POD i DMD dekompozicije u analizi i redukciji nelinearnih dinamičkih sustava

Kapetanović, Marta

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO–MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Marta Kapetanović

POD I DMD DEKOMPOZICIJE U ANALIZI I REDUKCIJI NELINEARNIH DINAMIčKIH SUSTAVA

Diplomski rad

Voditelj rada:
Prof. dr. sc. Zlatko Drmač

Zagreb, Studeni, 2017.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Mojim roditeljima i Brunu

Sadržaj

Uvod

Potreba za razumijevanjem prirodnih procesa potaknula je stvaranje matematičkih modela koji opisuju mnoge pojave, ne samo s područja prirodnih već i društvenih znanosti. Rezultat je mijenjanje perspektive, napredak od uloge pukog promatrača. Razvitak tehnologije omogućio je korak naprijed, vjernu i preciznu numeričku aproksimaciju modela. No zatim se, npr. kod upravljanja, javlja potreba da se ta aproksimacija računa gotovo instantno. Kada uzmemo u obzir da se matematički modeli sastoje od jedne ili pak sustava (općenito nelinearnih) parcijalnih diferencijalnih jednadžbi, jasno je da takva simulacija zahtijeva gustu diskretizacijsku mrežu, pa računanje postaje vremenski dugotrajno i samim time neiskoristivo u takvim situacijama. Rješenje ovog problema nalazimo u računanju s modelima nižeg reda. Modeli nižeg reda imaju glavna obilježja punog modela, no znatno niži trošak računanja. Za njhovu konstrukciju koristit ćemo dvije dekompozicije: POD i DMD. Vidjet ćemo da uloga ovih dekompozicija nije samo u redukciji sustava, već sadrže informacije u određenim svojstvima promatrane veličine.

Prava ortogonalna dekompozicija (POD) je metoda koja za skup podataka u određenom prostoru nalazi l vektora (prostornih modova) koji najbolje opisuju dani skup u smislu najmanjih kvadrata. Dobiveni POD modovi opisuju energiju sustava. Galerkinovom projekcijom sustava na POD bazu dobivamo aproksimacijski sustav nižeg reda. Dok za linearne sustave ova metoda daje dobre rezultate te čak dozvoljava predračunanje matrice sustava, u slučaju nelinearnosti sustava, POD-Galerkinova metoda zahtijeva računanje s punim sustavom.

Dekompozicija dinamičkih modova (DMD) pronalazi modove gibanja koji nemaju samo prostorne značajke (kao kod POD-a), već su usko povezani sa periodičnim gibanjima koji predstavljaju njihovu vremensku evoluciju. Zasniva se na teoriji Koopmanovog operatora, kojim nelinearni dinamički sustav možemo prikazati kao beskonačnodimenzionalan linearan sustav. Konstruiranje modela nižeg reda predstavlja veći izazov nego kod POD dekompozicije, no DMD ima drugu važnu ulogu: predviđanje stanja sustava u vremenskim trenutcima nakon intervala mjerenja.

U poglavlju 1 za početak ponavljamo osnovna svojstva dekompozicije singularnih vrijednosti. Potom dajemo definiciju POD dekompozicije, njenu vezu s SVD-om i

neka osnovna svojstva. Pokazujemo i kako dobiti model nižeg reda Galerkinovom projekcijom. Za kraj dajemo ocjenu greške.

U poglavlju 2 definiramo DMD dekompoziciju. Detaljanije opisujemo tri algoritma za računanje DMD modova. Uz jedan jednostavan primjer opisujemo značenje DMD modova i pokazujemo kako napraviti rekonstrukciju rješenja. Naposljetku dajemo vezu s Koopmanovim operatorom.

U zadnjem poglavlju 3 na primjerima iz mehanike fluida primijenjujemo dane metode te diskutiramo rezultate.

Poglavlje 1

POD

Pretpostavimo da je zadana funkcija $z(x,t)$, gdje je x prostorna a t vremenska varijabla, te da tu funkciju želimo nad nekom domenom aproksimirati kao konačnu sumu u separiranim varijablama:

$$z(x,t) \approx \sum_{k=1}^{N} a_k(t)\phi_k(x), \tag{1.1}$$

uz uvjet da aproksimacija teži egzaktnom rješenju kada N teži u beskonačnost (osim eventualno na skupu mjere 0). Takva reprezentacija nije jedinstvena već ovisi o izboru funkcija $\phi_k(x)$. Za svaki izbor niza $\phi_k(x)$, takav da tvore bazu za određenu klasu funkcija $z(x,t)$, postoji jedinstveni niz funkcija $a_k(t)$. Poželjno je da baza bude ortonormirana, jer u tom slučaju $a_k(t)$ ovisi samo o $\phi_k(x)$, ne i o preostalim funkcijama ϕ. Drugi kriterij pri odabiru baze je da ona minimizira srednju kvadratnu pogrešku aprokismacije za svaki N. Dakle želimo pronaći niz ortonormiranih funkcija ϕ_k takvih da prvih N funkcija daje najbolju N-članu aproksimaciju. Funkcije koje zadovoljavaju navedene kriterije zovemo pravim ortogonalnim modovima funkcije $z(x,t)$ (napomenimo samo da je poredak funkcija ϕ_k bitan). Za takve ϕ izraz (1.1) zovemo prava ortogonalna dekompozicija (POD) od $z(x,t)$.

Podatci čiju aproksimaciju ili model nižeg reda tražimo obično dolaze iz eksperimentalnih mjerenja ili numeričkih rješenja dinamičkih sustava. Prostor u kojem žive rješenja diferencijalnih jednadžbi koje opisuju dinamički sustav mogu biti beskonačne dimenzije, no rješenja dobivena diskretizacijom istih žive u konačnodimenzionalnom prostoru. Iz tog razloga se fokusiramo na konačnodimenzionalni slučaj.

1.1 SVD

Teoriju započinjemo SVD dekompozicijom, koja ima vrlo važnu ulogu u POD i DMD dekompoziciji. Štoviše, SVD dekompozicija pronalazi primjenu u mnogim slučajevima analize podataka. Ona omogućava faktorizaciju matrice na jednostavnije djelove koji nose određeno značenje. Za početak ćemo dati definiciju dekompozicije. Ona proizlazi iz sljedećeg teorema koji daje definiciju i egzistenciju singularnih vrijednosti

Teorem 1.1.1. *Neka je $Y \in \mathbb{R}^{m\times n}$ matrica ranga $d \leq min\{m,n\}$. Tada postoje ortogonalne matrice $U \in \mathbb{R}^{m\times m}$ i $V \in \mathbb{R}^{n\times n}$ takve da je*

$$U^T Y V = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} \coloneqq \Sigma, \quad D = diag(\sigma_1, ...\sigma_d), \tag{1.2}$$

pri čemu vrijedi

$$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_d > 0.$$

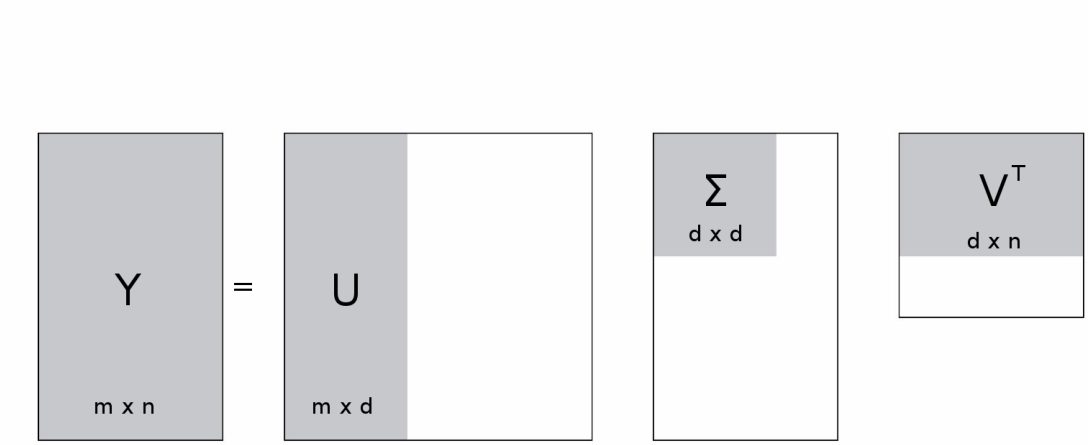
Realne brojeve $\sigma_1, ..., \sigma_d$ (zajedno sa još $\min\{m,n\} - d$ nula) zovemo singularne vrijednosti matrice Y. Stupce matrice U zovemo lijevi, a stupce od V desni singularni vektori matrice Y.

Vratimo se sada na značenje elemenata dekompozicije. Matrice U i V su ortogonalne pa u realnom prostoru predstavljaju rotaciju, dok je Σ dijagonalna pa je zaslužna za skaliranje pojedine koordinate. Iz prethodnog teorema dakle slijedi da se svako linearno preslikavanje može prikazati kao kombinacija skaliranja i dvije rotacije. Shematski prikaz dekompozicije dan je na slici 1.1.

SVD dekompozicija zapravo je poopćenje dekompozicije svojstvenih vrijednosti. Dok svojstvene vrijednosti posjeduju samo kvadratne matrice, singularne vrijednosti ima svaka matrica. Singularne vrijednosti matrice Y jednake su kvadratnim korijenima vlastitih vrijednosti matrica YY^T i Y^TY. Iz

$$\begin{aligned} YY^T &= U\Sigma \underbrace{V^T V}_{=I} \Sigma^T U^T = U \underbrace{\Sigma\Sigma^T}_{=\Lambda} U^T = U\Lambda U^T \\ YY^T U &= U\Lambda \end{aligned} \tag{1.3}$$

slijedi da su vektori $\{u_i\}_{i=1}^d$ svojstveni vektori matrice YY^T, sa pripadnim svojstvenim vrijednostima $\lambda_i = \sigma_i^2 > 0$, $i = 1, ..., d$. Također, vrijedi $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d > 0$. Analogno dobijemo da su $\{v_i\}_{i=1}^d$ svojstveni vektori matrice Y^TY. Preostali vektori, $\{u_i\}_{i=d+1}^m$ i $\{v_i\}_{i=d+1}^n$ imaju svojstvenu vrijednost 0.



Slika 1.1: *Shematski prikaz SVD dekompozicije*

Iz sheme na slici 1.1 jasno je da su sve potrebne informacije sadržane u prvih d stupaca matrica U i V, te u matrici D koju smo definirali ranije[1]. Stoga ima smisla definirati matrice $U^d \in \mathbb{R}^{m\times d}$ i $V^d \in \mathbb{R}^{n\times d}$ takve da vrijedi:

$$\begin{aligned} U^d_{ij} &= U_{ij} \quad \text{za1} \leq i \leq m,\ 1 \leq j \leq d, \\ V^d_{ij} &= V_{ij} \quad \text{za1} \leq i \leq m,\ 1 \leq j \leq d. \end{aligned} \tag{1.4}$$

Sada umjesto $Y = U\Sigma V^T$ možemo pisati

$$Y = U^d D (V^d)^T. \tag{1.5}$$

Zaključujemo da se stupci od Y mogu zapisati u bazi koju čine stupci od U^d tako da su koeficijenti raspisa dani u retcima od $D(V^d)^T$. U kontekstu prikaza (1.1), funkcije ϕ_k su reprezentirane stupcima matrice U, a a_k u retcima od $D(V^d)^T$. Svojstveni vektori međusobno su ortogonalni i norme $\|u_i\|_2 = 1$ pa je zadovoljeno i svojstvo ortonormiranosti te vrijedi:

$$y_j = \sum_{i=1}^{d} \langle y_j, u_i \rangle_{\mathbb{R}^m} u_i \quad \text{za } j = 1, ..., n, \tag{1.6}$$

[1]Primijetimo da je $D \in \mathbb{R}^{d\times d}$ a $\Sigma \in \mathbb{R}^{m\times n}$.

gdje je sa $\langle \cdot, \cdot \rangle_{\mathbb{R}^m}$ označen skalarni produkt u $\mathbb{R}^m$. U ovom trenutku jasno je da pomoću SVD dekompozicije možemo dobiti i funkcije Φ_k i koeficijente a_k. Preostalo nam je pokazati da je aproksimacija dobra. Tome ćemo se posvetit u sljedećem ulomku, kao i ulozi SVD dekompozicije u aproksimaciji matrica matricama nižeg ranga.

1.2 SVD i POD

Možemo reći da je cilj POD-a konstruirati potprostor koji najbolje aproksimira dane vektore u smislu najmanjih kvadrata. U (1.5) smo pokazali kako pronaći bazu potprostora (punog ranga) u kojem leže vektori $y_1, ..., y_n$. Sljedećim teoremom ćemo, u konačnoj dimenziji, pokazati da je baza u (1.5) optimalna u smislu najmanjih kvadrata i da to vrijedi i ukoliko je dimenzija baze manja od broja elemenata skupa $\{y_1, ...y_n\}$, tj. ukoliko tražimo aproksimaciju nižeg ranga. Teorem su originalno dokazali Eckart i Young za Frobeniusovu normu u [10],a kasnije je Mirsky pokazao da dani dokaz vrijedi za proizvoljnu unitarno invarijantnu normu u [11]. Ovdje ćemo dati dokaz u Frobeniusovoj normi, no teorem vrijedi i za operatorsku normu. Prisjetimo se, Frobeniusova norma je definirana s:

$$\|Y\|_F = \sqrt{Tr(Y^*Y)} = \sqrt{\sum_{i=1}^{d} \sigma_i^2}.$$

Teorem 1.2.1. *Neka je X proizvoljna $m \times n$ matrica ranga d, čija je dekompozicija singularnih vrijednosti $X = U\Sigma V^T$, gdje je Σ dijagonalna matrica singularnih vrijednosti $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_d$. Za $k \in \{1, ..., d-1\}$ definirajmo matricu*

$$X^k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

Tada je

$$\min_{rang(Y)=k} \|X - Y\|_F = \|X - X^k\|_F = \sqrt{\sum_{i=k+1}^{d} \sigma_i^2} \tag{1.7}$$

Dakle, od svih $m \times n$ matrica ranga k, X^k je najbolja aproksimacija matrice X.

Dokaz. Koristimo Weylov teorem za svojstvene vrijednosti, tj. njegov oblik za singularne vrijednosti. Neka su A i B $m \times n$ matrice i neka su njihove singularne vrijednosti u padajućem poretku.

Iz *Teorema 3.3.16* u [1] vrijedi

$$\sigma_{i+j-1}(A+B) \leq \sigma_i(A) + \sigma_j(B) \quad \text{za } 1 \leq i,j \leq n, i+j-1 \leq n. \tag{1.8}$$

Ako je Y ranga k slijedi $\sigma_{k+1}(Y) = 0$. Uzmimo $j = k+1$, $B \coloneqq Y$ i $A \coloneqq X - Y$. Slijedi

$$\sigma_{i+k}(X) \leq \sigma_i(X-Y) \quad \text{za } 1 \leq i \leq n-k \tag{1.9}$$

Za Frobeniusovu normu vrijedi

$$\|X-Y\|_F^2 \geq \sum_{i=1}^{n-k} \sigma_i^2(X-Y) \geq \sum_{i=k+1}^{n} \sigma_i^2(X) \tag{1.10}$$

Uzmimo $Y = X^k$. Dobivamo jednakost

$$\|X-X^k\|_F^2 = \sum_{i=1}^{n-k} \sigma_i^2(X-X^k) = \sum_{i=k+1}^{n} \sigma_i^2(X) \tag{1.11}$$

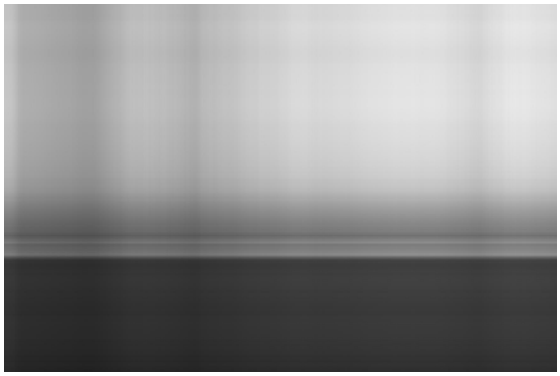
Slijedi

$$\|X-Y\|_F \geq \|X-X^k\|_F$$

□

Prethodno smo radili sa proizvoljnom matricom Y, koja nije sadržavala nikakvo fizičko značenje. Ovdje ćemo dati primjer aproksimacije matrice koja predstavlja crno bijelu sliku. Svaki element matrice predstavlja piksel slike i sadrži informaciju o intenzitetu sive boje u intervalu $[0,1]$, gdje 0 predstavlja bijelu, a 1 crnu boju. Slika se sastoji od 1168×1752 piksela te pripadna matrica ima 1168 singularnih vrijednosti. Na slici 1.2 prikazana je aproksimacija s matricama reda 1, 5, 10, 50, 200 te originalna matrica. Vidljivo je da je već prikaz sa 200 najvećih vrijednosti dovoljno jasan.

Iz (1.11) slijedi da je greška aproksimacije jednaka korijenu sume kvadrata odbačenih singularnih vrijednosti. Ova ocjena bit će korisna kod odabira broja vektora baze, čime smo opravdali odabir norme. Iz teorema slijedi da se nalaženje POD baze u konačnodimenzionalnom slučaju svodi na računanje SVD dekompozicije. No, računanje SVD-a, u slučaju velikih dimenzija matrice Y, je skupa operacija. Zato ćemo potražiti drugačiji pristup, koji će dati efikasniji algoritam za rješavanje problema.



(a) Aproksimacija sa 1 singularnom vrijednosti

(b) Aproksimacija sa 5 singularnih vrijednosti

(c) Aproksimacija sa 10 singularnih vrijednosti

(d) Aproksimacija sa 50 singularnih vrijednosti

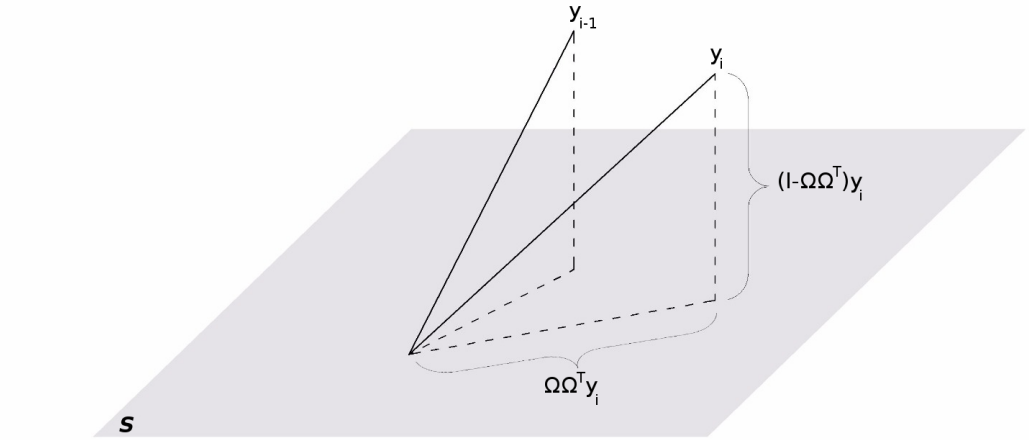
(e) Aproksimacija sa 200 singularnih vrijednosti

(f) Polazna matrica

Slika 1.2: Primjer aproksimacije pomoću SVD-a

1.3 Metoda uzoraka

Metoda uzoraka usko je vezana uz konkretne probleme. Pretpostavimo da imamo skup od n mjerenja neke veličine (npr. brzina fluida) u k prostornih točaka. Ta mjerenja nazivamo uzorcima. Neka je svaki uzorak reprezentiran jednim stupcem matrice Y. Ukoliko se mjerenje odvija u dvodimenzionalnom prostoru, kojeg razapinju vektori x_1 i x_2, brzinu moramo mjeriti u obe dimenzije. Dakle jedan uzorak sadrži $2k$ vrijednosti, tako da je prvih k vrijednosti u stupcu jednako izmjerenoj veličini u smjeru x_1 a sljedećih k veličini u smjeru x_2. Ovdje se nećemo ograničiti na dimenziju prostora pa ćemo za prostornu dimenziju uzeti m, koji može biti jednak k, $2k$ ili $3k$. Matricu $Y \in \mathbb{R}^{m\times n}$ nazivamo matrica uzoraka. Tražimo linearani potprostor za koji je ukupna greška aproksimacije uzoraka na taj potprostor minimalna. Neka je traženi potprostor, označimo ga sa $\mathcal{S}$, dimenzije k te neka je Ω matrica u kojoj se nalaze ortonormirani vektori $[\omega_1, ..., \omega_k]$ takvi da je $\mathcal{S} = span\{\omega_1, ..., \omega_k\}$. Ukoliko je baza ortonormirana, što je poželjno, matrica Ω je ortogonalna pa vrijedi $\Omega^T\Omega = I_k$. Također, znamo da je tada projektor na potprostor oblika $P = \Omega\Omega^T$



Slika 1.3: *Projekcija uzoraka na potprostor* $\mathcal{S}$

($P^2 = \Omega\Omega^T\Omega\Omega^T = \Omega\Omega^T = P = P^T$ pa je projektor dobro definiran). Dakle, projicirani uzorci su jednaki $\Omega\Omega^T y_i$, a greška projekcije jednaka je $(I - \Omega\Omega^T)y_i$, (pogledaj sliku 1.3). Ukupna greška jednaka je

$$\sum_{j=1}^{n} \|(I - \Omega\Omega^T)y_j\|_2^2 = \sum_{j=1}^{n} \|y_j - \Omega\Omega^T y_j\|_2^2 = \|Y - \Omega\Omega^T Y\|_F^2. \tag{1.12}$$

Želimo ju minimizirati po Ω pa imamo sljedeći minimizacijski problem

$$\min_{\Omega \in \mathbb{R}^{m\times k}} \|Y - \Omega\Omega^T Y\|_F^2. \tag{1.13}$$

Primijetimo da je $\Omega\Omega^T$ maksimalno ranga k. Ako je $U\Sigma V^T$ SVD dekompozicija od Y, iz teorema (1.2.1) znamo da je najbolja aproksimacija matrice Y matricom ranga k upravo $Y_k = U_k\Sigma_k V_k^T$. Slijedi

$$\|Y - \Omega\Omega^T Y\|_F^2 \geq \|Y - Y_k\|_F^2. \tag{1.14}$$

Pokažimo da u (1.14) vrijedi jednakost za $\Omega = U_k$. U možemo pisati na sljedeći način:

$$U = [U_k \; U_{n-k}], \tag{1.15}$$

gdje je U_{n-k} matrica koja se sastoji od posljednjih $n-k$ stupaca od U. Sada je

$$U_k^T U = \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix}, \tag{1.16}$$

gdje je I_k jedinična matrica dimanzije k, a nule su odgovarajućih dimenzija. Konačno, imamo

$$U_k U_k^T Y = U_k U_k^T U\Sigma V^T = U \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix} \Sigma V^T = U_k\Sigma_k V_k^T. \tag{1.17}$$

Time smo pokazali poveznicu između aproksimacije matrica matricama nižeg reda i optimalne projekcije vektora na linearni potprostor. Također, pokazali smo da je tvrdnja teorema kojeg ćemo u nastavku dati već dokazana u teoremu 1.2.1, no želja nam je dati dokaz kojeg je lakše proširiti na općenitije probleme, npr. na prostore funkcija. Problemu pronalaska POD baze pristupit ćemo kao optimizacijskom problemu. U tom slučaju tražimo $u_1, ..., u_l$ takve da je:

$$\min_{u_1,\dots u_l \in \mathbb{R}^m} \sum_{j=1}^{n} \|y_j - \sum_{i=1}^{l} \langle y_j, u_i\rangle_{\mathbb{R}^m} u_i\|_{\mathbb{R}^m}^2 \quad \text{za } \langle u_i, u_j\rangle_{\mathbb{R}^m} = \delta_{ij},\ 1 \leq i, j \leq l. \tag{1.18}$$

Vrijedi

$$\|y_j - \sum_{i=1}^{l} \langle y_j, u_i\rangle u_i\|_{\mathbb{R}^m}^2 = \big\langle y_j - \sum_{j=1}^{n} \langle y_j, u_i\rangle_{\mathbb{R}^m} u_i \rangle_{\mathbb{R}^m}, y_j - \sum_{j=1}^{n} \langle y_j, u_i\rangle_{\mathbb{R}^m} u_i \big\rangle_{\mathbb{R}^m} \tag{1.19}$$

$$= \|y_j\|^2 - 2\sum_{i=1}^{l} |\langle y_j, u_i\rangle|^2 + \sum_{i=1}^{l}\sum_{k=1}^{l} \langle y_j, u_i\rangle\langle y_j, u_k\rangle\delta_{ik} \tag{1.20}$$

$$= \|y_j\|^2 - \sum_{i=1}^{l} |\langle y_j, u_i\rangle|^2, \tag{1.21}$$

pa minimizacijski problem u (1.18) možemo promatrati kao maksimizacijski problem, tj u_i su rješenje od

$$\max_{u_1,...u_l \in \mathbb{R}^m} \sum_{i=1}^{l} \sum_{j=1}^{n} |\langle y_j, u_i \rangle_{\mathbb{R}^m}|^2 \quad \text{uz uvjet} \quad |\langle u_i, u_j \rangle_{\mathbb{R}^m}|^2 = \delta_{ij} \text{ za } 1 \leq i, j \leq l$$

Rješenje gornjeg problema daje sljedeći teorem.

