Usporedba implementacija Bühlmannovog modela dekompresije u ronjenju

Student: Fran Borčić

Mentor: Nikola Mišković

FER, 28. siječnja 2019.

# Sadržaj

Sadržaj 2

Uvod 3

Dekompresija u ronjenju 4

Bühlmannov model dekompresije 5

Implementacije Bühlmannovog modela 7

Numerička integracija jednadžbe 8

Implementacija pomoću Schreinerove jednadžbe 10

Implementacija podjelom na konstantne intervale 12

Implementacija cjelobrojnim računanjem 14

Odjeljak kao LTI sustav 17

Zaključak 19

Dodaci 20

Dodatak: Ispitno sučelje 20

Dodatak: Kod za izračun dekompresijskog profila 23

Dodatak: Definicije konstanti korištenih u kodu 25

Popis literature 26

# Uvod

Ronioci u današnje vrijeme uvelike ovise o tehnologiji koja im omogućuje da uz minimalne rizike dosežu dubine i trajanja ronjenja koja nikada prije nisu bila dohvatljiva. Vjerojatno najvažnije dostignuće u ronilačkoj tehnologiji su modeli dekompresije i ronilačka računala. Cilj ovog projekta je razmotriti različite implementacije Bühlmannovog dekompresijskog modela koji je danas vjerojatno najrašireniji model u tehničkom ronjenju.

Bühlmannov model unatoč svojoj jednostavnosti uživa veliku popularnost jer je 80tih godina prošlog stoljeća objavljen javno za razliku od većine drugih modela čiji detalji nisu dostupni javnosti. Zbog svoje otvorenosti, Bühlmannov je model intenzivno ispitan u praksi i empirijska saznanja pokazuju da je izrazito siguran.

Model je moguće implementirati na više različitih načina, od kojih se neke razmatraju u ovom projektu. Ne postoji univerzalno najbolji način implementacije, već vrsta implementacije ovisi o primjeni, o čemu će detaljnije biti riječi kasnije.

Ideja ovog projekta je da postane podloga za razvoj ronilačkog računala i računalne podrške za planiranje urona.

# Dekompresija u ronjenju

Krajem 19. stoljeća, primjećeno je da radnici u tzv. kesonima[[1]](#footnote-1) koji su izloženi visokom tlaku nakon povratka na atmosferski često obolijevaju od dotad neopisane bolesti. U oboljelih su primjećeni bolovi u zglobovima, osip, a u težim slučajevima i trajna oštećenja živčanog sustava. Liječnici su ubrzo shvatili da se radi o emboliji uzrokovanoj mjehurićima plinova koji se stvaraju u tjelesnim tekućinama uslijed pada tlaka, ali dugo vremena nije bilo poznato kako tome doskočiti.

Danas spomenutu bolest zovemo dekompresijska bolest i predmet je mnogih istraživanja i radova. Kroz posljednjih stotinjak godina, sakupljena su mnoga saznanja o uzrocima, vrstama i prevenciji dekompresijske bolesti, ali je nepoznanica i dalje puno i prostora za daljnja istraživanja ne nedostaje.

Ključni alat u prevenciji dekompresijske bolesti je dekompresijski model. Zadatak dekompresijskog modela je da roniocu pruži proceduru izrona koja će mu umanjiti rizik od oboljenja na prihvatljivu razinu. Kroz povijest istraživanja dekompresije, razvili su se različiti dekompresijski modeli, ali se u praksi zadržala nekolicina. Razlog tome leži u činjenici da je razvoj modela prilično skup s obzirom na ekstremno tešku validaciju – modele moraju ispitati različite vrste ronioca na različitim profilima urona, pritom prihvaćajući rizik od oboljenja od dekompresijske bolesti.

Srećom, s popularizacijom ronjenja, etablirani su modeli ispitani izrazito veliki broj puta i poznato je kolika je incidencija dekompresijske bolesti u ljudi koji ih koriste. Općenito se smatra da rizik od dekompresijske bolesti nikada nećemo potpuno eliminirati zbog individualnih fizioloških razlika, kao i da je rizik za one koji prate dekompresijske režime prema nekom od prihvaćenih modela prihvatljivo mali.

U nastavku ćemo razmotriti Bühlmannov model dekompresije koji je najrašireniji model u tehničkom ronjenju.

# Bühlmannov model dekompresije

Bühlmannov model dekompresije razvio je švicarski profesor medicine Albert Bühlmann (1923-1994) početkom druge polovice dvadesetog stoljeća nastavljajući istraživanja svojih prethodnika Johna Scotta Haldanea i Roberta Workmana.

Osnovna ideja njegovog modela, kao i modela njegovih prethodnika je sljedeća. Dušik[[2]](#footnote-2) se otapa u tjelesnim tkivima brzinom proporcionalnom razlici parcijalnog tlaka dušika u zraku kojeg ronioc udiše i parcijalnog tlaka dušika otopljenog u tkivu. Po istom principu se dušik i oslobađa pri izronu, ali u slučaju da se počne oslobađati prebrzo, nastaju mjehurići koji mogu prouzročiti dekompresijsku bolest.

Faktor ovisnosti brzine otapanja plina i razlike tlakova različit je za različita tkiva, pa se u modelu modelira niz hipotetskih tkiva koja će se u daljnem tekstu zvati odjeljci. Odjeljak je, kao što je to uobičajeno u eksponencijalnim modelima, definiran vremenom koje je potrebno da na konstantnom tlaku razlika tlaka plina u tkivu i okolini padne na pola. Osim toga, Bühlmann za svaki odjeljak uvodi dvije konstante, a i b, koje u ovisnosti o parcijalnom tlaku otopljenog plina određuju minimalni apsolutni tlak okoline koji odjeljak može podnjeti bez nastanka mjehurića. U daljnjem tekstu ćemo taj tlak zvati *tlak dekompresijskog stropa*, a dubinu pri kojoj se taj tlak ostvaruje *dekompresijski strop.*

Formalno, ponašanje svakog odjeljka definirano je diferencijalnom jednadžbom:



Gdje je *P* parcijalni tlak otopljenog plina u odjeljku, Pi parcijalni tlak plina u udahnutom zraku ili smjesi, a *k* konstanta definirana kao:



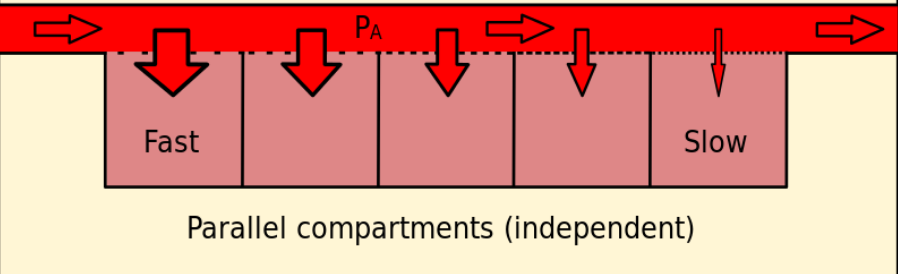
gdje je *th* gore opisano poluvrijeme.

Pri izronu, za odjeljak u kojem je parcijalni tlak inertnog plina jednak *Psat*, tlak dekompresijskog stropa određen je izrazom:



Gdje su *a* i *b* konstante koje je Bühlmann za svaki odjeljak definirao empirijski kroz svoje istraživanje.

U Bühlmannovom modelu opisanih je odjeljaka 16, a svaki predstavlja neku vrstu tkiva u tijelu. Odjeljci s kratkim vremenima zasićenja predstavljaju “brza tkiva” poput krvi, dok odjeljci s dugim vremenima zasićenja predstavljaju “spora tkiva” poput masnog tkiva. Odjeljci su međusobno neovisni sustavi, a ovisno o profilu urona ukupni dekompresijski strop određuje onaj odjeljak čiji je dekompresijski strop najdublji.



