Tallinna Tehnikaülikool
Meresüsteemide Instituut

Mere operatiivmudelisüsteemi NEMO kasutuselevõtt ja töölerakendamine mereala operatiivprognooside parandamiseks

Lõpparuanne

Keskkonnaministeeriumi käsunduslepingu 4-1/20/33 lõpparuanne

Ilja Maljutenko
Jüri Elken
Priidik Lagemaa
Svetlana Verjovkina

Tallinn 2022

Annotatsioon

Käesoleva arendustöö ülesandeks on uue põlvkonna mere operatiivmudelisüsteemi NEMO kasutuselevõtt ja töölerakendamine mereala operatiivprognooside parendamiseks. Kolmandas etapis oli ettenähtud mudelsüsteemi lõplik häälestamine ja üleandmine Keskkonnaagentuuri (KAUR) ilmateenistusele. Töö tulemused on esitatud käesolevas aruandes, mis täiendab esimese ja teise etapi aruannet. Aruandes on esitatud operatiivse süsteemi ülesehitus ning talitus. Lisaks on toodud ka vaatlusandmete assimileerimise metodoloogiline ja tehniline kirjeldus.

Uus mudelsüsteem NEMO-Est on operatiivses režiimis tööle rakendatud Keskkonnaagentuuri ilmateenistuse hallatavas kobararvutis. Prognoos arvutatakse igapäevaselt tunnise salvestus intervalliga 3 päeva ette, kasutades operatiivseid CMEMS (Euroopa Copernicus merekeskkonna seire teenus) okeanograafia ja ECMWF (European Centre for Medium-Range Weather Forecasts) meteoroloogiaandmeid. Mudeli käimashoidmiseks ja andmete automaatseks allalaadimiseks ning konverteerimiseks on üles seatud skriptide ja failide süsteem, mis tegeleb ka esmaste tõrgete lahendamisega. Väljundandmete kvaliteeti parandab vaatlusandmete assimilatsioon igapäevastesse mudeli taaskäivitamise failidesse. Eraldi hoitakse käimas assimileerimata simulatsioon ja versioon, kus assimilatsioon on aktiveeritud.

Sisukord

Annotatsioon **1**

1. Sissejuhatus **3**

2. Prognoosisüsteemi kirjeldus **5**

2.1 Prognoosisüsteemi arhitektuur **5**

2.1.1 Failisüsteem 5

2.1.2 Skriptide süsteem 8

2.1.3 Täiendav tarkvara 11

2.1.4 Mudelsüsteemi käivitamine 12

2.2 NEMO Mudeli kirjeldus **13**

2.2.1 Mudeli üldine kirjeldus ja seadistus 13

2.2.2 Mudeli väljundparameetrid 14

2.3 Vaatlusandmete assimileerimine **14**

2.3.1 Tehniline kirjeldus 15

2.3.2 Operatiivse prognoosi tulemused 16

3. Prognoosisüsteemi tõrked **20**

3.1 Tehnilised vead 20

3.2 Mudeli sisemised vead 20

4. Kokkuvõtte **24**

Kasutatud kirjandus **25**

Lisad **27**

Lisa 1 Operatiivse süsteemi peakataloogi alamstruktuur 27

Lisa 2 Algväljade interpoleerimine 29

Lisa 3 NEMO väljundite deklareerimine 31

Lisa 4 Assimileerimise metodoloogia ja rakendus (inglise keeles) 32

Lisa 5 Amplituudide filtreerimine 47

1. Sissejuhatus

Läänemere operatiivsete prognooside süsteemi uuendamiseks algatati 2020. a. kevadel Keskkonnaministeeriumi, TTÜ meresüsteemide instituudi ja Keskkonnaagentuuri (KAUR) koostööna projekt "Mere prognoosisüsteemide arendamine NEMO mudelisüsteemi kasutuselevõtuks". Käesolev aruanne annab ülevaate nimetatud projekti III etapis tehtud töödest ja saadud tulemustest.

NEMO (Nucleus for European Modelling of the Ocean, https://www.nemo-ocean.eu/) on rahvusvahelise koostöö tulemusena arendatud meredünaamika mudelite süsteem, mis kätkeb endas mitmeid alammudeleid, võimaldades neid kasutada paljudes mereuuringute ja prognooside rakendustes. Algselt ookeani mudeliks arendatud mudelit on edukalt uuringuteks rakendatud ka Läänemeres (Hordoir et al., 2019) ning selle alambasseinides Soome lahes (Vankevich et al., 2016; Westerlund et al., 2018, 2019) ja Botnia meres (Westerlund and Tuomi, 2016). Uuringuliste rakenduste põhjal töötati Euroopa merekeskkonna seireteenusesse Copernicus (CMEMS) Läänemere seire- ja prognoosikeskuse (BALMFC) poolt välja kogu Läänemerd hõlmav NEMO operatiivne seadistus (Kärna et al., 2021), mida testiti 2014-2016 vaatlusandmetega. CMEMS-BALMFC koostöös osaleb Eestist TTÜ Meresüsteemide instituut.

NEMOt arendatakse ja jagatakse CECILL litsentsi alusel, mis on Prantsusmaa adaptsioon GNU litsentsist. Avatud lähtekoodiga tarkvara võivad kõik mitteärilistel eesmärkidel kasutada ja modifitseerida. Käesolevas aruandes on kasutusel NEMO versiooni 4.0.0 alamversioon NEMO-NORDIC.

Senise mere prognoosisüsteemi HBM uuendamine NEMO põhiseks kannab endas mitut eesmärki:

- võtta kasutusele laiemalt levinud ning kiiremini arenev mudelsüsteem,
- parandada prognooside kvaliteeti ja töökindlust,
- võimaldada biokeemia mudeli kasutuselevõttu.

Projekti kolmandal etapil on järgmised põhilised ülesanded:

- Kasutajate koolitus ja arendatud prognoosisüsteemi kasutamise kirjeldus.
- Mudelisüsteemi lõplik häälestus ning kasutusjuhendi loomine.

Käesolevas aruandes on esitatud projekti kolmanda etapi *(NEMO prognoosisüsteemi valideerimine, NEMO töölepanek KAURis, koolitused)* tulemused. Aruande teises peatükis kirjeldatakse operatiivset mudelit, selle viimaseid arendusi ning operatiivsüsteemi, mis NEMO prognoose toodab. Kolmandas peatükis on välja toodud esinenud vead preoperatiivsel perioodil ning juhised nende kõrvaldamiseks.

Käesolev aruanne on põhimaterjaliks KAUR inimeste koolitamisel. Lisaks toimus kolmanda etapi tööde raames eraldi üritusena kasutajakoolitus 4. märtsil 2022 TTÜ meresüsteemide instituudis (ruum 327), kus kohale tulnud spetsialistidele tutvustati NEMO mudelsüsteemi paigaldamist ja prognoosisüsteemi toimimist. Selleks valmistati ette käesolevas aruandes kirjeldatud ühtne juurkataloogil põhinev süsteem, mida kasutajad said hõlpsalt kopeerida oma kasutajakontole ning käivitada kogu prognoosisüsteem oma kasutaja alt. Ülejäänud koolitus toimus konsulteerimise teel, kasutades elektroonilisi sidevahendeid.

2. Prognoosisüsteemi kirjeldus

Arendatud prognoosisüsteemi ülesanne on tagada NEMO mudeli perioodiline käigushoidmine ning selleks vajalike andmete haldamine ja teisendamine. Süsteem koosneb mitmetest omavahel seotud shell-skriptidest, mis on peaasjalikult kirjutatud Bash süntaksis (Bourne-again Shell). Paljude skriptide aluseks on võetud HBM prognoosisüsteemi skriptid (Lagemaa, 2015). Skriptid on väljatöötatud ilmateenistuse klastri jaoks ja neid hoiustatakse ka eraldiseisvana koodihaldussüsteemis, mis asub keskkonnaministeeriumi siseveebis https://gitlab.sise.envir.ee/Ilja.Maljutenko/nemo_op_sys .

2.1 Prognoosisüsteemi arhitektuur

2.1.1 Failisüsteem

Kolmandas etapis on prognoosisüsteem arendatud KAURi ilmateenistuse klastris juurkataloogis **/scratch/tmp** asuvas **nemo_op_sys** kataloogis. Selle all on järgmised alamkataloogid:

model/

Sisaldab NEMO mudeli (**nemo4_dev**) ja XIOS (**xios-2.5**) lähtekoode ning masin-faile kompileerimiseks erinevatel platvormidel.

main_scripts/

Sellesse kausta on kogutud prognoosisüsteemi üldised skriptid ning mõned ilmateenistuse kobararvutile kohandatud programmid kataloogides **bin**, **lib**, **inc** (**cdo**, **ncpdq**). Eraldi alamkaustadesse on salvestatud lähtekoodid abiprogrammidele, mille ülesandeks on assimileerimiseks vajalike andmete allalaadimine ning rekonstruktsiooniks vajalike mood-amplituudide välja arvutamine.

forcing/

Peamine sisendandmete kaust, mis sisaldab endas nii prognoosiks vajalike meteoroloogia, rajatingimuste kui ka jõgede sisendeid. Selles kaustas hoitakse ka erinevate simulatsioonide taaskäivitusfaile.

bathy_meter.nc*:* mudeli batümeetria ja 2D mask andmed. Sisaldab koordinaadistikku, mida kasutatakse mudeli võrgule interpoleerimiseks.

domain_cfg_EST_0.5nm_V110.nc*:* mudeli arvutusvõrgud (u,v,w,t,f).

rean_nrt_V110.<kuupäev>.nc*:* mudeli arvutusvõrgule interpoleeritud soolsuse ja temperatuuri algväljad CMEMS kataloogist. Faili jaluses on cdo käsud, millega on fail genereeritud.

forcing/**boundary**/

Avatud radadega seotud failid.

gridfile_bdy_est05: NEMO_Est avatud rajale vastav võrgu fail cdo'ga interpoleerimiseks.

forcing/boundary/**cmems_nrt/**

Alamkataloogis salvestatakse / kopeeritakse CMEMS serverist alla laetud rajatingimuste faile. Peale vormindamist NEMO sisendiks on rajatingimuste failid struktureeritud järgmistesse kataloogidesse (yyyy, mm, dd , hh vastavad aasta, kuu, päev, tund numbrile), mis vastavad vastava kuupäeva prognoosi sisendile.