Teorem 1.3.1. *Neka je $Y = [y_1, ...y_n] \in \mathbb{R}^{m \times n}$ matrica ranga $d \leq min\{m, n\}$. Neka je $Y = U\Sigma V^T$ dekompozicija singularnih vrijednosti od Y, gdje su $U = [u_1, ..., u_m] \in \mathbb{R}^{m \times m}$ i $V = [v_1, ..., v_n] \in \mathbb{R}^{n \times n}$ ortogonalne matrice a $\Sigma \in \mathbb{R}^{m \times n}$ je oblika*

$$\Sigma \coloneqq \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} = U^T Y V$$

gdje je $D = diag(\sigma_1, ..., \sigma_d) \in \mathbb{R}^{d \times d}$, $\sigma_1, ..., \sigma_d$ su singularne vrijednosti, a 0 u gornjem izrazu označavaju nul-matrice odgovarajućih vrijednosti. Tada, za sve $l \in \{1, ..., d\}$ rješenje problema

$$\max_{\overline{u}_1,...\overline{u}_l \in \mathbb{R}^m} \sum_{i=1}^{l} \sum_{j=1}^{n} |\langle y_j, \overline{u}_i \rangle_{\mathbb{R}^m}|^2 \quad \textit{uz uvjet} \quad |\langle \overline{u}_i, \overline{u}_j \rangle_{\mathbb{R}^m}|^2 = \delta_{ij} \textit{ za } 1 \leq i, j \leq l \tag{1.22}$$

je dan singularnim vektorima $\{u_i\}_{i=1}^{l}$, tj. sa prvih l stupaca matrice U. Vrijedi

$$\text{argmax}(1.22) = \sum_{i=1}^{l} \sigma_i^2 = \sum_{i=1}^{l} \lambda_i. \tag{1.23}$$

Prije dokaza navodimo Key Fanov teorem o maksimumu matrice, čiji dokaz se može pogledati u [14].

Teorem 1.3.2. *Neka su svojstvene vrijednosti λ_i Hermitske matrice H takve da je $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$. Za svaki $l \in \mathbb{N}$, $l \leq n$ vrjedi*

$$\max_{U^*U = I_k} \text{Trag}(U^*HU) = \sum_{i=1}^{l} \lambda_i \tag{1.24}$$

Dokaz. Teorem ćemo dokazati indukcijom po $l \in \{1, , , d\}$[2]. Neka je $\{\overline{u}_i\}_{i=1}^{m}$ ortonormirana baza za $\mathbb{R}^m$. U je ortogonalna pa je i $U_l = [u_1 \dots u_l]$ ortogonalna. Neka je

[2]alternativni dokaz može se naći u [7]

$l = 1$. Uzmimo $\phi_1 \in \mathbb{R}^m$ takav da je $\|\phi_1\|_{\mathbb{R}^m} = 1$. Jer je $\{\overline{u}_i\}_{i=1}^m$ ONB za $\mathbb{R}^m$, ϕ_1 možemo zapisati u obliku $\phi_1 = \sum_{i=1}^m \langle \phi_1, \overline{u}_i \rangle_{\mathbb{R}^m} \overline{u}_i$.

Uvrstimo (1.22):

$$\begin{aligned}
\sum_{j=1}^n |\langle y_j, \phi_1 \rangle_{\mathbb{R}^m}|^2 &= \sum_{j=1}^n |\langle y_j, \sum_{i=1}^m \langle \phi_1, \overline{u}_i \rangle_{\mathbb{R}^m} \overline{u}_i|^2 \\
&= \sum_{j=1}^n \big(\sum_{i=1}^m \langle y_j, \overline{u}_i \rangle_{\mathbb{R}^m} \langle \phi_1, \overline{u}_i \rangle_{\mathbb{R}^m} \big) \big(\sum_{k=1}^m \langle y_j, \overline{u}_k \rangle_{\mathbb{R}^m} \langle \phi_1, \overline{u}_k \rangle_{\mathbb{R}^m} \big) \\
&= \sum_{j=1}^n \sum_{i=1}^m \sum_{k=1}^m \langle y_j, \overline{u}_i \rangle_{\mathbb{R}^m} \langle y_j, \overline{u}_k \rangle_{\mathbb{R}^m} \langle \phi_1, \overline{u}_i \rangle_{\mathbb{R}^m} \langle \phi_1, \overline{u}_k \rangle_{\mathbb{R}^m} \\
&= \sum_{i=1}^m \sum_{k=1}^m \big\langle \underbrace{\sum_{j=1}^n \langle y_j, \overline{u}_i \rangle_{\mathbb{R}^m} y_j}_{= YY^T \overline{u}_i}, \overline{u}_k \big\rangle_{\mathbb{R}^m} \langle \phi_1, \overline{u}_i \rangle_{\mathbb{R}^m} \langle \phi_1, \overline{u}_k \rangle_{\mathbb{R}^m} \\
&\overset{(1.3)}{=} \sum_{i=1}^m \sum_{k=1}^m \langle \lambda_i \overline{u}_i, \overline{u}_k \rangle_{\mathbb{R}^m} \langle \phi_1, \overline{u}_i \rangle_{\mathbb{R}^m} \langle \phi_1, \overline{u}_k \rangle_{\mathbb{R}^m} \\
&= \sum_{i=1}^m \sum_{k=1}^m \lambda_i \delta_{ik} \langle \phi_1, \overline{u}_i \rangle_{\mathbb{R}^m} \langle \phi_1, \overline{u}_k \rangle_{\mathbb{R}^m} = \sum_{i=1}^m \lambda_i |\langle \phi_1, \overline{u}_i \rangle_{\mathbb{R}^m}|^2 \\
&\leq \lambda_1 \sum_{i=1}^m |\langle \phi_1, \overline{u}_i \rangle_{\mathbb{R}^m}|^2 = \lambda_1 \underbrace{\|\phi_1\|_{\mathbb{R}^m}^2}_{=1} = \lambda_1
\end{aligned}$$

Također vrijedi:

$$\begin{aligned}
\sum_{j=1}^n |\langle y_j, \overline{u}_i \rangle_{\mathbb{R}^m} \rangle|^2 = \sum_{j=1}^n \langle y_j, \overline{u}_i \rangle_{\mathbb{R}^m} \langle y_j, \overline{u}_i \rangle_{\mathbb{R}^m} &= \big\langle \underbrace{\sum_{j=1}^n \langle y_j, \overline{u}_i \rangle_{\mathbb{R}^m} y_j}_{YY^T \overline{u}_i}, \overline{u}_i \big\rangle_{\mathbb{R}^m} \\
= \langle YY^T \overline{u}_i, \overline{u}_i \rangle_{\mathbb{R}^m} = \lambda_i \langle \overline{u}_i, \overline{u}_i \rangle_{\mathbb{R}^m} &= \lambda_i \|\overline{u}_i\|_{\mathbb{R}^m}^2 = \lambda_i
\end{aligned} \tag{1.25}$$

za svaki $i \in \{1, ..., m\}$. Dakle za $l = 1$ vektor u_1 je rješenje (1.22). YY^T je hermitska matrica pa vrijedi

$$\sum_{i=1}^l \sum_{j=1}^n |\langle y_j, \overline{u}_i \rangle_{\mathbb{R}^m} \rangle|^2 = \sum_{i=1}^l \langle \underbrace{YY^T}_{H} \overline{u}_i, \overline{u}_i \rangle_{\mathbb{R}^m} = tr(U_l^* H U_l). \tag{1.26}$$

Sada iz tvrdnje teorema 1.3.2 slijedi (1.23). Pretpostavimo da je za $l \geq 1$ rješenje zadaće (1.22) dano vektorima $\{u_i\}_{i=1}^l$ te da vrijedi $\text{argmax}(1.22) = \sum_{i=1}^l \lambda_i$. Preostalo je pokazati tvrdnju za $l + 1$. Neka je $\phi_{l+1} \in \mathbb{R}^m$ takav da je $\|\phi_{l+1}\|_{\mathbb{R}^m} = 1$ i

$\langle \phi_{l+1}, \overline{u_i} \rangle_{\mathbb{R}^m} = 0$, za $i \in \{1, ..., l\}$. Postupamo kao i ranije. Prikažimo ϕ_{l+1} u bazi $\{\overline{u}_i\}_{i=1}^m$:

$$\phi_{l+1} = \sum_{i=1}^{m} \langle \phi_{l+1}, \overline{u}_i \rangle_{\mathbb{R}^m} \overline{u}_i = \sum_{i=l+1}^{m} \langle \phi_{l+1}, \overline{u}_i \rangle_{\mathbb{R}^m} \overline{u}_i$$

Uvrštavanjem u $\sum_{j=1}^{n} |\langle y_j, \phi_{l+1} \rangle_{\mathbb{R}^m}|^2$ dobivamo

$$\begin{aligned}
\sum_{j=1}^{n} |\langle y_j, \phi_{l+1} \rangle_{\mathbb{R}^m}|^2 &= \sum_{j=1}^{n} \Big| \langle y_j, \sum_{i=l+1}^{m} \langle \phi_{l+1}, \overline{u}_i \rangle_{\mathbb{R}^m} \Big|^2 \\
&= \sum_{j=1}^{n} \sum_{i=l+1}^{m} \sum_{k=l+1}^{m} \langle y_j, \overline{u}_i \rangle_{\mathbb{R}^m} \langle y_j, \overline{u}_k \rangle_{\mathbb{R}^m} \langle \phi_{l+1}, \overline{u}_i \rangle_{\mathbb{R}^m} \langle \phi_{l+1}, \overline{u}_k \rangle_{\mathbb{R}^m} \\
&= \sum_{i=l+1}^{m} \sum_{k=l+1}^{m} \Big\langle \sum_{j=1}^{n} \langle y_j, \overline{u}_i \rangle_{\mathbb{R}^m} y_j, \overline{u}_k \Big\rangle_{\mathbb{R}^m} \langle \phi_{l+1}, \overline{u}_i \rangle_{\mathbb{R}^m} \langle \phi_{l+1}, \overline{u}_k \rangle_{\mathbb{R}^m} \\
&\overset{(1.3)}{=} \sum_{i=l+1}^{m} \lambda_i |\langle \phi_{l+1}, \overline{u}_i \rangle_{\mathbb{R}^m}|^2 \leq \lambda_{l+1} \sum_{i=l+1}^{m} |\langle \phi_{l+1}, \overline{u}_i \rangle_{\mathbb{R}^m}|^2 = \lambda_{l+1}.
\end{aligned}$$

Iz (1.25) slijedi da je $\sum_{j=1}^{n} |\langle y_j, \overline{u}_{l+1} \rangle_{\mathbb{R}^m}|^2 = \lambda_{l+1}$. Ocjena (1.23) još jednom slijedi iz teorema 1.3.2.

□

U praksi, dimenzija prostorne veličine može biti i stotine tisuća, dok je broj uzoraka često mnogo manji. Ideja je iskoristiti vezu između svojstvenih vrijednosti od YY^T i singularnih vrijednosti od Y, te umjesto SVD-a matrice dimenzija $m \times n$ ($n \ll m$) tražiti rješenje svojstvenog problema dimenzija $n \times n$. Iz (1.3) slijedi:

$$Y Y^T u_i = \lambda_i u_i \quad \text{za } i = 1, ..., l. \tag{1.27}$$

Iako je puno manje zastupljen, korisno je znati da i za suprotan slučaj, kada je $m \ll n$, možemo rješavati svojstveni problem:

$$Y^T Y v_i = \lambda_i v_i \quad \text{za } i = 1, ..., l. \tag{1.28}$$

Sada POD bazu možemo računati na sljedeći način: ovisno o dimenzijama m i n definiramo korelacijsku matricu $Y^T Y \in \mathbb{R}^{n \times n}$, odnosno $YY^T \in \mathbb{R}^{m \times m}$ i rješavamo pripadni svojstveni problem (1.27) tj. (1.28). U posljednjem slučaju vektore baze u_i dobivamo iz relacije $Y v_i = \sigma_i u_i$:

$$u_i = \frac{1}{\sqrt{\lambda_i}} Y v_i \quad \text{za } i = 1, ..., l.$$

1.4 POD uz težinski skalarni produkt

POD baza koju smo definirali u prethodnom ulomku sastavljena je od modova koji najviše doprinose energiji sustava, no što u slučaju kada su nam bitni modovi koji sadrže manje energije? Također, mjerenja fizikalnih procesa mogu obuhvatiti više veličina, npr. brzinu, tlak, temperaturu. Te veličine nisu izražene u istim mjernim jedinicama pa njihove promjene nemaju jednaku važnost. Standardni skalarni produkt u ovim slučajevima nije najbolji izbor jer gubimo važne informacije o procesu kojeg promatramo. Promotrimo npr. slučaj Navier-Stokesovih jednadžbi koje opisuju gibanje nestlačivog fluida

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \Delta \mathbf{u}) = \nabla \cdot \sigma(\mathbf{u}, p) + f \tag{1.29}$$

$$\text{div}\mathbf{u} = 0. \tag{1.30}$$

Rješenje se sastoji od dvije komponente: brzine i pritiska. Vektor rješenja tada je u svakoj točki oblika

$$\begin{bmatrix} u_x \\ u_y \\ u_z \\ p \end{bmatrix}, \tag{1.31}$$

no značenje 2-norme dobivenog rješenja, koja iznosi $\|[u_x\ u_y\ u_z\ p]\|_2^2 = u_x^2 + u_y^2 + u_z^2 + p^2$, nije jasno. Stoga proširujemo teoriju na općenitiji težinski skalarni produkt koji je definiran sa simetričnom, pozitivno definitnom matricom. Neka je $W \in \mathbb{R}^{m \times m}$ jedna takva matrica. Tada vrijedi:

$$\langle u, v \rangle_W = u^T W v = \langle u, Wv \rangle_{\mathbb{R}^m} = \langle Wu, v \rangle_{\mathbb{R}^m} \quad \text{za } u, v \in \mathbb{R}^m$$

Norma inducirana danim skalarnim produktom je $\|u\|_W = \sqrt{\langle u, u \rangle_W}$ za $u \in \mathbb{R}^m$.

Za standardni skalarni produkt dovoljno je napraviti SVD matrice Y kako bi pronašli bazu. Kako postupiti kada je skalarni produkt težinski? Matrica W koja definira težinski produkt je realna, simetrična i pozitivno definitna pa posjeduje faktorizaciju Choleskog u obliku $W = LL^T$. Neka je Q proizvoljna ortogonalna matricu i G neka matrica koja zadovoljava $W = GG^T$. Iz

$$W = LL^T = LQQ^T L^T = (\underbrace{LQ}_{G})(LQ)^T = GG^T \tag{1.32}$$

vidimo da zapis W pomoću matrice G nije jedinstven. Za matricu L dobivenu faktorizacijom Choleskog zato postavljamo dodatan zahtjev, da je donje trokutasta s

jedinicama na dijagonali. Takva matrica L je jedinstvena. Po definiciji norme vrijedi

$$\begin{aligned}\|Y\|_W &= \max_{x\neq 0} \frac{\|Yx\|_W}{\|x\|_W} \\ &= \max_{x\neq 0} \frac{\sqrt{(Yx)^T LL^T (Yx)}}{\sqrt{x^T LL^T x}} \\ &= \max_{x\neq 0} \frac{\sqrt{(L^T Yx)^T (L^T Yx)}}{\sqrt{(L^T x)^T (L^T x)}} \\ &= \max_{L^T x\neq 0} \frac{\|L^T Yx\|_2}{\|L^T x\|_2}\end{aligned}$$

Uvedimo supstituciju $L^T x = z$ pa imamo

$$\|Y\|_W = \max_{z\neq 0} \frac{\|L^T Y(L^T)^{-1} z\|_2}{\|z\|_2} = \|L^T Y L^{-T}\|_2$$

Sada je jasno da POD bazu s težinskim skalarnim produktom možemo izračunati pomoću SVD-a na sljedeći način: prvo izračunamo faktorizaciju Choleskog matrice $W = LL^T$. Potom definiramo $\overline{Y} = L^T Y$ te računamo SVD te matrice. Neka je on oblika $\overline{Y} = \overline{U}\Sigma\overline{V}^T$. Nakon što odredimo dimenziju baze, k, vektore baze dobivamo iz $U^k = L^{-T}\overline{U}^k$.

Definicija optimizacijskog problema sa težinskim skalarnim produktom analogna je onoj sa standardnim skalarnim produktom:

$$\underset{\overline{u}_1,\ldots\overline{u}_l\in\mathbb{R}^m}{max} \sum_{i=1}^{l}\sum_{j=1}^{n} |\langle y_j, \overline{u}_i\rangle_W|^2 \quad \text{uz uvjet} \quad |\langle \overline{u}_i, \overline{u}_j\rangle_W|^2 = \delta_{ij} \text{ za } 1 \leq i,j \leq l \tag{1.33}$$

Pokažimo kako možemo iskoristiti rezultat teorema 1.3.1 u ovom slučaju. W je simetrična i pozitivno definitna matrica pa znamo da je SVD oblika $W = Q\Sigma Q^T$, gdje je $\Sigma = \text{diag}(\xi_1, ..., \xi_n)$ dijagonalna matrica čiji su svi elementi pozitivni. Time je osigurana egzistencija $W^{1/2} = Q\text{diag}(\xi_1, ..., \xi_n)Q^T$ pa možemo definirati matricu $\overline{Y} = W^{1/2}Y$. Neka je SVD dekompozicija od $\overline{Y}$ jednaka $\overline{U}\Sigma\overline{V}^T$. Tada je rješenje problema (1.33) dano vektorima $u_i = W^{-1/2}\overline{u}_i,\ i = 1, ..., l$ gdje su $\{\overline{u}\}_{i=1}^m$ stupci matrice $\overline{U}$. Slično kao i ranije, vrijedi:

$$\overline{YY}^T = \overline{U}\Lambda$$

gdje je $\Lambda = \text{diag}(\lambda_1, ..., \lambda_l)$, $\lambda_i = \sigma_i^2, i \in \{1, ..., l\}$, pa su $\overline{u}_i$ ujedno i svojstveni vektori matrice $\overline{YY}^T$. Dakle i ovdje možemo iskoristiti metodu uzoraka.

1.5 Primjena u dinamičkim sustavima

Nelinearne dinamičke sustave često promatramo u obliku sustava običnih diferencijalnih jednadžbi. Takav sustav obično je dobiven Garlekinovom aproksimacijom, o kojoj će biti riječi kasnije. Neka je $T > 0$, $y_0 \in \mathbb{R}^m$, $f : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ i

$$\begin{aligned} \dot{y}(t) =& f(t, y(t)) \quad , \ t \in [0, T] \\ y(0) =& y_0. \end{aligned} \tag{1.34}$$

Pretpostavimo da je f dovoljno glatka te da postoji i jedinstveno je rješenje $y : [0, T] \to \mathbb{R}^m$. Neka je dana ekvidistantna vremenska mreža $0 < t_1 < t_2 < ... < t_n \leq T$ sa korakom $\Delta t = \frac{T}{n-1}$. Pretpostavimo da znamo rješenja sustava (1.34) u trenutcima t_j, $j \in \{1, ..., n\}$, odnosno da imamo n eksperimentalno dobivenih mjerenja fizikalne veličine. Kako bi aproksimacija pomoću POD baze bila dobra, važno je odabrati dovoljno veliku dimenziju baze i važno je da uzorci dobro reprezentiraju dinamiku sustava. U nastavku ćemo opisati kako to postići.

Odabir dimenzije baze

Kako odrediti dimenziju $l \leq d$ takvu da je greška manja od željene tolerancije $\epsilon_{tol} > 0$? Iz (1.11) slijedi da je greška jednaka sumi kvadrata svojstvenih vrijednosti koje pripradaju vektorima baze koje odbacujemo. Slijedi da je dovoljno uzeti najmanji l takav da je

$$\mathcal{E}(l) = \frac{\sum_{i=1}^{l} \sigma_i^2}{\sum_{i=1}^{d} \sigma_i^2} = \frac{\sum_{i=1}^{l} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq 1 - \epsilon_{tol}. \tag{1.35}$$

U slučaju računanja baze metodom uzoraka treba nam ocjena izražena pomoću svojstvenih vrijednosti.