Slika 1 Ilustracija Bühlmannovog modela (izvor: Wikipedia)

Odjeljci Bühlmannovog modela određeni su sljedećim konstantama:

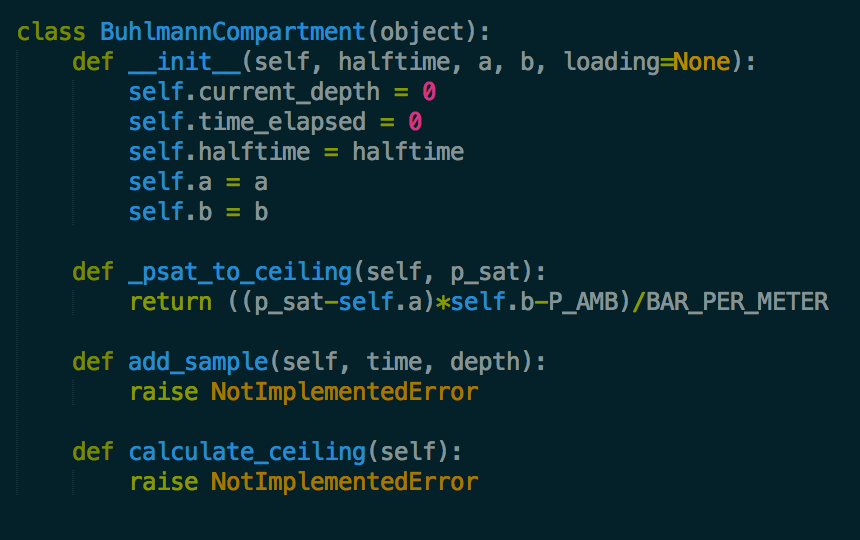
|  |  |  |
| --- | --- | --- |
| th (min) | a (bar) | b |
| 5.0 | 1.1696 | 0.5578 |
| 8.0 | 1.0 | 0.6514 |
| 12.5 | 0.8618 | 0.7222 |
| 18.5 | 0.7562 | 0.7825 |
| 27.0 | 0.62 | 0.8126 |
| 38.3 | 0.5043 | 0.8434 |
| 54.3 | 0.441 | 0.8693 |
| 77.0 | 0.4 | 0.891 |
| 109.0 | 0.375 | 0.9092 |
| 146.0 | 0.35 | 0.9222 |
| 187.0 | 0.3295 | 0.9319 |
| 239.0 | 0.3065 | 0.9403 |
| 305.0 | 0.2835 | 0.9477 |
| 390.0 | 0.261 | 0.9544 |
| 498.0 | 0.248 | 0.9602 |
| 635.0 | 0.2327 | 0.9653 |

Tablica 1 Konstante koje definiraju odjeljke (ZHL16C)

U praksi se pojavljuje još nekoliko varijanti navedenog skupa konstanti. U ovom sam se projektu odlučio za ovaj (poznat pod imenom ZHL16B) jer je najčešće korišten u računalnoj podršci za planiranje urona, pa tako i u programu *Subsurface* koji sam koristio za validaciju svoje implementacije.

# Implementacije Bühlmannovog modela

U nastavku je razmotreno pet pristupa implementaciji navedenog modela, uz prednosti i nedostatke, kao i prikaz rezultata u ispitnom slučaju za jednostavan profil urona. Implementacije se razlikuju u načinu implementacije odjeljka, dok je (trivijalni) algoritam za računanje profila zajednički u svim pristupima. Iz tog će razloga za svaki pristup biti izdvojen Python kod specifičan za taj odjeljak. Ispis koda vezanog za traženje profila dan je kasnije, a bazna klasa iz koje se izvodi svaki odjeljak je ovdje:



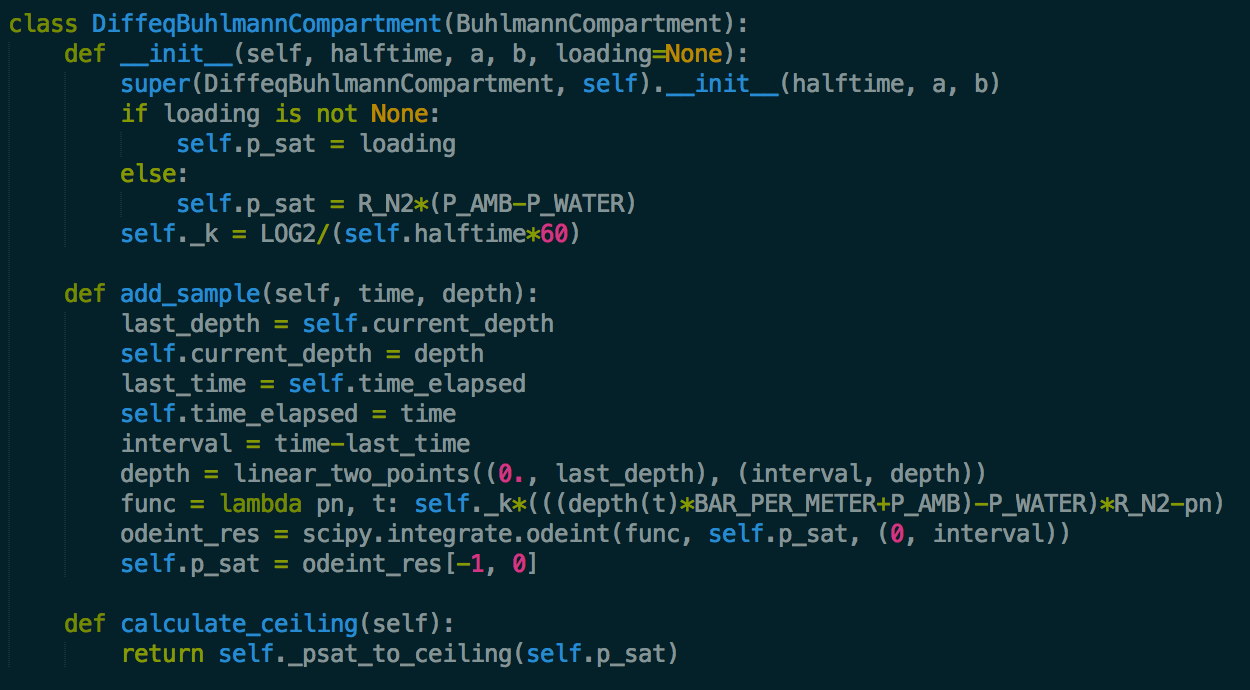
Slika 2 Bazna klasa odjeljka BuhlmannCompartment

Svaki odjeljak treba implementirati javne metode *add\_sample* i *calculate\_ceiling*. Prva metoda omogućava učitavanje para (*vrijeme*, *dubina*) koji se odnosi na dubinu izmjerenu neki broj sekundi nakon početka urona. Druga metoda treba vratiti trenutni dekompresijski strop u metrima. Bazna klasa definira i zaštićenu metodu *psat\_to\_ceiling* koja pretvara parcijalni tlak dušika u dekompresijski strop, koju izvedena klasa može i ne mora koristiti.

Definicije svih konstanti koje se koriste u kodu u nastavku dane su na kraju dokumenta.

## Numerička integracija jednadžbe

Prva implementacija služi se bibliotekom za numeričku integraciju *scipy.integrate* i direktno implementira gore opisanu diferencijalnu jednadžbu. Ovakva implementacija nije od prevelike praktične važnosti, ali implementira model prema temeljnom opisu i zbog toga može služiti za validaciju ostalih implementacija.



Slika 3 Implementacija direktnim integriranjem

Integriranje se, kao što je vidljivo iz priloženog koda, odvija svaki put kad se u odjeljak doda novi par (vrijeme, dubina). Odjeljak pamti zadnje vrijeme i dubinu te linearno interpolira između prethodnog i novog para. Osim toga, pamti vrijednost parcijalnog tlaka i pri svakom dodavanju novog mjerenja ažurira tu vrijednost.

Ovdje vrijedi napomenuti da se u svim implementacijama pri računanju tlaka dušika u udahnutom zraku mora kompenzirati za tlak vodene pare u plućima. Taj je tlak konstantan i ovdje definiran u konstanti *P\_WATER*.



