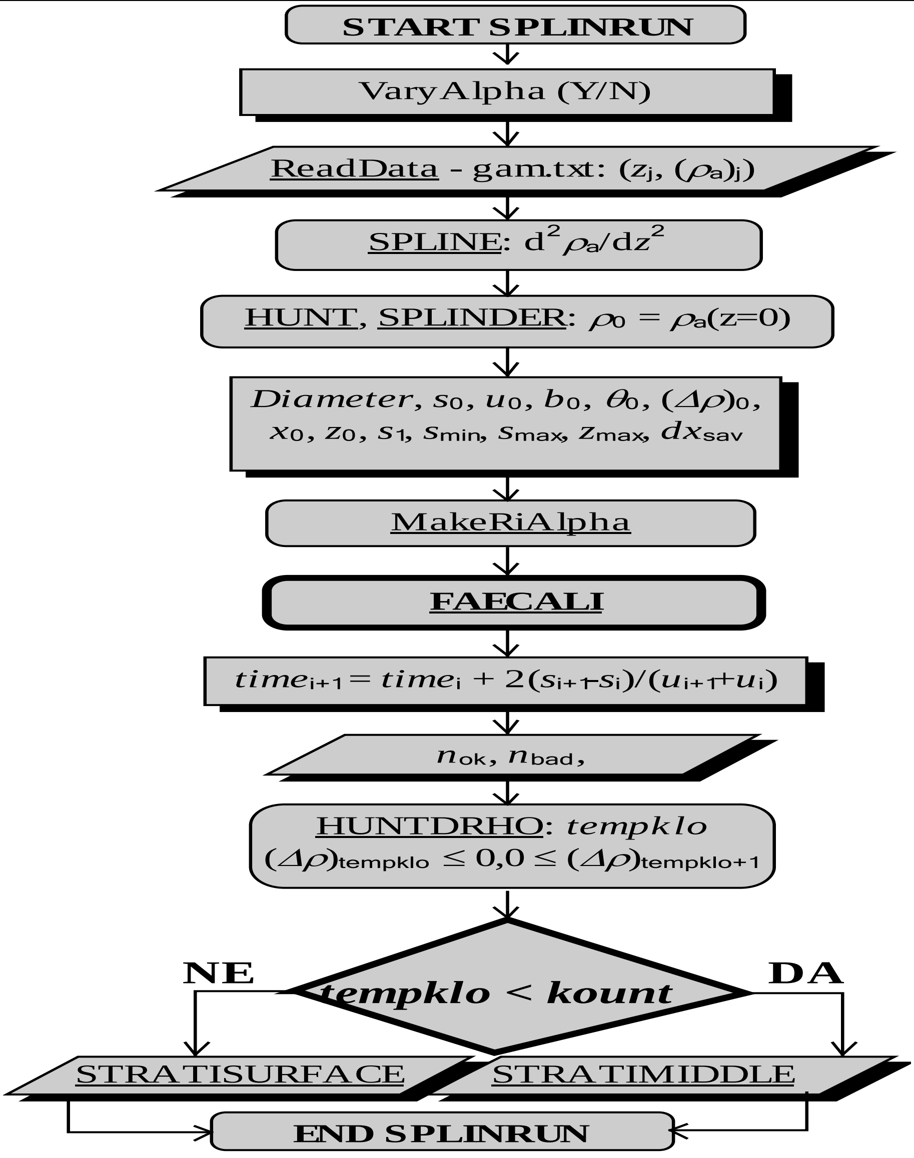
2.3 Pregled algoritma

Pregled algoritma skicirajmo le za glavni del programa brez podrobnosti, tako da shematično zajamemo bistvene sestavine. Glavni program je shematsko prikazan na Sl. 4, v Dodatku pa je izpis celotnega programa SPLINRUN.

3. KALIBRACIJA MODELA

3.1 Uvod v kalibracijo

Na osnovi desetih terenskih meritev z multisondo (CTD) v bližnji okolici difuzorjev piranskega izpusta smo zaključili, da je iz meritev, ki jih je možno izvesti, dokaj težko določiti višino dviga vzgonskega izliva odplak. Kljub indikacijam o globini in debelini plasti odpadne vode iz CTD meritev, ki so kot preliminarna ocena služile za določitev globin jemanja vzorcev za analizo hranilnih snovi in bakterij, pa sta se kot najzanesljivejši indikator globine razredčenih odplak pokazali vsebnost amoniaka in število fekalnih koliformov. Vendar pa je vertikalna ločljivost v tem primeru omejena tako s številom vzorcev, kot tudi z določevanjem globine jemanja vzorcev (na 0,3 m).



Sl. 4. Potek algoritma glavnega programa SPLINRUN.

Največji problem, ki pa se pojavi pri določevanju globine plasti odplak (običajno sta dve, po ena iz vsakega difuzorja), pa je v neznanem pretoku odplak skozi posamezen difuzor oz. izpust. Kot bomo videli iz kalibracije, je le-ta ključen podatek za napoved tako redčenja, kot tudi dviganja odplak v začetni (najpomembnejši) fazi redčenja. Zato kalibracijo modela nismo mogli in ne smeli izpeljati na osnovi terenskih meritev, pač pa na osnovi teoretičnih napovedi za dvig vzgonskega izliva ("buoyant plume"), kjer so bili ustrezni koeficienti določeni s poskusi. Kot bomo videli iz rezultatov simulacije, je premer debeline curka pri največji višini izliva (tisti premer, pri katerem pade hitrost curka na 1/e vrednosti v sredici izliva-curka) velikostnega reda 1 m. Ker so odprtine na medsebojni oddaljenosti 10 m, se vzgonski curki kot povsem samostojne enote dvigajo do vzgonsko nevtralne plasti, kjer pa se med seboj pomešajo. Zato smo privzeli analitične izraze za dvig iz majhne odprtine (krožnega preseka) in ne iz "linijskega" izvora (izjemno dolga odprtina - dolžine difuzorja).

3.2 Osnove kalibracijske metode

Osnova za določitev maksimalnega dviga vzgonskega izliva leži v dimenzijski analizi, ki je zaradi stratifikacije morske vode dodatno zapletena. Za kalibracijo modela predpostavimo linearno spreminjanje gostote z višino

$$\rho_{\mathrm{a}} = \rho_0 \left(1 - \delta(z)\right), \tag{3.1}$$

poleg tega zapišimo še zvezo za vzgonsko frekvenco N

$$N^2 = -\frac{g}{\rho_0}\frac{\mathrm{d}\rho_a}{\mathrm{d}z} = \frac{g\mathrm{d}\delta}{\mathrm{d}z} = g\delta' . \tag{3.2}$$

Pri stabilnem vodnem stolpu je $\delta' > 0$, pri linearnem upadanju gostote z višino pa sta tako N kot δ' konstantna. Iz dimenzijske analize sledi, da je značilna višina z_{M} za okrogel (vertikalni) curek, pri katerem sta pomembna pretok specifične gibalne količine M_0 in vzgonska frekvenca N

$$z_{\mathrm{M}} \propto \left(\frac{M_0}{N^2}\right)^{1/4}, \tag{3.3}$$

kjer je konstanta sorazmernosti blizu 3,8. Za curek, pri katerem prevladujejo vzgonske sile, pa se ohranja specifični pretok vzgona B_0 (Fischer in drugi, 1979, str. 342). Iz dimenzijske analize sledi za značilno višino z_{B}

$$z_{\mathrm{B}} = 3{,}98\left(\frac{B_0}{N^3}\right)^{1/4}, \tag{3.4}$$

kjer smo zapisali sorazmernostno konstanto 3,98 (Fischer in drugi, 1979, str. 396). Že v prejšnjem poglavju smo pri računu parametra zajetja α ocenili, da v našem primeru vzgonskega curka prevladuje vzgonski režim, zato bomo za linearno stratifikacijo za največjo višino uporabili kar (3.4).

Opredeliti treba še faktor redčenja. Pri izpeljavi masnega pretoka polutanta ψ_{C} po (1.21) smo zapisali pretok mase polutanta $\mathrm{d}\psi_{\mathrm{C}}$ skozi cev višine ds, v kateri ima polutant koncentracijo C, ki je odvisna od oddaljenosti od osi curka (polmera) in od opravljene poti s vzgonskega elementa na osi curka. Pretok ψ_c , ki je integral pretokov po vseh ceveh, pa zapišemo tudi kot $\langle C\rangle Q$, kjer je $\langle C\rangle$ po pretoku curka povprečena koncentracija. Od tod sledi

$$\langle C\rangle = \frac{\Psi_c}{Q} = \frac{\lambda^2}{\left(1+\lambda^2\right)} C(s), \tag{3.5}$$

kjer je $C(s)$ koncentracija na osi curka. Izraz (3.5) smo dobili z upoštevanjem pretoka ψ_c po (1.21) in Q po (1.8). Iz (3.5) sledi, da je $\langle C\rangle$ za faktor 0,59 (λ = 1,2), oz. za faktor 0,57 (λ = 1,16) manjša od koncentracije na osi curka $C(s)$. Faktor redčenja pa je definiran tudi kot razmerje med celotnim volumnom tekočine, v katerem se nahaja polutant, in volumnom, ki bi ga enaka masa polutanta zasedla sama, brez redčenja (Fischer in drugi, 1979). V primeru vzgonskega izliva opazujemo rezino (pravokotno na os izliva) volumna dV (višine ds) na določeni oddaljenosti od odprtine, skozi katero steče razredčeni polutant v času dt s pretokom Q, in jo primerjamo s podobno rezino volumna dV_0 tik pri odprtini difuzorja, skozi katero v enakem času steče enaka masa dm_c (nerazredčenega) polutanta s pretokom Q_0. Razmerje obeh volumnov nam dá prvo definicijo za faktor redčenja S_c:

$$S_c = \frac{\mathrm{d}V}{\mathrm{d}V_0} = \frac{Q}{Q_0} = \frac{C_0}{\langle C\rangle}. \tag{3.6}$$

Desno stran (3.6) smo dobili z upoštevanjem ohranitve mase polutanta dm_c = ψ_c dt, oz. masnega pretoka polutanta $\psi_c = \int Cu\mathrm{d}A = \langle C\rangle Q = C_0Q_0$, saj nič polutanta iz okolice naj ne bi prišlo v vzgonski curek. Za volumski pretok Q curka iz krožne odprtine uporabimo (1.8), $Q_0 = \pi u_0D^2/4$, od tod sledi

$$S_c = \frac{4ub^2}{u_0D^2}. \tag{3.7}$$

Fischer in drugi (1979, str. 330) so zapisali volumski pretok Q (pri njih μ) za izliv v vzgonskem režimu s spremenljivkama B in z (višine nad odprtino) na osnovi dimenzijske analize. Po (3.6) pa dobimo S_c kot

$$S_c = 0{,}15\frac{B^{1/3}z^{5/3}}{Q_0} = 0{,}15\left(\frac{g\Delta\rho z^5}{\rho_0 Q_0^2}\right)^{1/3}, \tag{3.8}$$

kjer smo upoštevali, da se v vzgonskem režimu izliva ohranja specifični pretok vzgona $B = B_0 = g(\Delta\rho/\rho_0)Q_0$, kjer je $\Delta\rho = \rho_0 - (\rho)_0$, če ni stratifikacije. Ta pomemben zaključek sledi iz izraza (1.6): dB/ds = 0, če je dρ_a /ds = 0. Relativna napaka sorazmernostne konstante v (3.8) pa je 10 %. Vendar pa bomo za kalibracijo modela uporabili rahlo spremenjeno definicijo za faktor redčenja, ki je tudi v veljavi in smo jo uporabili že pri zapisu spremljajočih enačb modela. Še enkrat zapišimo (1.26) malo drugače (Fan in Brooks, 1966):

$$S = \frac{C_0}{C(s)} = \frac{\lambda^2}{\left(1+\lambda^2\right)} \frac{C_0}{\langle C \rangle} = \frac{\lambda^2}{\left(1+\lambda^2\right)} S_c . \tag{3.9}$$

