UDK: 517.9:517.52(047)=861
COSATI: 12-01

Rutine za brzu interpolaciju tabelarno zadatih funkcija više promenljivih

Dr Milovan Živanović, dipl.inž.[1)]

Prikazan je algoritam za linearnu interpolaciju tabelarno zadatih funkcija proizvoljnog broja, monotono rastućih, nezavisno promenljivih veličina s promenljivim korakom tabele i različitim brojem tačaka u tabeli, za različite nezavisno promenljive veličine. Dat je kôd rutina koje realizuju algoritme pozicioniranja nezavisno promenljivih u njihovim tabelama i algoritme za linearnu interpolaciju tabelarnih funkcija više promenljivih. Date su ocene vremena izvršenja kôda svake rutine pojedinačno.

Ključne reči: **Funkcije više promenljivih, tabelarne funkcije, nizovi, pretraživanje nizova, linearna interpolacija.**

U v o d – Definicija problema

BILO koja aktivnost na projektovanju dinamike letelice i odgovarajućeg sistema upravljanja za letelicu podrazumeva poznavanje modela letelice u kome figurišu aerodinamičke sile i momenti (ASM). Zavisno od potreba konkretnih zadataka ASM jesu funkcije jedne ili više promenljivih. ASM se generišu iz tri izvora: gotovih baza podataka korišćenjem teorijskih metoda, eksperimentalnih podataka dobijenih merenjima u aerotunelu i merenjima u letu. Za isti fluid skalarne vrednosti ASM su invarijantne u istoj tački prostora stanja letelice. Postupak računanja ASM je različit za svaki izvor podataka, što se manifestuje različitim matematičkim formama funkcionalne zavisnosti ASM od izabranih nezavisnih argumenata, čiji je broj diktiran potrebama konkretnih zadataka. Bez obzira na izvor podataka, smatra se da su ASM dobro određeni, ako im je tačnost reda 90-95% od njihove stvarne vrednosti.

Za ASM dobijene teorijskim metodama i merenjem u letu, usvojena je koncepcija matematičkog predstavljanja u formi proizvoda aerodinamičkih koeficijenata, veličina stanja letelice i odgovarajućih članova, radi dimenzionog usaglašavanja. Odnosno, ASM predstavljaju proizvod Tejlorovog reda funkcije ASM i članova za dimenziono usaglašavanje. ASM se uzimaju kao funkcije stanja letelice

$$C^*=C^*(V(\text{iliMah}), \beta, \alpha, \dot{\beta}, \dot{\alpha}, p, q, r, \varphi, \theta, \psi, H(\rho, T), \delta_m, \delta_1, \delta_n)$$

gde su V(ili Mah) brzina letelice, $\beta, \alpha, \dot{\beta}, \dot{\alpha}$ ugao klizanja, napadni ugao i njihovi izvodi, p,q,r projekcije ugaone brzine na ose koordinatnog sistema vezanog za letelicu, φ, θ, ψ uglovi nagiba, propinjanja i skretanja, $H(\rho,T)$ visina leta kao funkcija gustine ρ i temperature T okoline i $\delta_m, \delta_1, \delta_n, \ldots$ otkloni komandnih površina. Aerodinamički koeficijenti dobijeni teorijskim metodama su obično funkcija od V(iliMah),β,α, a koeficijenti dobijeni iz leta su funkcije Mah i $H(\rho,T)$.

ASM iz aerotunela su funkcije (V,) α, β i za dalju primenu ne moraju se razvijati u red.

Za numeričku primenu funkcije aerodinamičkih koeficijenata i/ili ASM se prezentuju u obliku tabela s proizvoljnim korakom nezavisno promenljivih (NP) veličina.

Tako dobijeni koeficijenti se koriste za analizu performansi i dinamike letelice korišćenjem *off-line* i *on-line* metoda. *On-line* metode (najpoznatija je simulacija s hardverom u sprezi) zahtevaju računanje dinamike letelice u realnom vremenu, što postavlja zahtev za brzim računanjem (interpolacijom) aerodinamičkih koeficijenata, tj. ASM.

Za potrebe *off-line* analize postoji niz gotovih softverskih rešenja (u jeziku Fortran i C), kako za pretraživanje nizova, tako i za razne tipove interpolacije [1–5]. Teorijski su dobro pokrivene metode interpolacije za funkcije od jedne i dve NP. Teorijska osnova ponuđenih rutina za pretraživanje i pozicioniranje u monotonim nizovima su jednokoračna računanja (loc, atsg, atsm, atse [2]) i polovljenje intervala (locate, hunt [4]). Teorijska osnova ponuđenih rutina za interpolaciju su prvenstveno algoritmi Aitkena i Nevillea primenjeni za linearnu i nelinearnu interpolaciju tabelarnih funkcija jedne i dve NP (rutine: ali, ahi, acfi, … [2], icsccu, icsicu, ibcieu, ibcicu, … [3], polint, ratint, spline, splint, polin2, bccof, bcuint, … [4]). U softverskom paketu [5] razvijene su rutine (M fajlovi) za linearnu i nelinearnu interpolaciju funkcija jedne i više promenljivih (interp1 do interp6 i interpn), a za formiranje višedimenzionalnih nizova se koristi rutina ndgrid. Za svoje potrebe, služba aerodinamike je razvila rutine za linearnu interpolaciju za funkcije od jedne, dve i tri NP veličine (rutine intp1d do intp3d i augadi, koji ih objedinjuje[6,7]). Zajednička karakteristika raspoloživih rutina je da su rađene za primenu u okviru *off–line* metoda.

Shodno potrebama povezivanja podataka dobijenih teori-

[1)] Vojnotehnički institut VJ, 11000 Beograd, Katanićeva 15

jskim i empirijskim metodama, i potrebi za računanje vrednosti funkcije od više promenljivih brže od realnog vremena, formulisan je opštiji zadatak interpolacije tabelarnih funkcija. U ovom radu se rešava interpolacija tabelarno zadatih funkcija proizvoljnog broja, monotono rastućih NP veličina s promenljivim korakom tabele, i različitim brojem tačaka u tabeli za različite NP veličine. Rešenje treba da je prilagođeno za *off-line* i *on–line* aplikacije, uz obezbeđenje maksimalne slobode u zadavanju tabela. Kôd rešenja treba da bude nezavisan od tipa računara na kome će se kôd izvršavati.

Izbor metode interpolacije

U [5] je navedena brzina rada različitih interpolacionih metoda. Najbrža je metoda najbližeg suseda, čija se vrednost zavisno promenljive postavlja na njenu najbližu zadatu tabličnu vrednost u saglasnosti sa zadatim vrednostima NP. U rezultatu je dobijen stepenast oblik zavisno promenljive. Najtačnija je (i najsporija) metoda 'cubic spline', koja, kao rezultat, daje glatku promenu zavisno promenljive.

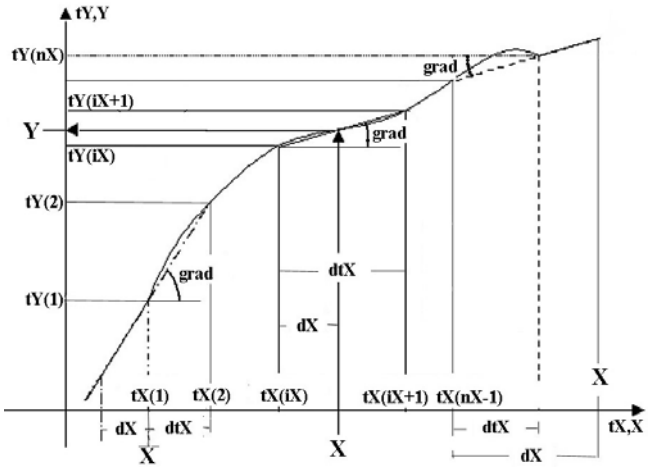
U inženjerskoj praksi mnogo funkcionalnih zavisnosti, u relativno širokom opsegu promene NP, imaju linearnu karakteristiku. Tabele se zadaju za mali broj tačaka, jer je taj broj ekvivalentan broju urađenih eksperimenata ili broju proračuna.