Slika 4 Prikaz rezultata direktnog integriranja u ispitnom sučelju

## Implementacija pomoću Schreinerove jednadžbe

Gore opisanu diferencijalnu jednadžbu za neke je slučajeve moguće riješiti analitički. Jedan od takvih slučajeva je kad je funkcija *Pi* linearna funkcija i tada rješenje glasi:

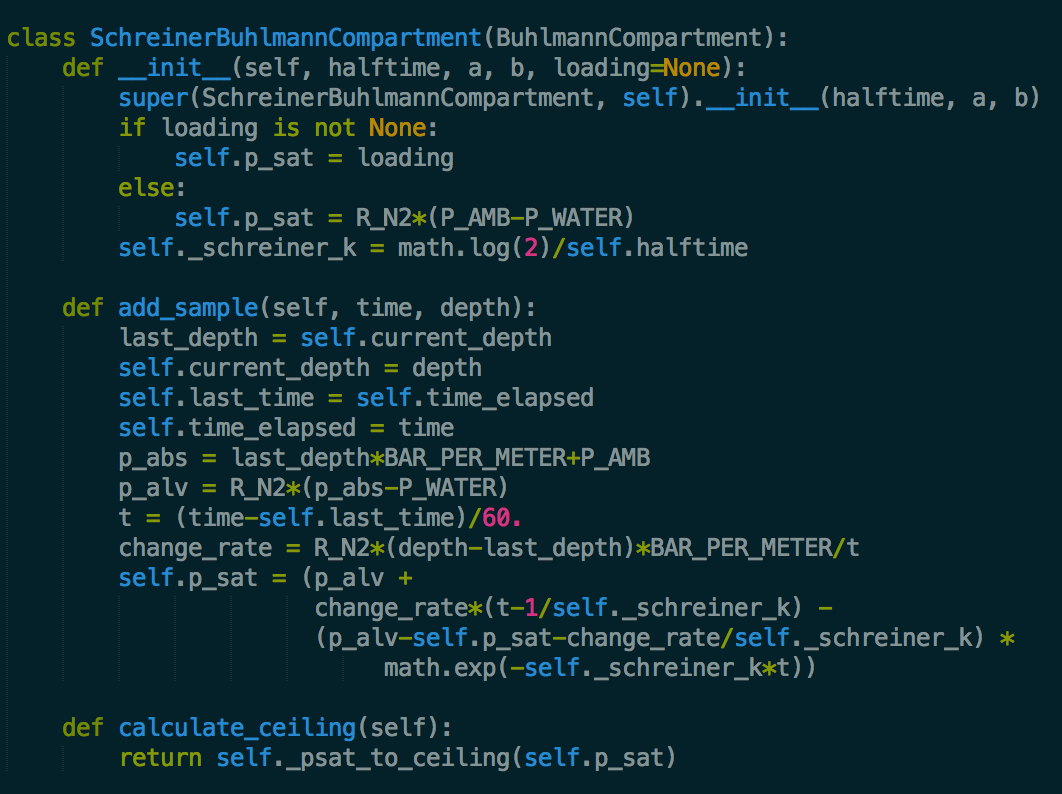


gdje je:

.

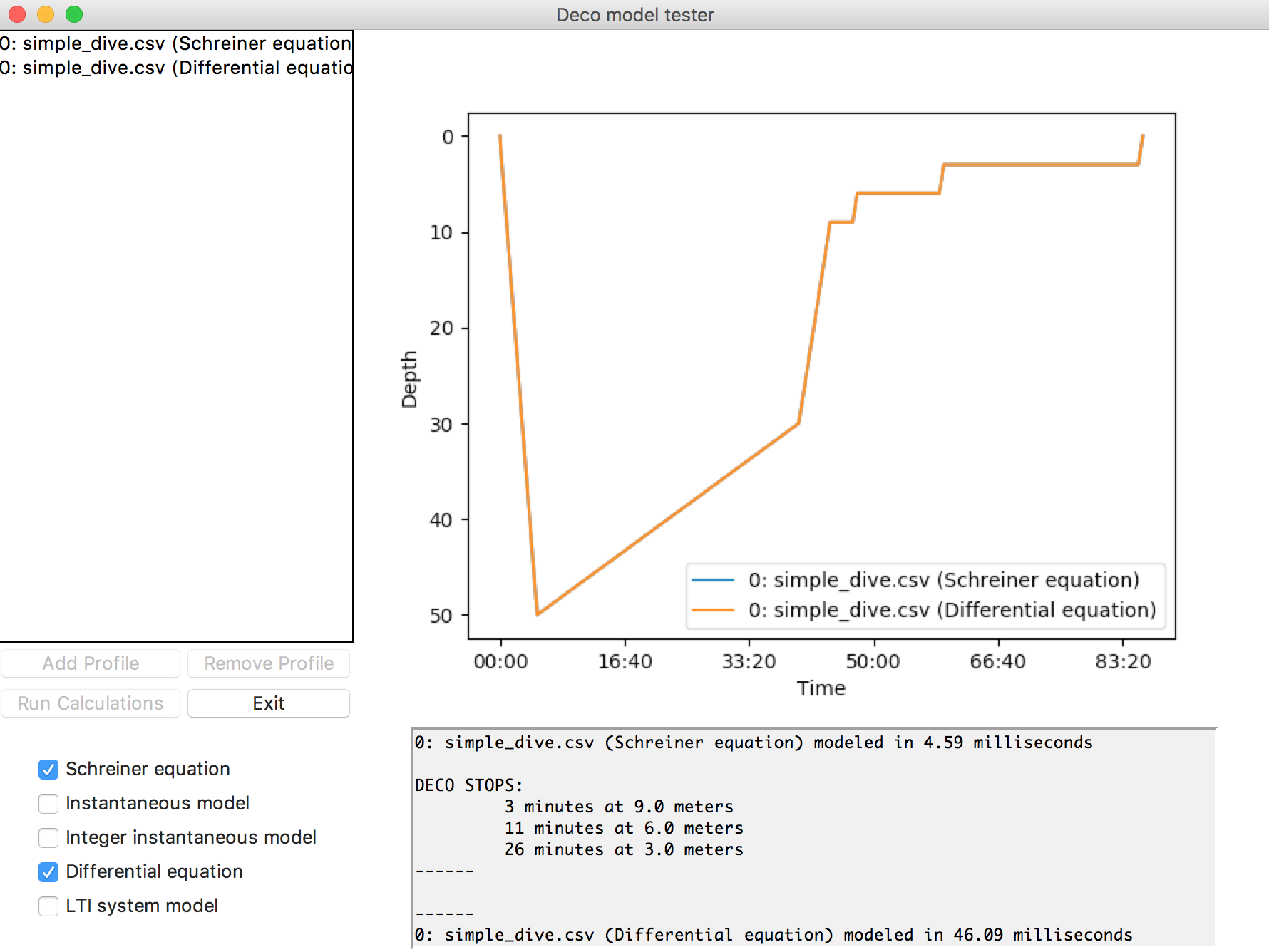
Taj je rezultat u literaturi iz ovog područja poznat kao Schreinerova jednadžba.

S obzirom na činjenicu da su planirani uroni najčešće po dijelovima linearne funkcije, ovaj je oblik idealan za računanje dekompresije u računalnim programima za planiranje urona. U implementaciji koja slijedi, za svaki se linearni dio profila računa završni parcijalni tlak dušika u odjeljku koji potom postaje vrijednost *P(0)* za idući linearni dio:



Slika 5 Implementacija odjeljka Schreinerovom jednadžbom

U ispitnom je sučelju vidljivo da se rezultati ove implementacije i implementacije numeričkim integriranjem potpuno poklapaju, ali da je ova implementacija za slučaj jednostavnog ispitnog profila deseterostruko brža:



Slika 6 Usporedba Schreinerovog modela s direktnim integriranjem

## Implementacija podjelom na konstantne intervale

Dok je prethodno opisana implementacija idealna za izračune pri planiranju urona, to nije slučaj za obradu realnih, izmjerenih profila urona. Pri snimanju profila urona, ronilačko računalo periodički, u vrlo kratkim vremenskim intervalima mjeri dubinu i zapisuje ju uparenu s dubinom. Pri tome bez prevelikog gubitka preciznosti može pretpostaviti da je ronioc za vrijeme cijelog intervala bio na izmjerenoj dubini.