Na desni strani (3.9) smo izrazili $C(s)$ s $\langle C \rangle$ po (3.5) in S_c po (3.6). Faktor $\lambda^2/(1+\lambda^2)$ pa smo že izračunali pri (3.5) in je 0,59 (λ = 1,2) oz. 0,57 (λ = 1,16). Fischer in drugi (1979, str. 396) so zapisali faktor redčenja vertikalnega izliva iz krožne odprtine v nestratificirano (homogeno) morje:

$$S = 0{,}089 \left(\frac{g \Delta \rho z^5}{\rho_0 Q_0^2} \right)^{1/3} . \tag{3.10}$$

Sorazmernostni faktor 0,089 pa dobimo iz sorazmernostnega faktorja 0,15, ki nastopa v (3.8), če slednjega množimo z $\lambda^2/(1+\lambda^2)$ = 0,59 (λ = 1,2 po Fischer in drugi, 1979). To pa je usklajeno z zvezo (3.9) med S in S_c. Tako smo z (3.10) izbrali, da je faktor redčenja S definiran kot razmerje med začetno koncentracijo in koncentracijo na osi vzgonskega curka. Tega smatramo kot analitično določen faktor redčenja na višini z nad odprtino za vzgonski izliv v homogeno (mirujoče) morje. Za faktor redčenja v linearno stratificirano morje so Fischer in drugi zapisali izraz podoben (3.10)

$$S = 0{,}071 \left(\frac{g \Delta \rho z_{max}^5}{\rho_0 Q_0^2} \right)^{1/3} \tag{3.11}$$

z manjšo sorazmernostno konstanto; največja višina z_{max}, do katere se vzgonski curek v linearno stratificiranem morju sploh dvigne, pa je kar enaka z_B za vzgonski režim po (3.4). Slednjo zapišemo malo drugače

$$z_{max} = 3{,}98 \left(Q_0 \frac{g \Delta \rho}{\rho_0} \right)^{1/4} \left(-\frac{g}{\rho_0} \frac{d\rho_a}{dz} \right)^{-3/8} . \tag{3.12}$$

3.3 Povzetek kalibracijske metode

Povzemimo: Za *homogeno* morje uporabimo faktor redčenja po (3.10), višina z pa je enaka celotni višini vodnega stolpa nad difuzorjem, saj se vzgonski element ne ustavi pri poti navzgor, ker nikoli ne more imeti gostote enake gostoti morja ρ_a. Za višino gladine je v našem primeru privzeta vrednost 20,7 m (Mavrič, 1979). Za *stratificirano* morje je najbolje privzeti, da gostota linearno upada z višino po (3.1), kjer je δ = konst. Tedaj ocenimo maksimalno višino dviga po (3.12), faktor redčenja pa iz (3.11). Omenjene tri zveze izračuna podprogram ANALYTIC v programu SPLINRUN. Iz rezultatov numeričnega modela pa izluščimo faktor redčenja po (1.26), ki je usklajen z izrazom (3.9), kjer v slednjega vstavimo S_c po (3.7). Pokazalo se je, da je pri linearni stratifikaciji za največjo višino izliva z_{max} najbolje uporabiti tisto višino, pri kateri je njegova hitrost u = 0, saj se vzgonski element na manjši višini, pri kateri je $\Delta\rho$ = 0, še vedno dviga zaradi vztrajnosti.

Poslednja pripomba se nanaša na relativno napako ocen redčenja. Pri redčenju vzgonskega izliva v homogenem morju zapišemo relativno napako z uporabo enačb (1.26) in (3.10), kjer faktor 0,089 pišemo kot $0{,}15\lambda^2/(1+\lambda^2)$

$$S_e = \frac{S(s) - S(z_{\max})}{S(z_{\max})} = \frac{\left| \dfrac{4ub^2}{u_0 D^2} - 0{,}15\left(\dfrac{g\Delta\rho z_{\max}^5}{\rho_0 Q_0^2}\right)^{1/3} \right|}{0{,}15\left(\dfrac{g\Delta\rho z_{\max}^5}{\rho_0 Q_0^2}\right)^{1/3}}, \tag{3.13}$$

od koder navidezno sledi, da je relativna napaka neodvisna od vrednosti za λ. Vendar pa sta tako upadanje hitrosti kot rast debeline curka odvisna od λ, kar sledi iz enačb (2.1) in (2.2). Še bolj je odvisna napaka S_e od λ za (linearno) stratificirano morje, saj je vrednost sorazmernega koeficienta v (3.11) določena zgolj eksperimentalno, poleg tega se v tem primeru ne ohranja specifični pretok vzgona B. V numeričnem modelu smo za relativno napako za faktor redčenja v homogenem morju uporabili (3.13), v oceni relativne napake za linearno stratificirano morje pa je (3.10) nadomeščena z (3.11), zato pa nastopi faktor $\lambda^2/(1+\lambda^2)$ v prvem členu števca količnika (3.13), koeficient 0,15 pa je nadomeščen z 0,071.

3.4 Rezultati kalibracije modela

Pri kalibraciji smo se v osnovi usmerili na primerjavo ocen za dvig in faktor redčenja odplak v primeru linearno stratificiranega morja ter na faktor redčenja v primeru homogenega morja, pri čemer pa smo kot prioritetno vzeli kalibracijo modela za primer stratificiranega morja. Morje je namreč stratificirano celo v zimskem času in je popolnoma homogeno morje relativno redek pojav. Pri linearno stratificiranem primeru smo izbrali, da gostota morske vode upada z višino z po zvezi (3.1)

$$\rho_{\mathrm{a}} = \rho_0 - \Gamma z, \tag{3.14}$$

kjer smo privzeli, da je ρ_0 = 1027,8232 kg/m^3 (gostota morja na nivoju odprtine difuzorja) in Γ = 0,233 kgm^{-4} = $\rho_0 N^2/g$ po (3.2), od koder sledi vzgonska frekvenca N = 0,05 s^{-1}. To je značilna poletna stratifikacija v Tržaškem zalivu, pri kateri se gostota spremeni za 4,66 kg/m^3 pri spremembi globine za 20 m. V vhodni datoteki LIMGAM.TXT zapis simuliranih podatkov (Tab. 1) izgleda podobno kot zapis podatkov, ki jih pridobimo iz meritev s CTD multi-sondo. Vhodni podatki so zato urejeni po globinah in ne po višinah.

V primeru homogenega morja lahko za gostoto morja ρ_{a} (= ρ_0) izberemo poljubno smiselno vrednost. Privzeli smo, da je

$$\rho_{\mathrm{a}} = 1025{,}48155 \text{ kg/m}^3.$$

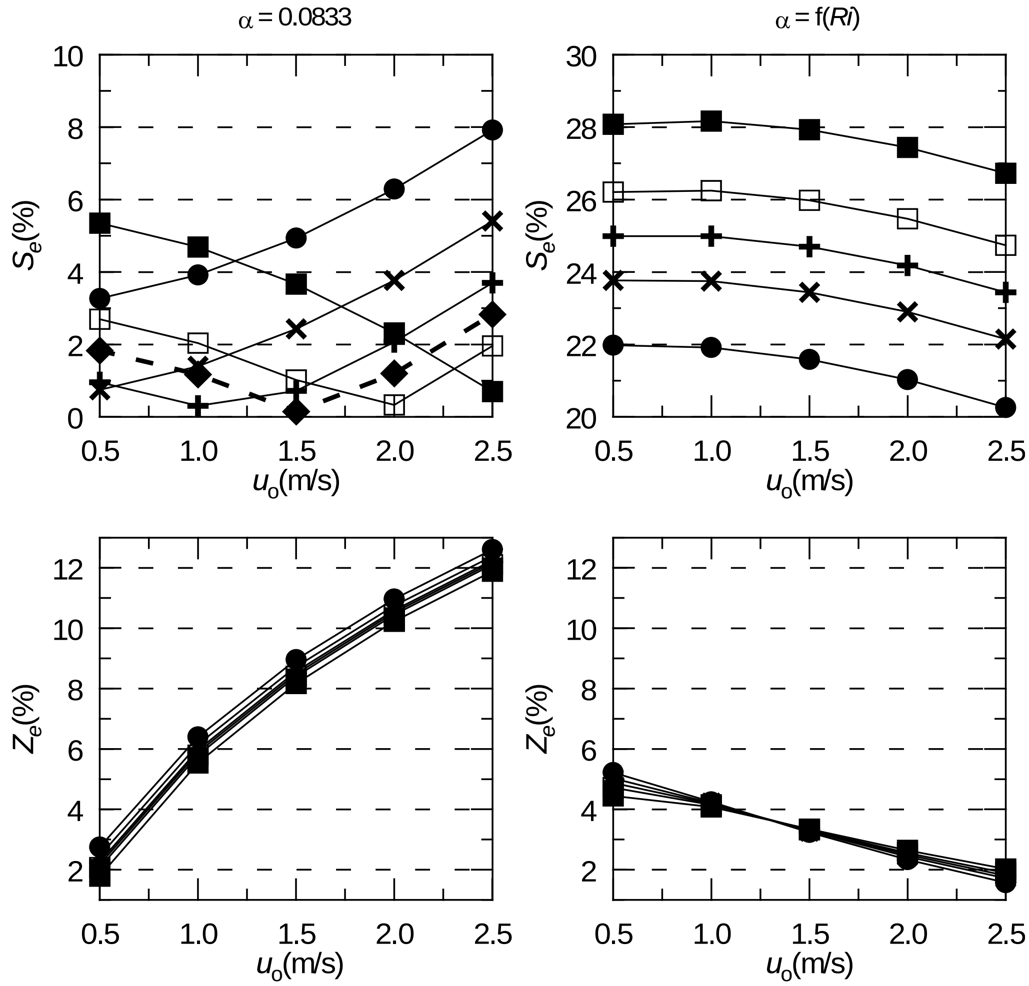
To je vrednost, ki leži med vrednostima gostote pri difuzorju in gostote na gladini pri prej izbrani linearni stratifikaciji.

0,1	999	23.0234
0.2	999	23.0467
…..	….	………..
20.6	999	27.7999
20.7	999	27.8232
…..	….	……….

Tab. 1. Pregled kalibracijskih podatkov v vhodni datoteki LINGAM.TXT. V prvem stolpcu se nahajajo globine (m), v drugem je število postaj, po katerih sicer povprečimo podatke o gostotah iz CTD meritev na terenu, v tretjem stolpcu pa so vrednosti gostot $\sigma = \rho - 1000$ (kg/m^3).

Kalibracijo smo pričeli z izračunom S_e odstopanja faktorja redčenja od predvidenega faktorja redčenja, ki je analitično določen z izrazom (3.13) za vzgonski curek. Poleg tega smo izračunali relativno odstopanje Z_e največje dosežene višine dviga, pri kateri je hitrost $u = 0$, od predvidene višine z_{max} po (3.12). Pri kalibraciji smo privzeli vertikalno brizganje iz odprtine ($\theta_0 = 90^0$). Opazovali smo odvisnost odziva numeričnega modela na različne vrednosti začetne hitrosti curka u_0 pri različnh vrednostih parametra λ (Sl. 5). Pri zapisu osnovnih enačb modela smo zapisali, da gostotna razlika $\Delta\rho$ (oz. koncentracija C_e) upade na 1/e tiste na osi curka pri polmeru λb ($1 < \lambda < 2$), medtem ko hitrost upade pri polmeru b na 1/e tiste na osi curka.

Razlikovali smo dva načina delovanja numeričnega modela: Pri prvem (leva polovica na Sl. 5) je koeficient vnosa okolne tekočine (morske vode) konstanta, $\alpha = 0{,}0833$. Pri drugem načinu pa je α funkcija Ri števila po (1.38). Takoj opazimo bistveno nižje vrednosti relativne napake za faktor redčenja S_e, ko je α = konst. Po drugi strani pa so odmiki najvišje višine curka Z_e od pričakovane višji, ko je α = konst., vendar so razlike med enim in drugim načinom zajemanja okolne vode manjše. Relativen odmik Z_e pri α = konst. monotono raste z začetno hitrostjo curka u_0 od pribl. 2 % (u_0 = 0,5 m/s) do 12 % (u_0 = 2,5 m/s), skorajda neodvisno od parametra λ. V primeru $\alpha = f(Ri)$ pa Z_e monotono pada z u_0, od pribl. 5 % (u_0 = 0,5 m/s) do pribl. 2 % (u_0 = 2,5 m/s). Pri manjših vrednostih hitrosti (do 1,0 m/s), ki so pogoste za oba difuzorja piranskega izpusta, je v prednosti način z α = konst. V tem primeru S_e monotono upada pri $\lambda = 1{,}1$ in monotono raste pri $\lambda = 1{,}17$ ter pri $\lambda = 1{,}2$. Ko je λ = 1,13 do 1,15 pa S_e doseže minimum pri določeni hitrosti.