Za interpolaciju bi mogla da se upotrebi i metoda najbližeg suseda, ako bi se tabele dopunile s dovoljno podataka (tačaka) radi minimizacije greške između stvarne vrednosti zavisno promenljive i njene vrednosti kao najbližeg suseda. To je jednostavno uraditi u fazi inicijalizacije softvera. Za brze aplikacije na računarima, čiji operativni sistemi rade sa segmentnom organizacijom memorije, potrebno je da se broj podataka tabele zavisno promenljive spakuje u okviru jednog segmenta. Broj podataka obične preciznosti (četiri bajta za jedan podatak) ne treba da bude veći od 16000. To znači, da proizvod broja podataka po svakom argumentu mora da bude manji od broja tabličnih vrednosti zavisno promenljive. Na primer, za zavisno promenljivu s pet argumenta, izbor bi mogao da bude po 10 tačaka za prve dve promenljive, i redom 8, 5 i 4 podatka za ostale promenljive (10*10*8*5*4=16000). Rezultat primene metode najbližeg suseda, npr. neki koeficijent koji zavisi od napadnog ugla kao druge promenljive, jeste da njegova vrednost uzima samo deset tabličnih vrednosti na ukupnom opsegu promene napadnog ugla, što je za većinu inženjerskih zadataka neprihvatljivo. Radi postizanja nezavisnosti u izboru broja tačaka, povećanja tačnosti, i radi izbegavanja problema s konvergencijom metoda koje traže niz uzastopnih tačaka (npr. metoda Runge-Kuta za integraciju), a da pritom brzina rada procesa interpolacije nije bitno ugrožena, kao interpolaciona metoda za brza računanja razmatra se samo metoda linearne interpolacije.

Ekstrapolacija zavisno promenljive može da se vrši nezavisno za gornju i donju granicu. Za vrednosti NP van tabelarnog opsega, ekstrapolacija može da se vrši na jedan od sledeća tri načina (sl.1):

a) zadržavanjem granične vrednost zavisno promenljive
b) dodeljivanjem nulte zavisno promenljive i
c) linearnim ili nelinearnim produžavanjem zavisno promenljive.

Najbrži rad rutina se postiže jednoobraznošću rada rutina za sve zadate vrednosti NP, tj. ako je isti postupak računanja zavisno promenljive za NP koja je i unutar i van opsega njenih tabličnih vrednosti. Ekstrapolacija se vrši linearnim produžavanjem pomoću prve dve i poslednje dve tablične vrednosti, a jednoobraznost se postiže izborom metode pretraživanja i pozicioniranja u nizu, u odnosu na tabličnu vrednost, pomoću koje je vršena interpolacija u prethodnom pozivanju rutine.



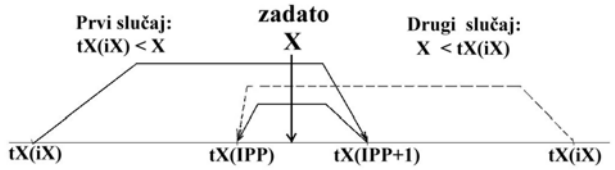
Slika 1. Linearna interpolacija i ekstrapolacija funkcije zavisne od jedne NP

Pozicioniranje u tablici

Karakteristika aerodinamičkih koeficijenata je da više koeficijenata zavise od istih promenljivih. Stoga je za interpolaciju više različitih funkcija, zavisnih od istih NP, dovoljno jednom odrediti pozicije NP u njihovim tabelama. Povećanje brzine procesa interpolacije se postiže izdvajanjem procesa pozicioniranja NP van rutina za interpolaciju i računanjem novih pozicija u tabeli samo u odnosu na poziciju, iz prethodnog pozivanja rutina za pretraživanje i pozicioniranje.

U raznim fazama projektovanja i ispitivanja, deo koeficijenata se procenjuje kao dominantan u funkciji samo pojedinih NP veličina. Ostali koeficijenti i promene po određenim veličinama se zanemaruju i kasnije proračunavaju. To znači, da su neke fukcije konstantne za sve promene NP. Stoga su tablice tih funkcija 'prazne', odnosno imaju samo jednu vrednost (obično jednaku 0) definisanu za proračunsku ili eksperimentalnu vrednost NP veličina. U tom slučaju, pozicioniranje se svodi na ispitivanje broja tabličnih vrednosti NP i postavljanjem brojača na jedan. Dozvoljeni zadati broj tabličnih vrednosti može da bude 1 ili 0, koji u fazi pripreme tabela treba postaviti na 1. Za dalji rad se pretpostavlja da je tablica svake NP data kao monotono rastuća funkcija u odnosu na njenu poziciju u nizu. Budući da su NP fizičke veličine, to je njihova promena mnogo sporija od softverskih zahteva numeričkih rutina. U primenama se retko dešava da tražena vrednost NP bude 'dalja' od dve tablične vrednosti u odnosu na njenu tabličnu vrednost korišćenu u prethodnom pozivanju interoplacionih rutina. Zbog prisustva velikog broja If petlji u algoritmima za pretraživanje i pozicioniranje koji se baziraju na polovljenju intervala [4], ne postiže se povećanje brzine izvršenja rutina za

pretraživanje. "Do while" petlja je brža, i daje veću efikasnost pretraživanja za mala pomeranja u tablici. Usvojeni algoritam pretraživanja se sastoji u jednokoračnom pomeranju po indeksu tablice u odnosu na zadati indeks iz prethodnog pretraživanja, prema sledećoj šemi pretraživanja:



iX - indeks iz prethodnog pretraživanja
IPP - traženi indeks

Postoje dva slučaja. Prvi slučaj, ako je zadata NP manja od njene tablične vrednosti u prethodnom pretraživanju, tada se indeks uvećava sve dok zadata vrednost ne postane manja od tablične vrednosti NP, a zatim se indeks smanjuje da tablična vrednost postane manja od tražene vrednosti NP. Drugi, ako je zadata NP veća od njene tablične vrednosti u prethodnom pretraživanju, indeks se smanjuje sve dok tražena vrednost ne postane manja od tablične vrednosti NP. Odvijanje ciklusa unutar tablice se ostvaruje promenom indeksa od 1 do maksimalno zadatog broja tabličnih vrednosti. U rezultatu se dobija indeks IPP prve tablične vrednosti tX(IPP), manje od zadate vrednosti X NP (tX(IPP) <X<tX(IPP+1)). Radi nesmetanog rada linearne ekstrapolacije, za zadate vrednosti NP izvan njihovog tabličnog opsega, indeks se postavlja na jedan za donju granicu, odnosno na indeks predzadnje tablične vrednosti NP za gornju granicu. U fazi inicijalizacije softvera (jednokratan *off-line* rad) treba inicijalizovati poziciju u nizu, polazeći od indeksa proizvoljne tablične vrednosti (obično jedan).