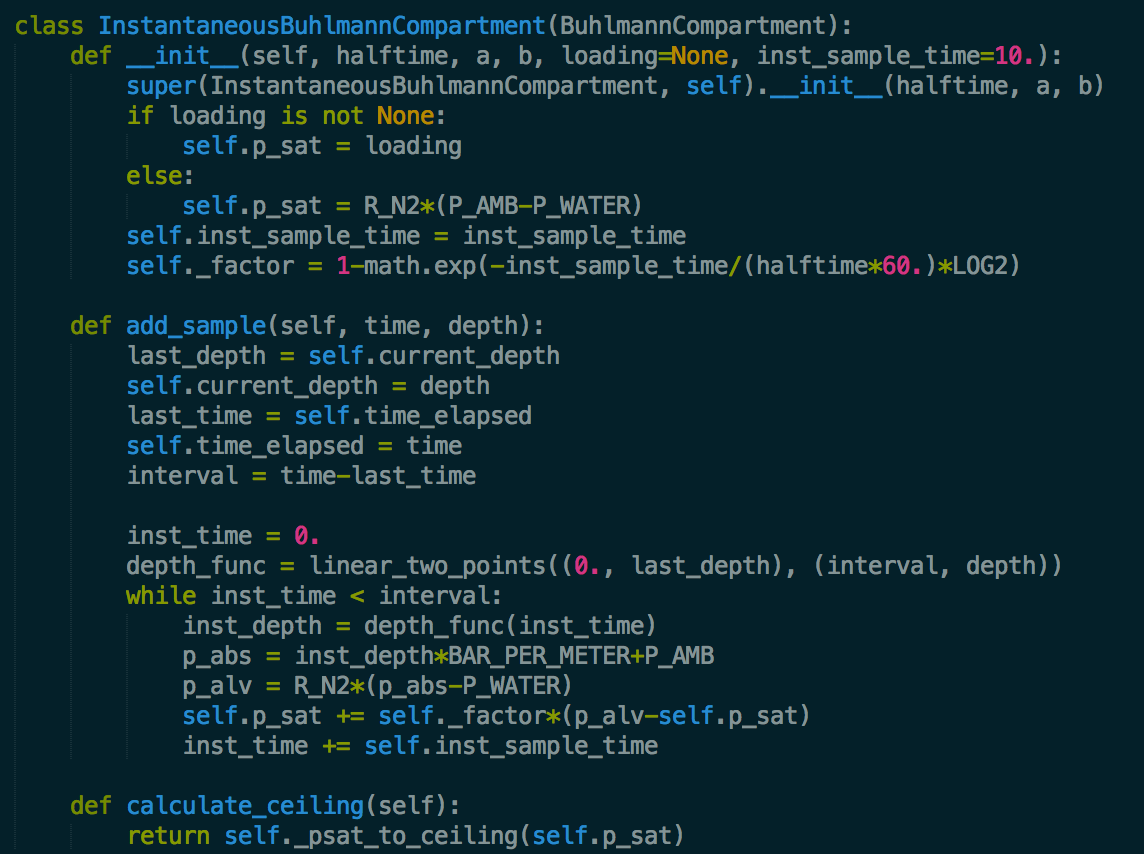
U opisanom slučaju funkcija dubine više nije samo po dijelovima linearna, već i po dijelovima konstantna. Za konstantnu funkciju u navedenoj Schreinerovoj jednadžbi imamo:



i tada ona poprima oblik:

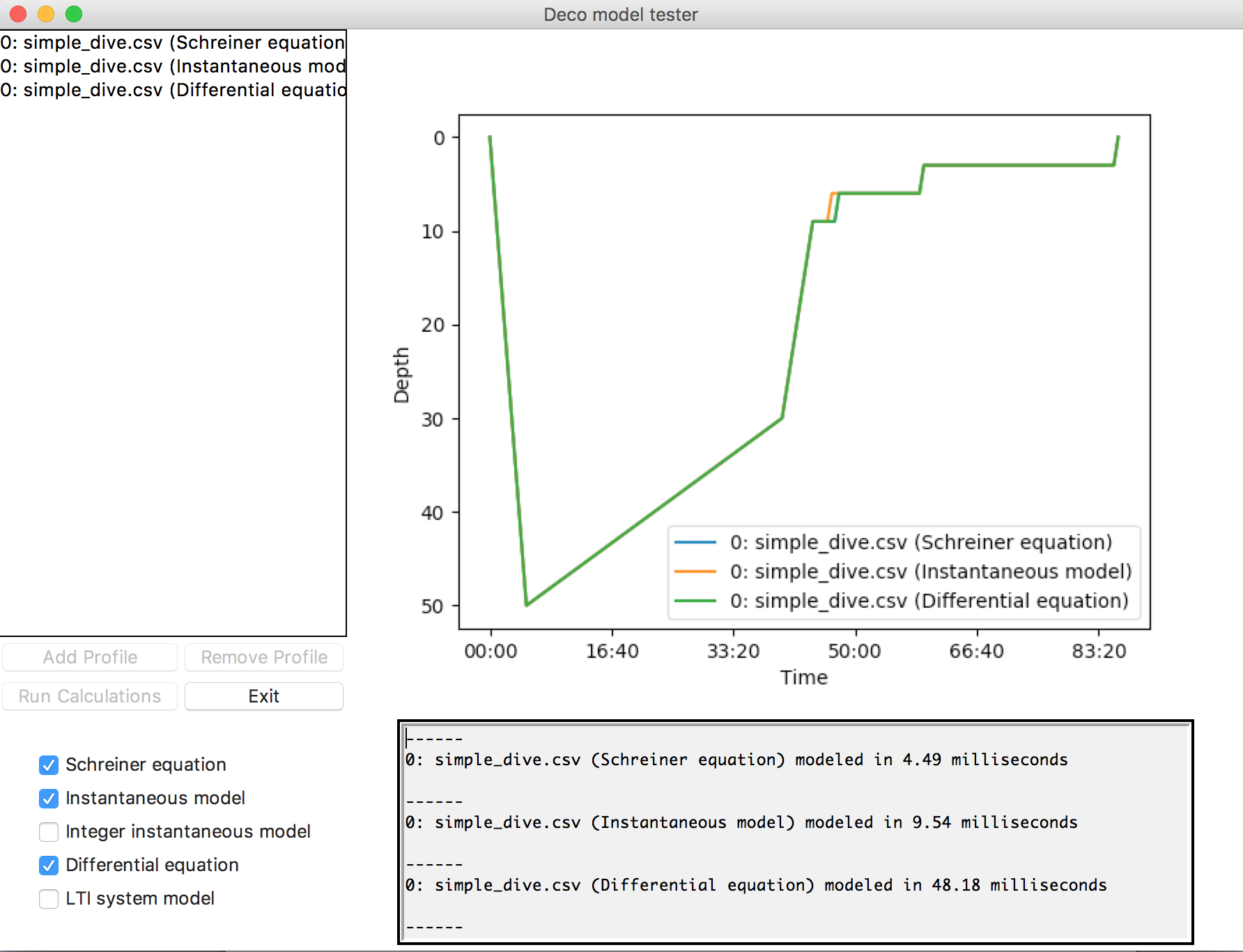
.

Ovaj je oblik izuzetno praktičan jer u opisanom slučaju *t* uvijek poprima istu vrijednost (razmak između uzoraka). Iz tog je razloga eksponencijalni dio jednadžbe moguće izračunati unaprijed za određeni odjeljak i tako izbjeći računanje eksponencijalne funkcije koje je računski skupo.



Slika 7 Implementacija podjelom na konstantne intervale

Mjerenjem u ispitnom slučaju vidljivo je da je za slučaj jednostavnog profila urona ovakav način računanja sporiji od računanja Schreinerovom jednadžbom. Razlog tome leži u činjenici da je profil na kojem je implementacija ispitana planirani, po dijelovima linearni profil, a ne izmjereni.



Slika 8 Računanje konstantnim intervalima u ispitnom sučelju

Ovaj je model pogodan za implementaciju u ronilačkim računalima s mogućnošću računanja s brojevima s pomičnom točkom (floating point arithmetic).

## Implementacija cjelobrojnim računanjem

Prethodno opisani model, iako pogodan za uređaje koji mogu računati s brojevima s pomičnom točkom, nije moguće implementirati na uređajima koji nemaju tu mogućnost. S obzirom na činjenicu da većina ronilačkih računala mora minimizirati potrošnju električne energije, pri njihovoj se proizvodnji često koriste primitivni mikrokontroleri koji računaju isključivo s cijelim brojevima.

Uz neka praktična ograničenja moguće je prilagoditi gore opisani algoritam tako da koristi isključivo cijele brojeve. U ovoj implementaciji se pokazuje da su 32 bitni cijeli brojevi dovoljne preciznosti za takvo računanje.