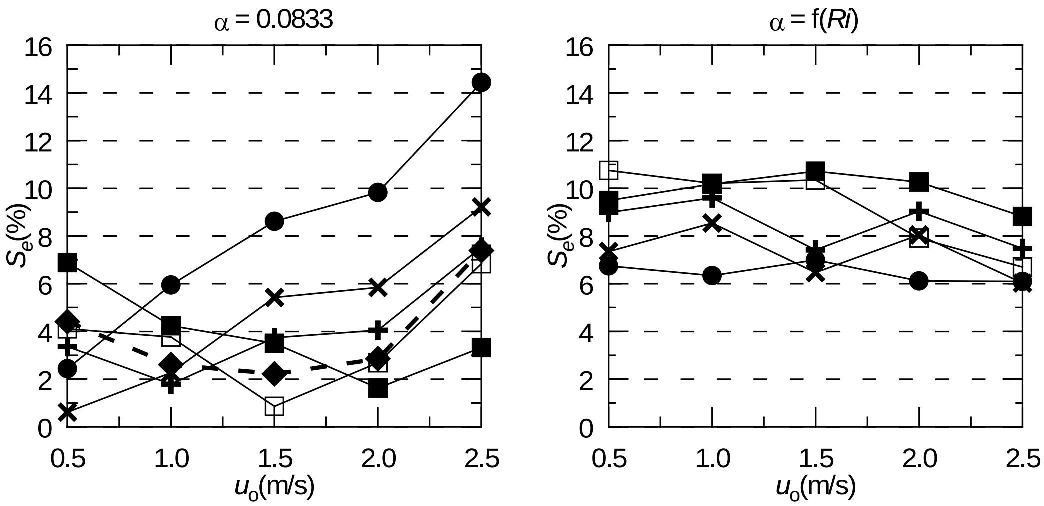


Sl. 5. Absolutne vrednosti relativnih odstopanj numerično dobljenih faktorjev redčenja od predvidenih faktorjev redčenja S_e (zgoraj), kot tudi odstopanj Z_e numerično dobljenih največjih višin od predvidenih vrednosti (spodaj) za linearno stratifikacijo po (3.14) pri konstantnem koeficientu vnosa okolne tekočine α (levo) in spremenljivem koeficientu (desno), odvisnem od lokalnega *Ri* števila po (1.38). Parameter λ ima vrednosti 1,10 (■), 1,13 (□), 1,14 (◆), 1,15 (+), 1,17 (×) in 1,20 (●). Najugodnejši rezultati za faktor redčenja so dobljeni pri λ = 1,14 (črtkano).

Kot najugodnejša je sprejeta možnost za λ = 1,14. Tedaj S_e sprva upada z u_0 od 1,8 % do 0,2 % pri u_0 = 1,5 m/s, ko prične rasti do vrednosti 2,8 % pri u_0 = 2,5 m/s. Tako majhno relativno odstopanje od predvidenih vrednosti za faktor redčenja je skorajda idealno za numeričen model. Vedeti pa moramo, da je S_e po (3.13) absolutna vrednost relativnega odstopanja numerično določenega faktorja redčenja od analitičnega. Pri u_0 = 0,5 m/s je namreč numerično določen faktor redčenja S = 46,8, pričakovana vrednost po (3.11) pa je višja, in sicer 47,7. Pri u_0 = 2,5 m/s pa je numerično določen S = 32,8, analitična vrednost pa je enaka 31,9, torej manjša od numerične. Zato zaključimo, da je pri λ = 1,14 numerično določen faktor redčenja

S pri majhnih hitrosti manjši od predvidene analitične vrednosti, blizu hitrosti u_0 = 1,5 m/s sta obe vrednosti enaki (do 0,2 %), pri višjih hitrostih pa odstopanje S_e raste z u_0, pri čemer je numerična vrednost večja od predvidene. Kadar so pri vrednostih parametra λ različnih od 1,14 vrednosti za S_e manjše od tistih pri λ = 1,14 pri manjših hitrostih (u_0 < 1,5 m/s), pa so vrednosti za S_e bistveno višje od tistih pri λ = 1,14 pri večjih hitrostih (u_0 > 1,5 m/s).

Pri homogenem morju smo opazovali le odmik faktorja redčenja od pričakovanega faktorja, saj se vzgonski curek tedaj dvigne na gladino. Sl. 6 prikazuje, da S_e pri λ = 1,2 monotono raste z u_0, če je α = konst. Tedaj so vrednosti



Sl. 6. Absolutne vrednosti relativnih odstopanj numerično dobljenih faktorjev redčenja od predvidenih faktorjev redčenja S_e za homogeno morje (ρ_a = 1025,48155 kg/m^3) pri konstantnem koeficientu vnosa okolne tekočine α (levo) in pri koeficientu, ki je odvisen od lokalnega *Ri* števila (desno) po (1.38). Parameter λ ima vrednosti 1,10 (■), 1,13 (□), 1,14 (◆), 1,15 (+), 1,17 (×) in 1,20 (●). Najugodnejši rezultati za faktor redčenja so dobljeni pri λ = 1,14 (črtkano).

za S_e pri λ = 1,13-1,14 dokaj nizke in malo odvisne od hitrosti u_0.

Za vrednost λ =1,14, ki je bila izbrana kot najugodnejša v primeru linearne stratifikacije, vidimo, da se vrednosti za S_e za homogeno morje nahajajo v intervalu med 2 in 4,5 % za u_0 med 0,5 in 2 m/s, ki se povzpnejo na skoraj 7 % pri u_0 = 2,5 m/s. Kadar je koeficient vnosa okolne tekočine α = f(*Ri*), pa so odstopanja od pričakovanih vrednosti višja, na celotnem intervalu hitrosti so praktično nad 6 % za vse izbrane vrednosti za λ in se s hitrostjo u_0 malo spreminjajo. Tako zaključimo, da je vrednost 1,14 za koeficient λ sprejemljiva tudi za homogeno morje.

4. OBČUTLJIVOST MODELA NA ZAČETNE POGOJE

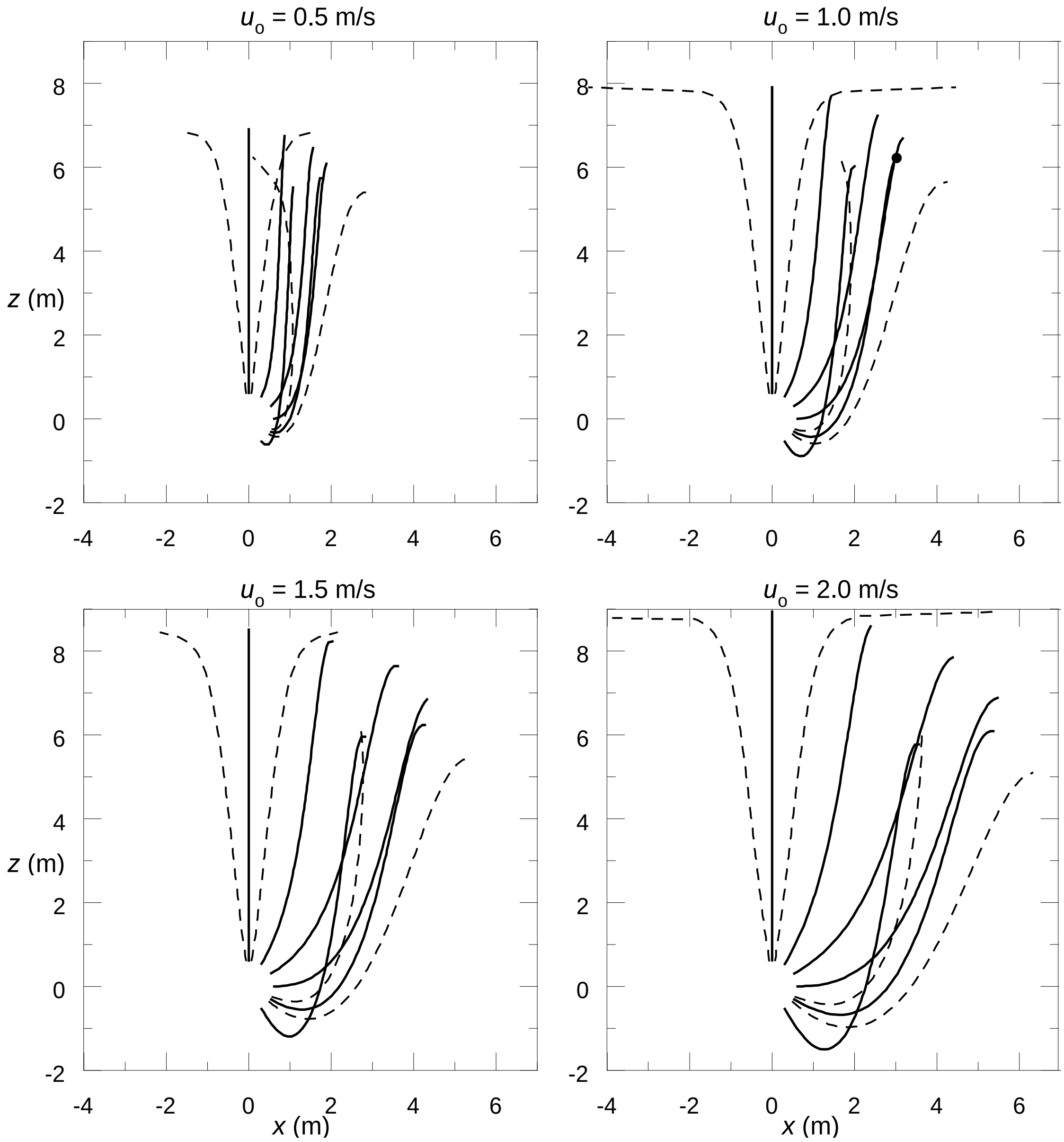
Ker sta parametra α (= 0,0833) in λ (= 1,14) s kalibracijo opredeljena, lahko opazujemo odziv modela pri spremenljivih začetnih pogojih. V primeru piranskega izpusta so začetne vrednosti (u_0, θ_0 in $(\Delta\rho)_0$) treh odvisnih spremenljivk v naravi (pri difuzorjih) variabilne; začetna širina curka, ki je opredeljena s premerom odprtine D po (1.24), pa je seveda konstantna. Od začetnih vrednosti treh količin pa se še najmanj spreminja začetna gostotna razlika $(\Delta\rho)_0$, ki je po (1.31) opredeljena z razliko med gostoto morja pri difuzorjih ρ_0 in gostoto odplak $(\rho)_0$ pri odprtini. Slednja se med letom bistveno ne spreminja, zagotovo pa so njene vrednosti med 997,1 kg/m^3 pri temperaturi 25,0 ^{0}C in 1000,1 kg/m^3 pri temperaturi 5,0 ^{0}C; v obeh primerih smo upoštevali tlak 20 dbar. Zato je prej privzeta vrednost 1000,0 kg/m^3 za $(\rho)_0$ po (1.32) blizu zgornji meji možnih vrednosti. Gostota morja pri dnu ρ_0 pa ima običajno vrednosti nad 1025 kg/m^3 in pod 1028 kg/m^3. Zato bo gostotna razlika ρ_0 - $(\rho)_0$ med 25 in 31 kg/m^3, kar pomeni relativno spremembo od ± 10,7 % okoli srednje vrednosti 28 kg/m^3 . Kvečjemu tolika pa je tudi relativna sprememba gostotne razlike $(\Delta\rho)_0$ po (1.31), ki jo moramo upoštevati v modelu ($(1+ \lambda^2)/2\lambda^2 = 0{,}89$). V simulacijah smo privzeli $(\Delta\rho)_0$ = 0,89*27,82 kg/m^3 = 24,76 kg/m^3.Tako preostaneta še dve začetni vrednosti u_0 in θ_0, ki zahtevata ločeno obravnavo.