Navedeni algoritam izvršava fortranska funkcija IPP (sl.2). Njen poziv je usaglašen s pozivima interpolacionih rutina. Tipizovane su karakteristike NP, koje se navode pri pozivu svih rutina datih u ovom radu. Navode se tX, nX, iX, X - tabela, broj članova u tabeli, tekuća pozicija u tabeli i zadata vrednost NP. Nesmetan rad rutine je obezbeđen deklaracijom niza i kontrolom indeksa u odnosu na deklarisanu dužinu niza.

```
C-----------------------------------------------------------------
C   namena: POZICIONIRANJE VARIJABLE U TABELI
C   Za tX(1)<X<tX(nX) trazi poziciju IPP zadate promenljive
C   X u zadatoj tabeli tX polazeci od zadate pozicije I,
C   tako da bude tX(IPP)<X<tX(IPP+1)
C                  *****************
C   Za tX(1) < X i X > tX(nX) setuje IPP na 1 odnosno nX-1.
C-----------------------------------------------------------------
      FUNCTION IPP (tX, nX, i, X )

      DIMENSION tX(nX)
      Real X
      Integer nX,i

      IF (nX.le.1) then                    ! NIJE F-JA OD X
          IPP = 1                  ! zadato nX=0 ili nX=1
        else                               ! JESTE F-JA OD X
          If ( X.lt.tX(1) ) then
              IPP = 1              ! za donju ekstrapolaciju
            else if( tX(nX).lt.X ) then
              IPP = nX-1           ! za gornju ekstrapolaciju
            else                   ! tX(1) < X < tX(nX)
              Do 1 While ( X.Ge.tX(i) .AND. i.le.nX-1)
1                 i = i+1
              Do 2 While ( i.gt.1 .and. X.Lt.tX(i) )
2                 i = i-1
              IPP = i
          end if
      end if
      end
```

Slika 2. Rutina za pretraživanje i pozicioniranje u nizu

Interpolacija funkcije jedne promenljive

Neka je funkcija Y(x) data tabelom tY definisanom za nX tačaka NP x date u tabeli tX. Zadatak je da se nađe vrednost funkcije Y(x) za proizvoljno zadatu vrednost NP x=X (sl.1 i 4).

Problem linearne interpolacije je trivijalan i svodi se na određivanje položaja tačke na pravoj provučenoj kroz dve zadate tačke u ravni. Jezgro funkcija Srazmer i Hiper1 izvršava taj zadatak, uz uslov da su zadate NP medusobno različete (slike 3 i 4). Rutina Srazmer direktno koristi promenljive, a rutina Hiper1 koristi indeksirane članove niza.

Logičke operacije imaju višestruku namenu. Zadatak ovih operacija je sprečavanje deljenja s nulom, obezbeđenje pripadnosti nizu i rukovanje konstantnim funkcijama na osnovu informacija dobijenih iz nivoa odakle su pozvane.

Rutina Srazmer (sl.3) će u slučaju istih tabličnih vrednosti NP (X1=X2) da zaključi, da je funkcija konstantna. Neće da vrši interpolaciju (izbeći će deljenje s nulom), već će da vrati vrednost prve navedene vrednosti zavisno promenljive Y1. To znači, da na nivou odakle se funkcija poziva, treba da se obezbedi da zadati parametri funkcije imaju smisla.

```
C-----------------------------------------------------------------
C   namena: SKALARNA JEDNODIMENZIONA INTERPOLACIJA
C        Izmedu dve zadate tacake (X1,Y1) i (X2,Y2), X1=\X2 u ravni
C        Oxy trazi vrednost zavisno promenljive Y = Srazmer
C        za zahtevanu vrednost nezavisno promenljive X.
C       Ako su iste tablicne vrednosti nezavisne promenljive
C       X1=X2 vraca se vrednost zavisno promenljive Y1(=Y2)
C-----------------------------------------------------------------
       FUNCTION Srazmer ( X1,Y1, X2,Y2, X)

       If (X1.ne.X2) then              ! Srazmer
              dtY  = Y2 - Y1
              dtX  = X2 - X1
              grad = dtY/dtX
              dX   = X - X1
              dY   = grad*dX
              Srazmer = Y1 + dY
         else                          ! X1=X2
              Srazmer = Y1
        end if
        end
```

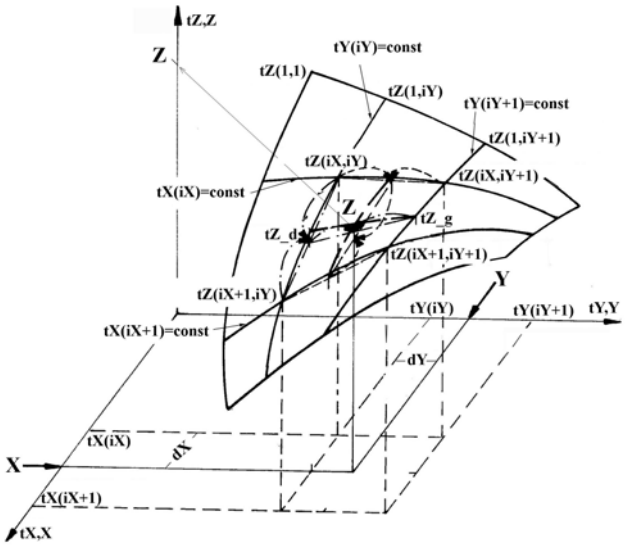
Slika 3. Rutina za direktno računanje srazmera

```
C-----------------------------------------------------------------
C  namena: LINEARNA JEDNODIMENZIONA INTERPOLACIJA
C  Iz zadatih tabela zavisne tY od nezavisne tX promenljive (nX tacaka)
C  za X - zadato              tX(iX)≤X≤tX(iX+1)
C  tX:   t X(1)  tX(2)   ....   tX(iX)↓  tX(iX+1) .....  tX(nX)
C  tY:   t Y(1)  tY(2)   ....   tY(iX) ⇓  tY(iX+1) .....  tY(nX)
C  trazi vrednost                Y=Hiper1.
C  Za X < tX(1) i X > tX(nX) vrsi se ekstrapolacija.
C  Za tX(ix)=tX(i+1) vraca se Y1(ix) (=Y(ix+1))
C-----------------------------------------------------------------
       FUNCTION Hiper1 ( tY, tX, nX, iX, X )

       DIMENSION tY(Nx), tX(Nx)

       If ( nX.ge.ix+1 .and. tX(ix+1).ne.tX(ix) ) then    ! Srazmer
              dtY  = tY(ix+1)-tY(ix)
              dtX  = tX(ix+1)-tX(ix)
              grad = dtY/dtX
              dX   = X - tX(ix)
              dY   = grad*dX
              Hiper1 = tY(ix) + dY
         Else                                  ! tX(ix+1)=tX(ix)
              Hiper1 = tY(ix)
       end If
       end
```

Slika 4. Računanje srazmera pomoću indeksiranih članova niza

Rutina Hiper1 (sl.4) direktno koristi rezultate rutine IPP. Pri pozivu rutine Hiper1 niz tX treba da je pozicioniran na član koji odgovara najmanjoj vrednosti NP x, a niz tY na član koji je korespondentan prvom članu niza NP x. Ako postoji promena tablične vrednosti NP veličine (tX(iX)≠tX(iX+1)), rutina vraća interpoliranu vrednost zavisno promenljive Y za zadatu vrednost NP X. U suprotnom, vraća se tablična vrednost zavisno promenljive Y sa kojom je rutina pozvana (tY(iX)). U rutini se sprečava izvršenje interpolacije za slučaj da je ukupan broj članova tabele NP manji od većeg indeksa NP (nX<iX+1). Ova operacija je postavljena radi sprečavanja grešaka proizvedenih zahtevom za obradu članova niza, izvan zadatog opsega. Taj slučaj greške bi se pojavio za nX=1.