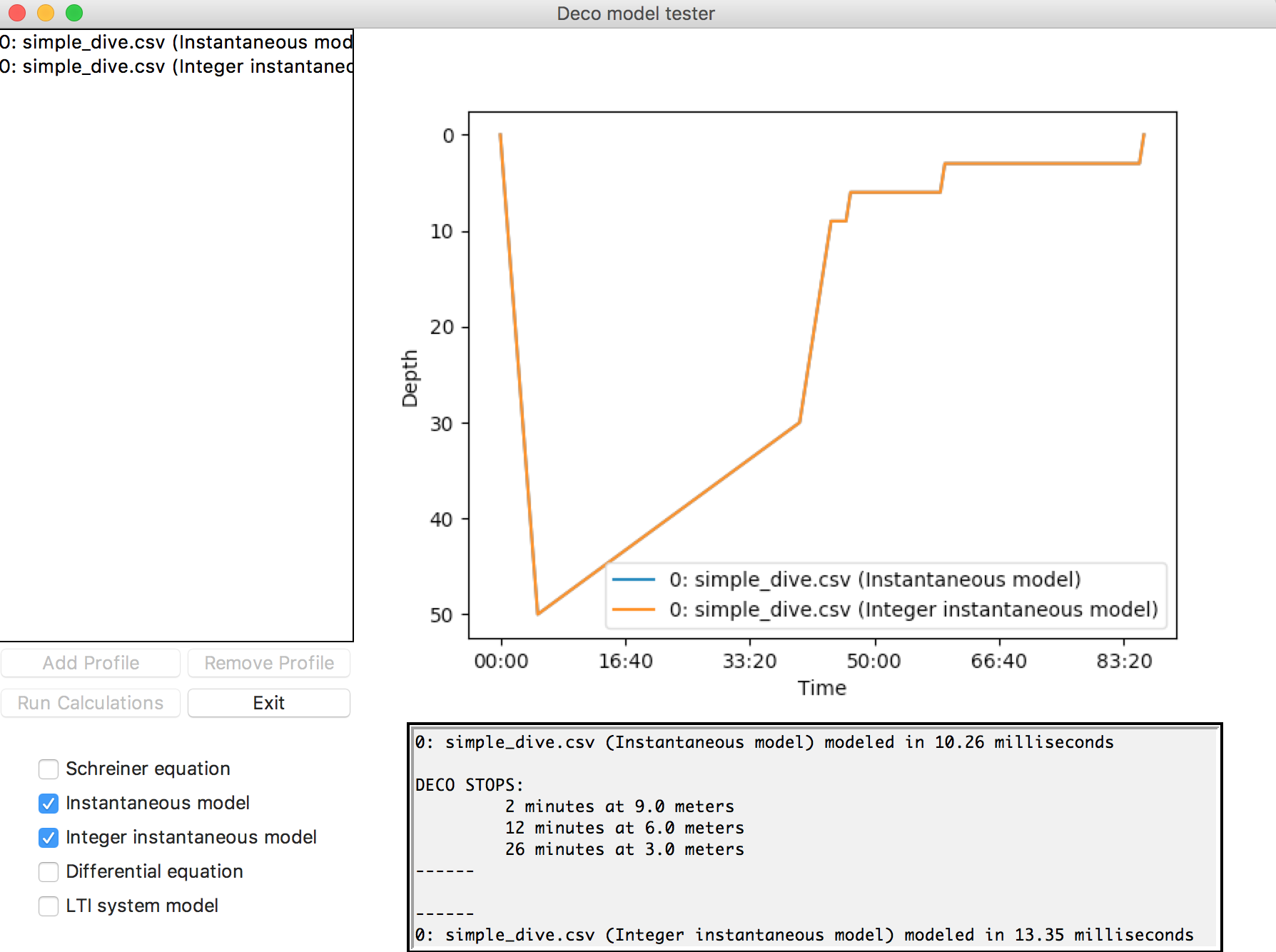
Prilagodbu je moguće izvršiti na sljedeći način – odredi se maksimalni tlak koji je moguć u bilo kojem od odjeljaka i njemu se dodijeli neka velika vrijednost koju je moguće zapisati u cijeli broj unaprijed određene veličine (u ovom slučaju 32 bita). Svi tlakovi u računu se skaliraju na novoodređenu jedinicu, a faktori koji se koriste u računu pretvaraju u cjelobrojne razlomke kojima je nazivnik unaprijed određen tako da je brojnik moguće zapisati u dvostruko manji cijeli broj (u ovom slučaju 16 bita).

U sljedećem ulomku koda prikazana je takva implementacija, s tim da su cjelobrojne konstante unaprijed definirane i prikazane s ostatkom konstanti na kraju ovog dokumenta.



Slika 9 Cjelobrojna implementacija

Iz prikaza u ispitnom sučelju vidi se da ova implementacija ni po čemu ne zaostaje za prethodnom za profil u realnim granicama.



Slika 10 Usporedba cjelobrojne i floating-point implementacije

## Odjeljak kao LTI sustav

Posljednja implementacija razmotrena unutar ovog projekta je implementacija u kojoj se odjeljak razmatra kao linearan vremenski nepromjenjiv (LTI) sustav. Promotrimo li model, vidimo da odjeljak odgovara LTI sustavu prvog reda kojemu je prijenosna funkcija:

.

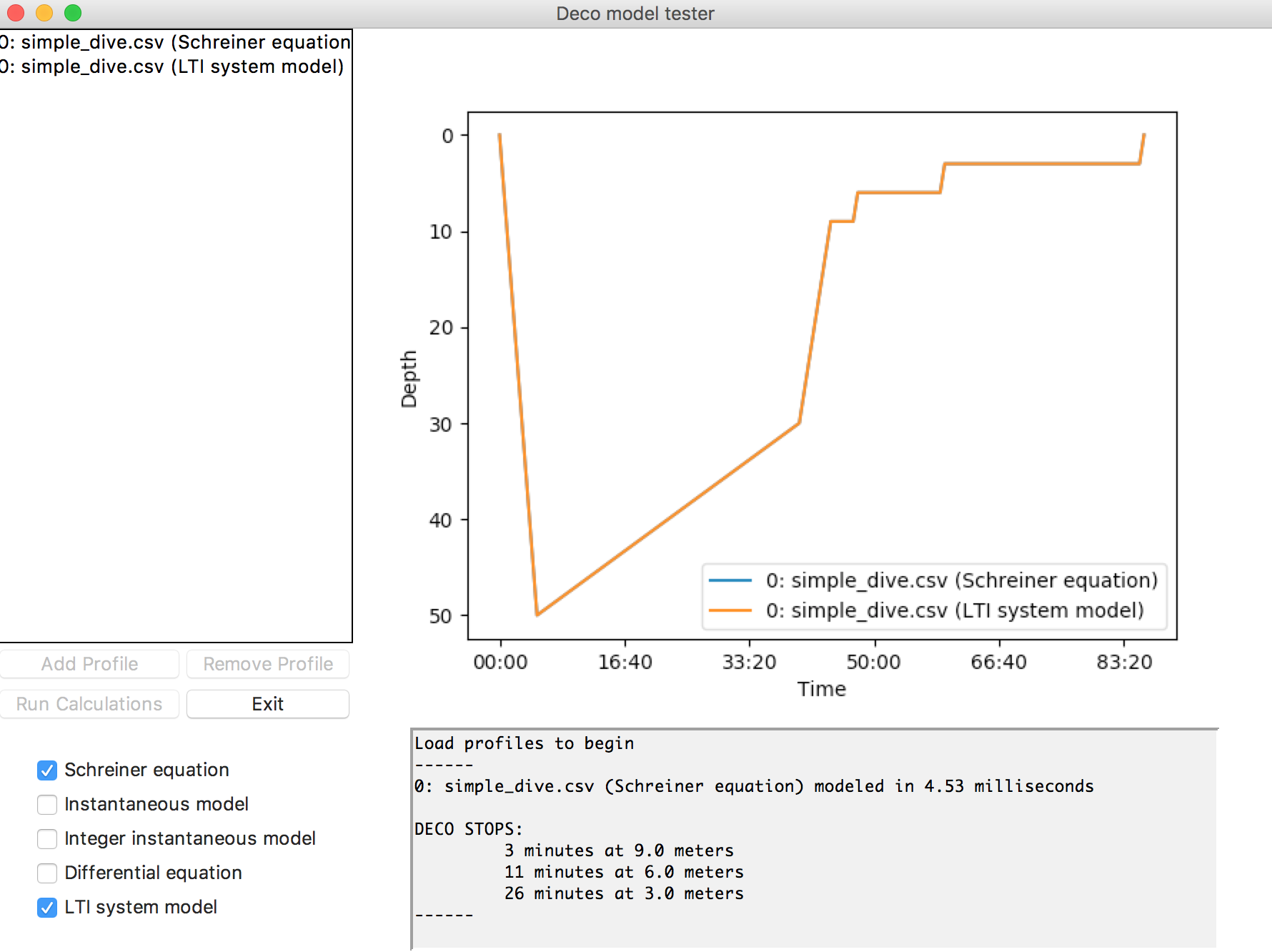
Pobuda sustava je parcijalni tlak dušika u udahnutom plinu, a izlaz sustava parcijalni tlak plina otopljenog u odjeljku.