Odabir uzoraka

Primijetimo da uzorci, pa time i vektori POD baze, ovise o trenutcima u kojima uzimamo uzorke. Kako odrediti u kojim trenutcima uzeti uzorak? Za početak, promatramo trajektoriju $y(t) \in \mathbb{R}^m$. Ona je neprekidna po vremenskoj varijabli i vrijedi $\{y(t) | t \in [0, T]\} \subset \mathbb{R}^m$. Neka je $\{u_i\}_{i=1}^m$ ortonormirana baza za prostor $\mathbb{R}^m$, na kojem imamo skalarni produkt $\langle \cdot, \cdot \rangle_W$. Operator $\mathcal{R} : \mathbb{R}^m \to \mathbb{R}^m$, definiran s

$$\mathcal{R}u = \int_0^T \langle y(t), u \rangle_W y(t) dt \in \mathbb{R}^m \tag{1.36}$$

može se prikazati matricom, čiji svojstveni vektori $\{u_i\}_{i=1}^m$ i pripadne svojstvene vrijednosti $\{\lambda_i\}_{i=1}^m$ zadovoljavaju:

$$\mathcal{R}u_i = \lambda_i u_i, \quad \text{za sve } i \in \{1, ..., m\} \quad \text{i} \quad \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_m \geq 0. \tag{1.37}$$

Budući je $\{u_i\}_{i=1}^m$ ONB, $y(t)$ možemo pisati kao

$$y(t) = \sum_{i=1}^{m} \langle y(t), u_i \rangle_W u_i. \tag{1.38}$$

Vrijedi

$$\int_0^T \|y(t)\|_W^2 dt = \int_0^T \langle \sum_{i=1}^{m} \langle y(t), u_i \rangle_W u_i, \sum_{k=1}^{m} \langle y(t), u_k \rangle_W u_k \rangle_W dt \tag{1.39}$$

$$= \int_0^T \sum_{i=1}^{m} \sum_{k=1}^{m} \langle y(t), u_i \rangle_W \langle y(t), u_k \rangle_W \delta_{ik} dt \tag{1.40}$$

$$= \sum_{i=1}^{m} \int_0^T |\langle y(t), u_i \rangle_W|^2 dt \tag{1.41}$$

$$= \sum_{i=1}^{m} \lambda_i, \tag{1.42}$$

pa zbog

$$\sum_{i=1}^{m} \int_0^T |\langle y(t), u_i \rangle_W|^2 dt = \sum_{i=1}^{m} \int_0^T \langle y(t), u_i \rangle_W \langle y(t), u_i \rangle_W \tag{1.43}$$

$$= \sum_{i=1}^{m} \int_0^T \langle \langle y(t), u_i \rangle_W y(t), u_i \rangle_W dt \tag{1.44}$$

$$= \sum_{i=1}^{m} \int_0^T \langle \mathcal{R} u_i, u_i \rangle_W dt \tag{1.45}$$

slijedi

$$\int_0^T \|y(t)\|_W^2 = \sum_{i=1}^{m} \langle \mathcal{R} u_i, u_i \rangle_W = \sum_{i=1}^{m} \lambda_i. \tag{1.46}$$

Budući da je promatrana trajektorija neprekidna u varijabli t postavlja se pitanje kako numerički provesti račun. Formule za numeričku integraciju često sadrže dodatne varijable: težine. Iz tog razloga ćemo svakom uzorku y_j pridružiti nenegativnu težinu α_j, za $1 \leq j \leq l$. Sada su vektori POD baze rješenje minimizacijskog problema koji glasi:

$$\min_{u_1, \ldots u_l \in \mathbb{R}^m} \sum_{j=1}^{n} \alpha_j \|y_j - \sum_{i=1}^{l} \langle y_j, u_i \rangle_W u_i\|_W^2 \quad \text{za } \langle u_i, u_j \rangle_{\mathbb{R}^m} = \delta_{ij},\ 1 \leq i, j \leq l. \tag{1.47}$$

Ujedno su i rješenje svojstvenog problema

$$YDY^TWu_i = \lambda_i u_i \quad \text{za } i = 1, ..., l. \tag{1.48}$$

gdje je $D = diag(\alpha_1, ..., \alpha_n)$. Postupak je analogan postupku u teoremu 1.3.1. Matrica YDY^TW u (1.48) nije simetrična, ali može se pokazati da je slična simetričnoj matrici. Ranije smo pokazali da je $W^{1/2}$ dobro definirano. Imamo:

$$W^{1/2}YDY^TWW^{-1/2} = (W^{1/2}YD^{1/2})((W^{1/2})^TY^T(D^{1/2})^T).$$

Uvedimo oznake $\overline{Y} = W^{1/2}YD^{1/2}$ i $\overline{u}_i = W^{1/2}u_i$ uz uvjet $\langle u_i, u_j\rangle_W = \delta_{ij},\ 1 \leq i, j \leq l$. Zaključujemo da rješenje problema (1.47) možemo ga izračunati iz simetričnog svojstvenog problema

$$\overline{Y}\overline{Y}^T\overline{u}_i = \lambda_i\overline{u}_i, \quad 1 \leq i \leq l. \tag{1.49}$$

Iz diskretnog problema (1.48) imamo

$$\sum_{j=1}^{n} \alpha_j \|y(t_j)\|_W^2 = \sum_{i=1}^{m} \lambda_i \tag{1.50}$$

za svaki $n \in \mathbb{N}$. Sada je očito da težine α_j možemo uzeti tako da vrijedi

$$\lim_{n\to\infty} \sum_{j=1}^{n} \alpha_j \|y(t_j)\|_W^2 \to \int_0^T \|y(t)\|_W^2 dt. \tag{1.51}$$

Dakle neprekidni sustav možemo dobro aproksimirati diskretnim ukoliko imamo dovoljan broj uzoraka. Osvrnimo se na par mogućih načina uzimanja uzoraka. Ukoliko radimo numeričku simulaciju dinamičkog sutatva svakako je moguće uzeti uzorak u svakom vremenskom koraku postupka. Prednost ove metode je što se, kako se broj uzoraka n približava broju prostornih točaka m, greška aproksimacije smanjuje. Za duže intervale promatranja proizvesti ćemo previše uzoraka pa će računanje SVD-a matrice uzoraka biti računski i vremenski neefikasno. Veliku dimenziju možemo izbjeći tako da spremamo svako $j-$ ti vektor rješenja, no time gubimo na točnosti aproksimacije. Naprednije metode uključuju rješavanje uvjetnih problema minimizacije, jednu takvu promatrali su Kunisch i Volkwein u [5]. Uz inicijalnih n uzoraka u vremenima $t_1, ...t_n$ cilj je odrediti vremenske trenutke u kojima je potrebno uzeti dodatnih m uzoraka kako bi imali što točniju sliku dinamike sustava. Pravim odabirom treunutaka u kojima uzimamo uzorke možemo unaprijed bitno reducirati vrijeme računanja POD dekompozicije. U trenutku kada smo izračunali POD bazu možemo je primjeniti za dobivanje modela nižeg reda.

1.6 Model nižeg reda. Galerkinova projekcija

Često se rješavanje složenih dinamičkih sustava svodi na numeričku aproksimaciju. Kako bi metoda bila postigla određenu točnost moramo uzeti male korake diskretizacije, što dovodi do velikih dimenzija aproksimacijskog prostora, pa onda i vremenski zahtijevnog računa. Takvi modeli nisu prikladni za upotrebu u određenim primjenama, npr. kod optimizacije po određenom broju parametara, budući da nam informacija treba u jako kratkom vremenskom roku. Iz tog razloga tražimo model nižeg reda, koji ima znatno manji broj stanja, ali koji sadrži većinu energije prvobitnog modela. Srećom, trajektorije rješenja dinamičkih sustava često žive na mnogostrukostima koje se mogu vrlo dobro aproksimirati s linearnim mnogostrukostima a koje su znatno niže dimenzije. Dakle prvi korak u konstrukciji modela nižeg reda je provjera istinitosti gornje pretpostavke. To možemo jednostavno napraviti. Uzorke od manjeg broja trajektorija spremimo u matricu i izračunamo singularne vrijednosti. Ukoliko njihove vrijednosti brzo opadaju, sustav je prikladan za redukciju.

Za dobivanje modela nižeg reda u "realnom" vremenu nećemo računati bazu, već ćemo iskoristiti POD bazu koju smo izračunali u tzv. "offline" fazi. Offline faza je faza predračunanja. Pretpostavimo da dani sustav ovisi o nekom parametru λ čije vrijednosti se nalaze unutar intervala $[\alpha_1, \alpha_2]$. Tada u offline fazi računamo POD bazu za n različitih vrijednosti parametra $\lambda \in [\alpha_1, \alpha_2]$ i od dobivenih baza istim algoritmom (vektori dobivenih baza su uzorci) tražimo novu POD bazu. Ta baza će biti "dobra" za svaki od tih n slučajeva, ali i za druge vrijednosti $\lambda \in [\alpha_1, \alpha_2]$, koje nismo imali u predračunu. Pretpostavimo da smo izračunali prvih l funkcija POD baze $\{u_1, ...u_l\}$ za neki fiksan l. One razapinju konačnodimenzionalan prostor

$$\mathcal{U} = \text{span}\{u_1, ..., u_l\} \subset \mathbb{R}^m \tag{1.52}$$

Neka je dan sustav (1.34). Želimo odrediti sustav koji "živi" na $\mathcal{U}$ i dovoljno dobro aproksimira (1.34). Definiramo aproksimaciju (reda $l < m$) od $y(t)$ sa

$$y^l(t) = \sum_{i=1}^{l} \langle y^l(t), u_j \rangle_W u_j \quad \text{za sve } t \in [0, T]. \tag{1.53}$$

Dakle $y^l(t) \in \mathcal{U}$. Ako su zadovoljeni uvjeti Galerkinove projekcije mora vrijediti

$$\langle \frac{d}{dt} y^l(t) - f(t, y^l(t)), u_i \rangle_W = 0, \quad \text{za } i = 1, ..., l, \tag{1.54}$$

tj.

$$\langle \frac{d}{dt} y^l(t), u_i \rangle_W = \langle f(t, y^l(t)), u_i \rangle_W \tag{1.55}$$

$$\frac{d}{dt} \sum_{j=1}^{l} \langle y^l(t), u_j \rangle_W \langle u_j, u_i \rangle_W = \langle f(t, \sum_{j=1}^{l} \langle y^l(t), u_j \rangle_W u_j), u_i \rangle_W \tag{1.56}$$

Uvedimo oznake $\mathbf{y}_j^l = \langle y^l(t), u_j \rangle_W \quad$ za $j = 1, ..., l, \mathbf{y}^l = [\mathbf{y}_1^l, ..., \mathbf{y}_l^l]^T \in \mathbb{R}^l$ i $[F(\mathbf{y}(t))]_i = \langle f(\sum_{j=1}^{l} \mathbf{y}_j u_j), u_i \rangle_W$ za $i = 1, ..., l$. Iz $\mathbf{y}^l(0) = \mathbf{y}_0^l$ slijedi $\mathbf{y}_0^l = (\langle y_0^l, u_1^l \rangle_W \ ... \ \langle y_0^l, u_l \rangle_W)^T$. Sada reducirani sistem zapisujemo kao

$$\frac{d}{dt} \mathbf{y}^l(t) = F(t, \mathbf{y}^l(t)) \tag{1.57}$$

$$\mathbf{y}^l(0) = \mathbf{y}_0^l = U^T W y_0 \tag{1.58}$$

U slučaju kada je f linearna vrijedi

$$\frac{d}{dt} \mathbf{y}_i^l(t) = \langle f(\sum_{j=1}^{l} \mathbf{y}_j^l u_j), u_i \rangle_W = \sum_{i=1}^{l} \mathbf{y}_i^l \langle f(u_i), u_i \rangle_W. \tag{1.59}$$

Vrijednosti $\langle f(u_i), u_i \rangle_W$. Sustav tada$1 \leq i \leq l$, ne ovise o vremenu t i možemo ih odrediti prije rješavanja sustava. Spremimo vrijednosti u dijagonalnu matricu $A \in \mathbb{R}^{n \times n}$ tako da je $A_{ii} = \langle f(u_i), u_i \rangle_W$. Sustav tada poprima oblik:

$$\frac{d}{dt} \mathbf{y}^l(t) = A \mathbf{y}^l(t) \tag{1.60}$$

$$\mathbf{y}^l(0) = \mathbf{y}_0^l \tag{1.61}$$

Neka je $U = [u_1, ..., u_l]$ matrica koja sadrži vektore POD baze. Definirajmo aproksimaciju iz (1.53) pomoću POD modova:

$$y(t) \approx U \mathbf{y}^l. \tag{1.62}$$

Tada je reducirani oblik sustava (1.34), za linearnu funckiju f jednak:

$$\frac{d}{dt} \mathbf{y}^l(t) = \tilde{A} \mathbf{y}^l(t) \tag{1.63}$$

$$\mathbf{y}^l(0) = \mathbf{y}_0^l, \tag{1.64}$$

gdje je $\tilde{A}_{ij} = \langle A u_i, u_j \rangle_W \in \mathbb{R}^{l \times l}$ i $\mathbf{y}_0^l = U^T \mathbf{y}_0 \in \mathbb{R}^l$. Ukoliko je funkcija f nelinearna imamo:

$$F(t, \mathbf{y}^l(t)) = U^T W f(t, U \mathbf{y}^l(t)) = \langle f(t, y(t)), U \rangle_W \tag{1.65}$$

pa je sustav nižeg reda:

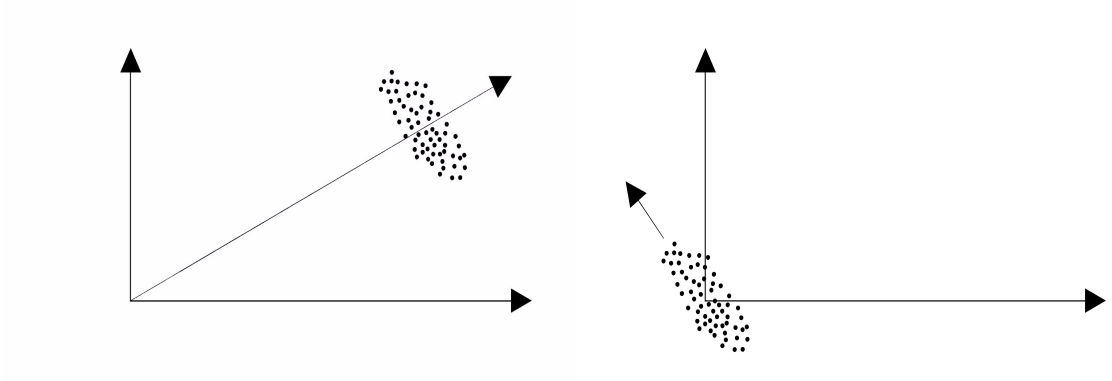
$$\frac{d}{dt}\mathbf{y}^l(t) = U^T W f(t, U\mathbf{y}^l(t)) \tag{1.66}$$

$$\mathbf{y}^l(0) = \mathbf{y}_0^l = U^T W f(U\mathbf{y}^l). \tag{1.67}$$

Kako bismo izračunali skalarni produkt u (1.65) potrebno je prvo vektor $\mathbf{y}^l \in \mathbb{R}^l$ množenjem s U proširiti na m-dimenzionalni vektor $U\mathbf{y}^l$, potom izračunati vrijednosti $f(t, U\mathbf{y}^l)$ te naposljetku množenjem s U^T dobiti reducirani model. Računski, ovaj proces je jako nepovoljan, jer zahtjeva računanje punog sustava. Bolje rezultate možemo postići korištenjem DEIM[3] metode.

1.7 Ocjena greške

Za ocjenu greške korisno je, umjesto projekcije na linearan potprostor, tražiti projekciju na afini potprostor. Linearan potprostor ćemo translatirati za srednju vrijednost stupaca matrice uzoraka. Translacija nema utjecaja na računanje već samo na interpretaciju rješenja. Pogledajmo zašto je ona korisna. Neka je dan skup mjerenja kao na slici 1.5 Pri računanju SVD-a dominantni svojstveni vektor sustava na lijevoj slici



(a) Početni sustav (b) Translatirani sustav

Slika 1.4

biti će u smjeru "oblaka" u kojem se nalaze vrijednosti mjerenja što u stvarnosti nije dominantan smjer. Translacijom sustava izbjegavamo taj problem (slika 1.4b) i dominantan smjer je dobro prepoznat. Neka je $\overline{y} = \frac{1}{nT}\sum_{i=1}^{m}\int_0^T y_i(t)dt$ srednja vrijednost

[3]pogledati npr. u [6]

m promatranih trajektorija i neka je $\mathcal{U}$ $l-$dimenzionalan linearan potprostor definiran u (1.52). Tada optimalan afin potprostor koji prolazi kroz $\overline{y}$, nazovimo ga $\mathcal{U}_{\overline{y}}$, dobivamo tako da linearan potprostor $\mathcal{U}$ translatiramo za vrijednost $\overline{y}$. Ovdje nam je cilj dati ocjenu na grešku pri kontrukciji modela nižeg reda za općeniti nelinearni sustav

$$\dot{y} = f(y,t) \quad , \; t \in [0,T] \tag{1.68}$$

$$y(0) = y_0. \tag{1.69}$$

Projekciju vektorskog polja f na potprostor $\mathcal{U}_{\overline{y}}$ dobivamo tako da prvo za svaku točku $p \in \mathcal{U}_{\overline{y}}$ računamo vrijednosti $f(p,t)$, te potom definiramo projekciju sa $f_{\overline{y}} = Pf(p,t)$. Ako su $y \in \mathbb{R}^m$ koordinate točke koja pripada dinamičkom sustavu u polaznom koordinatnom sustavu, projekcijom te točke na afini potprostor pridružujemo joj nove koordinate $z \in \mathbb{R}^l$, koje dobivamo na sljedeći način:

$$z = P(y - \overline{y}), \tag{1.70}$$

gdje je P marica projekcije s $\mathbb{R}^m$ na $\mathcal{U}$. Iz prethodno pokazanog, znamo da je ta matrica jednaka $U_l^T = [u_1 \dots u_l]^T$. Jasno je da pri ovoj projekciji radimo grešku. Kako bismo izračunali projekciju vektorskog polja f na potprostor $\mathcal{U}_{\overline{y}}$, za svaku točku $p \in \mathcal{U}_{\overline{y}}$ potrebno je izračunati vrijednosti $f(p,t)$. Potom računamo projekciju koju definiramo sa $f_{\overline{y}} = Pf(p,t)$. U ovom koraku također radimo grešku.

Dakle, kada tražimo model nižeg reda dva su koraka u kojima radimo grešku. Vektor rješenja $y \in \mathbb{R}^m$ možemo zapisati na sljedeći način:

$$y(t) = P^T u(t) + (P^C)^T v(t) + \overline{y}, \tag{1.71}$$

gdje je $u(t) \in \mathbb{R}^l$ a $v \in \mathbb{R}^{m-l}$. Znamo da je $P^T u(t) + \overline{y}$ upravo projekcija od $y(t)$ na $\mathcal{U}_{\overline{y}}$. Tada je greška projekcije jednaka

$$e_1(t) = P^T u(t) + \overline{y} - y(t) = -(P^C)^T v(t) \tag{1.72}$$

i okomita je na $\mathcal{U}_{\overline{y}}$. Slično, rješenje nižeg reda možemo zapisati kao:

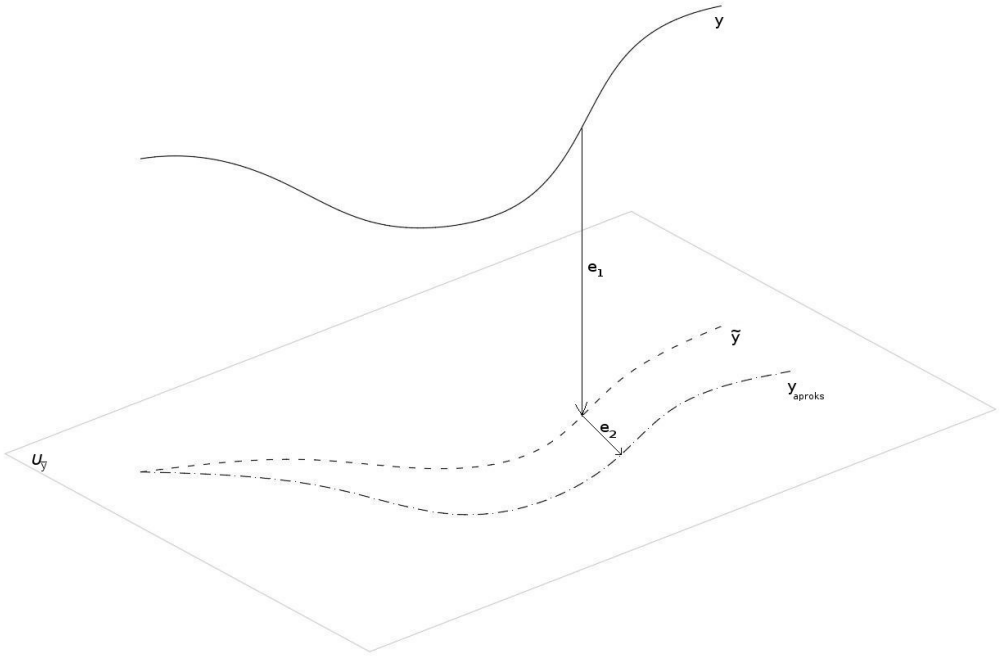
$$y_{\text{aproks}} = P^T u(t) + P^T w(t) + \overline{y}, \tag{1.73}$$

gdje je $w(t) \in \mathbb{R}^l$. Član $P^T w(t)$ rezultat je greške koja nastaje pri projekciji vektorskog polja na $\mathcal{U}_{\overline{y}}$. Dakle

$$e_2(t) = P^T w(t) = y_{\text{aproks}}(t) - (P^T u(t) + \overline{y}) \tag{1.74}$$

leži u $\mathbb{R}^l$, točnije u $\mathcal{U}_{\overline{y}}$. Zaključujemo da ukupna greška iznosi

$$e(t) = y_{\text{aproks}} - y(t) = -(P^C)^T v(t) + P^T w(t). \tag{1.75}$$



Slika 1.5: *Greška pri računanju modela nižeg reda*

Sljedeći teorem, dokazan u [2], daje ocjenu na ukupnu grešku reduciranog sustava u ovisnosti o grešci aproksimacije. Prije iskaza treba nam definicija logaritamske norme: logaritamska norma pridružena 2-normi definirana je s

$$\mu(Y) = \lim_{h \to 0^+} \frac{\|I + hY\|_2 - 1}{h}$$

Teorem 1.7.1. *Neka je dan sustav* $\dot{y}(t) = f(t, y(t))$, $y(0) = y_0$ *na* $[0, T]$. *Pretpostavimo da rješenje tražimo u obliku modela nižeg reda dobivenog POD-om. Neka je* $P \in \mathbb{R}^{l \times n}$ *matrica projekcije i* $\mathcal{U}_{\overline{y}}$ *afin potprostor na koji projiciramo. Neka vrijede* (1.71),(2.6)(1.72) *i* (1.74) *te neka je* $\tilde{y}(t) = P^T u(t) + \overline{y}$. *Neka je* $\gamma \geq 0$ *Lipschitzova konstanta td. vrijedi*

$$\|Pf(\tilde{y}(t) + (P^C)^T v(t), t) - Pf(\tilde{y}(t), t)\| \leq \|v\|$$

za sve $(v, t) \in D \subset \mathbb{R}^{n-l}$ *područje takvo da* $\tilde{D} = \{(\tilde{y}(t) + (P^C)^T v, t) : (v, t) \in D\} \subset \mathbb{R}^n \times [0, T]$ *sadrži* $(\tilde{y}(t), t)$ *i* $(y(t), t)$ *za sve* $t \in [0, T]$. *Neka je* $\mu(P\frac{\partial f}{\partial y}(y_{aproks}(t) + P^T z, t)P^T) \leq \overline{\mu}$ *za* $(z, t) \in V \subset \mathbb{R}^l \times [0, T]$, *gdje je* V *područje takvo da se za sve*

$t \in [0,T]$, $(u(t),t)$ i $(u(t)+w(t),t)$ nalaze untar V. Tada vrijedi:

$$\|e_2\|_\infty \leq \|e_1\|_2 \frac{\gamma}{\sqrt{2\overline{\mu}}} \sqrt{e^{2\overline{\mu}T} - 1} \tag{1.76}$$

$$\|e\|_2 \leq \|e_1\|_2 \sqrt{1 + \frac{\gamma^2}{4\overline{\mu}^2}(e^{2\overline{\mu}T} - 1 - 2\overline{\mu}T)}. \tag{1.77}$$

Primijetimo da greška ne ovisi o f. U idealnom slučaju, P i $\overline{y}$ računamo iz analitičkog rješenja y pa za grešku projekcije vrijedi $\|e_1\|_2 = \sqrt{\sum_{i=l+1}^{d} \lambda_i}$. U suprotnom, kada je dani skup podataka izračunat numerički, moramo pridodati i grešku računanja samih ulaznih podataka.