Začetni kot θ_0 med osjo curka pri odprtini in vodoravno osjo je seveda opredeljen s samo konstrukcijo difuzorja in se zato časovno ne spreminja - če ne pride do nepričakovanih "naravnih posegov", kot je npr. vleka difuzorja z ribiško globinsko mrežo. Za difuzorja na konceh obeh izpustnih cevi je predviden začetni kot pri vseh odprtinah $\theta_0 = 0$. Vendar pa smo s podvodnimi ogledi difuzorjev ugotovili, da pri novejšem difuzorju, skozi katerega se pretaka večina odplak, ta kot je pribl. 25-30^0 (na drugi strani cevi pa -30^0 do -25^0), medtem ko je razpon vrednosti za θ_0 pri starejšem difuzorju mnogo večji, od pribl. 45^0 na konici difuzorja do 90^0 nekje na sredi difuzorja. To pomeni, da je starejši difuzor torzijsko zasukan, poleg tega, da je bil z vleko mreže tudi premaknjen in da brizga odplake tudi navpično navzgor in navzdol - odprtine so navrtane izmenično z ene in druge strani cevi difuzorja. Nedoločenost vrednosti začetnega kota pomeni, da moramo obravnavati občutljivost modela na izliv pri različnih θ_0.

Za začetne vrednosti hitrosti pa ocenjujemo, da se spreminjajo v ritmu sprememb pretoka odplak. Dnevne meritve obratov tlačnih črpalk so pokazale, da se pretok lahko spreminja od 40 l/s (okrog štirih zjutraj) do 130 l/s (od 8^h do 10^h zjutraj in od 20. do 24. ure zvečer), kar pomeni pribl. 100 % dnevne spremembe pretoka. Prav tako se spreminja tudi začetna hitrost efluenta u_0 skozi odprtino (Malačič, 1997).

Zato smo opazovali odziv numeričnega modela v primeru linearno stratificiranega morja po (3.14) tako pri različnih vrednostih začetne hitrosti curka u_0 (od 0,5 m/s do 2 m/s s korakom 0,5 m/s), kot pri različnih vrednostih začetnega kota θ_0, ki ga curek oklepa z vodoravne osjo (od -60^0 do 90^0 s korakom 30^0), pri čemer pa kota $\theta_0 = -90^0$ žal nismo mogli upoštevati. Začetna smer brizganja curka odplake je tedaj sicer navzdol, vendar se pri dnu obrne navzgor in objame cev difuzorja. To pa presega zmogljivost numeričnega modela, ki simulira pretok odplake skozi rezino vzgonskega curka, ki je pravokotna na njegovo os širjenja. Vendar pa so opravljene simulacije občutljivosti modela zelo poučne. Na Sl. 7 so prikazane trajektorije curkov z različnimi začetnimi vrednostmi hitrosti u_0 in kota θ_0. Odprtina je vedno postavljena v koordinatno izhodišče. V neposredni bližini odprtine je upoštevano območje vzpostavljanja toka (ZFE), šele za njim se prične območje vzpostavljenega toka (ZEF) s simulacijo dviga z numeričnim modelom.

Faktor redčenja odplak (ni na Sl. 7) je seveda najnižji pri največji hitrosti iztekanja iz odprtine (u_0 = 2,0 m/s), ko se njegove vrednosti nahajajo v območju od 34,2 ($\theta_0 = 90^0$) do 36,1 ($\theta_0 = 0^0$). Najvišje vrednosti dosega pri najmanjši opazovani začetni hitrosti (u_0 = 0,5 m/s), ko je v območju od 46,9 ($\theta_0 = -60^0$) do 47,1 ($\theta_0 = 0^0$). Tako lahko hitro zaključimo, da pri izbrani hitrosti začetni kot curka z vodoravno osjo malo vpliva na faktor redčenja (spremembe manjše od 6%); pomembnejšo vlogo ima na spremembe redčenja začetna hitrost (do 30 % pri hitrostih od 0,5 do 2 m/s) pri izbranem kotu.



Sl. 7. Trajektorije vzgonskega curka (polne črte), skupaj z mejami območja curka (črtkano), pri katerih upade hitrost širjenja na 1/e od tiste na osi curka (pri polmeru curka b). Posamezne slike vsebujejo različne začetne vrednosti hitrosti u_0, ki so navedene nad sliko, v vsaki sliki je prikazana trajektorija curka za vrednosti začetnega kota od -60^0 do 90^0 v intervalu po 30^0, širina curka pa je zaradi preglednosti prikazana le za kota -30^0 in 90^0. Simulacija dviga vzgonskega curka je bila opravljena za linearno stratificirano morje po (3.14) z začetnim polmerom curka $D/2^{1/2}$ = 0,07 m in z začetno vzgonsko razliko $(\Delta\rho)_0$ = 24,76 kg/m^3.

Višina največjega dviga vzgonskega curka pa je bolj odvisna od začetnega kota θ_0 kot faktor redčenja. Pri hitrosti u_0 = 0,5 m/s se višina giblje med 5,5 m (θ_0 = -60^0) in 6,9 m (θ_0 = 90^0), pri hitrosti u_0 = 2,0 m/s pa se giblje med 5,8 m (θ_0 = -60^0) in 8,3 m (θ_0 = 90^0). Torej je relativna sprememba največje višine med 23 % (u_0 = 0,5 m/s) in 35 % (u_0 = 2,0 m/s) zaradi sprememb začetnega naklona curka θ_0. Sprememba največje višine zaradi sprememb začetne hitrosti u_0 pa je pribl. 18 %.

Faza začetnega redčenja se hitro zaključi. Iz podatkov izhodne datoteke za risanje trenutnih vrednosti spremenljivk sledi, da doseže vzgonski curek pri hitrosti u_0 = 0,5 m/s največjo višino po času t_{max}, ki se nahaja v območju od 30,0 s ($\theta_0 = 90^0$) do 31,5 s ($\theta_0 = -60^0$). Pri veliki začetni hitrosti u_0 = 2,0 m/s pa je razpon časa malo večji, in sicer od 27,9 s ($\theta_0 = 90^0$) do 33,7 s ($\theta_0 = -60^0$). Tako ocenimo, da začetni kot θ_0, ki ga curek oklepa z vodoravno osjo, vpliva od 5,6 % pri majhnih hitrostih do 18,8 % pri velikih hitrostih. Pri izbranem začetnem kotu (npr. $\theta_0 = 90^0$) začetna hitrost manj vpliva na čas dviganja do največje višine (8,2 %). Kadar je začetna hitrost velika, le-ta hitro upade v prvih nekaj sekundah (manj kot pet sekund) in se nato skorajda linearno zmanjšuje do vrednosti nič. Pri manjših začetnih hitrostih pa je začetni upad hitrosti manjši. Lahko zaključimo, da je v našem testnem primeru pri vzgonski frekvenci N = 0,05 s čas dviganja pribl. 30 s in da zato pribl. velja

$$Nt_{max} \sim 1{,}5\,, \tag{3.15}$$

od tod sledi, da je vzgonska frekvenca primerno merilo za čas dviganja vzgonskega curka. Ta ocena je še boljša od ocene $Nt_{max} \sim$ 4-5, ki so jo nedavno dobili (Fonseka in drugi, 1998) pri laboratorijskem opazovanju dviga izpuščene vzgonske tekočine, ki je bila hitro in kratkotrajno vbrizgana v laboratorijsko posodo. To je dobra simulacija za kratkotrajno delovanje izpusta pri velikih hitrostih izliva.

Omeniti je potrebno, da širina (polmer) curka izjemno močno naraste (tudi za faktor 100) pri majhni spremembi poti (Δs < 0,1 m) pri koncu poti curka, ko se le-ta približa vzgonsko nevtralni plasti. Vodoravnih črt (črtkano), ki bi ponazarjale izjemno širok curek na koncu opravljene poti, zaradi preglednosti nismo narisali na Sl. 7, pač pa smo risanje zaključili s predzadnjo vrednostjo polmera curka b. Le-ta ni nikoli presegla vrednosti 2,3 m. To pa seveda pomeni, da smo upravičeno obravnavali izliv odplak iz posamezne odprtine. Vzgonski curki se namreč pred prihodom v vzgonsko nevtralno plast zagotovo ne združijo, saj so odprtine na obeh difuzorjih navrtane na medsebojni razdalji 10 m.

5. POVZETEK IN ZAKLJUČEK

Na osnovi primerjave med numeričnimi rezultati in pričakovanimi vrednostmi smo ugotovili, da je za faktor redčenja ujemanje numeričnih rezultatov s pričakovanimi boljše, če je koeficient vnašanja morske vode v curek odplak α konstanten. Ocenili smo, da smo dobili najboljše ujemanje, če smo za parameter λ izbrali vrednost 1,14. Tedaj se faktor redčenja razlikuje od pričakovane vrednosti za manj kot 5 %, če je okolna tekočina (morje) homogena ali linearno stratificirana. Pri stratificiranem morju pa je pomembna tudi največja višina, do katere se vzgonski curek dvigne. Odmik numerične vrednosti največje višine od pričakovane vrednosti pa se s spreminjanjem vrednosti parametra λ bistveno ne spremeni. Le-ta pri α = konst. raste z začetno hitrostjo u_0 od 2 % tja do 11 %, pri $\alpha = f(Ri)$ pa z u_0 upada od pribl. 5 % do 2 %, pri čemer se u_0 povečuje od 0,5 m/s do 2,5 m/s.

Simulacija občutljivosti modela je pokazala, da je v stratificiranem morju vrednost faktorja začetne dilucije od pribl. najmanj 34 (u_0 = 2,0 m/s, $\theta_0 = 90^0$), do največ 47 (u_0 = 0,5 m/s, $\theta_0 = 0^0$), kar pomeni, da je redčenje prve faze znotraj območja UNEP-ovih priporočil (1995), med 10 in 1000. Vendar pa se to nanaša na upad celotnega števila bakterij (Malačič, 1997), tako da bi bilo potrebno v našem primeru dobljene vrednosti za faktor redčenja še množiti s faktorjem za eksponentni upad števila bakterij, ki je tudi gotovo večji od ena. Ne glede na to pa se bo celoten faktor redčenja na koncu prve faze nahajal bližje spodnji meji navedenega intervala, kot zgornji meji. S konstrukcijo pa je to skorajda nemogoče izboljšati. Iz izraza (3.11) za stratificiran vodni stolp ter iz izraza (3.10) za homogeno morje je razvidno, da faktor redčenja raste z višino kot $z^{5/3}$, kar pomeni resno oviro za faktor redčenja v našem plitkem morju (pribl. 20 m) v primeru šibke stratifikacije, ko vzgonski curek lahko prispe na gladino. Kljub temu pa difuzor opravi svojo nalogo in znatno prispeva k redčenju odplak.

Začetni kot curka z vodoravno osjo bolj malo vpliva na faktor redčenja (manj kot 6%), mnogo bolj se ta spremeni zaradi začetne hitrosti, do 30 % (u_0 od 0,5 do 2 m/s). Nasprotno pa je višina največjega dviga vzgonskega curka precej odvisna od začetnega kota θ_0. Najmanjša je pri najmanjših začetnih hitrostih curka in začetni usmerjenosti curka navzdol (5,5 m pri u_0 = 0,5 m/s in kotu $\theta_0 = -60^0$), največja pa je pri največjih opazovanih hitrostih, ko je curek usmerjen navpično navzgor (8,3 m pri hitrosti u_0 = 2,0 m/s in kotu $\theta_0 = 90^0$). Relativna sprememba največje višine ne presega 35 % (u_0 = 2,0 m/s) zaradi sprememb začetnega naklona curka θ_0, medtem ko je njena vrednost precej manjša (pribl. 18 %) pri spremembah začetne hitrosti u_0. Vsekakor lahko ugotovimo, da v primeru izbrane stratifikacije, ki ustreza srednjemu gradientu gostote v poletnem obdobju, vzgonski curek odplak nikoli ne doseže gladine morja, pač pa se razredčena odplaka horizontalno širi proč od difuzorjev v plasti, ki se nahaja na višini od 5,5 m do 8,3 m nad difuzorjema. To je seveda zelo ugodno.