- veetaseme ja jää rajatingimused

forcing/boundary/**cmems_nrt_bc_V110**/<yyyy>/<mm>/<dd>/<hh>/**bdy_hourly_2d_y**<yyyy>**m**<mm>**d**<dd>**.nc**

- soolsuse ja temperatuuri rajatingimused

forcing/boundary/**cmems_nrt_bc_V110**/<yyyy>/<mm>/<dd>/<hh>/**bdy_hourly_3d_y**<yyyy>**m**<mm>**d**<dd>**.nc**

forcing/**meteo**/

Selles kataloogis asuvad kõik meteoroloogia sisendid.

Alamkataloogis forcing/meteo/**meteo_nemo_ecmwf_BAL/** asuvad interpoleerimiskaalud (vt. Elken et al., 2021):

meteo_ecmwf_est05_bicub.nc

meteo_ecmwf_est05_bilin.nc

Meteo sisendfailid on grupeeritud prognooskuupäeva (yyyy, mm, dd) ja tunni (hh) järgi

<yyyy>/<mm>/<dd>/<hh>/**FORCE_ecmwf_***

forcing/**runoff**/

Kataloog sisaldab jõgede sissevooludega seotud sisendandmeid.

forcing/runoff/**runoff_q_seasonal**/river_data_*

Jõgede klimatoloogilised vooluhulgad igaks aasta päevaks. Referentsaasta 2015. aasta (Elken et al., 2021).

forcing/runoff/**runoff_t_atmt2**/river_data_t_*

Jõgede veetemperatuur, mis on arvutatud 4-päevase viitega meteoroloogia sisendist.

forcing/**hotstart**/

Mudeli taaskäivitamise failid vastavalt OCE mooduli, jää ja biogeokeemia parameetritele.

<runid>/ <yyyy>/<mm>/<dd>/<runid>_<num>**_restart_out_**<DOM>.nc

<runid>/ <yyyy>/<mm>/<dd>/<runid>_<num>_restart_ice_out_<DOM>.nc

<runid>/ <yyyy>/<mm>/<dd>/<runid>_<num>_restart_trc_out_<DOM>.nc

<runid> e. jooksu identifikaator on defineeritud **run_model_op.sh**. <num> on arvutuse samm, mil taaskäivitusfail on eelmise prognoosi käigus tekitatud (240s mudeli sammu korral on 24h taaskäivitusfail numbriga 360).

forcing/**dataassim/**

Vaatlusandmetega seotud failisüsteem, kus asuvad assimileerimiseks vajalikud klimatoloogilised ja muutlikkuse EOF mood väljad. Alamkataloogis forcing/dataassim**/obs** on alla laetud ja vormindatud vaatlusandmed koos vajalike mood-amplituud failidega (vt Pt 2.3.2). EOF rekonstruktsioonid ning nendele vastavad amplituudid on salvestatud pinnakihi, põhjakihi, 3D välja assimileerimisele vastavalt SUR/, DEP/ ja DA3D/ alamkataloogides.

setup/

Selles kaustas on kirjeldatud NEMO mudeli seadistus NEMO-Est. Kõik simulatsiooniks vajalikud staatilised sisendid ja nimistufailide vormid. Prognoosi käigus tekitatakse sellest kaustast eraldiseisev mudeli jooksu kaust kujul <simulatsiooni_id>_<kuupäev>, kuhu kopeeritakse kõik simulatsiooniks vajalikud sisendfailid.

setup_DA/

Andmete assimileerimiseks vajalikud lisakaustad koos vajalike skriptidega, mis kopeeritakse samuti mudeli jooksu kausta.

op_output/

Kataloog, kuhu salvestatakse väljundist töötluse käigus tekkinud andmete produktid. Eraldi alamkataloogid on pinnakihi jaoks, jaamade aegread, päeva keskmised statistikud. Vastavalt vajadusele võib operatiivsüsteemi initsialiseerimise skriptis (*init_user_vars.sh*) määrata mõne teise asukoha.

op_arch/

Sellesse kataloogi kogutakse mahukamad 3D väljundid, perioodiliselt taaskäivitusfailid, jms, mis kuuluvad arhiveerimisele.

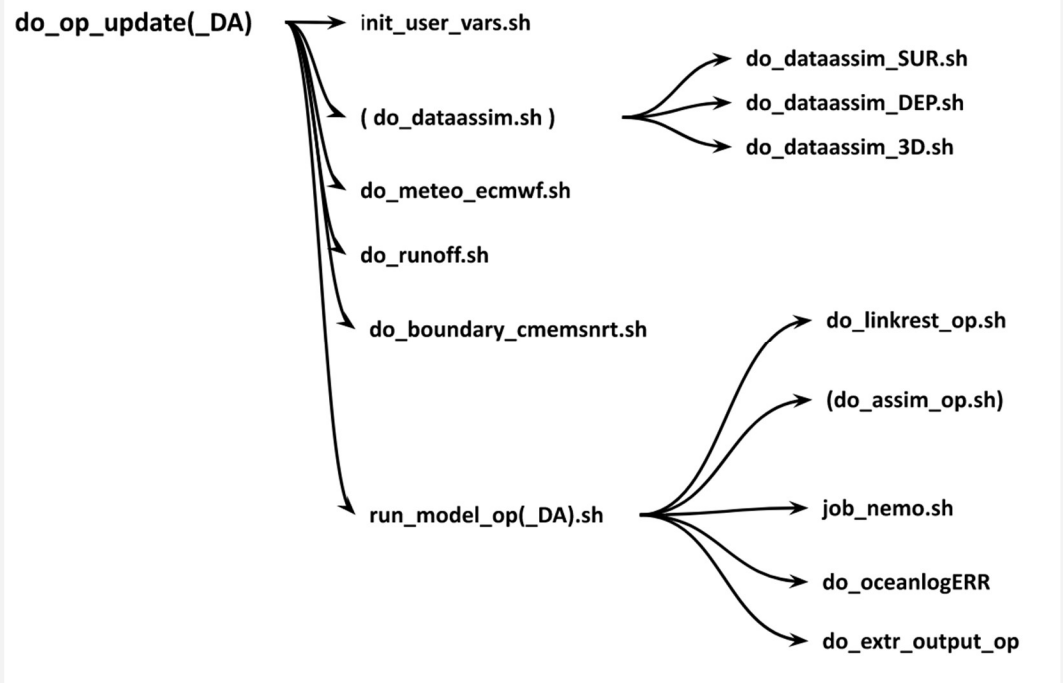
<RUNID>_<timestamp>/

Prognooskataloog, kuhu kopeeritakse kõik sisendid ning milles toimub mudeli käivitamine. Ruumi säästmise huvides luuakse kopeerimise asemel taaskäivitusfailidele otseteed (soft links) kaustast forcing/hotstart/RUNID/timestamp/ .

Lisas 1. on toodud failisüsteemi kaks esimest astet, koos faili nimetustega, mis neis kataloogides on.

2.1.2 Skriptide süsteem

Skriptide süsteem ning nende käivitamise järjekord on esitatud joonisel 2.1.1.



Joonis 2.1.1. Operatiivses režiimis käivitatavad shell-skriptid. Sulgudes on toodud skriptid, mis käivitatakse andmete assimileerimise korral.

do_op_update

Operatiivse süsteemi peaskript. Selle käivitamisega algab järjestikuste alamskriptide üleskutsumine. Esmalt seadistatakse kõik operatiivse süsteemi vajalikud virtuaalsed keskkonna muutujad (sisendandmete kataloogide asukohad) skriptis **init_user_vars.sh**. Ilmateenistuse klastris on selleks

source /fefs/scratch/tmp/nemo_op_sys/main_scripts/init_user_vars.sh

Üldiseks keskkonna muutujaks deklareerimist *export scrdir=$scrdir*, on vaja selleks, et alamskriptides saaks deklareeritud ja kasutatud samad parameetrid

source ${scrdir}init_user_vars.sh.

Kuupäeva (aja) genereerimine. Linux *date* käsuga genereeritakse prognoosi aasta, kuu ja päev. Neid saab lihtsa muutusega ka tsükliga defineerida (vt. kommentaare). Järjestikku kutsutakse üles skripte, mis genereerivad meteoroloogia sisendi (**do_meteo_ecmwf.sh**), jõgede sisendi (**do_runoff.sh**) ja rajatingimuste sisendi (**do_boundary_cmemsnrt.sh**). Mudeli simulatsiooni haldusskript (**run_model_op.sh**) ja võimalusel sõnumi saatmise skript kutsutakse viimases järgus.

do_op_update_DA

Operatiivne süsteem, mis lisaks eelmistele tegevustele kätkeb endas ka andmete assimileerimisega setud skripte. Vaatlusandmete laadimine ja neile vastava rekonstruktsiooni ettevalmistamine (do_dataassim.sh) ja mudeli haldusskript koos andmete assimileerimisega taaskäivitusfailidesse (**run_model_op_DA.sh**).

init_user_vars.sh

Skript, kus deklareeritakse sisend- ja väljundandmete kataloogid.

do_meteo_ecmwf.sh yyyy mm dd

Meteoroloogiat ettevalmistav skript, mis kutsutakse kolme kuupäeva argumendiga: 4-kohaline aasta yyyy, 2-kohaline kuu ja kuupäeva number. Skripti esimeses blokis toimub lähtefailide kontroll. Mudel otsib ECMWF meteo faile kataloogist **ECMWF_NES_dir** (määratud **init_user_vars.sh**) kujul NES${amm}${add}${ahh}00${pmm}${pdd}${phh}001 . a-algusega kuu, päev ja number tähistavad analüüsi, p-algusega muutujad prognoosi. Kui skript leiab vähemalt kaks prognoosfaili, siis järelikult on meteo laadimine ECMWF serverist alles pooleli ning skript ootab 2700 s (45 min) enne kui jätkab. Kui prognoosfaile on rohkem kui 72, siis on skriptil 3 päeva jagu andmeid ning jätkub andmete teisendamine.