Ova je implementacija zanimljiva jer daje teorijski pristup modelu kakav, koliko je autoru poznato, nije razmatran u literaturi. S praktičnog aspekta, promatranje odjeljka na ovaj način omogućava nam njegovo modeliranje analognim elektroničkim sklopom, pa tako i koncept analognog ronilačkog računala. Takvo računalo bi, unatoč ograničenoj funkcionalnosti, moglo predstavljati mnogo pouzdaniju alternativu onim digitalnima.

Za potrebe ostvarivanja LTI sustava u ovoj implementaciji se koristi programski paket za obradu signala *scipy.signal.*



Slika 11 Implementacija odjeljka kao LTI sustava



Slika 12 Usporedba LTI sustava sa Schreinerovim modelom

# Zaključak

Iz razmotrenih implementacija modela vidljivo je da ne postoji univerzalno optimalna implementacija te se može steći dojam kako ga je pogodno implementirati za koju primjenu. Različite implementacije mogu dati rezultate koji se u maloj mjeri razlikuju. Te razlike u praksi nisu važne jer se u praktičnoj primjeni ovog modela uvodi dodatni faktor sigurnosti u obliku faktora gradijenta koji roniocu omogućava da po volji određuje konzervativnost. Ta je modifikacija algoritma trivijalna i kao takva izvan opsega ovog projekta.

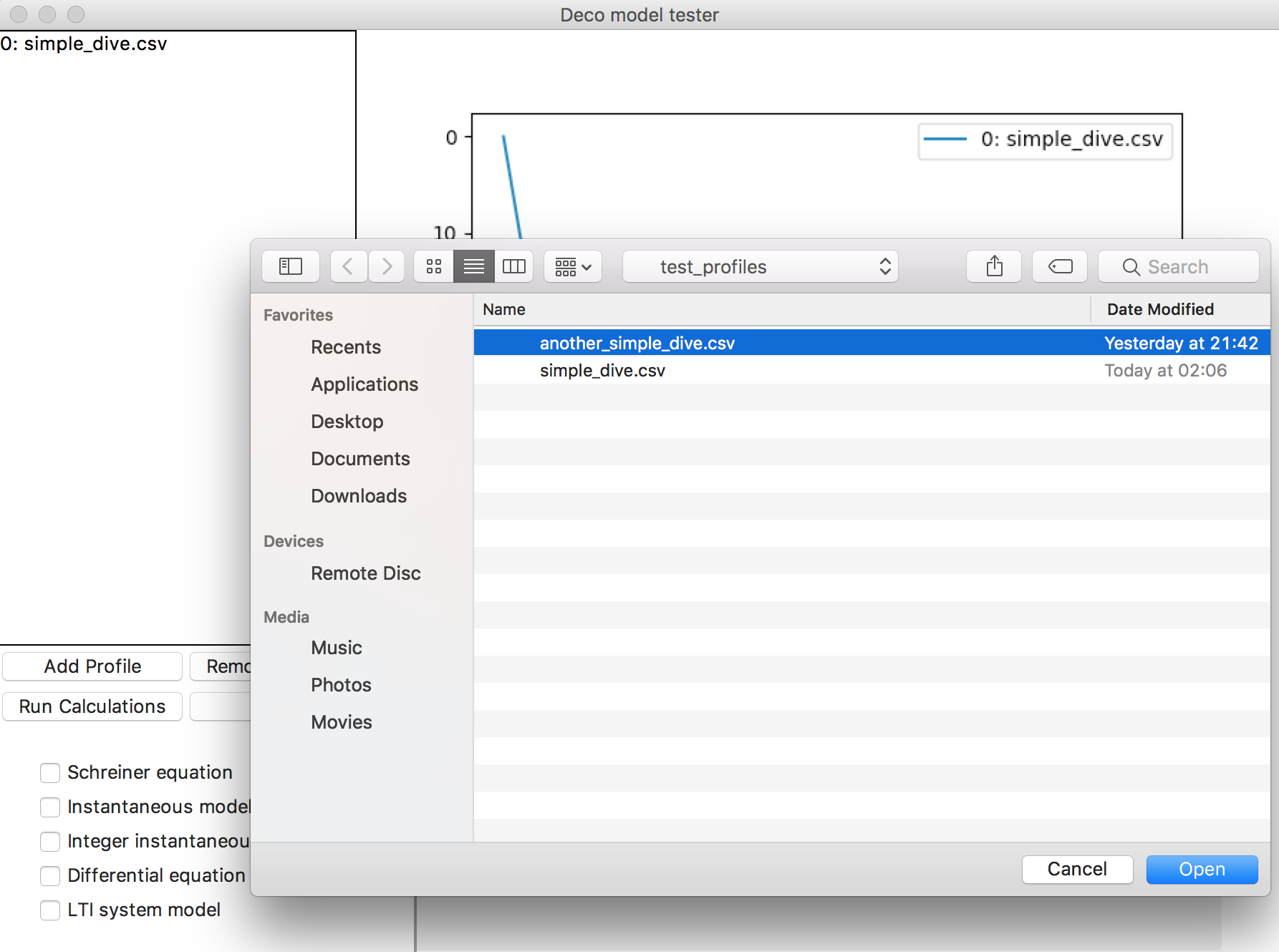
Razmatranja iz ovog projekta mogla bi poslužiti kao podloga praktičnoj implementaciji Bühlmannovog modela u ronilačkim računalima, izračunu tablica ili planiranju urona na osobnom računalu.

# Dodaci

## Dodatak: Ispitno sučelje

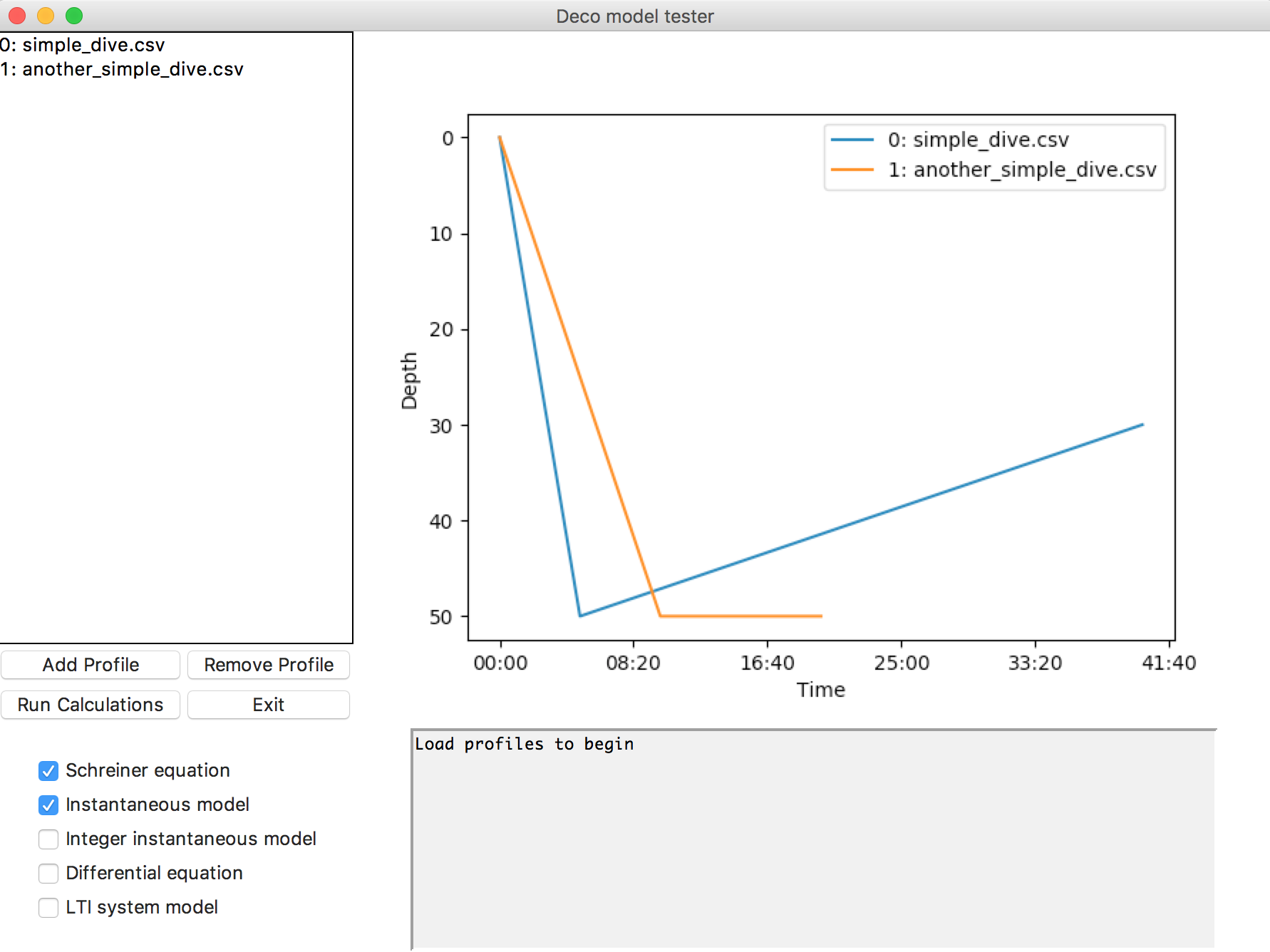
U sklopu ovog projekta razvijeno je jednostavno sučelje za ispitivanje modela i prikaz njihovih rezultata. Ispitno se sučelje pokreće pokretanjem python datoteke *gui.py* iz repozitorija projekta.