Zapišimo še, da je iz rezultatov modela razvidno, da so vrednosti končnega polmera vzgonskega curka (*b*) manjše od 2,3 m tik pred prihodom v vzgonsko nevtralno plast . To pomeni, da je bila začetna predpostavka, po kateri smo obravnavali začetno redčenje odplake, ki izvira iz posameznih okroglih odprtin difuzorja upravičena, saj se curki do prihoda v vzgonsko nevtralno plast ne prekrivajo, ker so na začetni medsebojni razdalji 10 m. Metodo, ki obravnava redčenje iz linijskega izvora z dolžino difuzorja, je zato potrebno zavreči. Začetna faza redčenja se zagotovo zaključi v manj kot eni minuti (30 s v testnem primeru).

Numerični model smo zgradili tako kot smo ga planirali. Primerjava s pričakovanimi vrednostmi za idealizirane pogoje v morju, ki pa so dokaj reprezentativni za realne razmere (stratifikacijo), je dala zelo vzpodbudne rezultate. Na koncu smo tudi preverili občutljivost modela na spremembe v začetnih pogojih (začetna hitrost curka in začetni kot, ki ga curek oklepa z vodoravno osjo). Tako poznamo vedenje modela in ga lahko uporabimo za poljubno realno stratifikacijo, ki je in še bo na morju izmerjena.

Dodajmo, da smo z modelom dobili vpogled le v redčenje v sredici vzgonskega curka na koncu prve faze, ko le-ta pristane v vzgonsko nevtralni plasti, ali pa na gladini pri mirujočem morju. Tokovi imajo vpliv že na začetno redčenje ob dviganju vzgonskega curka, če so dovolj izraziti in je stratifikacija relativno šibka. Poleg tega je na odprtem morju redčenje tudi posledica sekundarne faze, v kateri je pomembna disperzija, ki je posledica strukture advekcijskega polja. O tovrstnih efektih bo sicer še tekla beseda v prihodnjih poročilih, simuliranje ključnega dela redčenja, ki je še vezano na konstrukcijo difuzorja, pa je s tem delom zaključeno.

Zahvala

Nosilec naloge je iskreno hvaležen dr. Milanu Ambrožiču za skrben pregled poročila in koristne pripombe.

Seznam pomembnejših količin

b	polmer curka, pri katerem hitrost upade na 1/e vrednosti na osi curka
B, B_0	specifični pretok vzgona, začetni specifični pretok vzgona
C, C_0^*, C_0	koncentracija polutanta (na osi curka), koncentracija na razdalji s_0 od odprtine, koncentracija v odprtini
$\langle C \rangle$	koncentracija polutanta, povprečena po pretoku curka (= Ψ_c/Q)
D	premer odprtine (= 0,1 m)
E	prirastek volumskega pretoka na enoto dolžine curka (= dQ/ds)
g	težni pospešek (9,81 m/s^2)
G	gibalna količina
l_B, l_M, l_Q	značilna razdalja, pri kateri je pomembnen pretok B, M , oz. Q
m, m_c	masa, masa polutanta
M, M_0	specifični pretok gibalne količine, začetne gibalne količine
N	vzgonska frekvenca
Q, Q_0	volumski pretok, začetni volumski pretok curka (= pretok pri odprtini)
Re,	Reynoldsovo število
Ri	Richardsonovo število
s, s_0	opravljena pot elementa v curku, začetna oddaljenost elementa
S, S_0, S_c	faktor redčenja, začetni faktor redčenja (pri $s = s_0$), faktor redčenja definiran kot razmerje volumnov oz. pretokov (= Q/Q_0)
S_e	relativen odklon faktorja redčenja od pričakovane vrednosti
u, u_0	hitrost elementa tekočine (na osi curka), začetna hitrost curka
dV	volumen elementa tekočine (vzgonskega curka)
(x, z)	par koordinat središča rezine vzgonskega curka glede na odprtino difuzorja
z_M, z_B	značilna višina, pri kateri sta pomembna M_0 in N, oz. B_0 in N
Δ_n, Δ_s	željena in dejanska napaka numerične metode
Δs_n, Δs_s	novi in stari pomik neodvisne spremenljivke s
$\Delta\rho$, $(\Delta\rho)_0$	gostotna razlika (= ρ_a - ρ), začetna gostotna razlika
α	koeficient vnosa okolne tekočine v vzgonski curek
ε	zgornja vrednost za relativno napako (= 10^{-10})
Γ	koeficient linearnega upadanja gostote z višino (= 0,233 kg/m^4)
λ	razmerje med polmerom, pri katerem pade hitrost na 1/e vrednosti na osi curka, in polmerom, pri katerem pade koncentracija na 1/e osrednje vrednosti
ν	koeficient kinematične viskoznosti vode (10^{-6} m^2/s)
θ, θ_0	naklonski kot, ki ga os curka oklepa z vodoravno osjo, začetni naklonski kot
ρ, ρ_a, ρ_0	gostota odplak, gostota morske vode, začetna gostota morske vode
(ρ_0)	začetna gostota odplak
ω	brezdimenzijska funkcija, ki podaja radialno odvisnost za u in $\Delta\rho$ oz. C
Ψ, Ψ_c	masni pretok odplak, masni pretok polutanta