00:00 prognoos faili puudumist asendab eelmise päeva analüüsi +24h prognoosi samm. Kui juba teisendatud meteoandmeid ei ole, otsitakse eelmise analüüsi grib-faili (NES*).

Meteosisendi teisendamine NEMOle sobilikule kujule toimub kahes astmes: 1. skalaar- ja vektoriaalväljade ekstraheerimine; 2. kumulatiivsete väljade diferentseerimine voogudeks. Kuna kiirgus ja sademete parameetrid on ette antud integraalsetena üle aja, tuleb need teisendada voogudeks SI-ühikutes. Diferentseerimine toimub lahutades prognoosiajale vastavast suurusest eelmise ajahetke väljad ning jagatakse 3600 s (kui meteo väljad on tunnise sammuga). Kõik väljad kogutakse 24-tunni kaupa 1-päeva sisendiks kujule FORCE_ecmwf_y${pyy}m${pmm}d${pdd}.nc. Kuupäeva kuju <sisendi_nimi>**_y**<aasta>**m**<kuu>**d**<päev>**.nc** on NEMO standard sisendandmete lugemiseks. Kõik sisendfailid peavad sisaldama sama suure arvu ajasamme, seega viimase toiminguna on viimase päeva puuduolevate prognoostundide täitmine viimaste saadaval andmetega. Ehk +92h järgnevad viimased 6 tundi kopeeritakse 92h tunnist.

Tuleb meeles pidada, et see skript on arendatud teisendama etteantud konkreetset NES-meteovoogu. Kui toimub muudatus meteosisendis (muutub ala suurus, ühikud, prognoosi pikkus vms) tuleb muudatusi ettevaatlikult teha, kontrollides kõiki ühikuid ning joonistades välja parameetrite hetk-jaotusi ja aegridu.

do_runoff.sh yyyymmddHH

Jõgede veetemperatuuri arvutatakse möödunud päevade õhutemperatuuri põhjal ning salvestatakse forcing/runoff/runoff_t_atmt2/.

do_boundary_cmemsnrt.sh yyyymmddHH

Avatud radade sisendi ettevalmistamise skript. HBM süsteemist üle võetud skript, mis otsib **CMEMS_bdy_dir** kataloogist (määratud init_user_vars.sh) Copernicuse serveritest allalaetud rajatingimuste failid kujul **bc_est_**yyyymmddHH.nc . Kui skript faili ei leia, siis toimub pooletunnine paus, enne kui skript jätkab oma tööd (kui sisendit ei leita, skript väljub). Kasutades cdo tööriista viiakse allalaetud andmed NEMO_Est mudeli võrgule, esmalt interpoleerides horisontaalselt ja seejärel vertikaalselt. Eraldi tsüklis eraldatakse ühest mitme päeva andmeid sisaldavast failist üksikud päevad ning salvestatakse failinimedega, mis vastavad NEMO sisendi üldkujule (vt. meteo sisend).

run_model_op.sh yyyy mm dd

Skript, mis valmistab ette NEMO käivitamiseks vajalikud sisendid, nimistu failid (namelist_*) ning käivitab mudeli.

Esimeseks sammuks luuakse eraldiseisev prognooskaust, millesse kopeeritakse staatilised sisendid ja nimistu failide vormid (**setup/**). Järgmiseks luuakse taaskäivitusfailide sisendkaust (mudeli lähtestamiseks) ning taaskäivitusfailide väljundkaust (järgmiseks prognoosiks). Eraldi kataloogid luuakse meteo, avatud radade ja jõgede jaoks. NEMO ja XIOS binaarfailidele luuakse otseteed (**setup/**).

Simulatsiooni parameetritest defineeritakse esmalt ajaga seotud parameetrid (simulatsiooni algus, lõpp, ajasamm, arvutussammude arv). Nende põhjal luuakse nimistu failide vormidest *cat* ja *sed* käskude abiga (*sed* asendab _parameeter_ väljad shell muutujatega) mudeli poolt loetavad nimistufailid namelist*_ref.

Taaskäivitusfailid eelmisest simulatsioonist, mis peaks asuma forcing/hotstart/ kataloogis, ühendatakse otseteedega prognooskausta kataloogis initialstate/. Iga paralleeltöö tekitab taaskäivitusfaili oma arvutuspiirkonna e. alamdomeeni kohta. Samuti tekib iga alamdomeeni jaoks eraldi fail jää ja biogeokeemia mudeli kohta. Esmalt leitakse eelmise päeva aasta/kuu/päev ja üle-eelmise aasta/kuu/päev, sest kui mingil põhjusel ei ole eelmisest päevast taaskäivitusfaile +24h jaoks, rakendatakse üle-eelmise päeva +48h taaskäivitusfaile. Vaikimisi on eeldatud, et 240 s mudeli ajasammu korral kannab iga +24h taaskäivitusfail numbrit 360, kuid parem on lugeda see number eelmise päeva logi-failist (ocean.output). Restart failide linkimine toimub eraldi alamskriptiga **do_linkrest_op**, mis kopeeritakse alguses **setup/** kataloogist prognooskausta. Kontrollitakse, kas loodud otsetee esimesele alamdomeenile toimib, kui mitte, siis ei saa mudelit lähtestada, skript lõpetab töö ning kasutaja peab hakkama viga kõrvaldama.

Järgnevalt kopeeritakse kõik dünaamilised sisendid. Meteoroloogia sisendi kataloogi forcing_ecmwf/ kopeeritakse interpoleerimise kaalud (weights_ecmwf*) ning meteo sisendfailid (FORCE_*). Jõgede korral kopeeritakse kataloogi runoff_seas/ sesoonsed klimatoloogilised päeva vooluhulgad ning õhutemperatuurist arvutatud jõevee temperatuuri analüüsi päevaks kopeeritakse viiele järgnevale prognoospäevale. Kolmandana kopeeritakse Copernicusest varasemalt ettevalmistatud avatud rajatingimuste failid.

Enne mudeli käivitamist puhastatakse prognooskaust simulatsiooniga seotud logifailides, mis võivad olla sinna sattunud varasemast käivituskatsest, ning luuakse uued logifailid.

Ilmateenistuse klastris haldab töid PBS-haldussüsteem, mis jaotab erinevatele töödele vabasid arvutusressursse. NEMO tööfail (job_nemo.sh) lisatakse järjekorda käsuga *qsub*, millele lisatakse teegid Linux keskkonna muutujate edastamiseks (-V), logifailidele (-o ,-e) ja broneeritud ajakasutusele (-l walltime).

Kui mudel lisatakse järjekorda, hakkab skript 5 sekundi kaupa itereerima logifailide kontrollkäske, mis peaksid välja selgitama, kas simulatsioon on alanud (ilmuvad numbrid time.step logifaili) või on job_nemo.sh lõppenud (qsub logifailis sõna END). Kui iteratsioon on kestnud üle kolme tunni, siis lõpetab skript töö.

do_oceanlogERR alamskript kontrollib, kas NEMO logi fail `ocean.output` sisaldab "E R R O R" sõna, mis väljendab simulatsiooni ajal tekkinud viga. Kui veateade on positiivne, siis esmalt taaskäivitatakse mudel väiksema ajasammuga (180 s), aga kui ka sellele järgneb veateade, siis prognoos katkeb.

Viimase ülesandena viiakse kõik logid, väljundfailid ja taaskäivitusfailid prognooskaustas olevasse kausta, millisest omakorda kopeeritakse taaskäivitusfailid forcing/hotstart kataloogi. Kõige lõpuks käivitatakse skript (**do_extr_output_op**), mis ekstraheerib vajalikud väljad ning viib väljundi kasutaja poolt seadistatud prognoosi väljundi kausta (op_output) ja arhiivi (op_archive).

run_model_op_DA.sh yyyy mm dd

Skript, mis teeb lisaks **run_model_op.sh** ülesannetele ka andmete assimileerimist taaskäivitusfailidesse.

2.1.3 Täiendav tarkvara

Vajalikud teegid ja tarkvarad, mida eeskätt tasuks otsida linux süsteemi kataloogidest või moodulsüsteemis (käsk: *module avail*). Kui vajalike teeke süsteemis ei ole, tuleb vajalik tarkvara internetist laadida ja kasutajal ise kompileerida (juhendi vt. Lisa 1 I etappi Aruandes, Elken et al., 2020).

- Fortrani kompilaator GNU (v4.8+) / Intel (v12+) : NEMO ja XIOS, eof_reconstr_v1 kompileerimiseks. Sobivad ka Cray jt. kompilaatorid (vt. arhitektuur failide näidiseid).
- NetCDF -C, -F (v4.2+) : NEMO, XIOS , aga ka CDO, NCKS kompileerimiseks
- HDF5 (v10+) : NetCDF ja sellest sõltuvate teekide kompileerimiseks
- MPI (OpenMPI) : vajalikud NEMO kompileerimiseks

- Python3 + Motu 2.7 : rajatingimuste ja vaatluste alla laadimiseks (vt. võimalike uuenduste vajadusi Copernicuse kodulehel, https://marine.copernicus.eu/)

- cdo

NetCDF failide teisendamiseks ja töötlemiseks kasutatav tarkvara (Climate Data Operators, code.mpimet.mpg.de), mida kasutatakse süsteemis pea igal sammul (meteo, rajad, assim. rekonstr., jõgede sisend, väljundi vormindamine, jne ...). Kuna cdo sõltub paljudest teistest kohustuslikest ja valikulistest teekidest (grib, zlib, hdf, openmpi, jt.), siis tasub cdo installeerida süsteemi administraatoril või otsida cdo tarkvara pakutavast moodul-süsteemist.