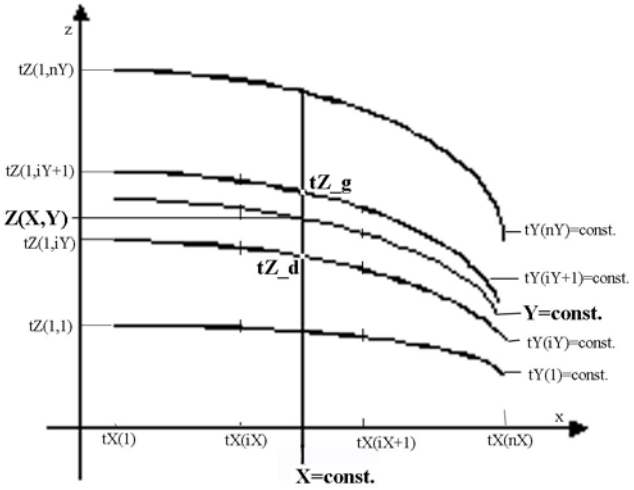
Ovako jednostavne rutine korektno vrše linearnu interpolaciju i ekstrapolaciju zahvaljujući izabranom načinu za određivanje indeksa u tabeli podataka NP. Interpolacija i ekstrapolacija će se korektno izvršavati samo ako je prethodno baš funkcijom IPP izvršeno pozicioniranje u tabeli NP.



Slika 5. Linearna interpolacija funkcije dve NP

Tablične funkcije dve promenljive

Funkcionalna zavisnost Z(x,y) sa dve NP x i y predstavlja hiperpovrš u trodimenzionalnom prostoru (sl.5). U inženjerskoj praksi je uobičajeno da se ta zavisnost predstavlja kao niz projekcija na ravan Ozx preseka hiperpovrši Z(x,y) s ravnima y=const. Praktično se daje više jednodimenzionalnih zavisnosti Z = Z(x) za niz vrednosti y=const kao parametrom (sl.6).



Slika 6. Parametarsko predstavljanje funkcije dve NP

Neka je funkcija Z(x,y) data tabelom tZ definisanom za nX vrednosti NP x date u tabeli tX i nY vrednosti NP y date u tabeli tY (ukupno nX*nY tačaka) (sl.7). Zadatak je da se nađe vrednost funkcije Z(x,y) za proizvoljno zadatu vrednost para NP (x,y)=(X,Y) (slike 5 i 6). Pri tome treba obezbediti korektno izvršenje rutine i kad funkcija Z(x,y) ne zavisi od jedne NP, tj. Z(x,y)= Z(x) ili Z(x,y)= Z(y).

tZ(nX*nY): 1 ⋯ nX nX+1 ⋯ 2nX ⋯ (nY-2)nX (nY-2)nX+1 ⋯ (nY-1)nX (nY-1)nX+1 ⋯ nYnX

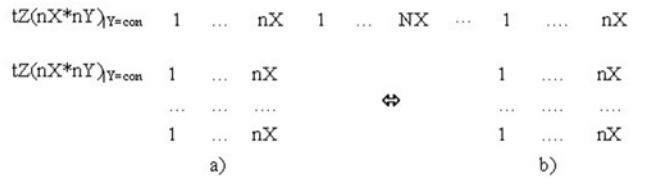
tZ(nX,nY):							
	1	….	nX		1,1	……	nX,1
	nX+1	….	2nX		1,2	……	nX,2
	….	….	….	⇔	…….	……	……
	(nY-2)nX+1		(nY-1)nX		1,nY-1	……	nX,nY-1
	(nY-1)nX+1		nYnX		1,nY	…….	nX,nY
		a)				b)	

Slika 7. Vektorsko tZ(nX*nY) i matrično tZ(nX,nY) deklarisanje istih podataka

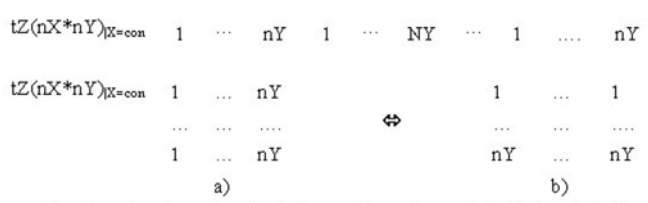
Tabela se formira tako što se prvo jedna promenljiva, npr. y, drži konstantnom, a druga se menja tako da dobije nX vrednosti. Na taj način dobijaju se tablične vrednosti krivih tZ(tX,iY+const), iY+1,…,nY zavisne od jedne promenljive x date tabelom tX, koje predstavljaju presek hiper površi Z(x,y) i ravni y=const. Ponavljajući postupak za nY različitih vrednosti y, dobiće se isto toliki broj tablica zavisnih od jedne promenljive x, koja uzima nX istih vrednosti datih tabelom tX za sve tablice tZ(tX,iY+const), iY+1,…,nY.

U primenama je čest slučaj da je tabela tX zadata (nX>1), ali da je definisana promena samo pojedinih koeficijenata po toj promenljivoj. Tada tabela zavisno promenljive tZ mora da bude formirana kao da je Z konstantna po X, što praktično znači da se u tabeli tZ zadaje nY grupa od po nX istih članova niza (sl.9). Na analogan se način formira tabela za tZ, ako promena nije definisana po y, samo što se u ovom slučaju nY puta ponavlja niz sa nX članova (sl.8).

Pakovanje tabličnih podataka tZ zavisno promenljive Z(x,y) u memoriji računara uvek je u vidu jednodimenzionog niza dužine nX*nY reči. U zavisnosti od potreba, taj niz u pojedinim rutinama može da se tretira kao jednodimenzioni ili kao višedimenzioni, što se reguliše odgovarajućim deklaracijama (sl.7). Ukoliko se podaci dvodimenzionog niza deklarišu kao podaci jednodimenzionog niza (tZ(nX*nY)), onda je nužno izračunati poziciju podatka u nizu. Taj način deklarisanja je pogodan za obradu svih članova niza (npr. sabiranje ili množenje matrica), ali je nepogodan za čest pojedinačan pristup različitim članovima niza, jer se pri svakom pristupu mora izračunavati njihova pozicija u nizu ([3], rutina loc). Ovde je ostavljeno da korisnik prema svojim postrebama na višem nivou programa (na nivou odakle poziva interpolacione rutine) deklariše nizove prema konkretnim potrebama, a da se u rutinama za interpolaciju nizovi deklarišu kao višedimenzioni tako da im je dimenzija korespondentna s brojem podataka za sva-ku NP veličinu (npr. deklaracija u rutini Hiper1 je tY(nX), u Hiper2 je tZ(nX,nY), itd.).