Dodatek A. Izvorna koda programa SPLINRUN

```
{$A+,B-,D+,E-,F-,G-,I+,L+,N+,O-,R-
,S+,V+,X+}
{N+}
{$M 16384,0,655360}
PROGRAM SPLINRUN;
{ "NATURAL SPLINE":}
{ je nic v prvi in zadnji tocki }
{ made from splindr4.pas, calibrated constants}
uses DOS;
LABEL 88,99;
const
     GG = 9.806; { for the TS Gulf }
     NN = 350; { maximum dimen. of data input}
     kmax = 200; {max. number of results < kmax}
     NV = 4; { Num. of Variables }
     DepthDif = 20.7; { Mavric V,'79:}
     { Konferencija o zastiti Jadrana, Prva Knjiga,}
     {Hvar 79.}
     { Sistem  dispozicije odpadnih vod v Piranu,}
     {165-174.}
     EPS = 1.0e-10; { accuracy }
     EPS_U = 1.0e-5;
     Alphj = 0.0535;
    { Fischer in drugi, 1979, page 371: }
     Alphp = 0.0833;
     Rip = 0.557;
     { An Introduction to Water Quality Model.
A.James (Ed.)}
     { Wiley, 1984, 234 pp., page 155: }
     { Lambda = 1.16;  = 1.2 Fischer in drugi,
1979, page 370}
     Lambda = 1.14; { best numerical results }
type DatArray = ARRAY [1..NN] of real;
    glnarray = ARRAY [1..NV] OF real;
    glarray = glnarray;
    results = ARRAY [1..NV,1..NN] OF real;
var HeightArr,GamArr,y2 : DatArray;
   NumDat,Tempklo : INTEGER;
   Alpha,Ri : real; { Entrain. coeff., Ri number }
   Sm : real ; { factor of dilution }
   ZZstra, SSstra : real;
  { height & dilut. for dgamdz = const.}
   Diameter,gam, dgamdz : real;
   u0,b0,theta0,gam0,Rho0,gama  : real;
   { starting velocity of the jet, b0 its width }
   { gam0 starting density difference }
   { theta0 starting angle of the jet}
   Height0  : real;
   { heigh = height above the outfall, XX
horizontal distance }
   inpf,outf,outf2,outf3 : text;
   inpname,outname,outnam2,outnam3 : string;
   dxsav,s1,smin,SS0,Zmax,SSmax : real;
   i,kount,nbad,nok : integer;
   jj : byte;
   ystart : glnarray;
   yp: results;  { dependent variables }
   time, xp, xxp, zzp, RiArr, AlphArr : DatArray;
   VaryAlpha : char;
   DirInfo : SearchRec;

PROCEDURE ReadData(var NumDat : integer;
            var HeightArr,GamArr :DatArray);
{**************************************}
var ii : integer;
   skip, skip1,skip2,skip3,skip4 : real;
Begin
  NumDat := 0;
  while not eof(inpf) do {find the number of data
}
  begin
    readln(inpf);
    Numdat := succ(Numdat);
  end;
  reset(inpf);
  for ii := Numdat downto 1 do
  begin
    { get the arrays of heights and densities }
    { read from lingam.txt: }
     readln(inpf,HeightArr[ii],skip,GamArr[ii]);
     HeightArr[ii] := DepthDif - HeightArr[ii];
  end;
End; { ReadData }

PROCEDURE HUNT(Depth: DatArray;
    n: integer;x: real; VAR jlo: integer);
{*** NUMERICAL RECIPIES BOOK *****}
{ calculates integer jlo for x to be between }
{ Depth[klo] and Depth[klo+1] }
(* Programs using routine HUNT must define
TYPE
  DatArray = ARRAY [1..n] OF real;
in the main routine. *)
LABEL 1,2,3,4;
VAR
  jm,jhi,inc: integer;
  ascnd: boolean;
BEGIN
  ascnd := Depth[n] > Depth[1];
  IF ((jlo <= 0) OR (jlo > n)) THEN BEGIN
    jlo := 0;
    jhi := n+1;
    GOTO 3
  END;
  inc := 1;
  IF ((x >= Depth[jlo]) = ascnd ) THEN BEGIN
1:      jhi := jlo+inc;
    IF (jhi > n) THEN BEGIN
      jhi := n+1
    END ELSE IF ((x >= Depth[jhi]) = ascnd )
THEN
    BEGIN
      jlo := jhi;
      inc := inc+inc;
      GOTO 1
    END
  END ELSE BEGIN
    jhi := jlo;
2:      jlo := jhi-inc;
    IF (jlo < 1) THEN BEGIN
      jlo := 0
    END ELSE IF ((x < Depth[jlo]) = ascnd )
THEN
    BEGIN
      jhi := jlo;
      inc := inc+inc;
      GOTO 2
```

```
      END
    END;
3:   IF ((jhi-jlo) = 1) THEN GOTO 4;
    jm := (jhi+jlo) DIV 2;
    IF ((x > Depth[jm]) = ascnd ) THEN BEGIN
      jlo := jm
    END ELSE BEGIN
      jhi := jm
    END;
    GOTO 3;
4:
END; { procedure HUNT }

PROCEDURE HUNTDRHO(var yy: results;
     index, n:integer;x: real; VAR jlo: integer);
{**** NUMERICAL RECIPIES BOOK ****}
{ calculates integer jlo for x to be between }
{ yy[index,klo] and yy[index,klo+1] }
{ index - index of the variable:}
{ index = 4 for the density delta_rho }
{ n - number of data (kount) }
(* Programs using routine HUNT must define
TYPE
  DatArray = ARRAY [1..n] OF real;
in the main routine. *)
LABEL 1,2,3,4;
VAR
  jm,jhi,inc: integer;
  ascnd: boolean;
BEGIN
  ascnd := yy[index,n] > yy[index,1];
  IF ((jlo <= 0) OR (jlo > n)) THEN BEGIN
    jlo := 0;
    jhi := n+1;
    GOTO 3
  END;
  inc := 1;
  IF ((x >= yy[index,jlo]) = ascnd ) THEN
BEGIN
1:     jhi := jlo+inc;
    IF (jhi > n) THEN BEGIN
      jhi := n+1
    END ELSE IF ((x >= yy[index,jhi]) = ascnd )
THEN
    BEGIN
      jlo := jhi;
      inc := inc+inc;
      GOTO 1
    END
  END ELSE BEGIN
    jhi := jlo;
2:     jlo := jhi-inc;
    IF (jlo < 1) THEN BEGIN
      jlo := 0
    END ELSE IF ((x < yy[index,jlo]) = ascnd )
THEN
    BEGIN
      jhi := jlo;
      inc := inc+inc;
      GOTO 2
    END
  END;
3:   IF ((jhi-jlo) = 1) THEN GOTO 4;
  jm := (jhi+jlo) DIV 2;
  IF ((x > yy[index,jm]) = ascnd ) THEN BEGIN
    jlo := jm
  END ELSE BEGIN
    jhi := jm
  END;
  GOTO 3;
4:
END; { procedure HUNTDRHO }

PROCEDURE SPLINE(VAR x,y: DatArray;
     n: integer; yp1,ypn: real;VAR y2: DatArray);
{*** NUMERICAL RECIPIES BOOK *******}
{*** with SPLINES of THIRD ORDER ******}
{**makes SECOND DERIVATIVES of y in y2 }
{ given the first derivative yp1 at the beginning, }
{ and at the end of data ypn }
{ IF yp1 or ypn > 1E20 then second derivative  }
{ of that point is zero! CALL SPLINE procedure}
{ with n = Numdat }
VAR
  i,k: integer;
  p,qn,sig,un: real;
  u: DatArray;
BEGIN
  IF (abs(yp1) > 0.99e20) THEN BEGIN
    y2[1] := 0.0;
    u[1] := 0.0
  END ELSE BEGIN
    y2[1] := -0.5;
    u[1] := (3.0/(x[2]-x[1]))*((y[2]-y[1])
         /(x[2]-x[1])-yp1)
  END;
  FOR i := 2 to n-1 DO BEGIN
    sig := (x[i]-x[i-1])/(x[i+1]-x[i-1]);
    p := sig*y2[i-1]+2.0;
    y2[i] := (sig-1.0)/p;
    u[i] := (y[i+1]-y[i])/(x[i+1]-x[i])
      -(y[i]-y[i-1])/(x[i]-x[i-1]);
    u[i] := (6.0*u[i]/(x[i+1]-x[i-1])-sig*u[i-1])/p
  END;
  IF (abs(ypn) > 0.99e20) THEN BEGIN
    qn := 0.0;
    un := 0.0
  END ELSE BEGIN
    qn := 0.5;
    un := (3.0/(x[n]-x[n-1]))*(ypn-(y[n]-y[n-1])
        /(x[n]-x[n-1]))
  END;
  y2[n] := (un-qn*u[n-1])/(qn*y2[n-1]+1.0);
  FOR k := n-1 DOWNTO 1 DO BEGIN
    y2[k] := y2[k]*y2[k+1]+u[k]
  END
END; { procedure SPLINE }

PROCEDURE MakeRiAlpha(var y : GlnArray;
       VaryAlpha : char;var Ri, Alpha : real);
{*************************************}
var sqrRi : real;
Begin
 { y[1]:=u; y[2]:=b; y[3]:=theta;y[4]:=gam0;}
 SqrRi                                          :=
4.0*Sqr(Lambda)*sqrt(2.0*PI)*GG*y[2]*y[4]/
 (Rho0*(1.0+sqr(Lambda))*sqr(y[1]));
 if (y[2]*y[4] > 0) then
   Ri := Sqrt(SqrRi)
 else
   Ri := -Sqrt(-SqrRi);
 If UpCase(VaryAlpha) = 'Y' then
```

```
    If SqrRi < -200.0 then
      Alpha:= 0.0
    else
      Alpha:=
Alphj*exp(ln(Alphp/Alphj)*SqrRi/Sqr(Rip))
  else
    Alpha := Alphp;
END; { MakeRiAlpha }

PROCEDURE SPLINDER(Xcut,xklo,xkhi, yklo,
      ykhi,y2klo,y2khi: real; VAR yy,dydx: real);
{**************************************}
{ Numerical Recipes, page 110, and 123, new
edition}
{ calculates the interpol. value of the integrand }
{ for argument Xcut, saves in the variable yy, }
{ calculates the derivative dydx for Xcut. }
VAR
  kk: integer;
  hh,aa,bb : real;
BEGIN
  hh := xkhi-xklo;
  IF (hh = 0.0) THEN BEGIN
    writeln ('pause in routine SPLINT');
    writeln (' ... bad hh input; hh:',hh,
           ' xkhi:',xkhi,' xklo:',xklo);
  readln END;
  aa := (xkhi-Xcut)/hh;
  bb := (Xcut-xklo)/hh;
  yy := aa*yklo+bb*ykhi+
    ((aa*aa*aa-aa)*y2klo+(bb*bb*bb-
bb)*y2khi)*(hh*hh)/6.0;
  dydx := (Ykhi-Yklo)/hh - ((3.0*aa*aa-
1.0)*hh*y2klo/6.0)
         + ((3.0*bb*bb-1.0)*hh*y2khi/6.0);
END; { procedure SPLINDER }

PROCEDURE DERIVS(x: real; y: glnarray;
              VAR dydx: glnarray);
{**************************************}
(* CALL SPLINDER for dgamdz first.
Programs using routine DERIVS must define the
TYPE
  glnarray = ARRAY [1..4] OF real;
in the calling routine. *)
BEGIN
  dydx[1] :=
2.0*GG*Sqr(Lambda)*y[4]*sin(y[3])
          /(y[1]*Rho0) - 2.0*y[1]*Alpha/y[2];
  dydx[2] := 2.0*Alpha-
(y[2]*GG*Sqr(Lambda)*y[4]*sin(y[3])
          /(Sqr(y[1])*Rho0));
  dydx[3] :=
2.0*GG*Sqr(Lambda)*y[4]*cos(y[3])
          /(Sqr(y[1])*Rho0);
  dydx[4] :=
(1.0+Sqr(Lambda))*sin(y[3])*dgamdz
          /(Sqr(Lambda)) - 2.0*Alpha*y[4]/y[2]
END;

PROCEDURE rk4(y,dydx: glnarray; n: integer;
       x,h: real; VAR yout: glnarray);
{**************************************}
(* Programs using routine RK4 must provide
DERIVS(x:real; y:glnarray; VAR dydx:glnarray);
which returns the derivatives dydx at location x,
given both x and the function values y. The
calling
program must also define the types
TYPE
  glnarray = ARRAY [1..nvar] OF real;
where nvar is the number of variables y. *)
VAR
  i: integer;
  xh,hh,h6: real;
  dym,dyt,yt: glnarray;
BEGIN
  hh := h*0.5;
  h6 := h/6.0;
  xh := x+hh;
  FOR i := 1 to n DO BEGIN
    yt[i] := y[i]+hh*dydx[i]
  END;
  DERIVS(xh,yt,dyt);
  FOR i := 1 to n DO BEGIN
    yt[i] := y[i]+hh*dyt[i]
  END;
  DERIVS(xh,yt,dym);
  FOR i := 1 to n DO BEGIN
    yt[i] := y[i]+h*dym[i];
    dym[i] := dyt[i]+dym[i]
  END;
  DERIVS(x+h,yt,dyt);
  FOR i := 1 to n DO BEGIN
    yout[i] :=
y[i]+h6*(dydx[i]+dyt[i]+2.0*dym[i])
  END
END;

PROCEDURE RKQC(VAR y,dydx: glarray; n:
integer;
          VAR x: real;htry,eps: real; yscal:
          glarray; VAR hdid,hnext: real);
{**************************************}
(* Programs using routine RKQC must provide a
PROCEDURE DERIVS(x:real; y:glnarray; VAR
dydx:glnarray);
which returns the derivatives dydx at location x,
given
both x and the function values y. They must also
define
TYPE
  glarray = ARRAY [1..n] OF real;
in the main routine.  *)
LABEL 1;
CONST
  pgrow=-0.20;
  pshrnk=-0.25;
  fcor=0.06666666666666666666;    (* 1.0/15.0
*)
  one=1.0;
  safety=0.9;
  errcon=6.0e-4;
VAR
  i: integer;
  xsav,hh,h,temp,errmax: real;
  dysav,ysav,ytemp: glarray;
BEGIN
  xsav := x;
  FOR i := 1 to n DO BEGIN
    ysav[i] := y[i];
    dysav[i] := dydx[i]
```

```
  END;
  h := htry;
1:   hh := 0.5*h;
  { evaluate first Runge-Kutta for a step h/2:}
  rk4(ysav,dysav,n,xsav,hh,ytemp);
  x := xsav+hh;
  DERIVS(x,ytemp,dydx);
  { second step starts at x=xsav+0.5*h. }
  { Evaluate second Runge-Kutta for a step h/2:}
  rk4(ytemp,dydx,n,x,hh,y); { you have  y }
  x := xsav+h;
  IF (x = xsav) THEN BEGIN
    writeln('pause in routine RKQC');
    writeln('stepsize too small'); readln
  END;
  { Evaluate Runge-Kutta for a full step h:}
  rk4(ysav,dysav,n,xsav,h,ytemp);   {   ytemp   by
large step}
  errmax := 0.0;
  FOR i := 1 to n DO BEGIN
    ytemp[i]   :=   y[i]-ytemp[i];{   difference   in
methods}
    temp := abs(ytemp[i]/yscal[i]); { rel. error }
    IF (errmax < temp) THEN errmax := temp
  END;
  errmax := errmax/eps;
  IF (errmax > one) THEN BEGIN
    h := safety*h*exp(pshrnk*ln(errmax));
    GOTO 1 END
  ELSE BEGIN
    { h was OK, find hnext }
    hdid := h;
    IF (errmax > errcon) THEN BEGIN
      hnext := safety*h*exp(pgrow*ln(errmax))
    END ELSE BEGIN
      hnext := 4.0*h
    END
  END;
  FOR i := 1 to n DO BEGIN
    { include fifth's order correction }
    y[i] := y[i]+ytemp[i]*fcor
  END
END; { RKQC }

PROCEDURE FAECALI(VAR ystart: glnarray;
              nvar:integer; VaryAlpha : char;
              x1,x2,eps,s1,smin: real;
              VAR nok,nbad: integer);
{*************************************}
{ made of odeint, s1 guessed first step, smin
min.step }
(* Programs using routine ODEINT must provide
a
PROCEDURE DERIVS(x:real; y:glnarray; VAR
dydx:glnarray);
which returns the derivatives dydx at location x,
given
both x and the function values y. They must also
define
TYPE
  glnarray = ARRAY [1..nvar] OF real;
and declare the following parameters
VAR
  kount: integer;
  dxsav: real;
  xp,yp:   ARRAY   [1..NV,1..NN],   xxp,   zzp:
DatArray;
in the MAIN PROGRAM  *)
LABEL 99;
CONST
  maxstp=10000;
  two=2.0;
  zero=0.0;
  tiny=1.0e-30;
VAR
  nstp,i: integer;
  xsav,x,dsnext,dsdid,ds: real;
  yscal,y,dydx: glnarray;
  { Alphst : real; }
BEGIN
  x := x1;
  IF (x2 > x1) THEN ds := abs(s1) ELSE ds := -
abs(s1);
  nok := 0;
  nbad := 0;
  kount := 0{1};
  FOR i := 1 to nvar DO BEGIN
    y[i] := ystart[i]
  END;
  xp[1] := x1; { inserted on 15 September 98 }
  xsav := x-dxsav*two;
(******     START BIG FOR LOOP:     *****)
  FOR nstp := 1 to maxstp DO BEGIN
  { call spline for dgamdz(Height=zzp[kount]), }
  { which you need in DERIVS: }
    HUNT(HeightArr,        NumDat,        zzp[kount],
tempklo);
    SPLINDER(zzp[kount],HeightArr[tempklo],
          HeightArr[succ(tempklo)],

GamArr[tempklo],GamArr[succ(tempklo)],

y2[tempklo],y2[succ(tempklo)],gam0,dgamdz);
    MakeRiAlpha(y,VaryAlpha,Ri,Alpha);
    DERIVS(x,y,dydx);
    FOR i := 1 to nvar DO BEGIN
 { Scaling for monitoring accuracy - modify if
necessary }
      yscal[i] := abs(y[i])+abs(dydx[i]*ds)+tiny
    END;
    IF (abs(x-xsav) > abs(dxsav)) THEN BEGIN
      IF (kount < kmax-1) THEN BEGIN
      { store intermediate results }
        kount := kount + 1;
        xp[kount] := x;
        FOR i := 1 to nvar DO BEGIN
          yp[i,kount] := y[i]
        END;
        xsav := x;
        If kount > 1 then begin
          xxp[kount] := xxp[pred(kount)] +
              (xp[kount]-
xp[pred(kount)])*cos(y[3]);
          zzp[kount] := zzp[pred(kount)] +
              (xp[kount]-
xp[pred(kount)])*sin(y[3]);
          RiArr[kount] := Ri;
          AlphArr[kount] := Alpha;
        End; { IF kount > 1 }
      END;  { IF kount < kmax-1 }
    END; { IF (abs(x-xsav) > abs(dxsav)) }
```

```
    { if stepsize can overshoot, decrease it }
    IF (((x+ds-x2)*(x+ds-x1)) > zero) THEN ds
:= x2-x;
    (******* CALL THE ENGINE: *******)

RKQC(y,dydx,nvar,x,ds,eps,yscal,dsdid,dsnext);
    { the step dsdid was really done }
    IF (dsdid = ds) THEN BEGIN
      nok := nok+1
    END ELSE BEGIN
      nbad := nbad+1
    END;
  { Are we done?: IF (((x-x2)*(x2-x1)) >= zero)
THEN BEGIN}
  { finish when y[1] = u < 0, NOT when theta
(y[3]) < 0 }
  { as supposed in Water Quality Modelling (p.
162) }
    IF ( (((x-x2)*(x2-x1)) >= zero) OR
    ((y[3] < 0.0) AND (zzp[kount] > 0)) OR
    ((y[1] < EPS_U) AND (zzp[kount] > 0)) OR
    (zzp[kount] > Zmax) )
    THEN BEGIN
      FOR i := 1 to nvar DO BEGIN
        ystart[i] := y[i] { ystart for the }
      END;
      { save final step }
      kount := kount+1;