- ncpdq

NCO (nco.sourceforge.net/) alamtööriist NetCDF failide töötlemiseks. Süsteemis kasutatakse nck tööriista dimensioonide restruktureerimiseks rajatingimuste failide töötlemisel.

Andmete assimileerimiseks on lisaks vajalikud järgmised tööriistad, mis on arendatud käesoleva projekti raames.

- eof_reconstr

FORTRAN programm, mis arvutab etteantud vaatlustest ja arvutusvõrgust EOF rekonstruktsioonid ja vastavad mood-amplituudid soolsuse ja temperatuuri väljadele.

- da_obs

Tööriist, mis laeb Copernicuse serverist vaatlusandmeid ning seab need EOF rekonstruktsiooniks vajaliku vormi. Koosneb shell ja Python (+Motu) skriptidest.

- da_edit_netcdf_(SUR)

FORTRAN programmid, mis ühildavad soolsuse ja temperatuuri EOF rekonstruktsioonväljad NEMO poolt eelmise prognoosi lõpuks arvutatud väljadega.

2.1.4 Mudelsüsteemi käivitamine

Mudelsüsteemi käivitamiseks on vaja linux käsureal käivitada peaskript **do_op_update (_DA)**, mis loeb süsteemi kellaajast käesoleva kuupäeva ning käivitab sellega kõik alamskriptid. Kuupäeva saab skriptis ka ise defineerida (või käivitada tsüklis), mis võimaldab varasemaid prognoose üle arvutada.

Kogu operatiivne prognoosisüsteem käivitatakse tsükliliselt automaatse *crontab* ülesandena. Crontab seadistuseks sisestada käsk *crontab -e* . Seejärel saab käsurea redaktorit kasutades sisestada sagedust ja käsku, mida soovitakse käivitada.

Näiteks crontab seadistus
*0 07 * * * /scratch/tmp/nemo_op_sys/main_scripts/do_op_update > /home/iljama/logs/OP_log 2>&1*

käivitab etteantud skripti *do_op_update* igapäevaselt kell 07:00 arvuti süsteemi kellaaja järgi (süsteemi kuupäeva/kellaaega näitab käsk: *date*).

2.2 NEMO Mudeli kirjeldus

2.2.1 Mudeli üldine kirjeldus ja seadistus

Operatiivne süsteem põhineb 3D hüdrodünaamika mudeli NEMO versioonil 4.0, mida on täiendatud Copernicus BALMFC modelleerimise rühma poolt (Kärna et al., 2021). NEMO on FORTRANi keelel põhinev okeanograafiliste võrrandite lahendamiseks loodud programm-moodulite kogum. OCE moodulis, mis on NEMO tuumikuks, toimub meredünaamika võrrandite lahendamine (Madec et al., 1998). Selles lahendatakase advektsiooni, difusiooni võrrandeid ning arvutatakse mere ja atmosfääri vahelisi soojavoogusid. Lisaks OCE moodulile on NEMO-Est konfiguratsiooni kaasatud ka jää (ICE) ja biogeokeemia moodul (TOP). Jää dünaamika modelleerimiseks kasutatakse SI3 (Aksenov et al. 2019) jäämudelit. SI3 lahendab merejää termodünaamikat, jää moondeid (kurrutusi, ladejää teket, rüsistumist) ning ka kinnisjää teket. Mudelsüsteemi on kaasatud ka mere ökosüsteemi mudel ERGOM (Lorkowski et al., 2021), mida saab vajadusel aktiveerida.

Mudeli NEMO-Est konfiguratsioon on välja töötatud projekti esimeses ja teises etapis. Horisontaalne arvutusvõrk Eesti mereala kohta jaotub ühtlaselt vahemikes: pikkus 21,55^{o} E kuni 30,35^{o} E, laius 56,94166666667^{o} N kuni 60,725^{o} N, kusjuures koordinaadid on võetud võrgupesa keskelt. Võrgusamm on geograafilise laiuse järgi 0,5’ ja pikkuse järgi 1’. Laiusel 60^{o} N on mõlemas suunas võrgusammu pikkuseks 0.5 meremiili, ehk ligikaudu 1 km. Vertikaalne

arvutusvõrk on jaotatud 110 kihiks, mille vertikaalne samm on 1 m vahemikus 0 - 80 m ja vahemikus 90 - 130 m on samm 2 m.

Mudel on lähtestatud Copernicuse Läänemere operatiivse mudeli väljundist 28. septembri 2021 jaoks (Skript toodud Lisas 2.). Samast produktist (BALTICSEA_ANALYSISFORECAST_PHY_003_006) saab mudel ka oma avatud rajatingimused soolsusele, temperatuurile ja veetasemetele (Elken et al., 2021). Meteoroloogia sisendiks kasutab mudelsüsteem ilmateenistuse klastrisse seadistatud ECMWF mudeli (NES*) andmevoogu (Elken et al., 2021).

NEMO mudelsüsteem on seadistatud KAUR kobararvutis, kus operatsioonisüsteemina on kasutusel Linux versioon *2.6.32-358.23.2.el6.x86_64 x86_64* (käsk *uname --kernel-name --kernel-release --machine*). Kompileerimiseks on kasutatud Inteli C ja Fortran kompilaatorit: *Intel(R) 64 Compiler XE, Version 15.0.2.164 Build 20150121*. NEMO ja XIOS kompileerimise juhend ja seaded on toodud projekti I etapi aruandes (Elken et al., 2020). Mudel on paralleliseeritud 179 alamdomeenil suurusega 25x28x110 võrgupunkti, mis on ühtlasi ka iga taaskäivitusfaili suuruseks. Mudeli 3D väljad on mõõtmetega 529x455x110 võrgupunkti.

Mudeli seadistustest on detailsemalt kirjeldatud käesoleva töö esimese ja teise etapi aruannetes (Elken et al., 2020, 2021).

2.2.2 Mudeli väljundparameetrid

NEMO mudeli väljundi eest vastutab XIOS server. Prognoosi simulatsiooni alguses käivitatakse mõlemad programmid koos (MPMD job), peale mida hakkab XIOS koguma NEMO väljund parameetreid ning salvestab need prognooskausta NetCDF failidesse. Seadete kaustas (**setup**/) olevate *xml nimistu failide abil hallatakse parameetrite loetelu, mida xios simulatsiooni ajal väljundiks kirjutab. Neis nimistu failides saab lisaks määrata kokkupakkimise astet (compression_level =1...10) või uute failide sagedust (split_freq – sagedus, mille tagant komplekteeritakse väljund). Seadistustööde käigus seadistati mudel kirjutama väljundit 24h kaupa tunniajase sammuga NetCDF failidess, mille kokkupakkimise aste =1. XIOS erinevad seaded ning võimalused on kirjeldatud XIOS kasutaja juhendis (Meurdesoif et al., 2016).

file_def_nemo-oce.xml failis kirjeldatakse väljundit - millise sagedusega, milliseid parameetreid soovitakse väljundfailides näha. Samuti saab kasutaja valida, millise faili nimega või kokkupressimise tasemega väljundit soovitakse. *field_def_nemo-oce.xml* on defineeritud parameetrite loetelu, mida on võimalik väljundfailidesse tellida. Sarnaselt on määratud ka jää ja biogeokeemia väljund vastavalt failides **ice.xml* ja **top.xml*.

Hüdrodünaamika ja jää moodulites salvestatakse:
2D: SSU SSV SST SSS SSH icethic icefrac uice vice snwthic /hoovuste kiirused, temperatuur, soolsus, meretase, jää paksus, jää kontsentratsioon, jää triivi kiirused, lume paksus/;
3D: temperature salinity uos vos wos /veetemperatuur, soolsus, hoovuse kolm komponenti/. Jäämudeli väljundis on lisaks jää triivi kiirustele uice ja vice kõigi 5 jää liigi jaoks eraldi jääkontsentratsioonid icefrac ja jää paksused icethic. Väljundandmete ruumi vajadus ning detailsem kirjeldus on toodud II etappi aruandes (Elken et al., 2021).

2.3 Vaatlusandmete assimileerimine

Mudelisse andmete assimileerimine toimub 00:00 taaskäivitusfailide modifitseerimisega. Uue simulatsiooni sisend S, T väljadele liidetakse juurde kolmemõõtmelised EOF mood-väljad, kasutades vaatlustest tuletatud mood-kaale. Kui saada on ainult FerryBox veepinna mõõtmised ja satelliitmõõtmised, siis kasutatakse kahemõõtmelisi mood-välju.

EOF moodväljad $F_k(x,y,z)$ või $G_k(x,y)$ on määratud CMEMS reanalüüsi andmetest 1993-2019 nii, et algväljad $T(x,y,z,t)$ saab esitada mood-kaalude ja mood-väljade korrutiste summana $T(x,y,z,t)=\Sigma^k A_k(t)F_k(x,y,z)$, kus k on moodi järjekorranumber. Kui meil punktides m on olemas vaatlusandmed $T_m=T(x_m,y_m,z_m,t_m)$, saab leida mood-kaalud $\underline{A}_k(t)$ nii, et rekonstruktsioon $\underline{T}(x,y,z,t)=\Sigma^k\underline{A}_k(t)F_k(x,y,z)$ on diskreetsete vaatlusandmetele väikseima ruutkeskmise veaga lähendus (Elken et al., 2020). Mood-väljadel $F_k(x,y,z)$ on see omadus, et nad kujutavad endast tüüpilisi ruumilise varieerumise mustreid. Kuna tegu on variatsioonidega ümber klimatoloogiliste väljade, tuleb lõplikuks rekonstruktsiooniks rehkendatud mood-väljadele liita juurde klimatoloogilised väljad. Erinevad mood-väljad on omavahel sõltumatud. Peamise osa rekonstruktsioonist saab arvutada vähese arvu mood-väljade ja mood-kaalude abil.