Sučelje se koristi u nekoliko koraka. Najprije se učita jedan ili više profila urona za koje je potrebno izračunati dekompresijski profil:



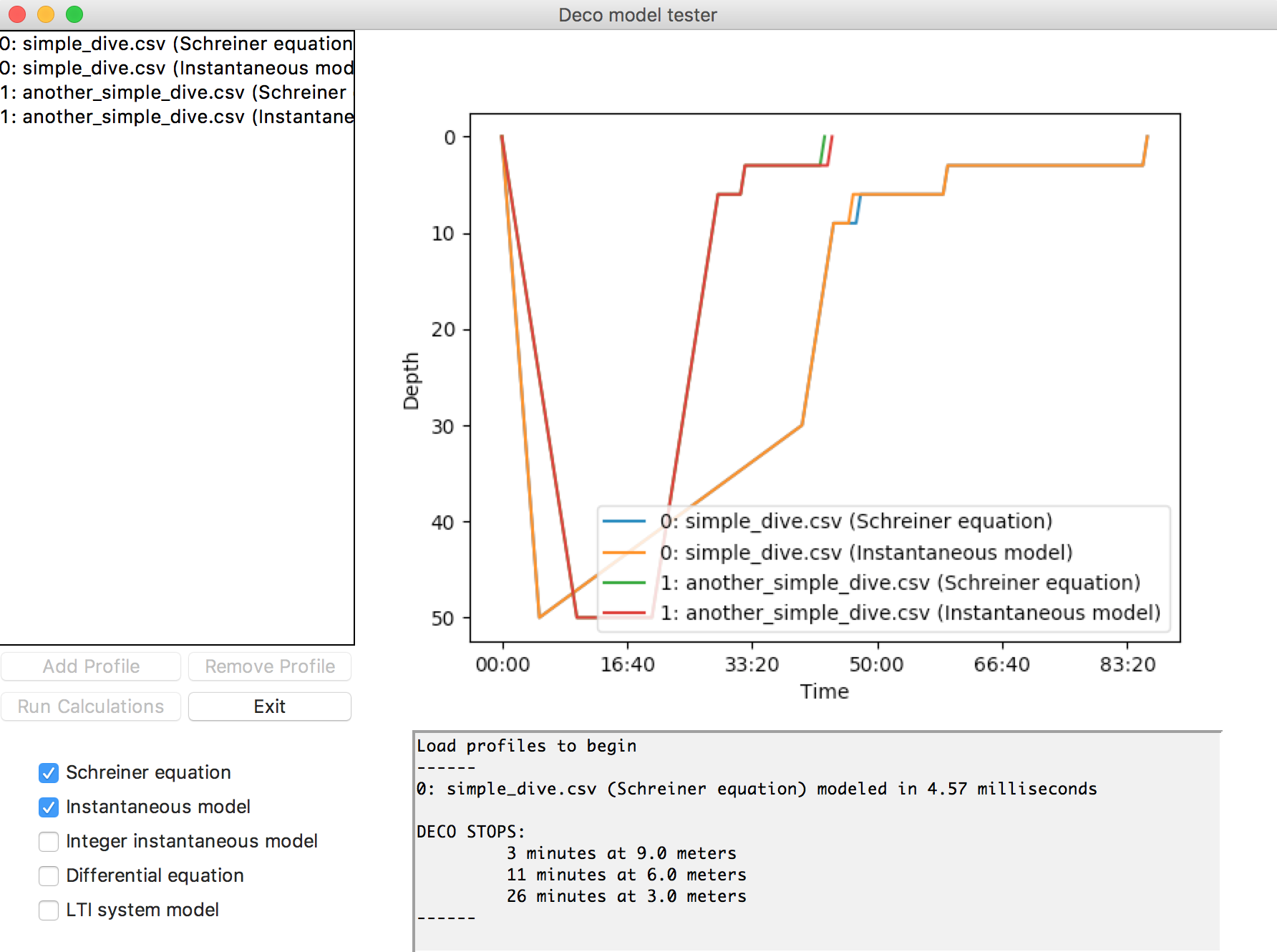
Slika 13 Učitavanje profila u ispitno sučelje

Potom se u donjem lijevom uglu odabere jedan ili više dekompresijskih modela. Dostupni dekompresijski modeli definirani su u datoteci *available\_models.py* koju je moguće proširiti.



Slika 14 Odabir dekompresijskih modela u ispitnom sučelju

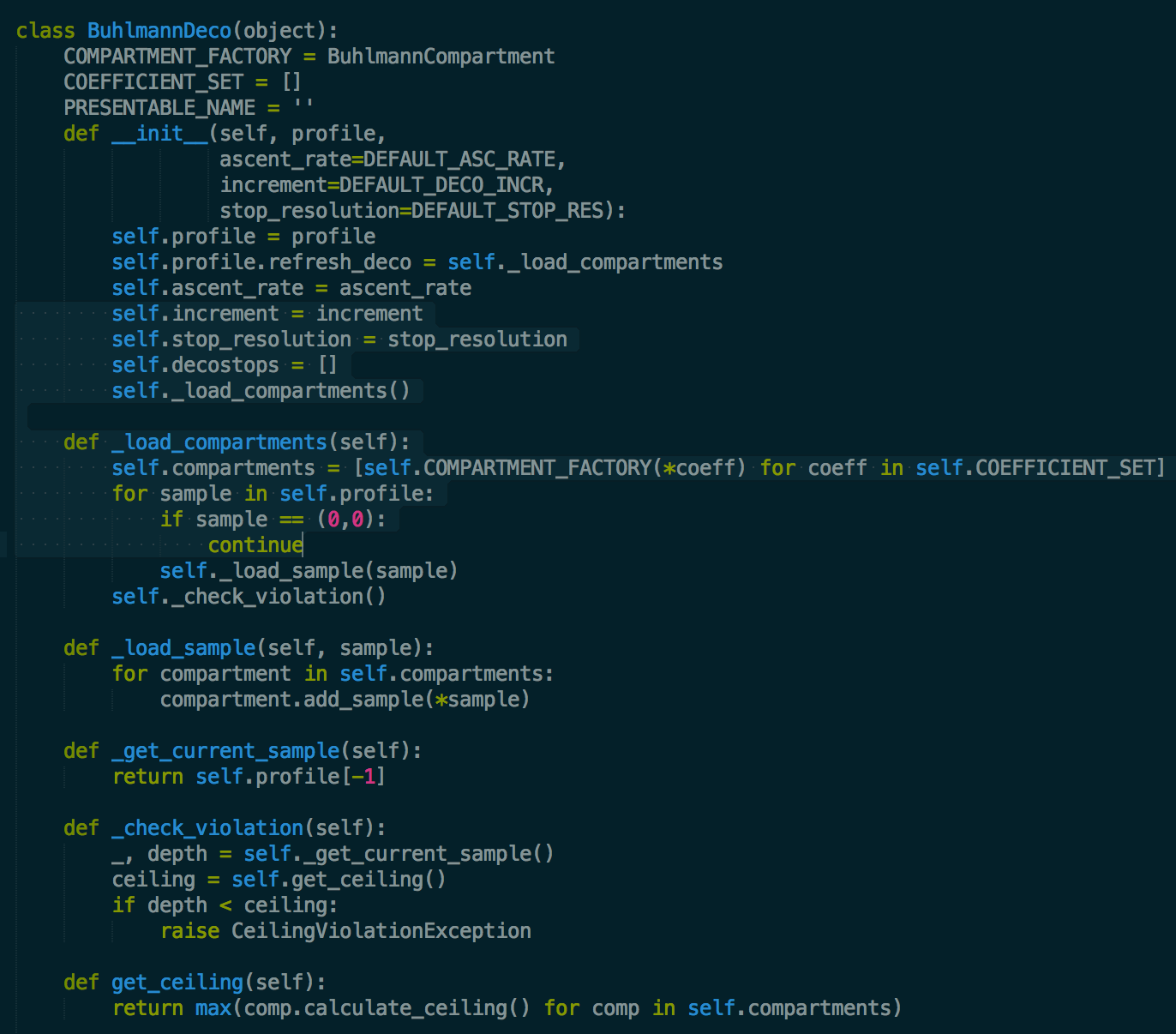
Pritiskom na gumb “Run Calculations” pokreće se računanje modela i rezultati se prikazuju na grafu i u tekstualnom polju ispod njega.