      xp[kount] := x;
      FOR i := 1 to nvar DO BEGIN
         yp[i,kount] := y[i]
      END;
      If kount > 1 then begin
         xxp[kount] := xxp[pred(kount)] +
           (xp[kount]-xp[pred(kount)])*cos(y[3]);
         zzp[kount] := zzp[pred(kount)] +
           (xp[kount]-xp[pred(kount)])*sin(y[3]);
         RiArr[kount] := Ri;
         AlphArr[kount] := Alpha;
      End;
      GOTO 99 { normal exit }
    END; { IF (()) >= 0 OR ... OR ... }
    IF (abs(dsnext) < smin) THEN BEGIN
      writeln('pause in routine ODEINT');
      writeln('stepsize too small'); readln;
    END;
    ds := dsnext;
  END; { for nstp }
(*******  END BIG FOR LOOP **********)
  writeln('pause in FAECALI - too many steps');
  readln;
  writeln(outf2,'pause in FAECALI - too many
steps');
99:   END; { FAECALI }

PROCEDURE
ANALYTIC(diameter,dgamdz,rho0,u0:real;
              var ZZstra,SSstra:real);
{************************************}
Const
    konst1 = 3.98;
    konst2 = 0.071;
    konst3 = 0.089;
Var gprime, N2, epsprime, QQ,delrho0 : real;
Begin
  delrho0 := rho0 - 1000.0;
  QQ := PI*u0*sqr(diameter)/4.0;
  writeln('QQ:',QQ:10:5);
  {gprime := GG*delrho0/rho0;}
  {Fischer in drugi 1979, p.345 rho0=rho fresh
water}
  {Lee H.W. and Neville-Jones, 1987,
J.Hydr.Eng., }
  { 113, 5, p. 616 rho0=rho_a=rho sea-water}
  gprime := GG*delrho0/rho0;
  writeln('delrho0:',delrho0:5:2,' rho0:',
        rho0:7:2,' gprime:',gprime:7:4);
  if abs(dgamdz) > 1e-10 then
  begin
    epsprime := -dgamdz/rho0;
    { Fischer in drugi, 1979, page 345 }
    N2 :=  GG*epsprime;
    writeln('N2:',N2:10:6);
    ZZstra := konst1*exp(0.25*ln(gprime*QQ))
    /exp(3.0*ln(N2)/8.0);
    SSstra := konst2*exp((ln(gprime) +
    5.*ln(ZZstra) - 2.*ln(QQ))/3.0);
  end
  else
  begin
   ZZstra := DepthDif;
   SSstra := konst3*exp((ln(gprime) +
   5.*ln(ZZstra) - 2.*ln(QQ))/3.0);
  end;

End;{ procedure ANALYTIC }

PROCEDURE STRATIMIDDLE;
{************************************}
Begin
  FOR i := 1 to kount DO BEGIN
    write(xp[i]:8:4,' ');
    write(outf,time[i]:5:2,' ',xp[i]:5:2);
    For jj := 1 to NV do
    begin
       write(yp[jj,i]:5:3,' ');
       if jj = 2 then
          write(outf,yp[jj,i]:6:2,' ')
       else
          write(outf,yp[jj,i]:5:2,' ');
    end;
    writeln(xxp[i]:5:2,' ',zzp[i]:5:2);
   { Sm - numerical dilution along the trajectory }
    Sm := yp[1,i]*sqr(yp[2,i])*4.0*Sqr(Lambda)/
         (U0*Sqr(Diameter)*(1.0+Sqr(Lambda)));
    writeln(outf,' ',xxp[i]:5:2,' ',zzp[i]:5:2,' ',
       xxp[i]+yp[2,i]*sin(yp[3,i]):5:2,' ',
       zzp[i]-yp[2,i]*cos(yp[3,i]):5:2,' ',
       xxp[i]-yp[2,i]*sin(yp[3,i]):5:2,' ',
       zzp[i]+yp[2,i]*cos(yp[3,i]):5:2,' ',
       Sm:5:2,' ',RiArr[i]:6:3,' ',AlphArr[i]:6:4);
  END;
  writeln(
  outf2,' zzp   Sm     y[1]    y[2]   y[3]   y[4]');
  i := 1;
  Sm := yp[1,i]*sqr(yp[2,i])*4.0*Sqr(Lambda)/
      (U0*Sqr(Diameter)*(1.0+Sqr(Lambda)));
  writeln(outf2,zzp[i]:5:2,' ',Sm:5:2,' ',
      yp[1,i]:6:4,' ',yp[2,i]:6:4,' ',yp[3,i]:6:4
      ,' ',yp[4,i]:6:4);
  write(outf3,yp[1,1]:4:2,' ');
{************************************}
```

```
  { INTERPOLATION of data for del_rho = 0 }
  {    HUNTDRHO(yp,4,    kount,0.0,tempklo);    -
already  }
  { called in the main program }
  for jj := 1 to pred(NV) do begin   { NV = 4 }
    yp[jj,tempklo] := yp[jj,tempklo] +
    (yp[4,succ(tempklo)]-
0.0)*(yp[jj,succ(tempklo)]
    -yp[jj,tempklo])/
    (yp[4,succ(tempklo)]-yp[4,tempklo]);
  end;
  yp[4,tempklo] := 0.0;
  i := tempklo;
  zzp[i] := zzp[i] +
  (yp[4,succ(i)]-0.0)*(zzp[succ(i)]-zzp[i])/
  (yp[4,succ(i)]-yp[4,i]);
  {******  END INTERPOLATION  *******}
  Sm := yp[1,i]*sqr(yp[2,i])*4.0*Sqr(Lambda)/
       (U0*Sqr(Diameter)*(1.0+Sqr(Lambda)));
  writeln(
  outf2,zzp[i]:5:2,' ',Sm:5:2,' ',yp[1,i]:6:4,' ',
     yp[2,i]:6:4,' ',yp[3,i]:6:4,' ',yp[4,i]:6:4);
  write(
  outf3,zzp[i]:7:2,' ',Sm:7:2,' ',yp[1,i]:7:2,' ',
   yp[2,i]:7:2,' ');
  Sm := yp[1,i]*sqr(yp[2,i])*4.0*Sqr(Lambda)/
        (U0*Sqr(Diameter)*(1.0+Sqr(Lambda)));
  { write a record for i = tempklo (del_rho = 0)}
  writeln
  (outf2,zzp[i]:5:2,' ',Sm:5:2,' ',yp[1,i]:6:4,' ',
          yp[2,i]:6:4,' ',yp[3,i]:6:4,' ',yp[4,i]:6:4);
  write(outf3,zzp[i]:7:2,' ',Sm:6:2,' ',yp[1,i]:5:2,' ',
        yp[2,i]:5:2,' ',yp[4,i]:7:2);
  i := kount; { write last record in other two files }
  Sm := yp[1,i]*sqr(yp[2,i])*4.0*Sqr(Lambda)/
        (U0*Sqr(Diameter)*(1.0+Sqr(Lambda)));
  writeln(outf2,zzp[i]:5:2,'  ',Sm:5:2,'  ',yp[1,i]:6:4,'
',
          yp[2,i]:6:2,' ',yp[3,i]:6:4,' ',yp[4,i]:6:4);
  write(outf3,zzp[i]:7:2,' ',Sm:7:2,' ',yp[2,i]:6:2,' ',
        yp[4,i]:4:2);
{*************************************}
{<dgamdz> from diffusser to max plume height}
  HUNT(
   HeightArr, NumDat,zzp[kount], tempklo);
  { get gama for zzp[kount] }
  SPLINDER(zzp[kount],HeightArr[tempklo],
    HeightArr[succ(tempklo)],GamArr[tempklo],
    GamArr[succ(tempklo)],y2[tempklo],
    y2[succ(tempklo)],gama,dgamdz);
  HUNT(HeightArr, NumDat,Height0, tempklo);
  { get gam0 for Height0 }
  SPLINDER(Height0,HeightArr[tempklo],
    HeightArr[succ(tempklo)],GamArr[tempklo],
    GamArr[succ(tempklo)],y2[tempklo],
    y2[succ(tempklo)],gam0,dgamdz);
  dgamdz := (gama-gam0)/(zzp[kount]-Height0);
  writeln('<dgamdz>:',dgamdz:10:5);
  {** call experimental estimation of ZZstra **}
  {** and SSstra - fixed dgamdz **}
  ANALYTIC(
  diameter,dgamdz,rho0,u0,ZZstra,SSstra);
  writeln(outf2,ZZstra:5:2,' ',SSstra:5:2);
  { i = kount }
  writeln(
  outf3,ZZstra:7:2,' ',abs(ZZstra-zzp[i])*100.0
  /ZZstra:5:2,' ',SSstra:5:2,' ',abs(SSstra-Sm)
  *100.0/SSstra:5:2);
  writeln('SSstra:',SSstra:8:2,'
ZZstra:',ZZstra:5:2);
  writeln('U0:',U0:4:2, ' theta0(deg):',theta0*180.
  /PI:5:2,' Zmax:',Zmax:5:2,' U:',yp[1,kount]:5:3,
  ' theta(deg):',yp[3,kount]*180./PI:5:2,' Zend:',
  zzp[kount]:5:2);
End; { STRATIMIDDLE }

PROCEDURE STRATISURFACE;
{*************************************}
var Sm_temp : real;
Begin
  HUNT(zzp,kount,    DepthDif,tempklo);{zzp    in
FAECALI}
  FOR i := 1 to tempklo DO BEGIN
    write(xp[i]:8:4,' ');
    write(outf,time[i]:5:2,' ',xp[i]:5:2);
    For jj := 1 to NV do
    begin
       write(yp[jj,i]:5:3,' ');
       if jj = 2 then
         write(outf,yp[jj,i]:6:2,' ')
       else
         write(outf,yp[jj,i]:5:2,' ');
    end;
    writeln(xxp[i]:5:2,' ',zzp[i]:5:2);
   { Sm - numerical dilution along the trajectory
}
    Sm := yp[1,i]*sqr(yp[2,i])*4.0*Sqr(Lambda)/
         (U0*Sqr(Diameter)*(1.0+Sqr(Lambda)));
    writeln(outf,' ',xxp[i]:5:2,' ',zzp[i]:5:2,' ',
         xxp[i]+yp[2,i]*sin(yp[3,i]):5:2,' ',
         zzp[i]-yp[2,i]*cos(yp[3,i]):5:2,' ',
         xxp[i]-yp[2,i]*sin(yp[3,i]):5:2,' ',
         zzp[i]+yp[2,i]*cos(yp[3,i]):5:2,' ',
         Sm:5:2,' ',RiArr[i]:6:3,' ',AlphArr[i]:6:4);
  END; { FOR i = 1 TO tempklo }
  Sm_temp := Sm;
  { Sm for zzp[tempklo] < depthDif - from FOR
loop above}
  {*  LINEAR INTERPOL. of last data for zzp =
DepthDif *}
  i := tempklo;
  for jj := 1 to NV do begin   { NV = 4 }
   yp[jj,i] := yp[jj,i] +
   (zzp[succ(i)]-DepthDif)*(yp[jj,succ(i)]-
yp[jj,i])
   /(zzp[succ(i)]-zzp[i]);
   writeln(jj,' ',yp[jj,i]);
  end;
  time[i] := time[i] + (zzp[succ(i)]-DepthDif)*
  (time[succ(i)]-time[i])/(zzp[succ(i)]-zzp[i]);
  xp[i] := xp[i] + (zzp[succ(i)]-DepthDif)*
  (xp[succ(i)]-xp[i])/(zzp[succ(i)]-zzp[i]);
  xxp[i] := xxp[i] + (zzp[succ(i)]-DepthDif)*
  (xxp[succ(i)]-xxp[i])/(zzp[succ(i)]-zzp[i]);
  writeln('xp:',xp[i],' xxp:',xxp[i]);
  writeln('zzp:',zzp[i],' succzzp:',zzp[succ(i)]);
  { calculate Sm for zzp > DepthDif }
  Sm := yp[1,succ(i)]*sqr(yp[2,succ(i)])*4.0*
    Sqr(Lambda)/(U0*Sqr(Diameter)*
    (1.0+Sqr(Lambda)));
  {**** interpolate Sm for zzp = DepthDif ***}
  Sm := Sm_temp + (zzp[succ(i)]-DepthDif)*
```

```
    (Sm-Sm_temp)/(zzp[succ(i)]-zzp[i]);
  write(xp[i]:8:4,' ');
  write(outf,time[i]:5:2,' ',xp[i]:5:2);
  For jj := 1 to NV do
    begin
      write(yp[jj,i]:5:3,' ');
      if jj = 2 then
        write(outf,yp[jj,i]:6:2,' ')
      else
        write(outf,yp[jj,i]:5:2,' ');
    end;
  writeln(xxp[i]:5:2,' ',DepthDif:5:2);
  writeln(outf,' ',xxp[i]:5:2,' ',DepthDif:5:2,' ',
       xxp[i]+yp[2,i]*sin(yp[3,i]):5:2,' ',
       DepthDif-yp[2,i]*cos(yp[3,i]):5:2,' ',
       xxp[i]-yp[2,i]*sin(yp[3,i]):5:2,' ',
       DepthDif+yp[2,i]*cos(yp[3,i]):5:2,' ',
       Sm:5:2,' ',RiArr[i]:6:3,' ',AlphArr[i]:6:4);
{*************************************}
  writeln(outf2,
  ' zzp   Sm     y[1]    y[2]   y[3]   y[4]');
  i := 1;
  Sm_temp                                    :=
yp[1,i]*sqr(yp[2,i])*4.0*Sqr(Lambda)/
    (U0*Sqr(Diameter)*(1.0+Sqr(Lambda)));      {
first record }
  writeln(outf2,zzp[i]:5:2,'            ',Sm_temp:5:2,'
',yp[1,i]:6:4,
    ' ',yp[2,i]:6:4,' ',yp[3,i]:6:4,' ',yp[4,i]:6:4);
  write(outf3,yp[1,1]:4:2,' ');
  i := tempklo;
  {  write  down  already  interpolated  data  for
DepthDif }
  writeln(
  outf2,DepthDif:5:2,' ',Sm:5:2,' ',yp[1,i]:6:4,' ',
    yp[2,i]:6:4,' ',yp[3,i]:6:4,' ',yp[4,i]:6:4);
  write(
  outf3,DepthDif:7:2,' ',Sm:7:2,' ',yp[1,i]:6:2,' ',
    yp[2,i]:7:2,' ');
 { <dgamdz> from diffusser to max plume height}
  HUNT(HeightArr,                NumDat,DepthDif,
tempklo);
  { tempklo no more related to zzp -the height of}
  { solution, but related to HeightArr - the height
  { of  the records of density. }
  { get gama for zzp[kount] }
  SPLINDER(zzp[kount],HeightArr[tempklo],
    HeightArr[succ(tempklo)],GamArr[tempklo],
    GamArr[succ(tempklo)],

y2[tempklo],y2[succ(tempklo)],gama,dgamdz);
  HUNT(HeightArr, NumDat,Height0, tempklo);
  { get gam0 for Height0 }
  SPLINDER(Height0,HeightArr[tempklo],
    HeightArr[succ(tempklo)],GamArr[tempklo],
    GamArr[succ(tempklo)],y2[tempklo],
    y2[succ(tempklo)], gam0,dgamdz);
  dgamdz := (gama-gam0)/(zzp[kount]-Height0);
  writeln('<dgamdz>:',dgamdz:10:5);
  {** call  estimate  of  ZZstra  and  SSstra  -  fixed
dgamdz **}
  ANALYTIC(
   diameter,dgamdz,rho0,u0,ZZstra,SSstra);
  writeln(outf2,ZZstra:5:2,' ',SSstra:5:2);
  HUNT(zzp,kount, DepthDif,tempklo);
   {get original tempklo }
  i := tempklo;
  writeln(outf3,
     ZZstra:7:2,' ',abs(ZZstra-zzp[i])*100.0
     /ZZstra:5:2,' ',SSstra:5:2,' ',abs(SSstra-Sm)*
     100.0/SSstra:5:2);
  writeln('SSstra:',SSstra:8:2,'
ZZstra:',ZZstra:5:2);
  writeln('U0:',U0:4:2, ' theta0(deg):',theta0*180.
     /PI:5:2,'
Zmax:',Zmax:5:2,'U:',yp[1,kount]:5:3,
     ' theta(deg):',yp[3,kount]*180./PI:5:2,' Zend:',
     zzp[kount]:5:2);
End; { STRATISURFACE }


BEGIN { MAIN program  }
{*************************************}
{**** BIGI-BIGI, NO BUGGY************}
  write('Vary Alpha (y/n):');
  readln(VaryAlpha);
  repeat
    write
    ('vstavi ime datoteke z gostotami (gam.txt):');
    readln(inpname);
    FindFirst(inpname, Archive,DirInfo);
  until DosError = 0;
  assign(inpf,inpname);
  reset(inpf);
  ReadData(NumDat,HeightArr,GamArr);
  close(inpf);
  {******   IMENA DATOTEK ZA IZPIS ***}
  write
  ('vstavi    ime    izhodne    datoteke    za    risanje
podatkov:');
  readln(outname);
  writeln('vstavil si:',outname,' . Pritisni enter:');
  assign(outf,outname);
  rewrite(outf);
  write('vstavi ime izhodne datoteke za bistvene
  podatke:');
  readln(outname);
  writeln('vstavil si:',outname,' . Pritisni enter:');
  assign(outf2,outname);
  rewrite(outf2);
  write('vstavi ime izhodne datoteke za
   primerjavo podatkov:');
  readln(outname);
  writeln('vstavil si:',outname,' . Pritisni enter:');
  assign(outf3,outname);
  FindFirst(outname, Archive,DirInfo);
  If DosError = 0 then
     append(outf3) { create it first }
  else rewrite(outf3);
 {****** ZACETEK Z ZLEPKI **********}
  SPLINE(HeightArr,GamArr,Numdat,
            1.1E20,1.1E20, y2);
  HUNT(HeightArr, NumDat, Height0, tempklo);
  { get gam0 for ZZ = 0.0: }
  SPLINDER(Height0,HeightArr[tempklo],
     HeightArr[succ(tempklo)],GamArr[tempklo],
     GamArr[succ(tempklo)],y2[tempklo],
     y2[succ(tempklo)],gam0,dgamdz);
 {*****  ZACETNI PODATKI ***********}
  Diameter := 0.1; { [m] diameter of the orifice }
  SS0 := 6.2*Diameter;
  u0 := 1.0; { [m/s] Malacic,'97: 'Geofi-ekoloski'}
```

```
  b0 := Diameter/Sqrt(2.0);
{ Water Quality Model, p.155 }
  theta0 := PI*0.0/180.;
 { SS0 starting distance [m] along the jet }
 { ystart[1] = velocity along the jet axis }
 { ystart[2] = half-width of a buoyant jet }
 { ystart[3] = angle of jet with respect to XX }
 { ystart[4] = delta_rho = rho_ambient - rho_jet }
  ystart[1] := u0;
  ystart[2] := b0;
  ystart[3] := theta0;
  ystart[4] := gam0*(1.0 + Sqr(Lambda))/(2.0*Sqr(Lambda));
  { suppose a sewage with the density of 1000 kg/m^3,  }
  { delta_rho0 = gam0, also 'Water Quality Model.',p.154 }
  Rho0 := 1000.0 + gam0;
  xxp[1] := 6.2*Diameter*cos(theta0);
{ starting distance }
  Height0 := 0.0;
{ starting height above the diffusser }
  zzp[1] := Height0+6.2*Diameter*sin(theta0);
  for i := 2 to NN do begin
     zzp[i] := 0.0;
     xxp[i] := 0.0;
     yp[4,i] := 0.0;{ del_rho=0 - for later IF tempklo }
  end;
  s1 := 0.001; { guessed first stepsize }
  smin := 0.0; {1.0E-5;} { minimum step size }
  SSmax := 100.0;{ not known final distance along the jet }
  Zmax := DepthDif; { HeightArr[NumDat] may be > DepthDif }
  dxsav := 0.1;{recorded points are separated by ds > 0.1 }
  MakeRiAlpha(ystart,VaryAlpha,Ri,Alpha);
  RiArr[1] := Ri; AlphArr[1] := Alpha;
(** CALL THE BEST FAECALI ENGINE **)
{*************************************}
FAECALI(
  ystart,NV,VaryAlpha,SS0,SSmax,eps,s1,smin,
  nok,nbad);
  writeln;{ nok = number of good steps, }
  { nbad = number of bad (repaired) steps }
  { kount = number of stored (written) values }
  writeln('successful steps:',' ':13,nok:3);
  writeln('bad steps:',' ':20,nbad:3);
  writeln('stored intermediate values:',' ',kount:3);
  writeln;
  For i := 1 to NN do
    time[i] := 0.0;
  For i :=1 to pred(kount) do
   time[succ(i)] := time[i] +
   (xp[succ(i)]-xp[i])*2.0/(yp[1,succ(i)]+yp[1,i]);
  { tempklo >= kount? }
  { If YES Then zzp(del_rho=0) IS NOT }
 {WITHIN the WATER COLUMN => }
 {STRATISURFACE, ELSE STRATIMIDDLE}
  HUNTDRHO(yp,4, kount,0.0,tempklo);
  IF TEMPKLO < KOUNT THEN begin
    writeln('REMAINED BELOW.
    Oki-Doki. Press Return!'); readln;
    STRATIMIDDLE
  end
  ELSE begin
    writeln('GONE TO THE SURFACE,
   Bye-bye. Press Return!'); readln;
    STRATISURFACE;
  end;
  writeln(tempklo);
  close(outf); close(outf2);close(outf3);
END. { program }
```