2.3.1 Tehniline kirjeldus

Assimileerimiseks vajalikud EOF moodväljad (G_k ja F_k) ja klimatoloogia väljad on salvestatud NetCDF failides `ST_coarse_clim_SUR.nc` ja `ST_coarse_modevars_SUR.nc` , selleks et EOF rekonstruktsioone saaks luua kasutades *cdo* avaldise operaatorit *expr*. Lisaks on vajalikud ka kataloogis obs/ olevad tekstifailid eigenvalues*, grid*, modes* ja seasonal* (v.t Lisa 1).

Vaatlusandmete assimileerimise etapid on järgmised.

1. DA ettevalmistuseks käivitatakse igapäevaselt kell Shell skript **main_da_obs.sh** (da_obs, täiendtarkvara), mis laadib CMEMS serverist alla viimase 30 päeva ferriboks andmed (INSITU_BAL_NRT_OBSERVATIONS_013_032) DA kvadraadis, ning kirjutab ette antud formaadis ASCII kujul andmed faili `ts_obs_yyyymmdd.dat`, kus `yyyy` on jooksva kuupäeva aasta, `mm` on kuu number ning `dd` on päeva number.

2. **do_dataassim_SUR.sh** skript käivitab DA programmi `eof_reconstr_v1` (kompileeritud FORTRAN failist `eof_reconstr_vX_Y.f90`, kus X ja Y tähendavad versiooni ning tema modifikatsiooni. Seadistused on failides `DA_Config.nml` (annab ette DA põhilised parameetrid) ja `DA_RunConfig.nml` (annab ette kuupäeva ning DA tüübi), mille detailid on toodud lisas 4. Seadistused luuakse sama skriptiga vastavate failide põhjadele (`*_template`).

3. Punkti 2 väljundiks on EOF amplituudid tekstifailides `t_ampl_yyyymmdd_SUR.txt` ja `s_ampl_yyyymmdd_SUR.txt`. Neid amplituude filtreeritakse ajas shellskripti **do_dataassim_SUR.sh** ühe osana (Lisa 5.). Ning salvestatakse sama nimega alamkataloogi SUR/amp_filt/.

4. Skripti **do_dataassim_SUR.sh** teise ülesandena loetakse sisse punktis 3 arvutatud ja filtreeritud `t_ampl_yyyymmdd_SUR.txt` ja `s_ampl_yyyymmdd_SUR.txt` ning *cdo* **expr** operaatori tarbeks konstrueeritakse amplituudide ja moodide avaldis (*estr = " so_Sma=* $\Sigma^k A_k(t)F_k(x,y,z)$ *")*.
cdo $opt expr,$estr ST_coarse_modevars_SUR.nc ST_coarse_SUR_Smxa.nc

5. Kasutades cdo tööriista liidetakse rekonstrueeritud muutlikkuse väljale juurde vastava päeva klimatoloogia (*so_c*) ning salvestatakse järgmisesse faili ST_SUR_coarse_R.nc.
cdo $opt expr,"so_R=so_c+so_Sma;thetao_R=thetao_c+thetao_Sma" ST_coarse_SUR_mergedCSmxa.nc ST_SUR_coarse_R.nc

6. Viimaseks sammuks on rekonstrueeritud väljade viimine mudeli arvutusvõrgule ning maapunktide katmine lähim-naaber meetodiga.
cdo $opt --reduce_dim setmisstonn -remapbil,$bathfile ST_SUR_coarse_R.nc ST_SUR_fine_R.nc

Sellega on loodud vajalikud rekonstrueeritud S ja T väljad (failis *ST_SUR_fine_R.nc*), mida saab siduda mudeli käivitamisel eelmise prognoosi S ja T väljadega.

Edasised sammud toimuvad vahetult enne mudeli käivitamist, kui kopeeritakse (või ühendatakse) prognooskaustas taaskäivitusfaile.

1. Prognoosmudeli käivitamis skriptis (**run_model_op_DA.sh**) otsitakse esmalt SUR, DEP, 3D tähisega rekonstruktsioon faile, andmete assimileerimise kataloogist. Johtuvalt saab muutuja *assim_case* väärtuse, 1, 2 või 3. Failide puudumise korral on muutuja 0 ning assimileerimist ei toimu - taaskäivitusfailid ühendatakse eelmise prognoosiga .

2. EOF Rekonstruktsioon failide olemasolu korral käivitub alamskript (**do_assim_op**), mis kopeerib eelmise päeva taaskäivitusfailid, teostab assimileerimise ning kopeerib modifitseeritud taaskäivitusfaili prognooskaustas olevasse sisend kataloogi initialstate/.

3. Assimileerimist teostatakse Fortran programmi abiga (*edit_netcdf_SUR , edit_netcdf_DEP* või *edit_netcdf_3D*), mis modifitseerib algvälju iga paralleel-domeeni kohta eraldi.

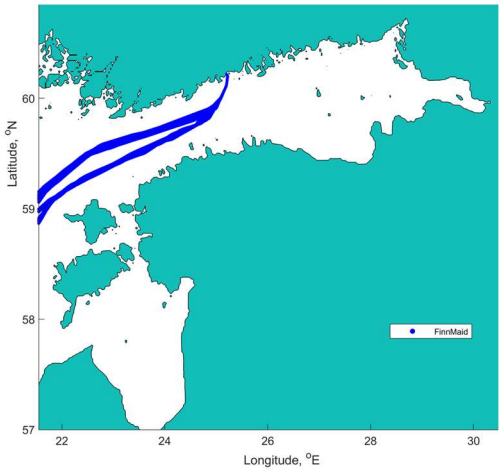
Kui kõik toimingud on edukad, siis ilmuvad prognooskaustas olevasse alamkataloogi initialstate/ kõikide alamdomeenide jaoks viited, mis on ühendatud assimileeritud taaskäivitusfailidele.

2.3.2 Operatiivse prognoosi tulemused

Operatiivne prognoosisüsteem, kuhu on lisatud vaatluste assimileerimine, on järgi arvutanud kõik prognoosid pre-operatiivse perioodi vältel vahemikus 01.10.2021 - 01.05.2022. Assimilatsiooni Operatiivse simulatsiooni tulemusi, kuhu on assimileeritud ka vaatlusi, on võrreldud FerryBox vaatlustega.

FerryBox

Assimileerimisega simulatsiooni on võrreldud FerryBox'ilt saadud soolsuse ja veetemperatuuri aegridadega, piki parvlaeva FinnMaid teekonda ((Joonis 2.3.1). Joonistel 2.3.2-3 on toodud laeval mõõdetud soolsuse ja temperatuuri aegridade vastavused laiuskraadiga. 59 ^{o}N vastab Hiiumaa lääne rannikule ning 60 ^{o}N vastab Soome lahe põhjakaldale (Helsinki). Nende vahele jäävad väärtused vastavad Soome lahe läänepoolsele avaosale.



Joonis 2.3.1. FerryBox laeva trass perioodil 24.09.2021 - 29.04.2022, tähistatud sinise joonega.

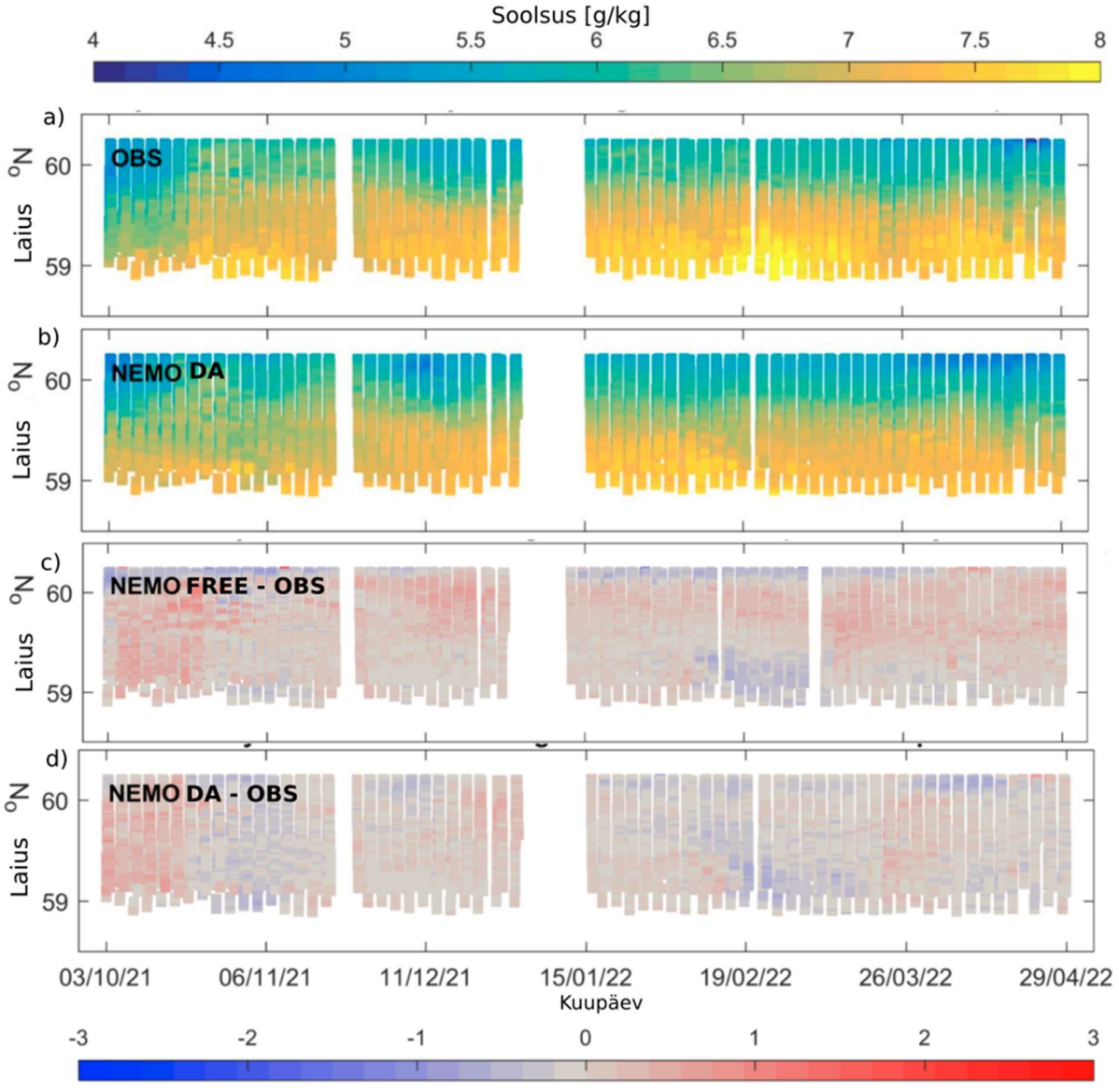
Soolsusest pinnakihis nähtub, et vabajooksus (ilma assimileerimiseta simulatsioon) pinnakihi soolsus on Soome lahe keskosas kuni ühiku võrra ülehinnatud. See erinevus on vaatluste assimilatsiooni käigus oluliselt vähenenud. Vähenenud on ka soolsuse alahindamise viga vahetult Soome ranniku ääres. Korrektsiooni käigus ei ole ilmnenud süstemaatilisi vastupidise märgiga vigu, mis tähendab, et EOF mustrid on tabanud vaatluste ruumilisi mustreid küllaltki realistlikult.