Slika8. Vektorsko tZ(nX*nY) i matrično tZ(nX,nY) deklarisanje istih podataka za Y=const=> tZ(iX,iY)=tZ(iX)



Slika 9. Vektorsko tZ(nX*nY) i matrično tZ(nX,nY) deklarisanje istih podataka za X=const=> tZ(iX,iY)=tZ(iY)

```
C-----------------------------------------------------------------------
C     namena: LINEARNA DVODIMENZIONALNA INTERPOLACIJA
C          Iz zadate tabele zavisne tZ od nezavisnih tY,tX promenljih
C          (nX*nY tacaka) trazi vrednost Z=Hiper2 za zahtevano
C          X > tX(ix) i Y > tY(iy), t#(i#)=\t#(i#+1), #=X,Y.
C          Za X < tX(1) i X > tX(nX), kao
C          i za Y < tY(1) i Y > tY(nY) vrsi se ekstrapolacija.
C          Za t#(ix)=t#(i+1), #=X,Y isituju se sve kombinacije
C          zavisnosti funkcije od nezavisnih argumenta i setuje
C          f(x,_)=tZ_d, f(_,y)=hiper1 (y) i f(_,_)=tZ(iX,iY)
C-----------------------------------------------------------------------
   FUNCTION Hiper2 ( tZ, tX,nX,ix,X, tY,nY,iy,Y )

   DIMENSION tZ(Nx,Ny), tX(Nx), tY(Ny)

   If ( nX.ge.iX+1 ) then                !f-ja X
      tZ_d = Srazmer(tX(ix),tZ(ix,iY), tX(ix+1),tZ(ix+1,iY), X )
      if ( nY.gt.iY+1) then              ! f-ja X i Y
         tZ_g = Srazmer (tX(ix),tZ(ix,iY+1),tX(ix+1),                +
                tZ(ix+1,iY+1),X)
         Hiper2=Srazmer (tY(iy),  tZ_d, !      iy         Y      y+1
+                    tY(iy+1),tZ_g, !    ix +...……..#. .....+
+                                  Y )!          .          .
                                          ! X-> tZ_d    *      tZ_g
       else  ! F-ja od X, a ne i od Y       !      .      Hiper2   .
         Hiper2 = tZ_d                     !      .          .
      end if                               ! ix+1 +……...#……..+
    else              ! NIJE f-ja od X, mozda od Y
      If ( nY.ge.iY+1) then               ! f-ja Y
         Hiper2 = Srazmer (tY(iy),tZ(1,iY),tY(iy+1),tZ(1,iy+1),Y)
       else
         Hiper2 = tZ(iX,iY)               ! nije f-ja ni od X, ni od Y
      end if
   end if
   end
```

Slika 10. Rutina za interpolaciju funkcije dve NP pomoću rutine Srazmer

```
C-----------------------------------------------------------------------
C     namena: LINEARNA DVODIMENZIONA INTERPOLACIJA
C          Iz zadate tabele zavisne tZ od nezavisnih tY,tX prom.
C          (nX*nY tacaka) trazi vrednost Z=Hiper2 za zahtevano
C          X > tX(ix) i Y > tY(iy), t#(i#)=\t#(i#+1), #=X,Y.
C          Za X < tX(1) i X > tX(nX), kao
C          i za Y < tY(1) i Y > tY(nY) vrsi se ekstrapolacija.
C          Za t#(ix)=t#(i+1), #=X,Y ispituju se sve kombinacije
C          zavisnosti funkcije od nezavisnih argumenta i setuje
C          f(x,_)=tZ_d, f(_,y)=Srazmer (y) i f(_,_)=tZ(iX,iY)
C-----------------------------------------------------------------------
      FUNCTION Hiper2 ( tZ, tX,nX,ix,X, tY,nY,iy,Y )

      DIMENSION tZ(nX,nY), tX(nX), tY(nY)

      tZ_d = Hiper1 (tZ(1,iY  ), tX, nx, ix, X )     ! f-ja od X  ?
      If ( nY.gt.iY+1) then                          ! f-ja od Y ?
         tZ_g = Hiper1 (tZ(1,iY+1), tX, nX, iX, X )
         Hiper2 = Srazmer (tY(iy), tZ_d, tY(iy+1),tZ_g, Y )
       else
         Hiper2 = tZ_d
      end if
      end
```

Slika 11. Rutina za interpolaciju funkcije dve NP pomoću rutine Hiper1

Pozicioniranje se obavlja pre pozivanja rutine za interpolaciju dvostrukim pozivanjem rutine IPP za pozicioniranje NP X i Y u nizovima tX i tY. U rezultatu se dobijaju indeksi iX i iY, takvi, da su ispunjeni sledeći uslovi (tX(iX)≤X≤tX(iX+1), iX∈[1,nX-1] i tY(iY)≤Y≤tY(iY+1), iY∈[1,nY-1]). Time je obezbeđeno da za NP, unutar tabela (tX(1) ≤ X≤tX(nX) i tY(1) ≤Y≤tY(nY)), tražena vrednost zavisno promenljive Z bude u ćeliji hiperpovrši Z(x,y) omeđene njenim presecima s ravnima x=tX(iX)=const., x=tX(iX+1)= const. i y=tY(iY)=const., y=tY(iY+1)=const. Taj deo hiperpovrši (ćelija) u tabeli tZ definisan je čvorovima tZ(iX,iY), tZ(iX+1,iY), tZ(iX,iY+1) i tZ(iX+1, iY+1) koji u opštem slučaju ne pripadaju jednoj ravni, i kroz koje može da se provuče šest mimoilaznih pravih (četiri stranice i dve dijagonale) (sl.5).

U ovom se radu za linearnu interpolaciju koristi algoritam, kojim se iz dve interpolisane vrednosti, takođe interpolacijom, dobija konačna interpolisana vrednost (algoritam tipa Nevillea). Takva konačno interpolisana vrednost zavisno promenljiva Z(x,y) može da se dobije na dva načina. Jedan način je, da se prvo za zadatu vrednost NP X nađu interpolisane vrednosti zavisno promenljive Z=tZ_d u ravni y=tY(iY)=const i Z=tZ_g u ravni y=tY(iY+1)=const, a da

se zatim na osnovu njih (u ravani X=const), za zadatu vrednost NP Y, odredi njena konačna interpolisana vrednost Z. Za ovaj način prvo se vrši interpolacija prema X pa zatim prema Y. Drugi način je, da se prvo izvrši interpolacija prema Y, a zatim prema X.

Linearnom interpolacijom interpolisana vrednost zavisno promenljive Z(x,y) je jednoznačno određena, jedino, kada čvorovi ćelije pripadaju istoj ravni. Stoga se za jednoznačno dobijenu interpolisanu vrednost zavisno promenljive dobija i jednoznačna vrednost greške interpolacije. Međutim, ako čvorovi ne pripadaju istoj ravni, osim navedenim postupcima, linearna interpolacija može da se realizuje i na druge načine. Ako čvorovi ne pripadaju istoj ravni, primena različitih postupaka, pa i prethodna dva postupka, daće različite interpolisane vrednosti zavisno promenljive Z. Očigledno je da će se ostvariti i različite greške interpolacije (sl.5).

U ovom radu je određen rekursivni postupak interpolacije koji se izvršava za najkraće vreme, a da proizvedena greška bude za red veličine manja od postojeće greške podataka kojima se rukuje. Ako je tačnost podataka reda deset posto od njihove apsolutne vrednosti, tada je greška interpolacionog postupka zadovoljavajuća, ako je ona reda jedan posto od apsolutne vrednosti tih podataka.

Na grešku interpolacije može da se utiče na tri načina: samim postupkom interpolacije, načinom pripreme poziva rutina i načinom pripreme tabelarnih podataka za interpolaciju, čemu se u ovom radu daje prednost.

Izbor redosleda veličina po kome će se vršiti interpolacija treba da bude tako odabran, da se interpolacija prvo obavlja po veličinama s kojima je zavisno promenljiva linearno povezana u većem delu njihovog opsega promene. Taj redosled, u svim ovde datim rutinama, je s leva na desno. Za svaku novu NP s desne strane se dopišu njene tipizirane karakteristike i broj rutine za interpolaciju se podigne za jedan. Na primer, za aerodinamičke koeficijente dobijene tunelskim ispitivanjem pogodan redosled MP je $V(Mah),\alpha,\beta,\delta_m,\delta_1,\delta_n$.