Poglavlje 2

DMD

Kod POD dekompozicjie glavni cilj je pronaći modove koji najviše pridonose energiji sustava. Međutim ti modovi nam ne govore previše o samoj dinamici sustava. Cilj DMD dekompozicije je otkriti strukturu gibanja. Od kakvih se gibanja sastoji, koji je njihov intenzitet i frekvencija. DMD nije osnovan na optimalnoj projekciji na svako od stanja dinamičkog sustava, kao POD, već na dekompoziciji na modove koji tvore dinamiku sustava. Rezultat je prostorno-vremenska dekompozicija podataka u skup DMD (ili dinamičkih) modova, uz poznate frekvencije i stupanj rasta tj. opadanja. Sve to ponekad je nužno bez eksplicitnog poznavanja operatora dinamičkog sustava, samo na osnovi empirijski dobivenih podataka. Analizom podataka možemo doći do aproksimacijskog linearnog modela (dinamičkog operatora), koji je optimalan u smislu najmanjih kvadrata. Prednost DMD dekompozicije je što do svih rezultata dolazimo linearnim operacijama bez rješavanja jednadžbi, čak i u slučaju neliearnog sustava. Pokazat ćemo kako je moguće iskoristiti zavisnost u danim podatcima kako bi taj aproksimacijski linearni model bio nižeg ranga.

2.1 DMD algoritam

Unatoč tome što je DMD dekompozicija relativno nedavno razvijena, već su se pojavile mnoge njene inačice. Ovdje ćemo ih razlikovati prema ograničenjima na skup ulaznih podataka. Neka je za početak taj skup dan uzastopnim mjerenjima s fiksnim vremenskim korakom. Za svaki trenutak $t_k \in [0, T]$ u kojem vršimo mjerenje (ili računanje) neke veličine, za svaku točku mjerenja, podatke o veličini spremamo u vektor stupac. Označimo sa x_k vektor dobiven u trenutku t_k. Definirajmo matricu X koja se sastoji od m uzoraka:

$$X_1^m = \begin{bmatrix} x_1 & x_2 & \dots & x_m \end{bmatrix} \tag{2.1}$$

i matricu X^{m+1} koja je pomaknuta za jedan vremenski korak:

$$X_2^{m+1} = \begin{bmatrix} x_2 & x_3 & \dots & x_{m+1} \end{bmatrix} \tag{2.2}$$

Pretpostavka je da su vektori x_k i x_{k+1}, pa onda i matrice X^m i X^{m+1}, povezani linearnim operatorom A, tj da vrijedi:

$$x_{k+1} = Ax_k \tag{2.3a}$$
$$X_2^{m+1} = AX_1^m. \tag{2.3b}$$

Ako se prisjetimo npr. Eulerove metode za rješavanje običnih diferencijalnih jednadžbi, možemo uočiti sličnosti. Rješenje problema

$$\begin{aligned} y'(t) =& f(t, y(t)) \\ y(t_0) =& y_0 \end{aligned}$$

računamo na sljedeći način:

$$y_{k+1} = y_k + hf(t_k, y_k), \tag{2.4}$$

gdje je $t_k = t_0 + k * h$, te $y_k = y(t_k)$ za vremenski korak h. U svakom koraku metode, vrijednosti u trenutku t_{k+1} računamo iz vrijednosti u trenutku t_k. Slično, $x_k + 1$ možemo izračunati iz x_k. Također, prvi DMD algoritam razvijen je kao inačica Arnoldijevog algoritma pa je metoda bila ograničena upravo na skup podataka koji se sastoji od niza uzoraka dobivenih s fiksnim vremenskim razmakom između mjerenja. Matrica A u (2.3b) približno je jednaka za sve vremenske korake što znači da je preslikavanje uzoraka konstantno. Primijetimo da tada vrijedi:

$$x_{k+1} = A^k x_1$$

pa od uzoraka x_k, $k = 1, \dots, m$ možemo konstruirati Krilovljev niz

$$\mathcal{K}^m = \{x_1, Ax_1, A^2x_1, ..., A^{m-1}x_1\}.$$

Cilj nam je na osnovu niza $\mathcal{K}^m$ odrediti karakteristike dinamičkog procesa opisanog matricom A. Uzorci će nakon nekog broja (neka je to m) postati linearno zavisni[1], pa se za svaki $k \geq m$ vektor x_k može prikazati kao linearna kombinacija vektora $x_1, \dots, x_m$. Imamo

$$x_k = \sum_{k=1}^{m} a_k x_k + \mathbf{r} = X_1^m \mathbf{a} + \mathbf{r}, \quad \text{za } k > m,$$

[1] to će se sigurno dogoditi budući da je prostor konačnodimenzionalan

gdje je $\mathbf{a} = [a_1\ a_2 \dots\ a_m]^T$ vektor koeficijenata a $\mathbf{r}$ vektor reziduala. Iz (2.3b) slijedi:

$$X_2^{m+1} = AX_1^m = \begin{bmatrix} x_2 & x_3 & \dots & x_m & X_1^m\mathbf{a} \end{bmatrix} + \mathbf{r}e_m^T. \tag{2.5}$$

Matricu $[x_2\ x_3 \dots\ x_m\ 0]$ možemo dobiti na način da na matricu X_1^m sa desne strane primijenimo matricu lijevog šifta, koja je oblika

$$L = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & & 0 & 0 \\ 0 & 1 & 0 & & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}.$$

Sada je očito da se $[x_2\ x_3 \dots\ x_m\ X_1^m\mathbf{a}]$ može prikazati kao $X_1^m C$ gdje je

$$C = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & a_1 \\ 1 & 0 & 0 & & 0 & a_2 \\ 0 & 1 & 0 & & 0 & a_3 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & 1 & a_m \end{bmatrix}.$$

C nazivamo matrica pratilica. Cilj nam je minimizirati rezidual $\mathbf{r}$, tj naći C za koji vrijedi

$$C = \arg\min_{B \in \mathbb{C}^{m\times m}} \|X_2^{m+1} - X_1^m B\|_2. \tag{2.6}$$

Rješenje gorenjeg problema jedinstveno je ako su stupci matrice X_1^m linearno nezavisni i možemo ga računati pomoću pseudo-inverza

$$C = (X_1^m)^\dagger X_2^{m+1} \tag{2.7}$$

gdje smo sa $(X_1^m)^+$ označili pseudoinverz od X_1^m, koji daje optimalno rješenje u smislu najmanjih kvadrata. C možemo dijagonalizirati na sljedeći način:

$$C = T^{-1}\Lambda T, \tag{2.8}$$

gdje je Λ dijagonalna matrica s vrijednostima $\lambda_j, j = 1, \dots, m^2$ na dijagonali a T Vandermondova matrica oblika

$$\begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \dots & \lambda_1^{n-1} \\ 1 & \lambda_2 & \lambda_2^2 & \dots & \lambda_2^{n-1} \\ 1 & \lambda_3 & \lambda_3^2 & \dots & \lambda_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_m & \lambda_m^2 & \dots & \lambda_m^{n-1} \end{bmatrix}. \tag{2.9}$$

[2] radimo pod pretpostavkom da su vrijednosti λ_j međusobno različite

Vrijednosti λ_j nazivamo Ritzove vrijednosti dok vektore $[v_1 \dots v_m]$ dane u matrici

$$V = X_1^m T^{-1} \tag{2.10}$$

nazivamo Ritzovim vektorima. Očito X_1^m možemo zapisati kao $X_1^m = VT$ pa za X_2^{m+1} vrijedi

$$\begin{aligned} X_2^{m+1} =& X_1^m C + \mathbf{r} e_m^T && (2.11)\\ =& X_1^m T^{-1} \Lambda T + \mathbf{r} e_m^T && (2.12)\\ =& V T T^{-1} \Lambda T + \mathbf{r} e_m^T && (2.13)\\ =& V \Lambda T + \mathbf{r} e_m^T. && (2.14) \end{aligned}$$

Slijedi da se vektori x_k, za $k = 1, ...m$ mogu zapisati kao

$$x_k = \sum_{j=1}^{m} \lambda_j^k v_j, \quad \text{za } k = 1, ..., m, \tag{2.15}$$

te za $k = m + 1$

$$x_k = \sum_{j=1}^{m} \lambda_j^k v_j + \mathbf{r}, \quad \mathbf{r} \perp \operatorname{span}\{x_1, ...x_m\}, \tag{2.16}$$

Svojstvene vrijednosti matrice C su aproksimacije nekih svojstvenih vrijednosti od A. Isto vrijedi i za svojstvene vektore. U praksi nije poželjno računati na ovakav način budući da je osjetljiv na nepravilnosti koje se javljaju kod eksperimentalnih mjerenja. To rezultira lošom uvjetovanosti algoritma pa, u praksi, dobijemo samo jednu ili dvije najveće svojstvene vrijednosti.

Standardni/Schimdtov DMD

Loša uvjetovanost opisanog algoritma može biti posljedica nedovoljnog ranga matrice X_1^m. Kako bi to izbjegli možemo iskoristiti SVD dekompoziciju[3] od X_1^m te odbaciti vektore čija je pripadna singularna vrijednost relativno mala. Neka je SVD dekompozicija od X_1^m jednaka $U\Sigma V^*$. Iz (2.5) slijedi

$$X_2^{m+1} = AX_1^m + \mathbf{r} e_m^T = AU\Sigma V^* + \mathbf{r} e_m^T = (U\Sigma V^*)C + \mathbf{r} e_m^T. \tag{2.17}$$

[3] u slučaju matrice kompleksnih vrijednosti SVD ne možemo vizualizirati kao i za realne matrice no teorem 1.1.1 vrijedi

Pretpostavimo da nakon odbacivanja imamo k vektora, tj. $X_1^m \approx U_k \Sigma_k V_k^*$. Uvedimo oznaku za odbačeni dio $\delta = \sum_{i=k+1}^{m} \sigma_i u_i v_i^*$. Umjesto računanja matrice pratilice C računat ćemo matricu $\tilde{A}_k$, koju nazivamo Rayleighov kvocijent. Djelovanje operatora A poznato nam je samo na vektorima $x_1, ..., x_m$ pa ima smisla tražiti projekciju od A na skup kojeg ti vektori čine. Znamo da je POD baza, dimenzije k, vektora $\{x_1, ..., x_m\}$ dana stupcima matrice $U = [u_1 \dots u_k]$. Ranije smo pretpostavili da se stupci od X_2^{m+1} mogu zapisati kao linearna kombinacija stupaca matrice X_1^m. Slično tome, ovdje zahtijevamo da se mogu zapisati kao linearna kombinacija vektora $u_1, ..., u_k$. Dobivamo

$$X_2^{m+1} = AX_1^m = A(U_k \Sigma_k V_k^* + \delta) = (U_k \Sigma_k V_k^* + \delta)C + \mathbf{r} e_m^T \tag{2.18}$$

Iz definicije U_k i δ slijedi da je $U_k^* \delta = 0$. Jednako dobivamo i $\delta V_k^* = 0$.

$$U_k^* X_2^{m+1} = U_k^* A U_k \Sigma_k V_k^* = \underbrace{U_k^* U_k}_{=0} \Sigma_k V_k^* C + U_k^* \mathbf{r} e_m^T. \tag{2.19}$$

Budući je $\mathbf{r} \perp \operatorname{span}\{x_1, \dots, x_m\}$ slijedi

$$U_k^* A U_k = \Sigma_k V_k^* C V_k \Sigma_k^{-1} \tag{2.20}$$

Matricu $\tilde{A}_k$ definiranu s $\tilde{A}_k = U_k^* A U_k$ nazivamo Rayleighov kvocijent od C. Neka je $\tilde{A}_k W_k = \Lambda_k W_k$, tj. neka je Λ_k matrica svojstvenih vrijednosti, a W_k matrica desnih svojstvenih vektora od $\tilde{A}_k$. Zbog $\tilde{A}_k = W_k \Lambda_k W_k^{-1}$ iz definicije $\tilde{A}_k$ slijedi

$$AU_k W_k = U_k W_k \Lambda_k \tag{2.21}$$

Zaključujemo da, ukoliko je w svojstveni vektor od $\tilde{A}_k$ tada je $U_k w$ aproksimacija svojstvenog vektora od A. Slijedi da su svojstvene vrijednosti Λ_k matrice $\tilde{A}_k$ aproksimacije svojstvenih vrijednosti od A. Elemente Λ_k nazivamo Ritzove vrijednosti, a vektore koji se nalaze u matrici $\Phi = U_k W_k$ Ritzovi vektori. Time smo pokazali da je dovoljno izračunati dekompoziciju svojstvenih vrijednosti matrice $\tilde{A}_k$. Pri računanju $\tilde{A}_k$ nećemo koristiti definiciju[4]. Iz prve jednakosti u (2.19) vrijedi

$$\tilde{A}_k = U_k^* A U_k = U_k^* X_2^{m+1} V_k \Sigma_k^{-1}. \tag{2.22}$$

Dakle Rayleighov kvocijent možemo izračunati poznavajući samo matrice uzoraka. Za detaljnije objašnjenje Rayleighovog kvocijenta vrijedi pogledati [9].

Napomena 1. *U slučaju da je $k = m$ matrice $\tilde{A}_k$ i A bile bi povezane transformacijom sličnosti, koja nam osigurava jednakost svojstvenih vrijednosti od $\tilde{A}_k$ i A, dok na svojstveni vektore od $\tilde{A}_k$ treba djelovati transformacijom sličnosti, u ovom slučaju množenje s matricom U, kako bi dobili svojstvene vektore od A.*

[4]kada bismo imali A koji je potreban za računanje $\tilde{A}_k$ iz definicije, ne bismo trebali računati $\tilde{A}_k$

Iako je Schmidtov DMD algoritam općeprihvaćen algoritam za primjenu na uzorcima koji su nastali iz vremenski uzastopnih mjerenja s fiksnim korakom, to svojstvo nigdje u računu nije iskorišteno, stoga Schimdtov algoritam vrijedi i za općenitiji slučaj.

Algorithm 1: Standard DMD

1: Definiraj matrice X_1^m i X_2^{m+1} kao u (2.1) i (2.2)
2: Izračunaj SVD dekompoziciju od X_1^m:
3: $$X = U\Sigma V^*,$$
4: gdje je $U \in \mathbb{C}^{n\times r}$, $\Sigma \in \mathbb{C}^{r\times r}$, $V \in \mathbb{C}^{n\times r}$ a r rang od X_1^m
5: Odredi rang k
6: Definiraj matricu $\tilde{A} = U_k^* X_2^{m+1} V_k \Sigma_k^{-1}$
7: Izračunaj svojstvene vrijednosti i vektore od $\tilde{A}_k$: $\tilde{A}_k W_k = \Lambda_k W_k$
8: DMD modove dobivamo iz:
9: $$\Phi = U_k W_k$$

Egzaktni DMD

Promatrajmo parove podataka $\{(x_1, y_1), ..., (x_m, y_m)\}$, za koje vrijedi

$$y_k = Ax_k. \tag{2.23}$$

Definiramo matrice $X, Y \in \mathbb{C}^{n\times m}$:

$$X = \begin{bmatrix} x_1 & x_1 & \dots & x_m \end{bmatrix} \quad \text{i } Y = \begin{bmatrix} y_1 & y_2 & \dots & y_m \end{bmatrix}. \tag{2.24}$$

Definicija 2.1.1. *Za skup podataka dan kao u* (2.24) *definiramo operator*

$$A = YX^\dagger. \tag{2.25}$$

Tada je dekompozicija na dinamičke modove para (X, Y) *dana dekompozicijom svojstvenih vrijednosti matrice* A*. DMD modovi i svojstvene vrijednosti su svojstveni vektori i svojstvene vrijednosti od* A*, respektivno.*

Ako je SVD dekompozicija od X oblika $U\Sigma V^*$, znamo da je Moore-Penroseov pseudoinverz oblika $V\Sigma^\dagger U^*$. Sada možemo eksplicitno izraziti A kao

$$A = YV\Sigma^\dagger U^*. \tag{2.26}$$

Ovdje, kao i kod Schmidtovog algoritma, možemo iskoristiti projekciju od A na POD bazu dimenzije $k \leq m$, koja je dana u vektorima stupcima matrice $U_k = [u_1 \dots u_k]$. Kao i ranije definiramo $\tilde{A}_k$ i dobivamo jednakost:

$$\tilde{A}_k = U_k^* A U_k = U_k^* Y V_k \Sigma_k^{-1}. \tag{2.27}$$

Neka je

$$\tilde{A}_k W_k = \Lambda_k W_k, \tag{2.28}$$

sprektralna dekompozicija matrice $\tilde{A}_k$, gdje je Λ_k matrica svojstvenih vrijednosti a W_k pripadnih svojstvenih vektora. Zaključujemo da su vrijednosti u Λ_k aproksimacije vrijednosti od Σ_k. Aproksimacija svojstvenih vektori od A, dana je sa

$$\Phi = Y V_k \Sigma_k^{\dagger} W_k. \tag{2.29}$$

Razlika između egzaktnog i Scmidtovog algoritma je u računanju DMD modova. Može se pokazati (Teorem 1 u [3]) da su DMD modovi sadržani u stupcima matrice $\Phi = YV\Sigma^{-1}W$ zapravo svojstveni vektori matrice A. Tako dobivene modove nazivamo ekgzaktnim. Modove dobivene Schidtovim algoritmom nazivamo projicirani modovi, budući da su projicirani na potprostor kojeg razapinju stupci matrice X.

Algorithm 2: Egzaktni DMD

1: Definiraj matrice X i Y kao u (2.24)
2: Izračunaj SVD dekompoziciju od X:
3: $X = U\Sigma V^*$,
4: gdje je $U \in \mathbb{C}^{n\times r}$, $\Sigma \in \mathbb{C}^{r\times r}$, $V \in \mathbb{C}^{n\times r}$ a r rang od X
5: Odredi rang k
6: Definiraj matricu $\tilde{A} = U_k^* Y V_k \Sigma_k^{-1}$
7: Izračunaj svojstvene vrijednosti i vektore od $\tilde{A}_k$: $\tilde{A}_k W_k = \Lambda_k W_k$
8: DMD modove dobivamo iz:
9: $\Phi = Y V_k \Sigma_k^{\dagger} W_k$

2.2 Rekonstrukcija

U trenutku kada su nam poznate svojstvene vrijednosti i svojstveni vektori operatora A, tj. njegove aproksimacije, spremni smo rekonstruirati početni skup podataka. Čak i više od toga, možemo izračunati stanje u nekom budućem vremenu koje je

izvan našeg intervala mjerenja $[0, T]$, no jasno je da s rastom vremenske varijable raste i pogreška. Rekonstrukcija je linearna kombinacija DMD modova, svojstvenih vrijednosti od $\tilde{C}$ i početne amplitude. Za svako vrijeme t, $x(t)$ možemo izračunati iz

$$x(t) \approx \sum_{k=1}^{r} \Phi_k \exp(\omega_k t) b_k, \tag{2.30}$$

gdje je b_k početna amplituda k-tog moda a ω_k pripadna frekvencija. Frekvencije su povezane sa svojstvenenim vrijednostima λ_i putem logaritamskog preslikavanja:

$$\omega_k = \frac{\log(\lambda_k)}{\Delta t} \tag{2.31}$$

gdje je Δt vremenski interval između dva uzastopna uzorka. Preostalo je samo odrediti početne frekvencije b_k. Neka je $\mathbf{b} = [b_1, ..., b_r]^T$. Za $t = 0$ vrijedi $x_1 = x(0) = \mathbf{\Phi b}$. Budući da $\mathbf{\Phi}$ općenito nije kvadratna matrica, u $\mathbf{b} = \mathbf{\Phi}^{-1} x_1$ koristimo pseudoinverz. Vrijednosti amplituda $\mathbf{b}$ određuju koji modovi su dominantni u sustavu. DMD modovi predstavljaju prostorne uzorke čije ponašanje u vremenu je određeno svojstvenim vrijednostima.