Ligikaudu paari nädalaga väheneb erinevus kogu FerryBox trassi ulatuses peaaegu olematuks. Novembri alguses ilmnevad ka mõned alahinnatud soolsuse väärtused. Novembris erinevad prognoosväljad üksnes Soome kalda ääres. Ka märtsi keskpaigas on Soome kalda ääres kahenädalane periood, mil soolsust on mudel alahinnanud. Soome lahe lääneosas (59. laiuskraadil) on näha üksikuid perioode (novembri algus, veebruari keskpaik), kus mudel on alahinnanud sissevoolanud veemasside soolsust.

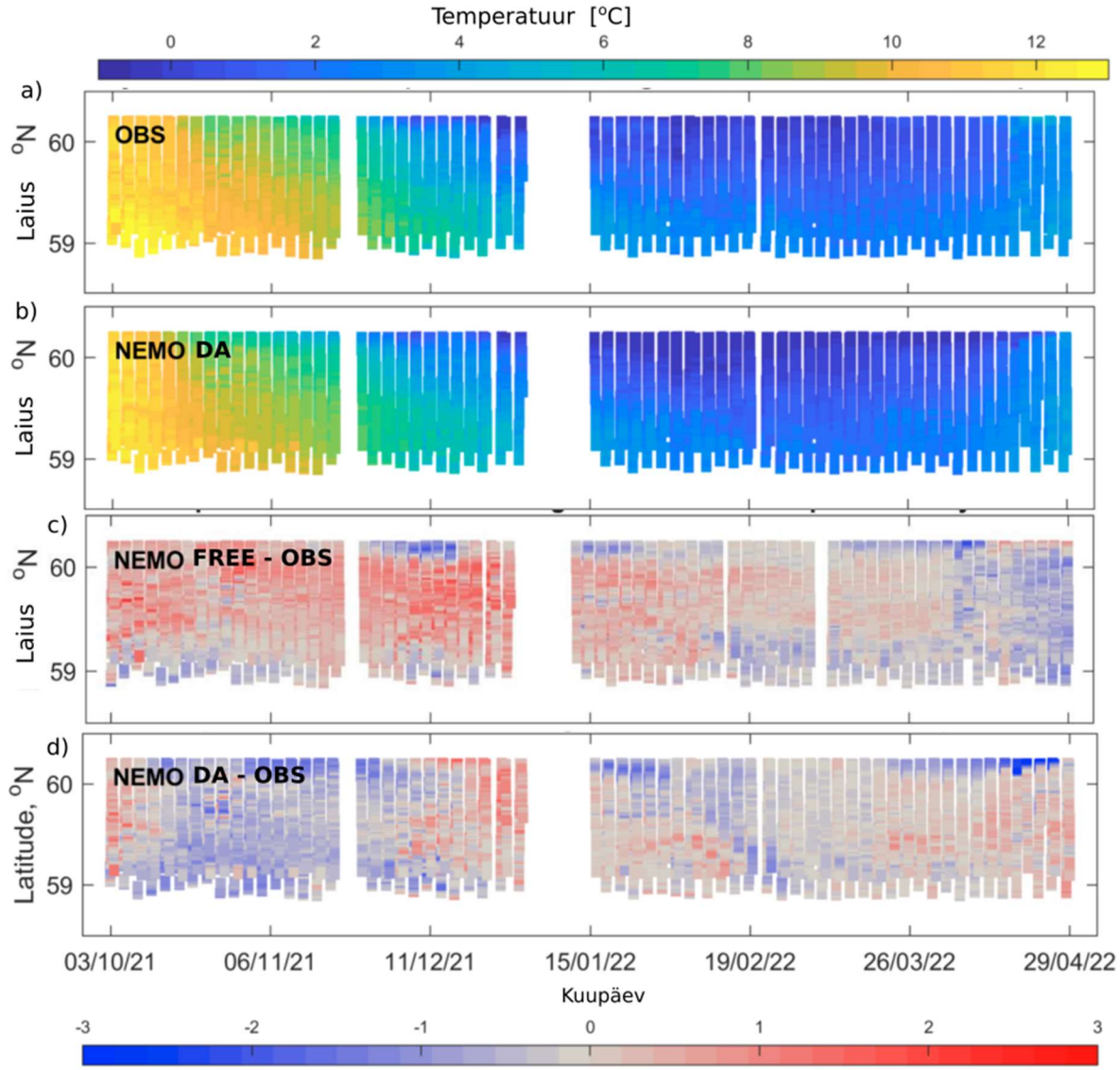
Temperatuuri ruumilistelt aegridadelt näeme, et vabajooksus on pinnakihi temperatuur sisuliselt märtsi lõpuni ülehinnatud. Eriti suur on viga novembris/detsembris, kui temperatuuri erinevus Soome lahe avaosas võrreldes vaatlustega on pea 2 kraadi. Samuti on vabajooksus aprillis temperatuur pea ühiku võrra alahinnatud. Vaatluste assimilatsiooni käigus toimub pinnakihi temperatuuri vigade kõrvaldamine juba kahe nädalaga, peale mida esineb prognoosis vaatlustest külmemaid tingimusi. Detsembri ja novembri (aga ka jaanuari ja veebruari) vabajooksus esinenud kõrgema temperatuuriga kõrvalekalded on viidud assimilatsiooni käigus nulli lähedale. Ka aprillis olnud püsivad temperatuuri alahindamised prognoosis on avamerelt eemaldatud. Vahetult Soome ranniku lähedal on temperatuuri

erinevus vaatlustega võrreldes suurenenud, mis võib olla põhjustatud muutunud jää kogusest ning seotud aeglasema jää taandumisega.

Olulisemad muutused on toimunud avamere parameetrite kvaliteedis, mis on avatud radadest eemal. Avatud radadest kaasnevad ka välise mudeli (Copernicus prognoosmudeli) kõrvalekalded, mis kohanevad vastavalt käesoleva mudeli assimileerimise algoritmidele. Vaatluste assimileerimine on nii temperatuuri kui ka soolsuse väljadest kõrvaldanud püsivad samamärgilised kõrvalekalded Soome lahe avaosas. Selliste vigade efektiivne parandamine tagab mudeli soolsuse ja temperatuuriväljade kvaliteedi ka juhul, kui mudelil (või sisendandmetes) hakkab esinema süstemaatilisi vigasid. Selleks, et võimalikele vigadele jälile jõuda, oleks hea kui paari aasta jooksul oleks paralleelselt assimileeritud simulatsiooniga võrdluseks kaasas kas prognoos, kuhu vaatluseid ei assimileerimata (e. vabajooks).



Joonis 2.3.2. Pinnakihi soolsuse aegrida parvlaev FinnMaid trassil (Joonis 2.3.1). a) paneelil on toodud vaatlusandmed, b) paneelil aktiivse assimileerimisega andmed, c) mudeli ja mõõtmiste erinevus vabajooksu (ilma assimileerimiseta) korral ja d) erinevused mudeli ja mõõtmiste vahel assimileeritud simulatsiooni korral.



Joonis 2.3.3. Pinnakihi temperatuuri aegrida parvlaev FinnMaid trassil (Joonis 2.3.1). Erinevate paneelide kirjeldused vastavad joonisele 2.3.2.

3. Prognoosisüsteemi tõrked

Järgnevalt on kirjeldatud mõned vead, mis on pre-operatiivses faasis esinenud, mille tagajärjel on prognoos ja selle järjepidevus katkenud. Iga tõrke juures on kirjeldatud ka võimalik lahendus. Esinenud vead jagunevad tehnilisteks ja numbrilisteks.