U pripremi tabela o izabranom se redosledu mora voditi računa, jer on diktira smeštaj podataka u memoriji računara (slike 5 do 9). Kako zavisno promenljive nisu linearne u celom opsegu NP, to u fazi pripreme tabela korak tabelarne promene NP treba izabrati tako, da linearni opseg bude pokriven samo s dve tačke, a nelinearni s više tačaka. Takvom postavkom, analiza i podešavanje greške interpolacije obavlja se u fazi pripreme tabela (*off-line* rad), što nije tema ovog rada.

Za interpolaciju funkcije dve nezavisno promenljive nudi se rutina Hiper2. Vrednosti donjeg i gornjeg čvora Z=tZ_d u ravni y=tY(iY)=const i Z=tZ_g u ravni y=tY(iY+1)=const za zadatu vrednost NP X i na osnovu njih, za zadatu vrednost NP Y, konačno interpolisana vrednost zavisno promenljive Z određuje se ili rutinom Srazmer (sl.10) ili rutinom Hiper1 (sl.11). Rutina Hiper2 ne vrši direktno interpolaciju, već samo analizira indekse i priprema ulazne parametre za rutinu koja vrši interpolaciju (ovde računa srazmer). Ispitivanjem odnosa indeksa NP i ukupnog broja članova u tabeli, obezbeđena je upotreba samo članova niza.

Ukoliko se tZ_d i tZ_g formiraju rutinom Srazmer (sl.10) ispitivanje indeksa NP X se obavlja radi sprečavanja poziva rutine Srazmer s nepostojećim članom niza (slučaj nX=1), što praktično znači da funkcija ne zavisi od te NP. Ako Z(x,y) zavisi od y , onda postoji samo tabela tY sa nY članova, a dvodimenzionalni niz zavisno promenljive postaje jednodimenzionalni tZ(nX=1,nY), pa se interpolisana vrednost Z nalazi između iY i iY+1-og člana tog niza. Ista logika se koristi i u slučaju nY=1.

Ukoliko se tZ_d i tZ_g formiraju rutinom Hiper1 (sl.11), onda se Hiper1 poziva s nizovima definisanim za tY(iY)=const. i tY(iY+1)=const. Ako je nX=1, biće i iX=1, pa će Hiper1 vratiti vrednosti njene lokalne promenljive tY(iX)=tY(1), koje su na nivou rutine Hiper2 vrednosti njene lokalne vrednosti tZ(1,iY) i tZ(1,iY+1). Ako je nY=1 rutina Hiper1 će biti pozvana samo jednom i njen odgovor će biti odgovor i rutine Hiper2, koji je za nX>1 interpolisana vrednost prema tabeli tX, a za nX=1 prvi i jedini član niza tZ(1,1).

Interpolacija funkcija zavisnih od tri i više promenljivih

Algoritam interpolacije funkcije zavisnih od tri i više NP je analogan algoritmu za interpolaciju funkcije zavisne od dve NP. Rekursija se ostvaruje na sledeći način. Traži se interpolacija funkcije sa N promenljivih. Potom se nađe interpolisana vrednost funkcije prvih N-1 promenljvih za konstantne vrednosti N-te NP. Konstantne vrednosti N-te NP su njene tablične vrednosti, između kojih se nalazi zahtevana vrednost te NP. Na primer, za interpolaciju funkcije S(x,y,z) od tri NP treća NP je z. Neka se zahtevana vrednost z=Z nalazi između tZ(iZ) i tZ(iZ+1). Prvo se nalaze tS_d i tS_g kao interpolisane vrednosti funkcija S(x,y,z=tZ(iZ)) i S(x,y,z=tZ(iZ+1)) za zahtevane vrednosti prve dve NP x=X i y=Y (sl.12). Dalje se na osnovu dobijenih tS_d i tS_g za treću (poslednje navedenu) NP z=Z, nađe konačna interpolisana vrednost S=S(X,Y,Z).

Forma i logika koda rutine Hiper2 koja poziva rutinu Hiper1 je baza za interpolacione rutine funkcija više od dve promenljive (slike 13 do 15). Broj NP određuje ime rutine za interpolaciju, npr. za interpolaciju funkcije sa četiri NP koristi se rutina Hiper4. Niz zavisno promenljive se u svakoj rutini lokalno deklariše prema njenom broju, npr. u Hiper5 sa NP X,Y,Z,U i V deklaracija će biti DIMENSION tQ(nX,nY,nZ,nU,nV), tX(nX), tY(nY), tZ(nZ), tU(nU), tV(nV). Gornja i donja vrednost zavisno promenljive u hiperravnima definisanim za konstantne vrednosti poslednje navedene NP, određuje se pozivom rutine, čije ime ima broj za jedan niži od broja rutine iz koje se poziv obavlja. U svakoj rutini ispitivanje indeksa vrši se samo za NP koja je poslednja navedena u deklaraciji. Broj NP limitiran je mogućnostima kompajlera koji se koristi za prevođenje rutina. Microsoftov Fortan 5.1 za 16-bitne aplikacije dozvoljava deklaraciju sa sedam promenljivih.

Upotreba rutina je jednostavna. Neka je na nekom nivou programa potrebno interpolisati neki koeficijent sa pet promenljivih. Na tom nivou nužno je imati deklarisane nizove koji su ulazni parametri za rutinu Hiper5. Pre poziva rutine Hiper 5, za svaku NP, treba pozvati rutinu IPP koja će odrediti indekse čvorova za omeđenje prostora u kome se nalazi NP. Posle toga se poziva jedino rutina Hiper5, čiji je odgovor tražena interpolisana vrednost zavisno promenljive. Rutina Hiper5 će automatski sama pozvati rutine nižeg reda. Za više funkcija zavisnih od istih NP, samo se ponavljaju pozivi interpolacionih rutina.

Samo najniža rutina (Srazmer i/ili Hiper1) obavlja interpolaciju. Ukoliko postoji potreba za promenu tipa interpolacije, onda treba promeniti samo te rutine i njih kodirati za obavljanje drugog tipa interpolacije.

```
      FUNCTION Hiper3 (tS, tX,nX,ix,X, tY,nY,iy,Y, tZ,nZ,iZ,Z)

      DIMENSION tS(Nx,Ny,Nz), tX(Nx), tY(Ny), tZ(nZ)

      tS_d = Hiper2 (tS(1,1,iZ  ), tX,nX,ix,X, tY,nY,iy,Y )

      If ( nZ.ge.iZ+1) then                    ! F-ja Z ?
         tS_g = Hiper2 (tS(1,1,iZ+1), tX,nX,ix,X, tY,nY,iy,Y )
         Hiper3 = Srazmer (tZ(iz),tS_d,tZ(iz+1),tS_g, Z )
       else                                    ! NIJE f-ja od Z
         Hiper3 = tS_d
      end if
      end
```