Primjer

Prikažimo značenje ovih vrijednosti na jednostavnom primjeru koji se sastoji od dva vremensko-prostorna signala[5]. Signali su opisani funkcijama

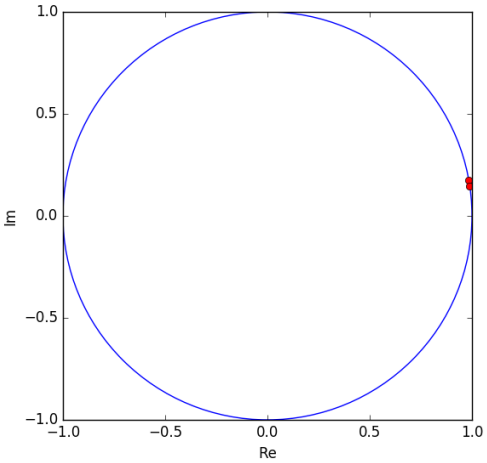
$$f_1(x,t) = \frac{1}{cosh(x+3)} e^{\mathbf{i} 2.3t} \tag{2.32}$$

$$f_2(x,t) = \frac{2tanh(x)}{cosh(x)} e^{\mathbf{i} 2.8t} \tag{2.33}$$

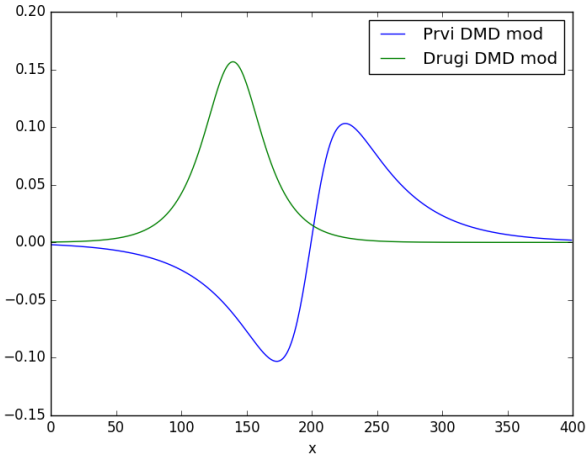
Promatramo miješani signal $f(x,t) = f_1(x,t) + f_2(x,t)$. Pokušajmo pomoću DMD dekompozicije odrediti jednostavne signale koji ga čine. Iz SVD-a matrice X_1^m vidimo da postoje samo dvije ne-nul singularne vrijednosti pa preostale vrijednosti odbacujemo i provodimo račun za $r = 2$. Na slici 2.1a možemo vidjeti da svojstvene vrijednosti matrice lineariziranog sustava $\tilde{A}$ leže na jediničnoj kružnici u kompleksnoj ravnini.

Položaj tih vrijednosti u odnosu na jediničnu kružnicu daje informaciju o rastu odnosno opadanju pripadnog moda gibanja. Ako svojstvena vrijednost ima imaginarni dio različit od nule, tada postoje oscilacije u DMD modu. Ukoliko se svojstvena

[5] primjer je preuzet iz [13]



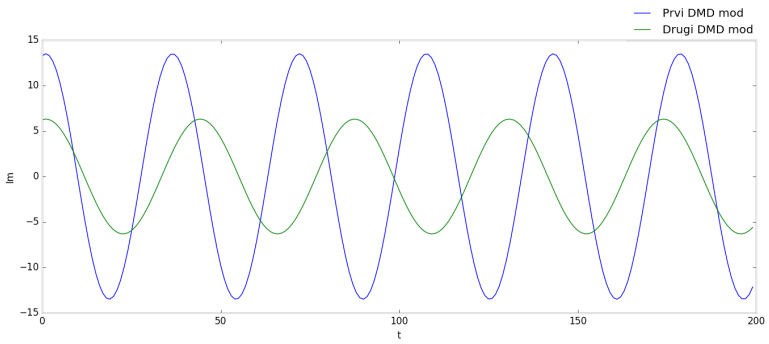
(a) Svojstvene vrijednosti



(b) Modovi gibanja

vrijednost nalazi unutar jedinične kružnice tada mod opada, a u slučaju da je izvan kružnice mod raste. Za svojstvenu vrijednost koja se nalazi na kružnici znamo da mod stagnira[6]. Modove danog sustava možemo vidjeti na slici 2.1b.

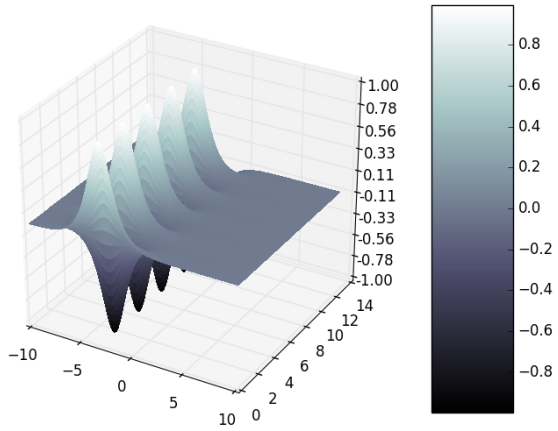
Kako bi pokazali da oni zaista stagniraju dovoljno je promotriti njihovo ponašanje kroz vrijeme dano na slici 2.2.



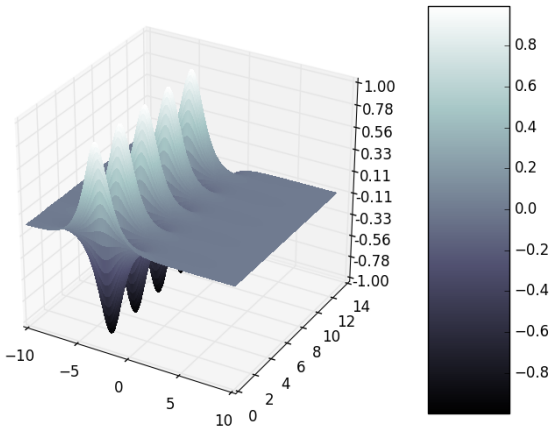
Slika 2.2: Ponašanje modova gibanja kroz vrijeme

Jasno je vidljivo da oba moda osciliraju no njihove amplitude ostaju iste. Iz početnih amplituda zaključujemo da je prvi DMD mod dominantan. Usporedbu signala f_1 i f_2 s DMD modovima, te njihovu kombinaciju f s DMD rekonstrukcijom možemo vidjeti na slici 2.3. Kod se može naći u dodatku B.

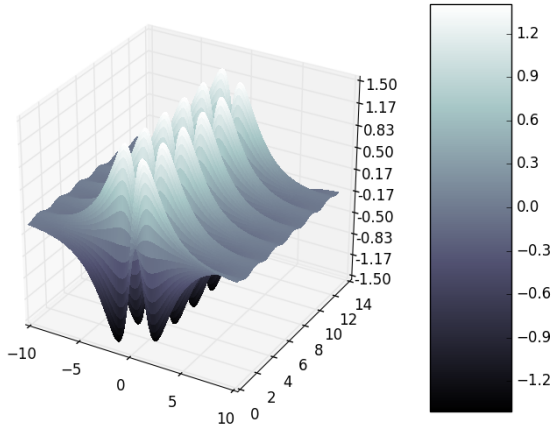
[6]tvrdnje vrijede u slučaju kada su X_1^m i X_2^{m+1} povezane diferencijalnim jednadžbama



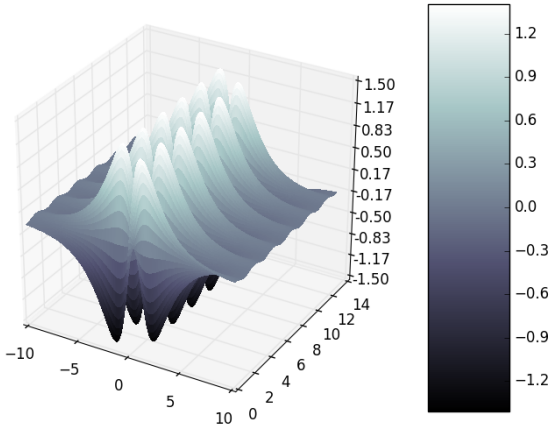
(a) Prvi signal, egzaktni prikaz



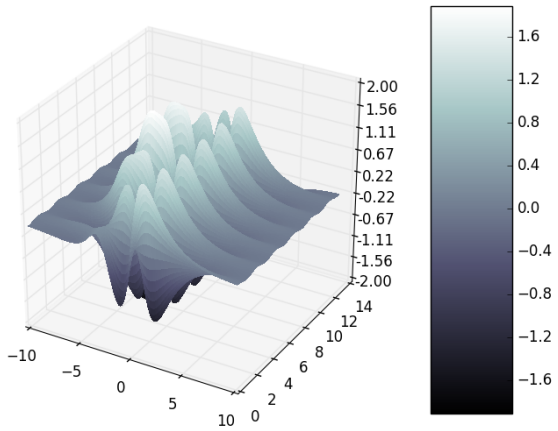
(b) Prvi signal kao prvi vektor DMD baze



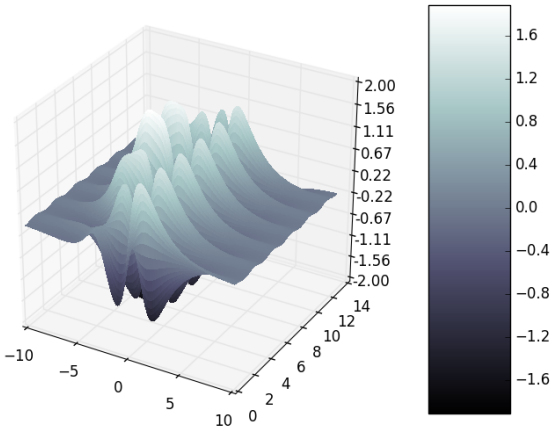
(c) Drugi signal, egzaktni prikaz



(d) Drugi signal kao drugi vektor DMD baze



(e) Kombinacija signala



(f) DMD rekonstrukcija kombinacije

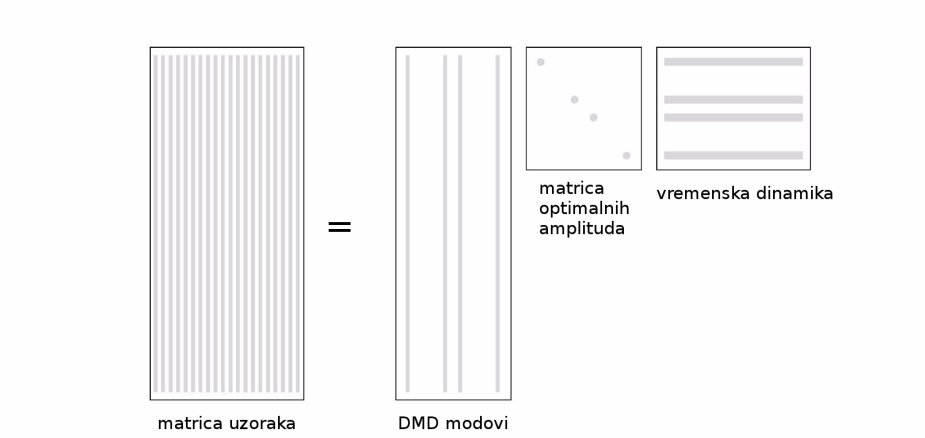
Slika 2.3: Primjer DMD rekonstrukcije

Model nižeg reda

Iz prethodnog primjera vidljivo je da je DMD dekompozicija dobar alat kada nam je potrebna rekonstrukcija sustava. No dinamički sustavi obično zahtijevaju mjerenja u malim vremenskim razmacima te velikom prostornom dimenzijom što računanje dekompozicije i rekonstrukcije čini računski zahtijevnim. DMD modovi općenito nisu ortogonalni pa je time pronalaženje modela nižeg reda teže. Kod algoritama koji se baziraju na korištenju SVD dekompozicije matrice X, već u tom koraku radimo redukciju, odbacujući svojstvene vektore čija je svojstvena vrijednost jednaka nuli. U daljnoj potrazi za načinom reduciranja sustava možemo iskoristiti činjenicu da DMD dekompozicija omogućuje prepoznavanje jednostavnih struktura koje čine kompleksni sustav. To nas dovodi do zaključka da u dinamici sustava postoje praznine, što znači da vektorski prikaz pojedinog moda gibanja u prikladnoj bazi sadržava velik broj nula. Tehnika koja, u nastojanju da rekonstruira signal sa što manje uzoraka, iskorištava upravo ovo svojstvo, naziva se "compressed sensing" i dolazi iz teorije signala. U teoriju ove metode ovdje nećemo ulaziti. Istaknuti ćemo samo rezultat da je za rekonstrukciju rijetkih signala dovoljan manji broj mjerenja. Time smo smanjili broj ulaznih podataka pa samim time i red sustava. Što ako želimo smanjiti red rekonstruiranog sustava, ali nakon što smo već izračunali sve DMD modove? Tada je potrebno odrediti podskup skupa DMD modova, koji će biti manjeg ranga ali koji će i dalje vjerno reprezentirati sustav. Jedna takva metoda dana je u [12]. Rješavanjem minimizacijskog problema dobivamo vektor optimalnih amplituda, koji se sastoji od određenog broja nula. Broj nula neposredno je određen parametrom γ, koji izražava odnos između točnosti aproksimacije i dimenzije reduciranog skupa modova. Shematski prikaz rekonstrukcije sa dobivenom matricom amplituda dan je na slici 2.4.

2.3 Koopmanov operator i veza s DMD

Kako je moguće nelinerni dinamički sustav opisati linearnom kombinacijom modova, čije su frekvencije sadržane u svojstvenim vrijednostima linearnog operatora? Pokazano je da je DMD usko povezan sa spektralnom dekompozicjiom Koopmanovog operatora. Dekompozicija na Koopmanove modove posljedica je nastojanja da se, općenito nelinearne, dinamičke sustave (u neprekidnoj vremenskoj varijabli) prikaže kao sumu linearnih beskonačnodimenzionalnih veličina. Dvije najpoznatije takve metode su razvoj u Taylorov i Fourierov red. No, za ove dekompozicije skup funkcija (u prostornoj varijabli) na koje projiciramo sustav da bi dobili vremenske koeficijente, unaprijed je određen. Druga mogućnost je POD, no njen nedostatak je što



Slika 2.4: *Model nižeg reda pomoću matrice optimalnih amplituda*

je u mogućnosti prepoznati velike promjene u energiji sustava, ali često ne i male promjene koje su ih potaknule. Mezić je u [2005] problem promatrao unutar teorije operatora. Ideja je bila projicirati sustav na svojstvene funkcije linearnog operatora, točnije Koopmanovog operatora, koji djeluje na dani nelinearni sustav a Rowley je u [2009] pokazao da DMD modovi sadrže podskup Koopmanovih modova. Koopmanov operator je linearan i beskonačnodimenzionalan operator koji prikazuje nelinearnu no konačnodimenzionalnu dinamiku sustava bez linearizacije . Zadržimo se na diskretnom vremenskom prostoru. Imamo sljedeću definiciju:

Definicija 2.3.1. *Neka je dan dinamički sustav (u diskretnj vremenskoj varijabli)*

$$\dot{\mathbf{y}} = \mathbf{F}(\mathbf{y}), \tag{2.34}$$

gdje je $\mathbf{y}$ *element* $n-$*dimenzionalne mnogostrukosti* $\mathcal{M}$*, a* $\mathbf{F}$ *općenito nelinearna vektorska funkcija. Koopmanov operator* $\mathcal{K}$ *djeluje na skupu skalarnih funkcija* $g : \mathcal{M} \to \mathbb{R}(\mathbb{C})$ *tako da*

$$\mathcal{K}g(\mathbf{y}) = g(\mathbf{F}(\mathbf{y})). \tag{2.35}$$

Očito je $\mathcal{K}$ linearan operator, čak i kad je $\mathbf{F}$ nelinearan. Štoviše, budući da djeluje na funkcijama $\mathcal{K}$ je beskonačnodimenzionalan i u slučaju kad je $\mathbf{F}$ konačne dimenzije. Ta činjenica ne predstavlja problem u slučaju da se djelovanje može ograničiti bez velikog gubitka točnosti. Promotrimo slučaj kada je promatrana veličina (g) vektorska funkcija. Tada je svaka njena komponenta (g_i) skalarna veličina. Možemo pretpostaviti da operator $\mathcal{K}$ ima svojstvene vrijednosti λ_j i pripadne svojstvene vektore ϕ_j za

koje vrijedi

$$\mathcal{K}\phi_j(\mathbf{y}) = \lambda_j\phi_j(\mathbf{y}) \quad \text{za } j = 0, 1, 2, ... \tag{2.36}$$

Uz pretpostavku da je g_i unutar prostora razapetog svojstvenim vektorima ϕ_j, g možemo pisati kao

$$g(\mathbf{y}) = \sum_{j=0}^{\infty} \phi_j(\mathbf{y})\psi_j, \tag{2.37}$$

gdje je $\psi_j \in \mathbb{R}(\mathbb{C})$. Vektore ϕ_j nazivamo Koopmanovi modovi. Promatramo dinamički sustav, u diskretnom vremenu,

$$\mathbf{y_{k+1}} = \mathbf{F}(\mathbf{y}), \tag{2.38}$$

te neka je g vektorska. Primjenom Koopmanovog operatora dobivamo:

$$\mathcal{K}g(\mathbf{y}) = g(\mathbf{F}(\mathbf{y}_k)) = g(\mathbf{y}_{k+1}). \tag{2.39}$$

Iskoristimo razvoj (2.36) i zapišimo $g(\mathbf{y}_k)$ u obliku (2.37):

$$g(\mathbf{y}) = \sum_{j=0}^{\infty} \lambda_j^k\phi_j(\mathbf{y}_0)\psi_j. \tag{2.40}$$

Ukoliko je veličina g koju mjerimo upravo stanje, tj. $g(\mathbf{y}) = \mathbf{y}$, te dinamički sustav kao u (2.38) jasna je veza između Koopmanovog operatora i DMD-a. Matrice X_1^m i X_2^{m+1} s početka poglavlja povezane su operatorom iz (2.38). Sada je pretpostavka o postojanju lineranog operatora A u (2.3) opravdana.

Poglavlje 3

Primjeri

3.1 Tok fluida oko cilindra

Pojava koju ćemo proučavati i na kojoj ćemo primijeniti DMD dekompoziciju je tok nestlačivog fluida kroz pravokutnu domenu s preprekom. Kao posljedica postojanja prepreke, ovisno o viskoznosti fluida, stvaraju se vrtlozi tj. tok postaje nestabilan. Gibanje nestlačivog fluida općenito je opisano Navier-Stokesovim jednadžbama:

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \Delta \mathbf{u}) = \nabla \cdot \sigma(\mathbf{u}, p) + f \tag{3.1a}$$

$$div\mathbf{u} = 0 \tag{3.1b}$$

gdje smo sa $\mathbf{u}$ označili brzinu, a s $\mathbf{p}$ pritisak. f je vanjska volumna sila a $\sigma(\mathbf{u}, p)$ tenzor naprezanja koji u danom slučaju iznosi

$$\sigma(\mathbf{u}, p) = 2\mu\epsilon(\mathbf{u}) - pI, \tag{3.2}$$

za mjeru deformacije

$$\epsilon(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T). \tag{3.3}$$

Gornje jednadžbe za promatrani problem možemo pojednostaviti na sljedeći oblik:

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{u} \cdot \Delta\mathbf{u} + \nu\nabla\mathbf{u} + \nabla p \tag{3.4a}$$

$$div\mathbf{u} = 0 \tag{3.4b}$$

Kinematička viskoznost ν omjer je dinamičke viskoznosti i gustoće fluida i usko je povezan s pojmom Reynoldsovog broja. Ona izražava omjer viskoznih naspram inercijalnih sila. Viskozne sile su posljedice trenja između slojeva fluida, dok inercijalne

sile nastaju zbog momenta fluida i za posljedicu imaju opiranje promjeni. Što je kinematička viskoznost manja tok je turbulentniji. Za potrebe DMD dekompozicije simulaciju radimo sa fluidom kinematičke viskoznosti $\nu = 0.001m^2/s$. Sustav rješavamo metodom konačnih elemenata, u Python sučelju za FEniCS. Potrebno je sustav napisati u varijacijskom obliku:

nađi $\mathbf{u} \in H_0^1(\Omega, \mathbb{R}^3)$, $p \in L^2(\Omega, \mathbb{R})$ *takve da vrijedi*

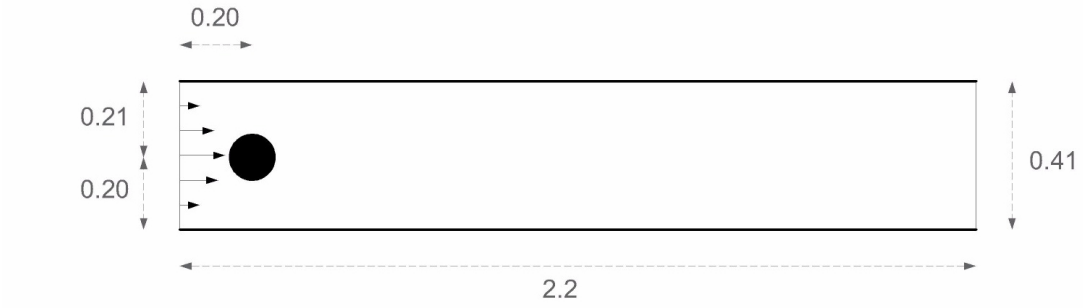
$$\int_\Omega \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} + \int_\Omega (\mathbf{u} \cdot \nabla)\mathbf{u} \cdot \mathbf{v} + \nu \int_\Omega \nabla \mathbf{u} \cdot \nabla \mathbf{v} - \int_\Omega p div(\mathbf{v}) = 0 \tag{3.5}$$

$$\int_\Omega div(\mathbf{u}) q = 0, \tag{3.6}$$

za svaki $\mathbf{v} \in H_0^1(\Omega, \mathbb{R}^3)$ i $p \in L^2(\Omega, \mathbb{R})$.
Podsjetimo se,

$$H_0^1(\Omega, \mathbb{R}^3) = \{v \in L^2(\Omega) : \partial_i v \in L^2 \text{ za } 1 \leq i \leq d \text{ i } v_{|\Gamma} = 0\}, \tag{3.7}$$

gdje je $\Gamma = \partial\Omega$. U nastavku ćemo se kratko upoznati s načinom rješavanja sustava i implementacijom.