3.1 Tehnilised vead

Kui prognoos ei ole õigeks ajaks valmis saanud, siis kõige tõenäolisemaks põhjuseks on olukord, kus vajalikke arvutusressursse ei jätku. Kõik toimingud, mis on seotud sisendandmete kogumisega, ei ole eriti ressursimahukad ning neid tehakse juhtmasinal. NEMO mudeli käivitatakse aga läbi järjekorra halduri (PBS) paljudel arvutusmasinatel. Kui vajalikku kogust ressursse ei ole, siis läheb protsess järjekorda ootele, kuni vajalik arv (179) arvutustuumi vabaneb. Kui ooteaeg ületab 3 tundi, siis r**un_model_op.sh** skript katkestab töö ning prognoosisüsteem peatub veateatega. Skript ei eemalda tööd järjekorrast, seega kui ressursid vabanevad peale kolme tundi, siis NEMO ja XIOS protsessid jätkavad oma tööd ning tekitavad väljundi ja logi-failid prognooskausta.

Sarnane tõrge võib esineda ka siis, kui mingil põhjusel arvutusteks määratud arvutussõlm langeb mõne tehnilise rikke tõttu kasutusest ära. Operatiivseid simulatsioone ei tohiks sestap siduda kindlate sõlmedega, vaid tuumade jagamine jätta järjekorrasüsteemi (PBS) kanda.

Lahend: Kui tehniline tõrge on kõrvaldatud, siis et väljund ja taaskäivitusfailid jõuaksid kasutaja poolt defineeritud kohtadesse (op_output/ , forcing/), oleks kõige lihtsam operatiivse prognoosi peaskript (**do_op_update**) peale vabade ressursside ilmnemist uuesti käivitada.

Puuduvad / hilinevad sisendandmed on teiseks tehniliseks prognoosi katkemise põhjuseks. Operatiivne mudelsüsteem eeldab, et prognoosi alguseks on süsteemi allalaetud ECMWF'i meteo ja Copernicuse operatiivse mudeli rajatingimused. Kui neid mingil põhjusel ei ole, siis skriptidega seda kontrollitakse ning oodatakse (meteo 45 min, rajad 30 min). Kui ka siis ei ole andmeid saabunud, jäävad NEMO mudeli sisendandmed genereerimata. Kui NEMO käivitub ning vajalikke sisendandmeid ei leita, siis simulatsioon katkeb ning sellekohane kirje ilmub NEMO logisse *ocean.output.*

Lahend: Kui sisendid siiski tekivad (kasutaja peab need hiljem uuesti alla laadima), tuleks operatiivne prognoos taaskäivitada.

3.2 Mudeli sisemised vead

Prognoosides võivad esineda äärmuslikud olukorrad, mida varem pole esinenud, ning mis võivad endaga kaasa tuua numbrilisi ebastabiilsusi. Tingimused ei pruugigi olla seotud mõne tormiga - ebastabiilsused võivad esineda ka vaiksete ilmastikutingimuste korral. Testperioodil on seadistatud mudeli difusiivsust ja ajasammu nii, et mudelis ebastabiilsusi ei esineks. Samas täielikult välistada ebastabiilseid olukordi ei saa. Võib esineda juhtumeid, kus prognoosisimulatsioon katkeb keset simulatsiooni, kui NEMO sisemine kontroll tuvastab parameetrites väärtusi, mis on väljaspool etteseatud realistlikest piiridest. Kui ilmnevad ebarealistlikud väärtused, lõpeb mudeli töö ning veateade koos kirjeldusega kirjutatakse NEMO logisse *ocean.output*.

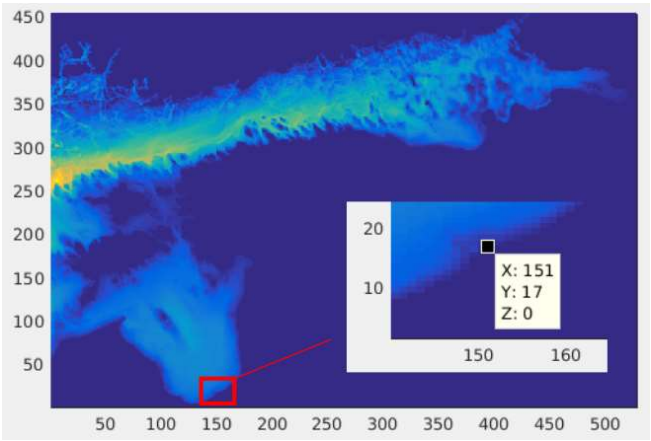
Põhjuseid, miks mudel on ebastabiilne, võib olla mitmeid erinevaid ning johtuvalt on ka nende mõistmine ja kõrvaldamine erinev. Sagedaseks põhjuseks on dünaamiliste sisendite (avatud rajad, jõed, meteoroloogia) füüsikaline vastuolu mudeli sisemise dünaamikaga. Üldjuhul on selliste vigade vältimiseks vajalik ajasammu vähendamine ning mudeli uuesti käivitamine. Kui aga ka see ei kõrvalda viga, on ajutiseks lahenduseks meteo-, jõe- või radade mõju eemaldamine üheks prognoosiks. Enamus tehnilisi vigu, mis on tingitud negatiivse soolsuse ilmnemisel seoses kompilaatori iseärasustega (vt NEMO juhend pt 14.4.1), on kõrvaldatud kompileerimisel (*key_nosignedzero*). Järgnevalt on toodud mõned näited ja lahendid ebastabiilsustest, mis esinesid preoperatiivses faasis.

Ebarealistlikud / absurdsed väärtused mudeli väljades.
Veateade **ocean.output** logis ===>>> : E R R O R

```
      ===========
 STOP
  stp_ctl: |ssh| > 20 m  or  |U| > 10 m/s  or  S <= 0  or  S >= 100  or  NaN en
 counter in the tests
  E R R O R message from sub-domain:          3
  kt=    903   |ssh| max:   20.68    , at  i j :   151   17
  kt=    903   |U|   max:   14.30    , at  i j k:   151   17    1
  kt=    903   S     min:  0.1000    , at  i j k:   152   17    1
  kt=    903   S     max:   11.43    , at  i j k:   151   18    5
      ===> output of last computed fields in output.abort.nc file
 huge E-R-R-O-R : immediate stop
```

kt ajasammul (903) on mudeli väljades ilmnenud ebarealistlik(ud) väärtus(ed). Antud näites on ilmnenud veetaseme absoluutväärtustes 20 m ületavaid veetasemeid ning hoovuse kiiruseid, mis ületavad 14 m/s. Viimasest ajasammust pärinevad väljad on salvestatud 3-ndasse output.abort.nc faili ehk faili **output.abort_0002.nc** (domeenide loendus algab indeksist 0).

Kasutades veateates viidatud indekseid (i,j,k), on võimalik leida ruumis üles vea allikas. Käesolevas näites peitub vea-allikas Daugava jõe suudmes (i=151, j=17) ning anomaalsed kiirused viitavad pinnakihile (k=1)



NB: NEMO automaatne vigade kontroll, mis katkestab simulatsiooni, ei kontrolli realistlikke temperatuurivahemikke, mistõttu võivad need avalduda temperatuurist sõltuvate parameetrite väljades (hoovuse kiirus, veetase, bioloogia väljad). Need piirid on defineeritud

NEMO koodis ning on mõeldud tuvastamaks suuri amplituudiga ebastabiilsusi, mitte väikseid numbrilisi vigu (näiteks numbrilisi malelaua mustreid).

Lahend: Konkreetne juhtum oli tingitud liigsoojast jõevee klimatoloogilisest temperatuurist, mis sattus vastuollu detsembri alguses toimunud külmalainega. Selle vea tarbeks on prognoosisüsteemis arvutatud jõevee temperatuur eelmiste päevade meteoprognoosidest. Sarnaste jõgedega esinevate vigade korral võib viga kõrvaldada, püüdes a) vähendada ajasammu dt=180, või 150; b) lülitada välja jõe temperatuur nimistu failis või c) lülitada üheks prognoostsükliks jõgede temperatuur sootuks välja.

Kui veateade viitab asukohale i = 1 ... 5 (ja k = 56, 54), siis on põhjuseks mudeli vastuolud avatud radadega, mis tekitavad sügavat konvektsiooni. Siin võivad põhjused peituda välismõjude füüsikaliste tingimuste erinevused. Kui näiteks surutakse radadest soojemat / magedat vett kui mudelis, on tulemuseks konvektsioon (e. veesamba kiire segunemine), mis mudeli sisemise dünaamika järgi ei peaks olema.

Välistada ei saa ka vigaseid meteoroloogia sisendeid, kus kriitilist ebastabiilsust võivad tingida tehnilised vead sisendandmete konverteerimisel.

```
STOP
  stp_ctl: |ssh| > 20 m  or  |U| > 10 m/s  or  S <= 0  or  S >= 100  or  NaN encounter in the tests
 E R R O R message from sub-domain:          90
 kt=   4782   |ssh| max:        NaN, at  i j  :   298  314
 kt=   4782   |U|   max:   0.000    , at  i j k:   298  314  110
 kt=   4782   S     min:        NaN, at  i j k:   298  314    1
 kt=   4782   S     max:        NaN, at  i j k:   298  314    1
      ===> output of last computed fields in output.abort.nc file
===>>> : E R R O R
       ===========
STOP
  stp_ctl: |ssh| > 20 m  or  |U| > 10 m/s  or  S <= 0  or  S >= 100  or  NaN encounter in the tests
 E R R O R message from sub-domain:          99
 kt=   4782   |ssh| max:        NaN, at  i j  :   166  334
 kt=   4782   |U|   max:   0.000    , at  i j k:   166  334  110
 kt=   4782   S     min:        NaN, at  i j k:   166  334    1
 kt=   4782   S     max:        NaN, at  i j k:   166  334    1
      ===> output of last computed fields in output.abort.nc file
```

Lahendused on sarnased jõgede ebastabiilsusega: a) vähendada ajasammu; b) lülitada ajutiselt välja temperatuuri rajatingimus või c) kogu radade mõju. Viimase kahe lahenduse korral on vältimatud veetasemete vead 1-3 päeva ulatuses, sestap tuleks neid kasutada viimasena.