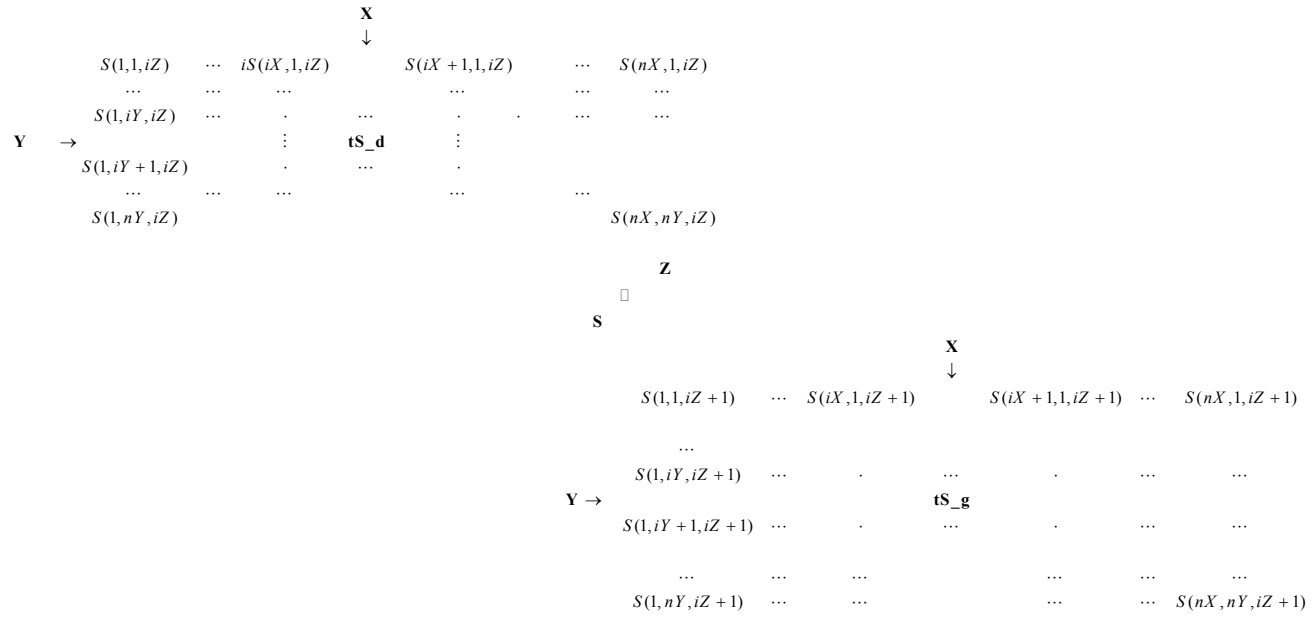
Slika 13. Rutina za interpolaciju funkcije tri NP

```
C-------------------------------------------------------------------------
C       namena: LINEARNA ČETVORODIMENZIONALNA INTERP.
C          Iz zadate tabele zavisne tR od nezavisnih tU,tZ,tY,tX
C          promenljivih (nX*nY*nZ*nU tacaka) trazi vrednost
C          R=Hiper4 za zahtevane X>tX(ix), Y>tY(iy), Z>tZ(iZ) i
C          U>tU(iU), t#(i#)=\t#(i#+1), #=X,Y,Z,U.
C          Za X<tX(1), X>tX(nX); za Y<tY(1), Y>tY(nY); za Z<tZ(1),
C          Z>tZ(nZ) i za U<tU(1), U>tU(nU) vrsi se ekstrapolacija.
C          Za t#(ix)=t#(i+1), #=X,Y,Z,U u okviru rutine isituje se
C          zavisnost funkcije od novog nezavisnog argumenta i setuje
C          f(*,*,*,_)=tR_d, a zavisnost f-je od drugih argumenata
C          se ispituje u okviru interpolacije nizeg reda.
C-------------------------------------------------------------------------
      FUNCTION Hiper4 ( tR, tX,nX,ix,X,
     +                  tY,nY,iy,Y, tZ,nZ,iZ,Z, tU,nU,iU,U )

      DIMENSION tR(Nx,Ny,Nz,Nu), tX(Nx), tY(Ny), tZ(nZ), tU(nU)

       tR_d = Hiper3 (tR(1,1,1,iU  ),  tX,nX,ix,X,
     +               tY,nY,iy,Y, tZ,nZ,iZ,Z )
       If ( nU.ge.iU+1 ) then                    ! f-ja U
          tR_g = Hiper3 (tR(1,1,1,iU+1), tX,nX,ix,X,
     +                   tY,nY,iy,Y, tZ,nZ,iZ,Z )
          Hiper4 = Srazmer (tU(iu),tR_d,tU(iu+1),tR_g, U )
        else                                     ! NIJE f-ja U
          Hiper4 = tR_d
       end if
       end
```

Slika 14. Rutina za interpolaciju funkcije četiri NP

```
C-------------------------------------------------------------------------
C       namena: LINEARNA TRODIMENZIONALNA INTERPOLACIJA
C          Iz zadate tabele zavisne tS od nezavisnih tZ,tY,tX prom.
C          (nX*nY*nZ tacaka) trazi vrednost S=Hiper3 za zahtevano
C          X > tX(ix), Y > tY(iy), Z > tZ(iZ), t#(i#)=\t#(i#+1),
C          #=X,Y,Z.
C          Za X < tX(1) i X > tX(nX), za Y < tY(1) i Y > tY(nY)
C          i za Z < tZ(1) i Z > tZ(nZ) vrsi se ekstrapolacija.
C          Za t#(ix)=t#(i+1), #=X,Y,Z u okviru rutine isituje se
C          zavisnost funkcije od novog nezavisnog argumenta i
C          setuje f(*,*,_)=tS_d, a zavisnost f-je od drugih argmenata
C          se ispituje u okviru interpolacije nizeg reda.
C-------------------------------------------------------------------------
```

```
C-------------------------------------------------------------------------
C       namena: LINEARNA PETODIMENZIONALNA INTERPOLACIJA
C          Iz zadate tabele zavisne tQ od nezavisnih tV,tU,tZ,tY,tX
C          promenljivih (nX*nY*nZ*nU*nV<16000 tac.) trazi vrednost
C          Q=Hiper5 za zahtevane X>tX(ix), Y>tY(iy), Z>tZ(iZ),
C          U>tU(iU) i V>tV(iV), t#(i#)=\t#(i#+1), #=X,Y,Z,U,V
C          za X<tX(1), X>tX(nX); za Y<tY(1), Y>tY(nY);
C          za Z<tZ(1), Z>tZ(nZ); za U<tU(1), U>tU(nU)
C          i za V<tV(1), V>tV(nV) vrsi se ekstrapolacija.
C          Za t#(ix)=t#(i+1), #=X,Y,Z,U,V u okviru rutine isituje se
C          zavisnost funkcije od novog nezavisnog argumenta i setuje
```



Slika 12. Šematski prikaz pakovanja tabelarnih podataka i interpolacije funkcije tri promenljive

```
C         f(*,*,*,*,_)=tS_d, a zavisnost f-je od drugih argumenata
C         se ispituje u okviru interpolacije nizeg reda.
C-------------------------------------------------------------------------
      FUNCTION  Hiper5 ( tQ, tX,nX,ix,X, tY,nY,iy,Y,
     +                   tZ,nZ,iZ,Z, tU,nU,iU,U, tV,nV,iV,V )

      DIMENSION tQ(Nx,Ny,Nz,Nu,Nv), tX(Nx),tY(Ny),
     +          tZ(nZ),tU(nU),tV(nV)

      tQ_d = Hiper4 (tQ(1,1,1,1,iV  ),  tX,nX,ix,X, tY,nY,iy,Y,
     +               tZ,nZ,iZ,Z, tU,nU,iU,U )
      If ( nV.ge.iV+1 ) then                          ! f-ja V
        tQ_g = Hiper4(tQ(1,1,1,1,iV+1), tX,nX,ix,X, tY,nY,iy,Y,
     +                tZ,nZ,iZ,Z, tU,nU,iU,U )
        Hiper5 = Srazmer (tV(iv),tQ_d,tV(iv+1),tQ_g, V )
      else                                            ! nije f-ja V
        Hiper5 = tQ_d
      end if
      end
```

Slika 15. Rutina za interpolaciju funkcije pet NP

Vreme izvršavanja rutina i njihova upotreba

Više faktora utiče na vreme izvršenja rutina. Oni se mogu svrstati u dve grupe. Jednu grupu faktora diktiraju osobine hardvera i karakteristike kompajlera za pripremu izvršenje rutina na tom hardveru. Drugu grupu čine faktori diktirani osobinama kôda izabranog algoritma.