Slika 3.1: Domena

Domenu možemo zamisliti kao 2D presjek cijevi kroz koju prolazi kružni cilindar, prepreka, koji je radijusa $r = 0.05$ i središta u $(0.2, 0.2)$. Dimenzije možemo vidjeti na slici 3.1. Kreiranje domene i triangulacije dano je sljedećim naredbama:

```
channel = Rectangle(Point(0, 0), Point(2.2, 0.41))
cylinder = Circle(Point(0.2, 0.2), 0.05)
domain = channel - cylinder
mesh = generate_mesh(domain, 64)
```

gdje broj 64 u posljednjoj liniji odznačava broj točaka po dužini domene. Sljedeći korak je definirati vektorski prostor konačnih elemenata. Koristimo Taylor-Hoodove elemente, tj. polinome stupnja 2 za brzinu i stupnja 1 za pritisak.

```
V = VectorFunctionSpace(mesh, 'P', 2)
Q = FunctionSpace(mesh, 'P', 1)
```

Na ulaznoj granici (vidi sliku 3.1) dan je parabolički profil brzine, koji je jednak

$$\mathbf{u}(x, y, t) = 1.5 \cdot \frac{4y(0.41 - y)}{0.41^2} \tag{3.8}$$

koji maksimalnu vrijednost 1.5 dostiže za $y = \frac{0.41}{2}$. Na desnom rubu cilindra pritisak je jednak nuli a na preostalim granicama imamo Dirichletov rubni uvjet $\mathbf{u} = 0$ za brzinu. Za svaku pojedinu granicu potrebno je napisati izraz koji određuje koje točke pripadaju toj granici a potom pridružiti vrijednost rubnog uvjeta.

```
inflow  = 'near(x[0], 0)'
outflow  = 'near(x[0], 2.2)'
walls  = 'near(x[1], 0) || near(x[1], 0.41)'
cylinder = 'on_boundary && x[0]>0.1 && x[0]<0.3 && x[1]>0.1 && x[1]<0.3'
inflow_profile = ('4.0*1.5*x[1]*(0.41 - x[1]) / pow(0.41, 2)', '0')
bcu_inflow = DirichletBC(V, Expression(inflow_profile, degree=2),inflow)
bcu_walls = DirichletBC(V, Constant((0, 0)), walls)
bcu_cylinder = DirichletBC(V, Constant((0, 0)), cylinder)
bcp_outflow = DirichletBC(Q, Constant(0), outflow)
bcu = [bcu_inflow, bcu_walls, bcu_cylinder]
bcp = [bcp_outflow]
```

Rješenje računamo verzijom Chorin metode(incremental pressure-correction) koja se sastoji od tri koraka koja računamo za svaki vremenski korak. Prvo računamo privremenu brzinu ($\tilde{u}$), diskretizirajući centralnim diferencijama u vremenskoj varijabli, iz jednadžbe (3.4a) tako da za pritisak i brzinu u konvektivnom članu ($u^n \cdot \nabla u^n$) iskoristimo vrijednost iz prethodnog vremenskog koraka. Vanjska sila jednaka je nuli pa u prvom koraku računamo[1]:

$$\int_\Omega \rho \frac{1}{\Delta t}(\tilde{u}^{n+1} - u^n) \cdot v dx + \int_\Omega \rho(u^n \cdot \nabla u^n) \cdot v dx + \int_{\partial\Omega} p^n \cdot n \cdot v d\mathcal{S} \tag{3.9}$$

$$+ \int_\Omega \sigma(\frac{u^n + \tilde{u}^{n+1}}{2}, p^n) \cdot \epsilon(v) dx - \int_{\partial\Omega} \mu\nabla(\frac{u^n + \tilde{u}^{n+1}}{2}) \cdot n \cdot v d\mathcal{S} = 0. \tag{3.10}$$

Posljednja dva člana dobivamo iz parcijalne integracije

$$-\int_\Omega \nabla \cdot \sigma \cdot v = \int_\Omega \sigma \cdot \epsilon(v) - \int_{\partial\Omega} T \cdot v, \tag{3.11}$$

[1]postoji i druga verzija Chorin metode, u kojoj u prvom koraku u potpunosti zanemarujemo pritisak

gdje je $T = \sigma \cdot n$ vektor naprezanja. Pretpostavljamo da je na izlaznoj granici derivacija brzine u tangencijalnom smjeru jednaka nuli pa na granici preostaje član $pn - \mu \nabla u$. Neka je `u` oznaka za probnu funckiju $\tilde{u}^{n+1}$, `u_` za u^{n+1} i `u_n` za u^n. Definiramo ih sa

```
# Funkcija za primvremeno rjesenje
u = TrialFunction(V)
# Funkcija za rjesenje u prethodnom koraku
u_n = Function(V)
# Funkcija za rjesenje u trenutnom koraku
u_ = Function(V)
# Test funkcija
v = TestFunction(V)
```

Analogno označavamo i definiramo varijable za računanje pritiska. Varijacijski problem za prvi korak dan je sljedećim kodom:

```
#Simetricni gradijent
def epsilon(u):
    return sym(nabla_grad(u))
#Tenzor naprezanja
def sigma(u, p):
    return 2*mu*epsilon(u) - p*Identity(len(u))

U  = 0.5*(u_n + u)
n  = FacetNormal(mesh)

F1 = rho*dot((u - u_n) / Dt, v)*dx \
+ rho*dot(dot(u_n, nabla_grad(u_n)), v)*dx \
+ inner(sigma(U, p_n), epsilon(v))*dx \
+ dot(p_n*n, v)*ds - dot(mu*nabla_grad(U)*n, v)*ds \
- dot(f, v)*dx
```

Bilinearnu formu $a(u, v)$ i linearnu formu $L(v)$ možemo jednostavno dobiti pomoću funkcija `lhs` i `rhs` implementiranih u FeniCS-u:

```
a1 = lhs(F1)
L1 = rhs(F1)
```

U sljedećem koraku koristimo privremenu brzinu za računanje pritiska iz jednadžbe:

$$\frac{u^{n+1} - \tilde{u}^{n+1}}{\Delta t} + \nabla p^{n+1} - \nabla p^n = 0. \tag{3.12}$$

Uzmemo li divergenciju cijelog izraza, zbog $div\ u = 0$, imamo:

$$\frac{div\ \tilde{u}^{n+1}}{\Delta t} + \Delta p^{n+1} - \Delta p^n = 0. \tag{3.13}$$

Vrijednost p^n je poznata pa se računanje p^{n+1} svodi na rješavanje Poissonove jednadžbe po p^{n+1}.

```
a2 = dot(nabla_grad(p), nabla_grad(q))*dx
L2 = dot(nabla_grad(p_n), nabla_grad(q))*dx - (1/Dt)*div(u_)*q*dx
```

Konačno, korektiranu brzinu u^{n+1} računamo iz

$$\int_\Omega u^{n+1} \cdot v dx = \int_\Omega \tilde{u}^{n+1} \cdot v dx - \Delta t \int_\Omega \nabla(p^{n+1} - p^n) \cdot v dx \tag{3.14}$$

```
a3 = dot(u, v)*dx
L3 = dot(u_, v)*dx - Dt*dot(nabla_grad(p_ - p_n), v)*dx
```

Bilinearna forma je za sva tri koraka neovisna o vremenu pa matrice sustava možemo postaviti izvan vremenske petlje. Također, za prva dva koraka potrebno je uvrstiti rubne uvjete.

```
# Asembliranje matrica
A1 = assemble(a1)
A2 = assemble(a2)
A3 = assemble(a3)
# Postavljanje rubnih uvjeta
[bc.apply(A1) for bc in bcu]
[bc.apply(A2) for bc in bcp]
```

Linearna forma ovisi o vremenskoj varijabli pa vektor desne strane, za sva tri koraka, moramo kreirati unutar vremenske petlje. Za prva dva koraka u desnu stranu potrebno je također uvrstiti rubne uvjete. Rješenje varijacijskog problema dobivamo jednostavnim pozivanjem funkcije `solve` kojoj prosljeđujemo matricu krutosti, vektor rješenja i vektor desne strane. Na kraju, spremamo dobiveno rješenje kako bi ga mogli iskoristiti u sljedećem koraku.

```
t = 0.0
for n in range(num_steps):
  # novo vrijeme
  t += dt
  # Korak 1: Racunanje privremene brzine
  b1 = assemble(L1)
  [bc.apply(b1) for bc in bcu]
  solve(A1, u_.vector(), b1, 'bicgstab','hypre_amg')
  # Korak 2: Korekcija pritiska
  b2 = assemble(L2)
  [bc.apply(b2) for bc in bcp]
  solve(A2, p_.vector(), b2, 'bicgstab','hypre_amg')
  # Korak 3: Korekcija brzine
  b3 = assemble(L3)
  solve(A3, u_.vector(), b3, 'cg','sor')
  # novo rjesenje
  u_n.assign(u_)
  p_n.assign(p_)
```

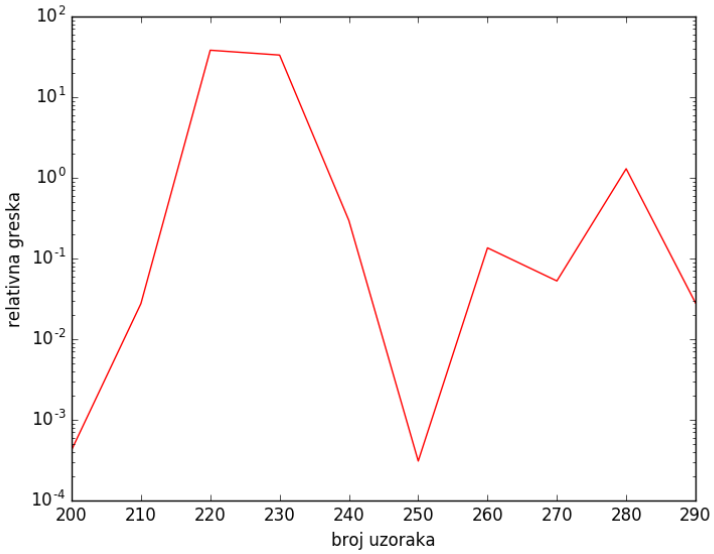
Spomenimo još ovdje dvije korisne mogućnosti, koje ćemo iskoristiti i u sljedećem primjeru i za stvaranje matrice uzoraka za POD i DMD dekompoziciju. Geometriju i triangulaciju domene možemo spremiti u datoteku jednostavnom naredbom

```
File('navier_stokes_cylinder_1dmd/cylinder.xml.gz') << mesh
```

Druga mogućnost je spremanje vektora rješenja u binarnu datoteku pomoću `TimeSeries` klase koja omogućuje učinkovito čitanje rješenja u određenim vremenskim trenutcima. Potrebno je kreirati datoteku

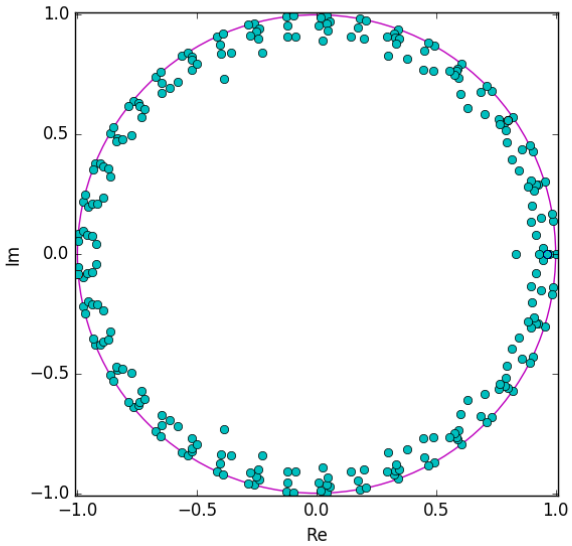
```
timeseries_u = TimeSeries('navier_stokes_cylinder/velocity_series')
```

u koju će vrijednosti biti spremljene te u svakom koraku metode zapisati vektor rješenja pomoću naredbe `timeseries_u.store(u.vector(), t)`. Brzinu toka fluida računamo za cijeli vremenski interval $[0, 7]$ no za potrebe DMD dekompozicije sa promatranjem sustava počinjemo u vremenu $t = 3.0$ kada se tok već razvio i formirali su se vrtlozi, i završavamo u vremenu $t = 7.0$. Algoritam kojim računamo je Egzaktni DMD algoritam. Kod DMD dekompozicije potreban je oprez pri odabiru broja uzoraka, budući da on određuje broj vektora baze. Želimo odabrati optimalan broj uzoraka, kako bi dobili realnu grešku pri predviđanju stanja sustava za vremenske trenutke nakon vremena $t = 7.0$. Na slici 3.2 pokazujemo odnos broja uzoraka, koji se kreće od 200 do 290, i relativne greške rekonstrukcije.



Slika 3.2: Realtivna greška rekonstrukcije s obzirom na broj uzoraka

U nastavku ćemo raditi sa 250 uzoraka unutar intervala $[3.0, 7.0]$. Time smo definirali $\Delta t = 0.016$. Na slici 3.3 dane su svojstvene vrijednosti matrice $\tilde{A}$.



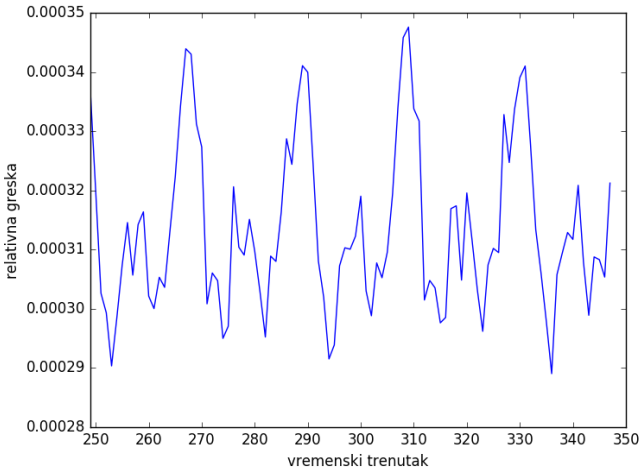
Slika 3.3: *Svojstvene vrijednosti od* $\tilde{A}$

Zanimljiva struktura koju tvore svojstvene vrijednosti uobičajena je za periodične sustave. Iz prikaza možemo zaključiti da modovi gibanja ili stagniraju ili njihove amplitude opadaju, što odgovara stvarnom stanju sustava. Vrijednosti koje se nalaze najbliže središtu su posljedica toga što se mjerenje sustava ne odvija u cijelom broju perioda. Vizualizaciju šest nasumično odabranih DMD modova možemo vidjeti na slikama 3.6 i 3.7. Iz prikaza 3.2 vidimo da je relativna greška pri rekonstrukciji sustava s 250 modova reda $\mathcal{O}(10^{-3})$. Usporedbu egzaktnog i rekonstruiranog rješenja u trenutku $t = 7.0$ možemo vidjeti na slici 3.8. Kod postupka nalazi se u dodatku D.

Sve vrijednosti potrebne za rekonstrukciju rješenja (formula za rekonstrukciju dana je u (2.30)) možemo izračunati neovisno o vremenu t. Iz toga slijedi da bi, sa Ritzovim vrijednostima, DMD modovima i početnim amplitudama izračunatim nad uzorcima iz nekog intervala $[a, b]$, mogli aproksimirati stanje sustava u trenutku $c = b + dt$, za neki razuman dt. Jasno je, najveći korak dt koji možemo uzeti, a da aproksimacija i dalje bude dobra, ovisi o samoj dinamici i turbulentnosti sustava. Ovdje ćemo pokušati izračunati stanje sustava u 50 trenutaka unutar intervala $[7.0, 8.6]$ na osnovu 250 uzoraka iz intervala $[3.0, 7.0]$ i usporediti sa egzaktnim rješenjem. U svakom vremenskom trenutku računamo relativnu grešku

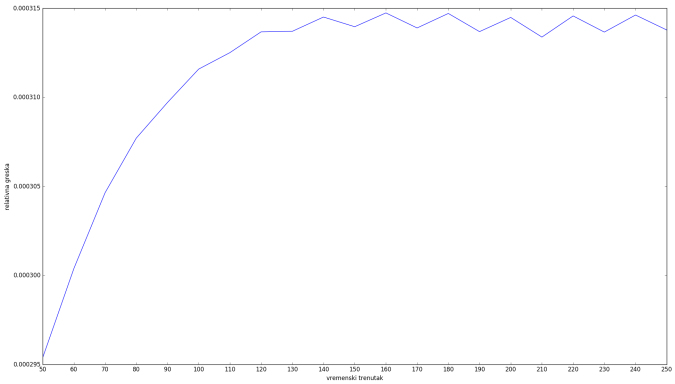
$$\frac{\|u(x) - u_{DMD}(x)\|_2}{\|u(x)\|_2}.$$

Iznos relativna greška u svakom vremenskom koraku dan je na slici 3.4. Iz prikaza

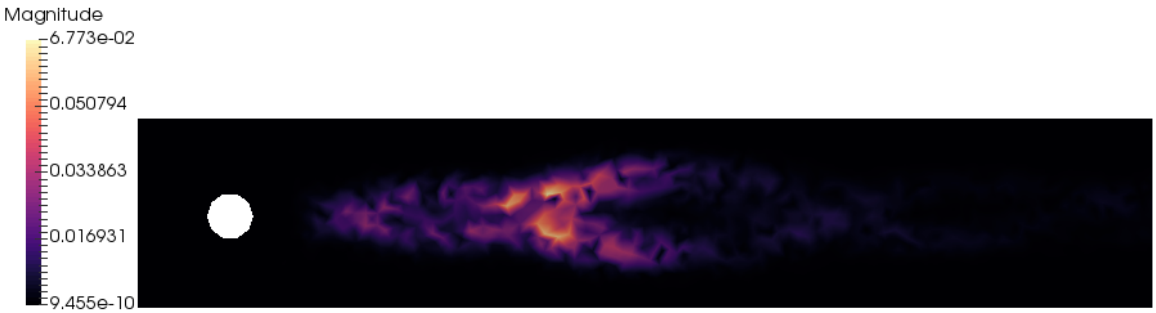


Slika 3.4: *Relativna greška predviđanja, koju računamo za* 100 *koraka unutar intervala* $[7.0, 8.6]$ *na osnovu* 250 *uzoraka iz intervala* $[3.0, 7.0]$

3.4 da se naslutiti da greška ne raste bitno s vremenom. Također je za očekivati da će greška biti manja ako se interval na kojem smo uzeli mjerenja i interval za koji želimo predvidjeti preklapaju. Ovdje ćemo i dalje računati vrijednosti za 100 koraka metode no trenutak u kojem počinjemo računati se mijenja. Na slici 3.5 dajemo relativnu grešku u odnosu na vremenski trenutak u kojem počinjemo računati predviđene vrijednosti.



Slika 3.5: *Relativna greška predviđanja u odnosu na trenutak u kojem smo počeli računati predviđene vrijednosti*



(a) 13. vektor DMD baze, realni dio

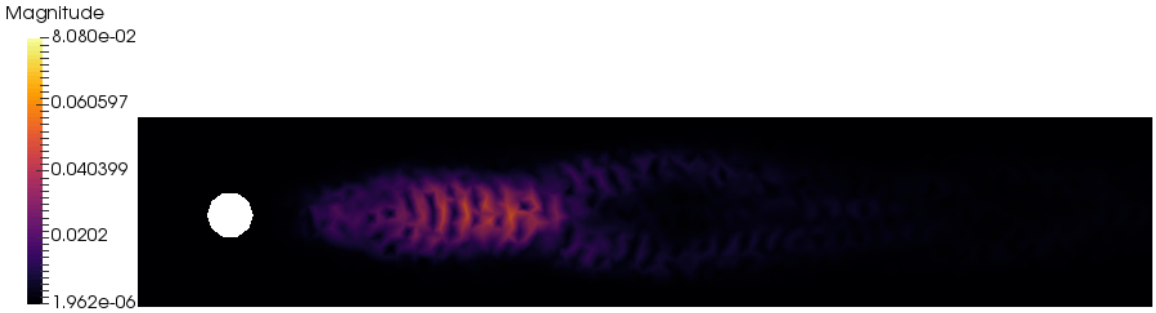
(b) 82. vektor DMD baze, realni dio

(c) 127. vektor DMD baze, realni dio

Slika 3.6: DMD modovi

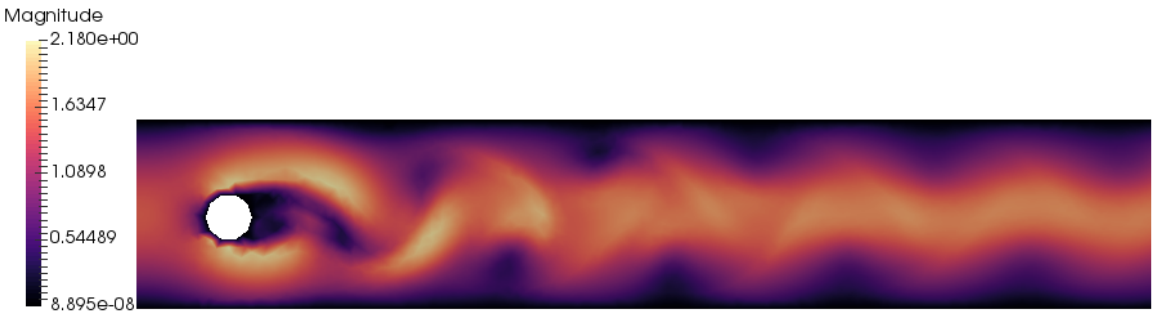
(a) 98. vektor DMD baze, realni dio

(b) 154. vektor DMD baze, realni dio

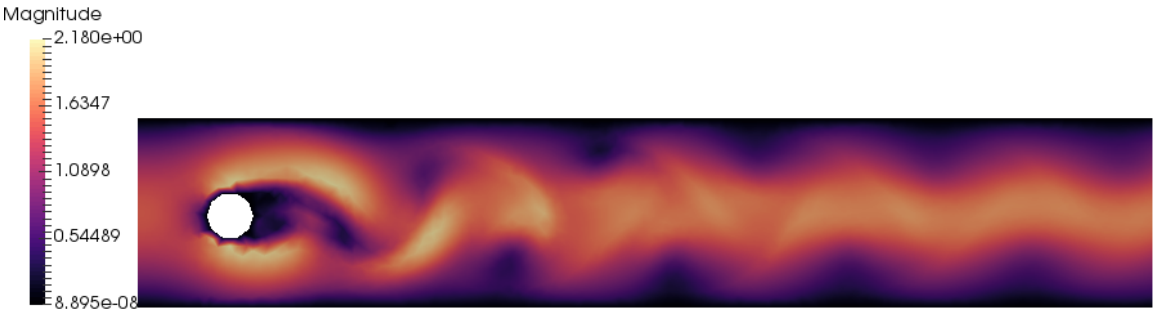


(c) 194. vektor DMD baze, realni dio

Slika 3.7: DMD modovi



(a) Egzaktno rješenje u trenutku $t = 6.984$



(b) DMD rekonstrukcija u trenutku $t = 6.984$

(c) Odstupanje u trenutku $t = 6.984$

Slika 3.8: Usporedba: egzaktno rješenje, DMD rekonstrukcija i odstupanje u trenutku $t = 7.0$