Literatura

Fan L. N., and N. H. Brooks, 1966. Horizontal jets in stagnant fluid of other density. J. Hydraul. Div. Proc. Am. Soc. Civ. Eng., 92, Hy2, 423-429.

Fischer, H. B., E. J. List, R. C. Y. Koh, J. Imberger, and N. H. Brooks, 1979. Mixing in Inland and Coastal Waters. Academic Press, New York, 483 pp.

Featherstone, R. E., 1984. Mathematical models of the discharge of wastewater into a marine environment. *An Introductory to Water Quality Modelling*, Ed. A. James, 1st ed., John Wiley, Chichester, 150-162

Fonseka, S. V., H. J. S. Fernando, and G. J. F. van Heijst, 1998. Evolution of an isolated turbulent region in a stratified fluid. J. Geophy. Res., 103, (C11), 24,857-24,868.

Kolar, J., 1983. Odvod odpadne vode iz naselij in zaščita voda. DZS, Ljubljana, str. 523.

Malačič V., 1991. Estimation of the vertical eddy diffusion coefficient of heat in the Gulf of Trieste (Northern Adriatic), Oceanol. Acta, 14, 23-32.

Malačič, 1997. Geofizikalno-ekološki pristop k disperziji odplak piranskega izpusta. Fazno poročilo I, Inštitut za biologijo, Morska biološka postaja Piran, 21 str.

Mavrič V., 1979. Sistem dispozicije odpadnih vod v Piranu. V: Konferencija o zašiti Jadrana, Vol. 1, Hvar 1979, 165-174.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1988. Numerical Recipes in C, Cambridge Univ. Press, 735 pp.

Quetin B., and M. de Rouville, 1986. Submarine sewer outfalls - a design manual. Mar. Poll. Bull., 17, 4, 133-183.

UNEP, 1995. Guidelines for Submarine Outfall Structures for Mediterranean Small and Medium-Sized Coastal Communities. Working document, UNEP(OCA)/MED WG. 89/Inf.6, 34 pp.