Vreme izvršenja rutina određeno je brojem taktova mašine potrebnim za izvršenje svih asemblerskih instrukcija, koje je za te rutine formirao upotrebljeni kompajler. Skup asemblerskih instrukcija je različit za različite kompajlere. Skup asemblerskih instrukcija je različit za različite opcije prevođenja za isti kompajler. Različit je broj taktova mašine potreban za izvršenje pojedinačne asemblerske instrukcije za različite procesore. Trajanje jednog takta određeno je brzinom rada procesora. Za poznat hardver, kompajler i opciju prevođenja, moguće je tačno odrediti vreme izvršenja rutina. Majoranta vremena izvršenja može da se lakše odredi prisiljavanjem procesora da istu rutinu poziva više puta. Na osnovu poznavanja realnog vremena pre i posle završetka određenog broja pozivanja rutina, određuje se vreme potrebno za izvršenje jednog ciklusa pozivanja. To vreme, sem vremena za izvršenje samih rutina, sadrži i vreme upotrebljeno za obradu ciklusa i pripremu ulaznih parametara za rutine. Vreme izvršenja rutina je manje od majorante vremena izvršenja dobijene na ovaj način.

Vreme izvršavanja kôda rutina koji sadrži IF petlje s različitim zahtevima računanja u pojedinim granama nije fiksno. U slučaju interpolacionih rutina razlike su posledica različitog broj zadatih tabelarnih podataka NP, različitog broja pomeranja indeksa u tabeli i različitog ponašanja indeksa tabele unutar tabele i na granicama tabela. Međutim, tabele su fiksne za jedan dinamički sistem i jedan konkretan zadatak. U tom slučaju, vreme izvršenja je približno isto i diktirano je samo pomeranjem u tabeli. U *off-line* radu može da se dobije informacija o maksimalnim pomeranjima u tabeli i na osnovu te informacije se definiše maksimalno potrebno vreme izvršenja rutina.

Rutine date u ovom radu su razvijene radi obavljanja poslova vezanih za vazduhoplove i brzu interpolaciju niza različitih funkcija sa istom bazom NP. Medutim, rutine bez ikakvih ograničenja mogu da se primene za interpolaciju bilo kakvih jednoznačnih funkcija.

Rutine su pisane po standardu Fortrana 77, i mogu da se primene na bilo kom procesoru i operativnom sistemu. Testiranje rada i provera vremena izvršenja rađena je na računarima sa DOS operativnim sistemom, koji u potpunosti mogu zadovoljiti potrebe HIL simulacije i eventualnog letnog računara. Za osnovne opcije prevođenja Microsoftovim kompajlerom za Fortran 5.1 (za šesnaestobitne aplikacije) određivana je majoranta vremena izvršenja interpolacionih rutina na procesorima 80386 i 80486 sa taktom od 40 MHz i 100 MHz za pomeranje po dva indeksa u tabeli pri svakom pozivanju IPP-a. Mereno je vreme izvršenja DO petlje sa 10.000 pozivanja svake pojedine rutine za interpolaciju. Najviše vremena se troši ako se interpolacija vrši za NP koje su jednake baš njihovim tabličnim vrednostima. Za taj slučaj majorante su određene iz ukupnog vremena izvršenja DO petlji i date su u sledećoj tabeli 1.

Tabela 1

Rutina	Vreme izvršenja DO i=1,10000 [sec]		Majoranta (vreme izvršenja jednog prolaza kroz petlju) [μsec]	
	80386	80486	80386	80486
IPP (sl.2)	0.55	0.01	55	1
Srazmer (sl.3)	0.39	0.06	39	6
Hiper1 (sl.4)	0.33	0.06	33	6
Hiper2 (sl.10) (Srazmer)	1.27	0.22	127	22
Hiper2 (sl.11) (Hiper1)	1.1	0.22	110	22
Hiper3 (sl.13)	3.18	0.50	318	50
Hiper4 (sl.14)	6.92	1.16	692	116
Hiper5 (sl.15)	14.71	2.47	1470	247

Rutine za interpolaciju date u ovom radu su korišćene za simulaciju leta aviona. Integracija punog nelinearnog modela dinamike aviona, kao krutog tela sa šest stepeni slobode, sa punom bazom aerodinamičkih podataka, vršena je metodom trapeza sa korakom integracije 0.01 sec. Simulacija 100 sec leta, zajedno s upisom podataka simulacije u fajl, se na procesoru 80386, 40 MHz završi posle 28.07 sec, odnosno 3.56 puta brže od realnog vremena. Isti posao se na procesoru 80486, 100 MHz završi posle 7.74 sec, odnosno 100/7.74=12.92 puta brže nego u realnom vremenu.

Zaključak

Analiziran je problem brze interpolacije tabelarno zadatih funkcija proizvoljnog broja monotono rastućih nezavisno promenljivih veličina, s promenljivim korakom tabele i različitim brojem tačaka u tabeli za različite nezavisno promenljive veličine. Time je omogućeno da se rezultati merenja i proračuna direktno koriste, bez bilo kakvog dodatnog izračunavanja. U radu je, kao rešenje, dat rekursivni algoritam tipa Nevillea za linearnu interpolaciju. Proces interpolacije je odvojen od procesa pretraživanja i pozicioniranja nezavisno promenljivih veličina u njihovim tabelama. Takvim rešenjem se za više funkcija, koje zavise od istih nezavisno promenljivih, samo jednom vrši pozicioniranje u tabelama nezavisno promenljivih. Za pozicioniranje u tabelama nezavisno promenljivih izabran je postupak jednokoračnog pomeranja po indeksu tabele u odnosu na zadati indeks iz prethodnog pozicioniranja. Dati su fortranski kodovi rutina koje realizuju izabrane algoritme pozicioniranja i interpolacije. Kodovi rutina su sačinjeni tako da obezbeđuju maksimalnu slobodu u zadavanju tabela, ali, i pored toga, zahtevi za računarske resurse su minimalni. Za svaku rutinu pojedinačno i simulaciju di-

namike aviona, u čijem kodu funkcionišu predložene rutine, date su procene (majorante) vremena izvršenja koda na personalnim računarima sa Intelovim procesorima 80386 i 80486. Vremena izvršavanja pokazuju da kodovi datih rutina zadovoljavaju potrebe i *on–line* aplikacija.

Literatura

[1] KORN,G., KORN,T. *Matematical handbook for scientists and engineers (in Russian).*Moskva, Nauka,1984.

[2] *IMSL Library*. Edition 8, vol.2, Houston, 1980.

[3] *System/360 Scientific Subroutine Package(SSP), Programmer's Manual.* (prog.num. 360A–CM– 03X), Fifth Edition, Version III, num. GH20–0205 –4, New York, IBM Corporation, 1970

[4] *Numerical Recipes in Fortran 77: The Art of Scientific Computing* (ISBN 0–521–43064–X), Cambridge University Press, 1988–1992

[5] *MATLAB, The Language of Technical Computing, Using MATLAB, Version 5.2,* The MathWorks, Inc., Natic, 1984-1998

[6] MAKSIMOVIĆ,A. i dr. *Program za aproksimaciju funkcija zavisnih od više parametara.* Vazduhoplovnotehnički institut, Beograd, int. dok. V3-1931, 1980.

[7] STOJIĆ,R i dr. *Programska podrška za simulaciju kretanja aviona*. Vazduhoplovnotehnički institut, Beograd, int. dok. V3-2090-A, 1984.

Rad primljen: 13.3.2001.god.