3.2 Sustav jednadžbi konvekcije-difuzije-reakcije

POD dekompoziciju primijenit ćemo na sustavu jednadžbi koje opisuju kemijsku reakciju dvije tvari u fluidu pri čemu se formira treća tvar. Veličina koja ih razlikuje je koncentracija. Pretpostavljamo da je brzina reakcije dviju tvari proporcionalna njihovoj koncentraciji. Također pretpostavimo da se formirana tvar raspada brzinom proporcionalnom njenoj koncentraciji. Označimo pripadne koncentracije s u_1, u_2 i u_3, pri čemu smo u u_3 označili koncentraciju nastale tvari. Neka je koeficijent difuzije jednak $\epsilon = 0.01$, brzina reakcije $K = 10$, te neka se fluid u kojem se kemijske komponente nalaze giba brzinom w. Tada je sustav opisan jednadžbama:

$$\begin{aligned}
\frac{\partial u_1}{\partial t} + w \cdot \nabla u_1 - \nabla \cdot (\epsilon \nabla u_1) &= f_1 - K u_1 u_2 \\
\frac{\partial u_2}{\partial t} + w \cdot \nabla u_2 - \nabla \cdot (\epsilon \nabla u_2) &= f_2 - K u_1 u_2 \\
\frac{\partial u_3}{\partial t} + w \cdot \nabla u_3 - \nabla \cdot (\epsilon \nabla u_3) &= f_3 + K u_1 u_2 - K u_3.
\end{aligned}$$

Ovu reakciju možemo smjestiti u domenu i vektorsko polje brzine fluida iz prethodnog primjera. Rješavamo sustav nelinearnih parcijalnih diferencijalnih jednadžbi:

$$\begin{aligned}
\rho(\frac{\partial w}{\partial t} + w \cdot \nabla w) &= \nabla \sigma(w, p) + f \\
\nabla \cdot w &= 0 \\
\frac{\partial u_1}{\partial t} + w \cdot \nabla u_1 - \nabla \cdot (\epsilon \nabla u_1) &= f_1 - K u_1 u_2 \\
\frac{\partial u_2}{\partial t} + w \cdot \nabla u_2 - \nabla \cdot (\epsilon \nabla u_2) &= f_2 - K u_1 u_2 \\
\frac{\partial u_3}{\partial t} + w \cdot \nabla u_3 - \nabla \cdot (\epsilon \nabla u_3) &= f_3 + K u_1 u_2 - K u_3.
\end{aligned}$$

Slabu formulacija sustava dobivamo množenjem s test funkcijom, parcijalnom integriracijom izraza $\nabla \cdot (\epsilon \nabla u_i)$ te konačno zbrajanjem jednadžbi. Za vremensku derivaciju koristimo implicitnu Eulerovu metodu: $\frac{\partial u_i}{\partial t}$ aproksimiramo s $\frac{u_i^{n+1} - u_i^n}{\Delta t}$. Sa v_1, v_2 i v_3 označimo test funkcije. Imamo:

$$\int_\Omega (\frac{1}{\Delta t}(u_1^{n+1} - u_1^n)v_1 + w \cdot \nabla u_1^{n+1} v_1 + \epsilon \nabla u_1^{n+1} \cdot \nabla v_1) dx$$
$$+ \int_\Omega (\frac{1}{\Delta t}(u_2^{n+1} - u_2^n)v_2 + w \cdot \nabla u_2^{n+1} v_2 + \epsilon \nabla u_2^{n+1} \cdot \nabla v_2) dx$$
$$+ \int_\Omega (\frac{1}{\Delta t}(u_3^{n+1} - u_3^n)v_3 + w \cdot \nabla u_3^{n+1} v_3 + \epsilon \nabla u_3^{n+1} \cdot \nabla v_3) dx$$
$$- \int_\Omega (f_1 v_1 + f_2 v_2 + f_3 v_3) dx$$
$$- \int_\Omega (-K u_1^{n+1} u_2^{n+1} v_1 - K u_1^{n+1} u_2^{n+1} v_2 + K u_1^{n+1} u_2^{n+1} v_3 - K u_3^{n+1} v_3) dx = 0.$$

```
# Varijacijski problem
F = ((u_1 - u_n1) / k)*v_1*dx + dot(w, grad(u_1))*v_1*dx \
+ eps*dot(grad(u_1), grad(v_1))*dx + K*u_1*u_2*v_1*dx  \
+ ((u_2 - u_n2) / k)*v_2*dx + dot(w, grad(u_2))*v_2*dx \
+ eps*dot(grad(u_2), grad(v_2))*dx + K*u_1*u_2*v_2*dx  \
+ ((u_3 - u_n3) / k)*v_3*dx + dot(w, grad(u_3))*v_3*dx \
+ eps*dot(grad(u_3), grad(v_3))*dx - K*u_1*u_2*v_3*dx + K*u_3*v_3*dx \
- f_1*v_1*dx - f_2*v_2*dx - f_3*v_3*dx
```

Iz Navier-Stokesovih jednadžbi potreban nam je iznos brzine u svakoj točki domene. Budući smo to već računali u prethodnom primjeru, dobivene vrijednosti ćemo iskoristiti. Kako bi mogli učitati vrijednosti brzine iz prethodnog primjera potrebno je definirati vektorski prostor brzine, na isti način kao i ranije na istoj triangulaciji. Vrijednosti učitavamo pomoću naredbe `TimeSeries` koja kreira objekt `timeseries_w` koji je potreban za dohvaćanje vektora iz HDF5 binarne datoteke unutar vremenske petlje.

```
# Ucitavanje triangulacije iz datoteke
mesh = Mesh('navier_stokes_cylinder_1/mesh.xml.gz')
W = VectorFunctionSpace(mesh, 'P', 2)
# Ucitavanje datoteke u kojoj su spremljeni vektori brzine
timeseries_w = TimeSeries('navier_stokes_cylinder_1/velocity_series')
```

Za koncentracije u_1, u_2 i u_3 želimo definirati prostor koji će sadržavati sve tri komponente. Iz tog razloga definiramo element kao produkt tri jednostavna elementa i koristimo ga za dobivanje željenog prostora.

```
# Definiranje funkcijskog prostora za koncentracije
P1 = FiniteElement('P', triangle, 1)
element = MixedElement([P1, P1, P1])
V = FunctionSpace(mesh, element)
```

Svakoj pojedinoj komponenti vektora rješenja možemo pristupiti sa:

```
u = Function(V)
u_1, u_2, u_3 = split(u)
```

Neka je $u_1 = u_2 = u_3$ u trenutku $t = 0$. Dodavanje kemijskih komponenti u sustav postižemo postavljanjem veličina f_1 i f_2 na nenul vrijednosti. Njih ćemo definirati blizu ulazne granice. Za f_3 uzimamo da je jednak nuli, budući da tvar koncentracije u_3 nastaje reakcijom prve dvije i nemamo njen direktan izvor.

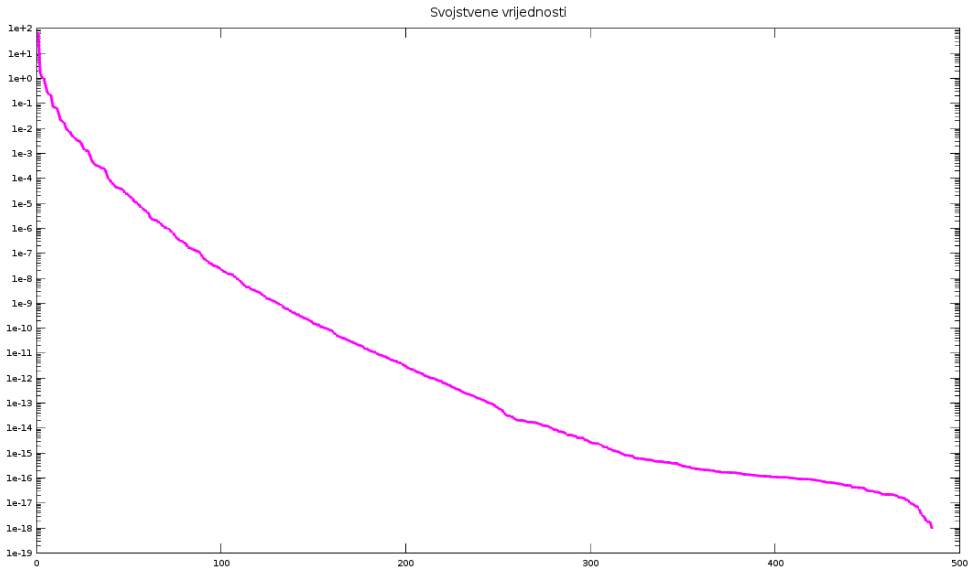
Na cijeloj granici, za u_1, u_2 i u_3, uzimamo homogeni Neummanov rubni uvjet $\frac{\partial u_i}{\partial n} = 0$. Sustav rješavamo tako da unutar vremenske petlje prvo dohvaćamo brzinu a zatim pozivamo funkciju `solve` koja u pozadini generira matricu krutosti i vektor desne strane, primijenjuje rubne uvjete te konačno rješava linearni sustav $Fu = b$. Ovdje nije potrebno definirati rubne uvjete posebno budući da je zadani uvjet paketa FeniCS upravo homogeni Neumannov uvjet.

```
t = 0.0
for n in range(num_steps):
  # novo vrijeme
  t += dt
  # Ucitaj brzinu iz datoteke
  timeseries_w.retrieve(w.vector(), t)
  # Rjesi varijacijski problem
  solve(F == 0, u)
  u_n.assign(u)
```

Napomenimo ovdje da tenutak t u kojem dohvaćamo brzinu putem naredbe `timeseries.retrieve` ne mora nužno odgovarati trenutku u kojem je brzina spremljena, budući da će se vrijednosti linearno interpolirati prema zadanom vremenu t. Promatrani vremenski interval je $[0, 5]$, korak $\Delta t = 0.01$, što generira 500 uzoraka. Broj prostornih točaka jednak je kao u prethodnom primjeru budući da moramo uzeti istu diskretizaciju da bi dohvatili vektor brzine. Viskoznost fluida jednaka je $\nu = 0.001 m^2/s$.

Na slici 3.9 prikazane su svojstvene vrijednosti matrice korelacije. Iz prikaza je jasno da jako brzo opadaju. Računski se dobije da već za $l = 7$ ukupna energija sustava zadovoljava $\mathcal{E}(l) \geq 0.99$. Prvih šest vektora POD baze možemo vidjeti na slikama 3.10 i 3.11.

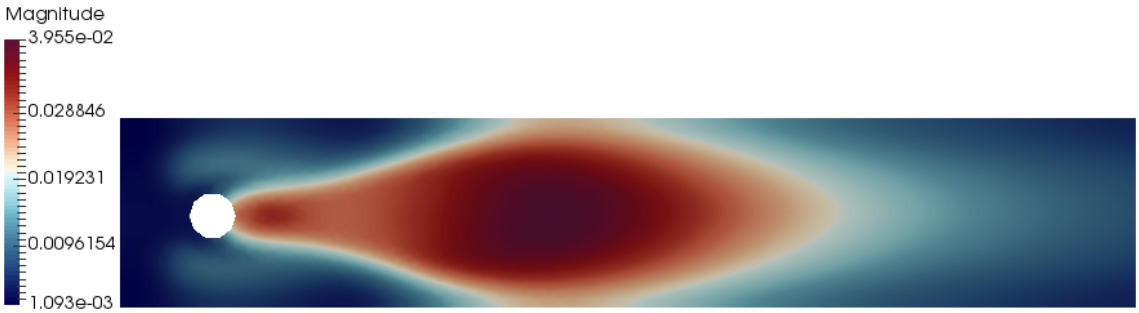
Za POD bazu od 50 vektora relativna greška je reda $\mathcal{O}(10^{-5})$ što odgovara ocjeni (1.11) budući da je 51. svojstvena vrijednost danog sustava jednaka $1.76e-05$. 50 uzoraka, uz jednaku prostornu diskretizaciju, sačinjava 0.1% početnog sustava, a daje aproksimaciju rješenja točnu do na treću decimalu. Na slikama 3.12 i 3.13 možemo vidjeti usporedbu egzaktnog i POD rješenja, te pripadna odstupanja u vremenima $t = 2.0$ i $t = 5.0$. Kod postupka može se naći u dodatku C.



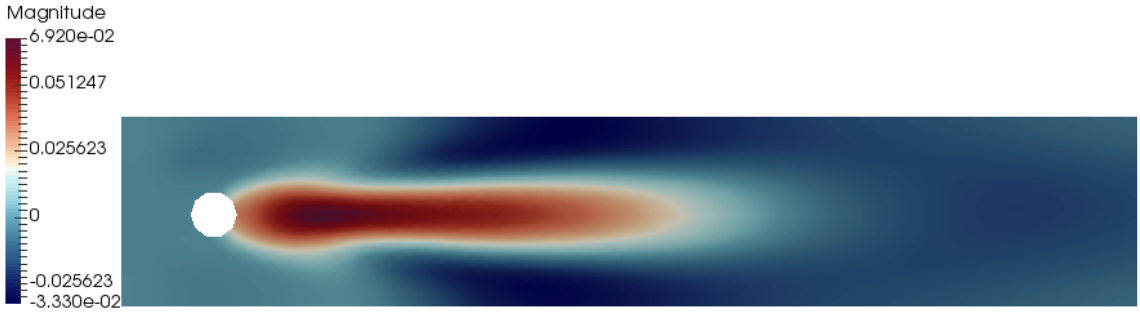
Slika 3.9: Svojstvene vrijednosti matrice korelacije

Kada smo govorili o Galerkinovoj projekciji kod POD metode, spomenili smo da se dio računanja može izvršiti u tzv. "offline" fazi. Ovdje ćemo demonstrirati jedan takav primjer. Izvršit ćemo 5 simulacija danog sustava, koje će se razlikovati u vrijednosti parametra ν, tj. viskoznosti fluida. Vrijednosti od ν uzimamo iz skupa $\{0.0009, 0.00095, 0.001, 0.00105, 0.0011\}$. Za svaku od ovih vrijednosti izračunat ćemo POD bazu pripadnog sustava sa 25 vektora. Sada vektore dobivenih baza možemo shvatiti kao uzorke, pa radimo POD dekompoziciju na ovih 125 uzoraka. Dobivamo novu POD bazu, za koju tvrdimo da je dovoljno dobra za sve vrijednosti parametra $\nu \in [0.0009, 0.0011]$. Testiramo na $\nu = 0.00103 m^2/s$. Relativnu grešku po vremenu možemo vidjeti na slici 3.14:

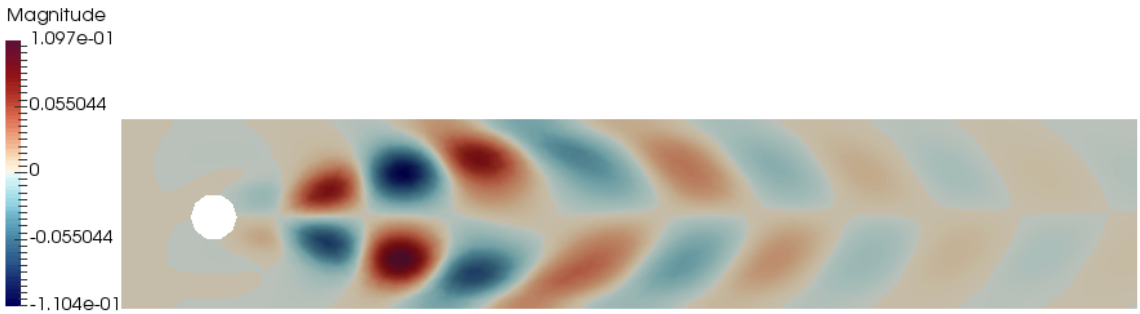
Vidimo da je u početnim trenutcima greška jako velika. Razlog tome je što se tada događa minimalna reakcija pa je iznos mjerene veličine (koncentracija tvari) tada blizu nule. Slijedi da je norma velčine jako mala što dovodi do velike relativne pogreške. Kada se reakcija već dogodila imamo stabilnu grešku reda $\mathcal{O}(3)$.