Slika 15 Prikaz rezultata u ispitnom sučelju

## Dodatak: Kod za izračun dekompresijskog profila

Pri izračunu dekompresijskog profila koristi se simulacija izrona konstantnom brzinom. Ako se pritom naiđe na dekompresijski strop koji je bliže od tri metra trenutnoj dubini, simulira se dekompresijski zastanak dok izron za sljedeća tri metra ne bude siguran.



Slika 16 Kod za izračun profila (prvi dio)

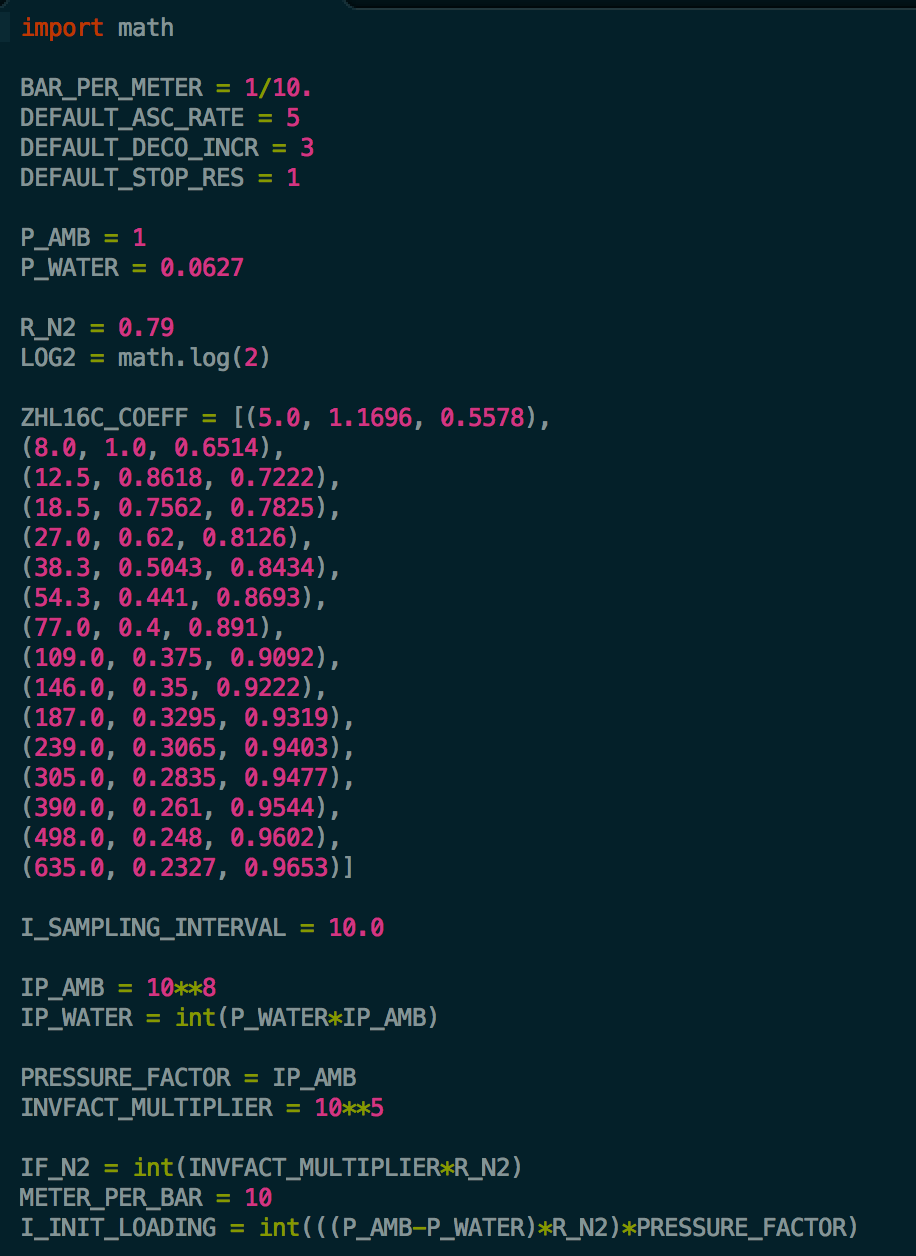


Slika 17 Kod za izračun profila (drugi dio)

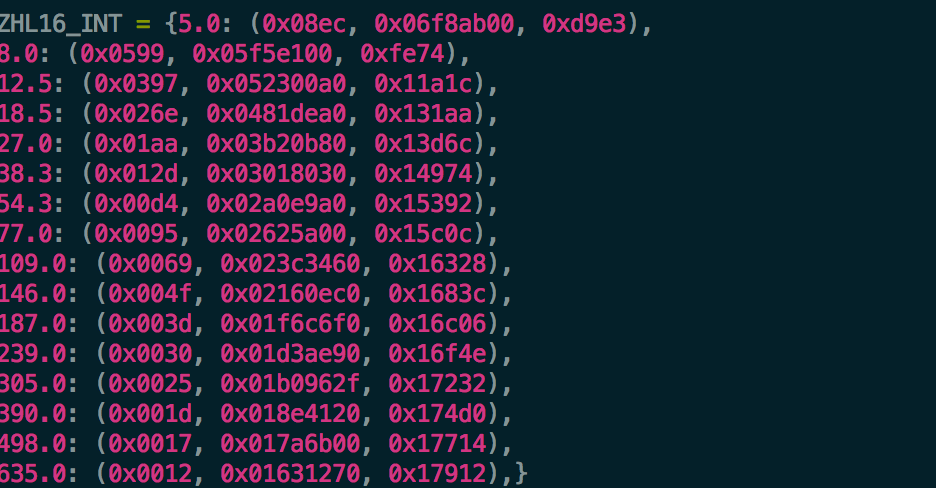


Slika 18 Kod za izračun profila (treći dio)

## Dodatak: Definicije konstanti korištenih u kodu



Slika 19 Definicije konstanti (prvi dio)



Slika 20 Definicije konstanti (drugi dio)

# Popis literature

1. *Deco for Divers*, Mark Powell, 2008.
2. *Decompression-Decompression sickness*, Albert A. Bühlmann, 1984.
3. *Some Introductory “Lessons” About Dissolved Gas Decompression Modeling,* Erik C. Baker, dohvaćeno s (<https://www.shearwater.com/wp-content/uploads/2012/08/Introductory-Deco-Lessons.pdf>) 10. siječnja 2019.
4. *Decompression theory,* Wikipedia*,* dohvaćeno 10. siječnja 2019.

1. podvodne konstrukcije koje su se koristile u izgradnji mostova [↑](#footnote-ref-1)
2. i ostali inertni plinovi u slučaju da su prisutni u plinu kojega ronioc udiše [↑](#footnote-ref-2)