(a) Prvi vektor POD baze

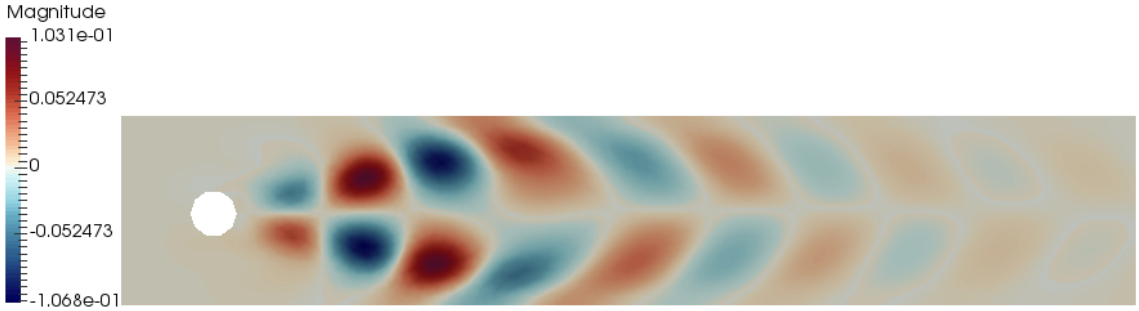


(b) Drugi vektor POD baze

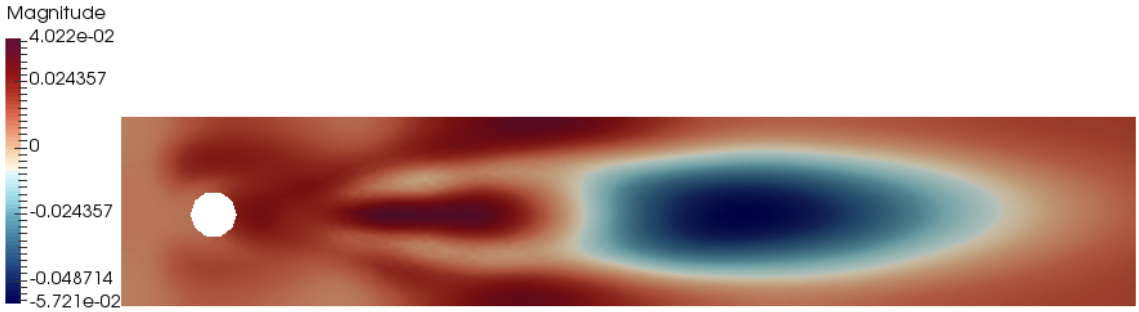


(c) Treći vektor POD baze

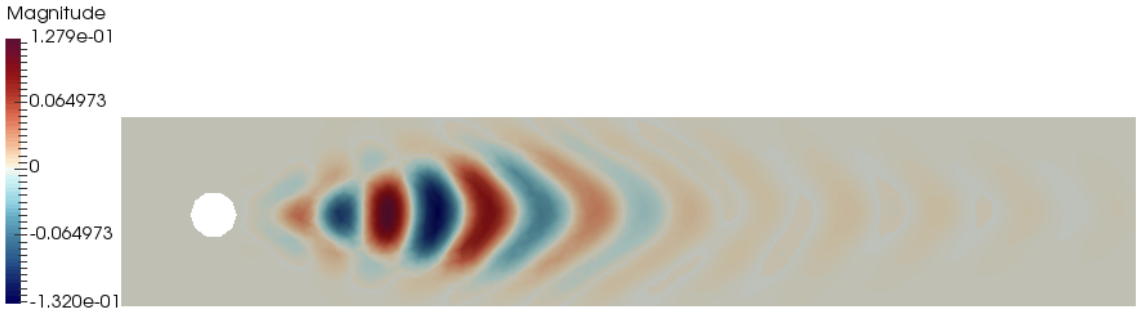
Slika 3.10: Vektori POD baze



(a) Četvrti vektor POD baze

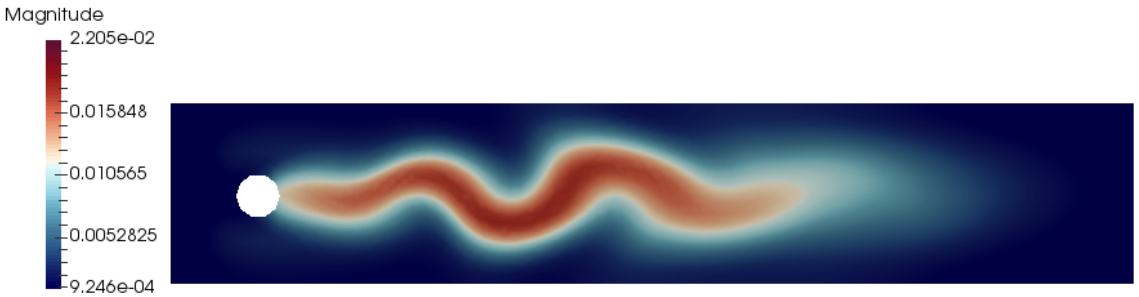


(b) Peti vektor POD baze

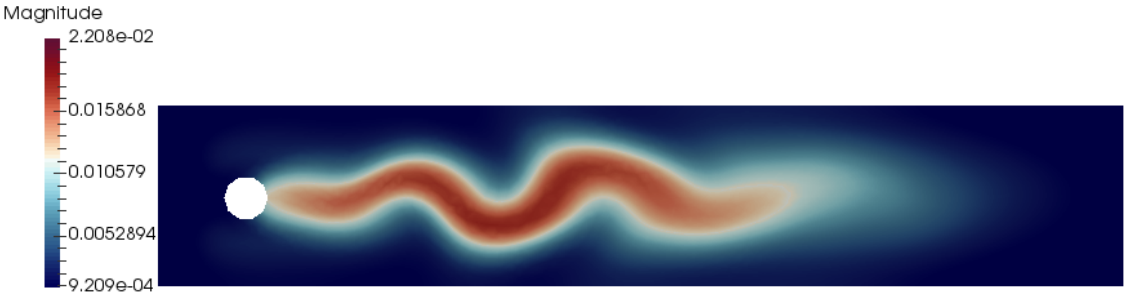


(c) Šesti vektor POD baze

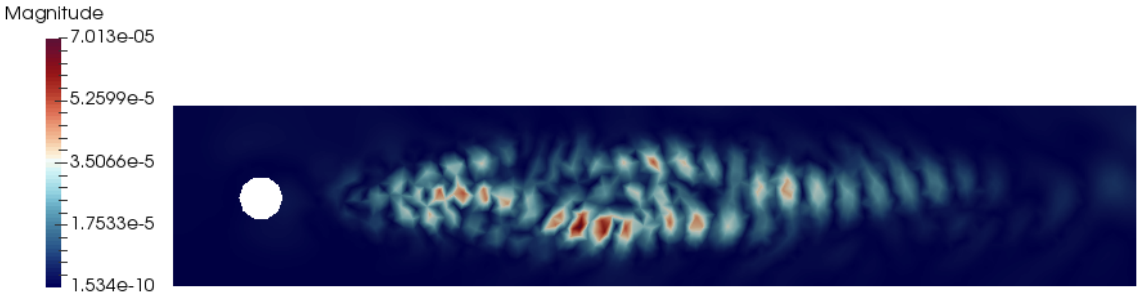
Slika 3.11: Vektori POD baze



(a) Egzaktno rješenje u $t = 2.0$

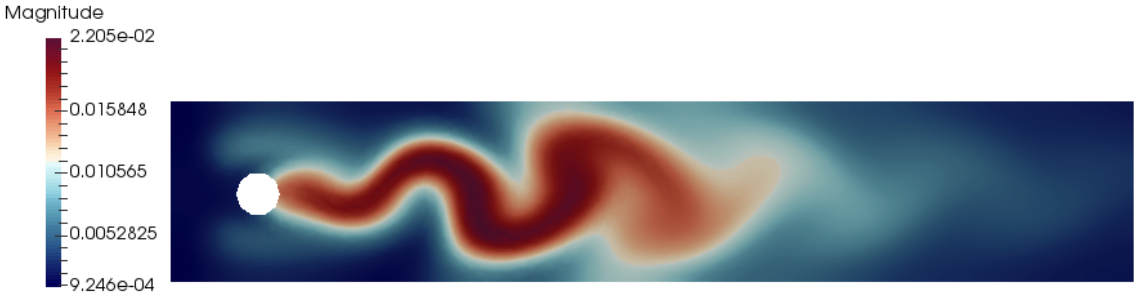


(b) POD rješenje u $t = 2.0$

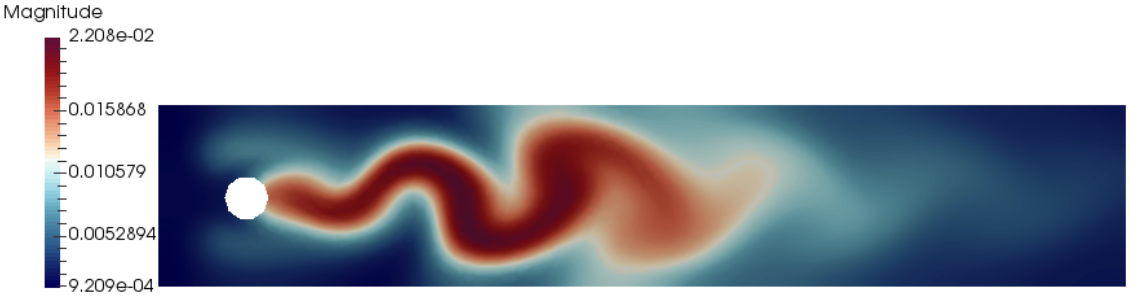


(c) Greška u $t = 2.0$

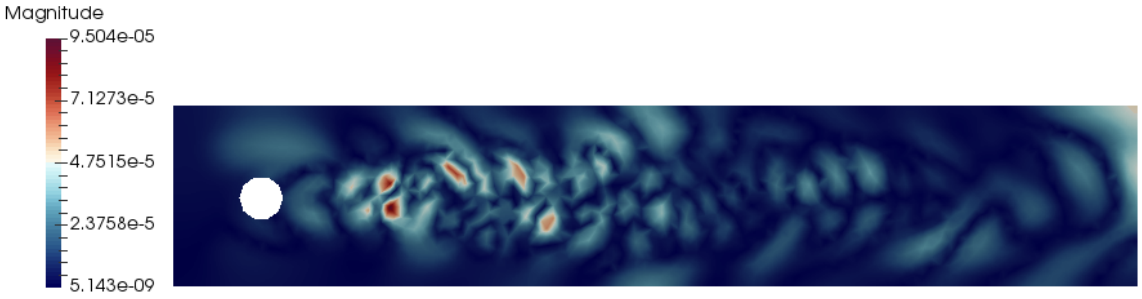
Slika 3.12: Usporedba: egzaktno rješenje, POD rješenje i odstupanje



(a) Egzaktno rješenje u $t = 5.0$

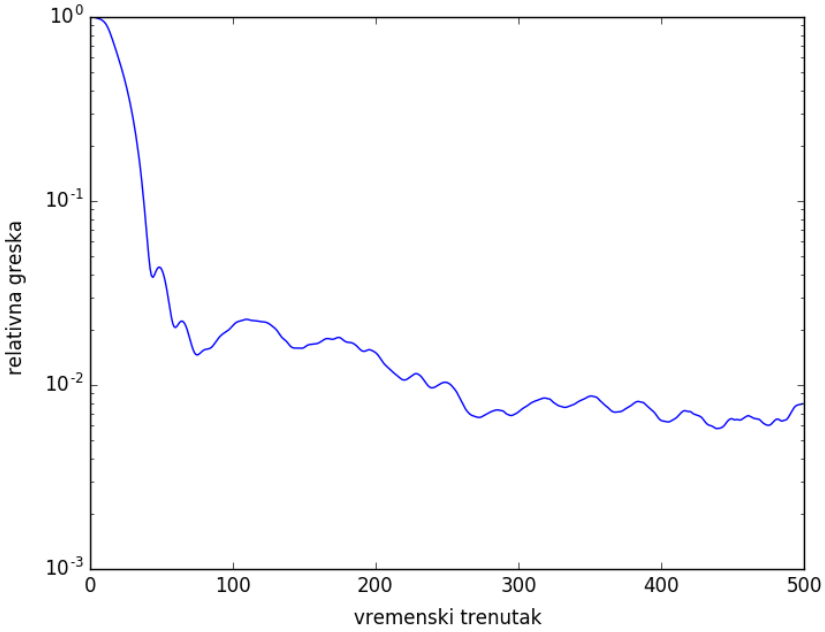


(b) POD rješenje u $t = 5.0$



(c) Greška u $t = 5.0$

Slika 3.13: Usporedba: egzaktno rješenje, POD rješenje i odstupanje



Slika 3.14: Relativna greška

Dodatak A

Softver

FEniCS je platforma otvorenog koda namijenjena rješavanju parcijalnih diferencijalnih jednadžbi. Sastoji se od više koomponenti, pisanih većinom u C++ i Python jeziku, koje imaju različite uloge. Nudi više formata za vizualizaciju, od kojih smo koristili `vtk` format. Za čitanje `vtk` formata odabrali smo program Paraview. Više o FEniCS-u može se saznati u [15] a o Paraview-u u [16]. U nastavku dajemo pojašnjenje nekih pojmova.

Ime	
bicgstab	Metoda stabiliziranih bikonjugiranih gradijena
hypre_amg	Hypre algebarski generator mreže(prekondicioner)
cg	Metoda konjugiranih gradijenata
sor	Metoda sukcesivne nadrelaksacije (prekondicioner)

Dodatak B

Kod za primjer rekonstrukcije signala

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.ticker import LinearLocator, FormatStrFormatter
from numpy import dot, multiply, diag, power
from numpy import pi, exp, sin, cos, cosh, tanh, real, imag
from numpy.linalg import inv, eig, pinv
from scipy.linalg import svd

# definiramo vremensku i prostornu domenu
x = np.linspace(-10, 10, 400)
t = np.linspace(0, 4*pi, 200)
dt = t[2] - t[1]
Xm,Tm = np.meshgrid(x, t)

# konstruiramo matricu podataka
f1 = multiply(1/np.cosh(Xm+3), exp((2.3j)*Tm))
f2 = multiply(multiply(1/cosh(Xm/2), tanh(Xm)), 2*exp((2.8j)*Tm))
D = (f1 + f2).T

# DMD ulazne matrice
X = D[:,:-1]
Y = D[:,1:]

U2,Sig2,Vh2 = svd(X, False)
r = 2 #rang
U = U2[:,:r]
Sig = diag(Sig2)[:r,:r]
V = Vh2.conj().T[:,:r]
```

```
30
31 # Rayleighov kvocijent
32 Atil = dot(dot(dot(U.conj().T, Y), V), inv(Sig))
33 mu,W = eig(Atil)
34
35 # racunanje DMD modova
36 Phi = dot(dot(dot(Y, V), inv(Sig)), W)
37
38 # racunanje vremenske evolucije
39
40 b = dot(pinv(Phi), X[:,0])
41 Psi = np.zeros([r, len(t)], dtype='complex')
42 for i,_t in enumerate(t):
43     Psi[:,i] = multiply(power(mu, _t/dt), b)
44
45 # DMD rekonstrukcija
46 D2 = dot(Phi, Psi)
```

Dodatak C

Kod za POD metodu

```
from fenics import *
import numpy as np
import scipy.linalg as la
from math import *
from pod import pod_ucitaj
from pod import pod_func
import matplotlib.pyplot as plt

T = 5.0
broj_vremenskih_koraka = 500
dt = T / broj_vremenskih_koraka
broj_elemenata_baze=0
tol=0.01

#ucitavanje mreze i konstrukcija prostora konacnih elemenata
mesh = Mesh('navier_stokes_cylinder_pod/cylinder.xml.gz')
P1 = FiniteElement('P', triangle, 1)
element = MixedElement([P1, P1, P1])
V = FunctionSpace(mesh, element)
u = Function(V)
u_1, u_2, u_3 = u.split(deepcopy=True)
v=u_3.vector()
v=as_backend_type(v)
broj_redaka = v.vec().size

#ucitavanje uzoraka
timeseries_w = TimeSeries('reaction_system_pod5/velocity_series_u3')
U=np.zeros((broj_redaka,broj_vremenskih_koraka))
t = 0.0
for n in range(broj_vremenskih_koraka):
  t += dt
  timeseries_w.retrieve(u_3.vector(), t)
```

```
33    U[:,n]=u_3.vector()
34
35  #matrica korelacije, eig
36  Ut=np.transpose(U)
37  K=np.matmul(Ut,U)
38  d,vl=la.eigh(K)
39  i=d.argsort()[::-1]
40  d=d[i]
41  vl=vl[:,i]
42
43  #odredivanje dimenzije baze
44  suma_d=d.sum()
45  menergy=d/suma_d
46  energija=0.0
47  l=0
48  while (1.0-energija) > tol:
49    energija=energija+d[l]/suma_d
50    l=l+1
51
52  #vektori baze
53  phi=np.zeros((broj_redaka,l))
54  for i in range(l):
55    tmp=np.matmul(U,vl[:,i])
56    phi[:,i]=tmp/sqrt(d[i])
57
58  #rekonstrukcija rjesenja
59  rj=np.zeros((broj_redaka,1))
60  rjesenje=np.zeros((broj_redaka,broj_vremenskih_koraka))
61  koef=np.zeros((l,broj_vremenskih_koraka))
62  for i in range(broj_vremenskih_koraka):
63    for j in range(l):
64      koef[j,i]=np.inner(U[:,i],phi[:,j])
65    rjesenje[:,i]=np.matmul(phi,koef[:,i])
66
67  #greska
68  greska= np.subtract(rjesenje,U)
69  rel_e=np.linalg.norm(greska)/np.linalg.norm(U)
70
71  #ispis u vtk formatu
72  vtkfile_u3 = File('pod_conc/u3_pod.pvd')
73  vtkfile_u3_exact = File('pod_conc/u3_pod_exact.pvd')
74  vtkfile_u3_error = File('pod_conc/u3_pod_error.pvd')
75  vtkfile_u3_phi = File('pod_conc/u3_pod_phi.pvd')
76  for i in range(broj_vremenskih_koraka):
77    u_3.vector()[:] = np.array(np.real(U[:,i]))
78    vtkfile_u3_exact << (u_3, i*dt)
79  for i in range(broj_vremenskih_koraka):
```

```
80    u_3.vector()[:] = np.array(np.real(rjesenje[:,i]))
81    vtkfile_u3 << (u_3, i*dt)
82  for i in range(l):
83    u_3.vector()[:] = np.array(np.real(phi[:,i]))
84    vtkfile_u3_phi << (u_3, i*dt)
85  for i in range(broj_vremenskih_koraka):
86    u_3.vector()[:] = np.array(np.real(greska[:,i]))
87    vtkfile_u3_error << (u_3, i*dt)
```

Dodatak D

Kod za DMD metodu

```
from fenics import *
import numpy as np
import scipy.linalg as la
from math import *
from pod import pod_ucitaj
from pod import pod_func
import matplotlib.pyplot as plt
import scipy.io

T_start= 3.0
T_end = 7.0
broj_vremenskih_koraka = 250
dt = (T_end-T_start) / broj_vremenskih_koraka

#ucitavanje mreze, konstrukcija prostora konacnh elemenata
mesh = Mesh('navier_stokes_cylinder_1dmd/cylinder.xml.gz')
W = VectorFunctionSpace(mesh, 'P', 2)
w = Function(W)
v=w.vector()
v=as_backend_type(v)
broj_redaka = v.vec().size
freeinds=W.dofmap().dofs()

#ucitavanje uzoraka
timeseries_w = TimeSeries('navier_stokes_cylinder_1dmd/velocity_series')
U=np.zeros((broj_redaka,broj_vremenskih_koraka))
t = T_start
for n in range(broj_vremenskih_koraka):
  # Update current time
  # Read velocity from file
   timeseries_w.retrieve(w.vector(), t)
  U[:,n]=w.vector()
```

```
33    t += dt
34 n=len(U[0])
35
36 X1=U[:,:-1]
37 X2=U[:,1:]
38
39 Uev,D,Vev=la.svd(X1,full_matrices=False)
40 r=n-1
41 Ur=Uev[:,0:r]
42 Sr=D[0:r]
43 Vr = Vev[:r, :]
44 Sr=np.diag(Sr)
45
46 At= Ur.T.dot(X2).dot(Vr.T)/Sr
47 Ev,Rev=la.eig(At)
48 Phi=np.dot(np.dot(np.dot(X2,Vr.T)),np.linalg.pinv(Sr)),Rev)
49
50 #vremenski razvoj
51 b,_,_,_=la.lstsq(Phi,X1[:,0])
52 omega = np.log(Ev) / dt
53 time=np.arange(0,r)*dt;
54 time_dy=np.zeros([r,len(time)],dtype='complex')
55
56 for i in range(broj_vremenskih_koraka-1):
57    time_dy[:,i]= np.multiply(b,np.exp(omega*time[i]))
58
59 #rekonstrukcija
60 U_rek=np.dot(Phi,time_dy)
61
62 #relativna greska
63 greska= np.subtract(np.real(U_rek),X1)
64 rel_e=np.linalg.norm(greska)/np.linalg.norm(X1)
65
66 #ispis u vtk formatu
67 up = Function(W)
68 vtkfile_dmd = File('dm/dmd.pvd')
69 vtkfile_phi = File('dm/phi_vtk.pvd')
70 vtkfile_greska3 = File('dm/greska3_vtk.pvd')
71 vtkfile_exact = File('dmd/exact_vtk.pvd')
72 for i in range(r):
73    up.vector()[:] = np.array(np.real(X1[:,i]))
74    vtkfile_exact << (up, i*dt)
75 for i in range(r):
76    up.vector()[:] = np.array(np.real(U_rek[:,i]))
77    vtkfile_dmd << (up i*dt)
78 for i in range(r):
79    up.vector()[:] = np.array(np.real(Phi[:,i]))
```

```
80    vtkfile_phi << (up, i*dt)
81  for i in range(r):
82    upvector()[:] = np.array(np.real(greska[:,i]))
83    vtkfile_greska3 << (up i*dt)
```

Bibliografija

[1] Roger A. Horn, Charles R. Johnson, *Topics in matrix analysis*, Cambridge University Press, 1994.

[2] Muruhan Rathinam, Linda R. Petzold, *A new look at proper orthogonal decomposition*, SIAM Journal on Numerical Analysis, 2003., Vol. 41, No. 5 : pp. 1893-1925.

[3] Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, J. Nathan Kutz, *On Dynamic Mode Decomposition: Theory and Applications*, Journal of Computational Dynamics,2013.

[4] Alessandro Alla, J. Nathan Kutz, *Nonlinear model order reduction via dynamic mode decomposition*, 2016.

[5] Karl Kunisch, Stefan Volkwein, *Optimal snapshot location for computing POD basis functions*, ESAIM: Mathematical Modelling and Numerical Analysis, 2010.

[6] S. S. Chaturantabut, D. Sorensen, *Nonlinear model reduction via discrete empirical interpolation*, SIAM Journal on Scientific Computing, 32 (2010), pp. 2737–2764.

[7] S. Volkwein, *Model reduction using proper orthogonal decomposition*, December 2011.

[8] Anindya Chatterjee, *An introduction to the proper orthogonal decomposition*, Current Science, Vol. 78, No. 7, 2000.

[9] Zlatko Drmač, Igor Mezić, Ryan Mohr, *Data driven modal decompositions: analysis and enhancements*, AIMdyn Technical Reports, No. 201708.004, Ver. 1, pp. 1- 37, 2017.

[10] G. Eckart,G. Young, *The approximation of one matrix by another of lower rank*, Psychotmetrika 1: 211-218, 1936.

[11] L. Mirsky, *Symmetric gauge functions and unitarily invariant norms*, Quart. J. Math. Oxford 11:50-59, 1960.

[12] M. R. Jovanović, P. J. Schmid, and J. W. Nichols, *Sparsity-Promoting Dynamic Mode Decomposition*, Physics of Fluids, vol. 26, no. 2, p. 024103, 2014.

[13] J. Nathan Kutz, Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*, 2016.

[14] Ky Fan, *On a Theorem of Weyl Concerning Eigenvalues of Linear Transformations: I*, Proceedings of the National Academy of Sciences of the United States of America, 1949.

[15] https://fenicsproject.org/

[16] https://www.paraview.org/

[17] Z. Drmač, V. Hari, M. Marušić, M. Rogina, Sanja Singer, Saša Singer *Numerička analiza*, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, Zagreb, 2003.

[18] K. K. Chen, Jo. H. Tu, C. W. Rowley *Variants of Dynamic Mode Decomposition: Boundary Condition, Koopman, and Fourier Analyses*, J Nonlinear Sci (2012) 22:887–915

[19] R. Pinnau, *Model Reduction via Proper Orthogonal Decomposition*, Technische Universität Kaiserslautern, Germany

Sažetak

U radu smo proučavali dvije dekompozicije dinamičkih sustava: POD i DMD. Fokus je bio na razumijevanju teorijske osnove u pozadini algoritma te primjeni u dobivanju modela nižeg reda (u slučaju POD) te rekonstrukciji i predikciji (kod DMD metode).

POD dekompozicija je metoda koja za dobivanje modela nižeg reda promatranog sustava traži projekciju sustava na ortonormiranu bazu, ali takvu da je projekcija optimalna u smislu najmanjih kvadrata. Spomenutu ortonormiranu bazu nazivamo POD baza. Opisali smo računanje POD baze u nekoliko mogućih slučajeva: konačnodimenzionalni slučaj, kada se svodi na računanje SVD-a, za težinski skalarni produkt te pomoću metode uzoraka. Aproksimacijski model nižeg reda dobili smo Galerkinovom projekcijom na prostor razapet POD bazom. Ukupnu grešku reduciranog sustava dali smo u ovisnosti o greški aproksimacije.

DMD metoda bazira se na pronalasku koherentnih prostornih struktura koje čine gibanje. One su dane svojstvenim vektorima matrice operatora A koji opisuje dinamiku sustava. Operator preslikavanja A općenito je nepoznat, no poznata su dva skupa uzoraka, početni i preslikani. Iz danih skupova uzoraka u stanju smo odrediti operator $\tilde{A}_k$ koji je linearna aproksimacija od A. Pokaže se da su svojstvene vrijednosti i svojstveni vektori operatora $\tilde{A}_k$ aproksimacije svojstvenih vrijednosti i vektora od A. Opisali smo tri algoritma za računanje DMD modova koji se razlikuju u ograničenjima na skup ulaznih podataka te računanju svojstvenih vektora od $\tilde{A}$. Vremensku evoluciju sustava određuju svojstvene vrijednosti od A. Potpunu rekonstukciju sustava dobivamo iz DMD modova, pripadnih svojstvenih vrijednosti i početnih amplituda.

Naposljetku smo opisane metode primijenili na dinamičkim sustavima koje opisuju Navier-Stokesove jednadžbe (za DMD), te jednadžbe konvekcije difuzije reakcije (za POD).

Summary

In this thesis we have studied two decompositions of dynamic systems: POD and DMD. The focus was on understanding the theoretical background defining the algorithm and applying it to obtain the low-order model (for POD) and reconstruction and prediction (for DMD method).

The POD decomposition is a method that seeks a projection of the system to an orthonormal basis to obtain a low-order model of the observed system, but such that the projection is optimal in the least-squares sense. Above mentioned orthonormal basis is called the POD basis. We have outlined the baseline POD calculation in several possible cases: in finite-dimension, when it's as simple as calculating an SVD of a matrix, in more complex space with weighted scalar product, and by snapshot method. The low-order model was obtained by Galerkin's projection to the space spanned by the POD basis. The total error of the reduced system is given as a function of the approximation error.

The DMD method is based on finding coherent spatial structures that define the dynamics of the system. These are given by the eigenvectors of the mapping matrix A. Matrix A is generally unknown, but two sets of snapshots are known. From the given sets of snaphots we can determine the operator $\tilde{A}$, which is a linear approximation of A. It can be shown that the eigenvalues and eigenvectors of $\tilde{A}_k$ are aproximations of eigenvalues and eigenvectors of A. We have provided three algorithms for computing DMD modes that differ in the constraints on the set of input data and the computation of eigenvectors of $\tilde{A}$. The system's time evolution is determined by eigenvalues of A. Complete reconstruction of the system is obtained from DMD modes, associated eigenvalues and initial amplitudes.

Finally, we applied the DMD method on dynamic system governed by Navier-Stokes equations, and POD on system governed by equations of convection diffusion reaction.

Životopis

Marta Kapetanović, rođena je 08.05.1992. godine u Splitu, gdje je pohađala Osnovnu školu "Split 3" te Opću gimnaziju "Marko Marulić". Školovanje nastavlja u Zagrebu gdje 2010. godine upisuje Preddiplomski sveučilišni studij Matematika a potom, 2015. godine Diplomski studij Primijenjene matematike.