Kuna mudel on võimeline lugema üle-eelmise päeva taaskäivitusfaile, siis ei ole ühe prognoosi vahele jätmine probleemiks ning prognoos võib jätkuda tõrgeteta.

Kui prognoosisüsteemis on esinenud tõrkeid rohkem kui üks päev, siis:

a) esinenud tõrke parandamine ning teha järel-arvutus nii, et tekiks vajalik järjepidevus taaskäivitusfailidega.

b) kui aeg venib pikemaks ning järelarvutus ei ole võimalik, tuleks mudel lähtestada uuesti Copernicuse prognoosist (tähtis on ühtsus rajatingimustega) või klimatoloogilistest väljadest (vt. andmete assimileerimine). Klimatoloogilised väljad saab genereerida, kui teha EOF rekonstruktsioon null-amplituutidega.

4. Kokkuvõte

Käesoleva töö raames on KAURi Ilmateenistuse kobararvutisse seadistatud operatiivne mere prognoosisüsteem, mis teostab igapäevaselt prognoosarvutusi, kasutades NEMO hüdrodünaamika mudelit. Andmete ettevalmistamine kätkeb endas sisendandmete konverteerimist NEMO mudelile sobivale kujule ning assimileerimiseks vajalike vaatlusandmete allalaadimist. Lisaks mudeli käitamisele teostab süsteem esmast prognoosandmete haldust.

Prognoosisüsteemi on kasutatud edukalt preoperatiivses režiimis alates 2021. oktoobrist, mil mudeli süsteem lähtestati Copernicuse operatiivprognoosist. Lisaks vabajooksul põhinevale prognoosile on preoperatiivse perioodi jooksul prognoosisüsteemi lisatud ka vaatlusandmete assimileerimise võimekus, mida on pre-operatiivse perioodi jaoks ka edukalt rakendatud.

4. märtsil 2022 toimunud kasutajakoolituse raames tutvustati kolmele KAURi spetsialistile (Ivar Ansper, Rain Elken, Sander Pukk) prognoosisüsteemi toimimist ja kasutamist. Läbiviidud lühikursuse raames installeerisid osalejad KAURi klastris NEMO mudelit ja seadistasid operatiivprognoosiks vajalikud lähteandmed/-failid. Lepiti kokku, et jooksev ja täiendav konsultatsioon toimub vastavalt vajadusele. Täiendavate küsimuste korral tuleks ühendust võtta aadressil ilja.maljutenko@taltech.ee .

Kasutatud kirjandus

Aksenov, Y., Blockley, E., Chevallier, M., Feltham, D., Fichefet, T., Garric, G., Holland, P., Iovino, D., Madec, G., Massonnet, F., Ridley, J., Rousset, C., Salas, D., Schroeder, D., Tietsche, S., Vancoppenolle, M., 2019, "Sea Ice modelling Integrated Initiative (SI3) – The NEMO sea ice engine", Scientific Notes of Climate Modelling Center, 31 — ISSN 1288-1619, Institut Pierre-Simon Laplace (IPSL), doi:10.5281/zenodo.1471689

Elken, J., Maljutenko, I., Lagemaa, P. & Verjovkina., S., 2021. Mere operatiivmudelisüsteemi NEMO kasutuselevõtt ja töölerakendamine mereala operatiivprognooside parandamiseks. II etapp: NEMO käivitamine pre-operatiivses režiimis, analüüsi- ja prognoosiväljade tootmine. TTÜ Meresüsteemide Instituut, 62 lk.

Elken, J., Maljutenko, I. and Lagemaa, P., 2020. Mere operatiivmudelisüsteemi NEMO kasutuselevõtt ja töölerakendamine mereala operatiivprognooside parandamiseks. I etapp: NEMO esmane seadistamine, kalibreerimine ja valideerimine. TTÜ Meresüsteemide Instituut, 82 lk.

Elken, J., Zujev, M., She, J. and Lagemaa, P., 2019. Reconstruction of large-scale sea surface temperature and salinity fields using sub-regional EOF patterns from models. Frontiers in Earth Science, 7, p.232.

Meurdesoif, Y., Caubel, A., Lacroix, R., Dérouillat, J. and Nguyen, M., 2016. XIOS tutorial. https://forge.ipsl.jussieu.fr/ioserver/raw-attachment/wiki/WikiStart/XIOS-tutorial.pdf (2022.05.20)

Gurvan, M., Bourdallé-Badie, R., Bouttier, P.-A., Bricaud, C., Bruciaferri, D., Calvert, D., Chanut, J., Clementi, E., Coward, A., Delrosso, D., Ethé, C., Flavoni, S., Graham, T., Harle, J., Iovino, D., Lea, D., Lévy, C., Lovato, T., Martin, N., Masson, S., Mocavero, S., Paul, J., Rousset, C., Storkey, D., Storto, A., Vancoppenolle, M., 2017. NEMO ocean engine. https://doi.org/10.5281/ZENODO.1472492

Hordoir, R., Axell, L., Höglund, A., Dieterich, C., Fransner, F., Gröger, M., Liu, Y., Pemberton, P., Schimanke, S., Andersson, H. and Ljungemyr, P., 2019. Nemo-Nordic 1.0: a NEMO-based ocean model for the Baltic and North seas–research and operational applications. Geoscientific Model Development, 12(1), pp.363-386.

Kärnä, T., Ljungemyr, P., Falahat, S., Ringgaard, I., Axell, L., Korabel, V., Murawski, J., Maljutenko, I., Lindenthal, A., Jandt-Scheelke, S., Verjovkina, S., Lorkowski, I., Lagemaa, P., She, J., Tuomi, L., Nord, A. and Huess, V., 2021. Nemo-Nordic 2.0: operational marine forecast model for the Baltic Sea. Geoscientific Model Development, 14(9), pp.5731-5749.

Lagemaa, P. 2015. Operatiivne meremudel HBM-EST. Tehniline juhend. TTÜ Meresüsteemide Instituut, KIK projekti nr 5484 aruanne, 24 lk.

Lorkowski, I., Schwichtenberg, F., Lindenthal, A. and Maljutenko, I., 2021, May. New developments of the operational biogeochemical model component in the Copernicus Marine Service (CMEMS) for the Baltic Sea. 9th EuroGOOS International conference, Shom; Ifremer; EuroGOOS AISBL, May 2021, Brest, France. pp.296-301. ffhal03335259v2f.

Madec, G., Delecluse, P., Imbard, M. and Levy, C., 1998. OPA 8.1. Ocean general circulation model reference manual. Note du Pôle de modélisation, Laboratoire d'Océanographie DYnamique et de Climatologie, 91 pp.

Vankevich, R.E., Sofina, E.V., Eremina, T.E., Ryabchenko, V.A., Molchanov, M.S. and Isaev, A.V., 2016. Effects of lateral processes on the seasonal water stratification of the Gulf of Finland: 3-D NEMO-based model study. Ocean Sci, 12, pp.987-1001.

Westerlund, A. and Tuomi, L., 2016. Vertical temperature dynamics in the Northern Baltic Sea based on 3D modelling and data from shallow-water Argo floats. Journal of Marine Systems, 158, pp.34-44.

Westerlund, A., Tuomi, L., Alenius, P., Miettunen, E. and Vankevich, R.E., 2018. Attributing mean circulation patterns to physical phenomena in the Gulf of Finland. Oceanologia, 60(1), pp.16-31.

Westerlund, A., Tuomi, L., Alenius, P., Myrberg, K., Miettunen, E., Vankevich, R.E. and Hordoir, R., 2019. Circulation patterns in the Gulf of Finland from daily to seasonal timescales. Tellus A: Dynamic Meteorology and Oceanography, 71(1), p.1627149.

Zujev, M., Elken, J. and Lagemaa, P., 2021. Data assimilation of sea surface temperature and salinity using basin-scale reconstruction from empirical orthogonal functions: a feasibility study in the northeastern Baltic Sea. Ocean Science, 17(1), pp.91-109.

Lisad

Lisa 1 Operatiivse süsteemi peakataloogi alamstruktuur

```
[iljama@lohe1 nemo_op_sys]$ tree -R -L 2  forcing/
forcing/
├── bathy_meter.nc
├── boundary/
│   ├── cmems_nrt/
│   ├── cmems_nrt_bc_V110/
│   └── gridfile_bdy_est05
├── dataassim/
│   ├── DA3D/
│   ├── DA_Config.nml
│   ├── DA_RunConfig.nml
│   ├── DA_RunConfig.nml_template
│   ├── DEP/
│   ├── do_check_ampl.m
│   ├── obs/
│   │      ├── eigenvalues_coarse_S3D_SUR_DA.txt
│   │      ├── eigenvalues_coarse_T3D_SUR_DA.txt
│   │      ├── grid_coarse_SUR_DA.txt
│   │      ├── modes_coarse_S3D_SUR_DA.txt
│   │      ├── modes_coarse_T3D_SUR_DA.txt
│   │      ├── seasonal_coarse_S3D_SUR_DA.txt
│   │      └── seasonal_coarse_T3D_SUR_DA.txt
│   ├── ST_coarse_clim_SUR.nc
│   ├── ST_coarse_modevars_SUR.nc
│   ├── ST_coarse_SUR_mergedCSmxa.nc
│   ├── ST_coarse_SUR_Smxa.nc
│   ├── ST_merge.nc
│   ├── ST_SUR_coarse_R.nc
│   ├── ST_SUR_fine_R.nc
│   └── SUR/
├── domain_cfg_EST_0.5nm_V110.nc
├── hotstart/
│   ├── EST05nm_op_run1/
│   └── EST05nm_op_run2/
├── meteo
│   └── meteo_nemo_ecmwf_BAL/
├── rean_nrt_V110.20210924.nc
└── runoff/
    ├── runoff_q_seasonal/
    └── runoff_t_atmt2/
```

```
[iljama@lohe1 nemo_op_sys]$ tree -R -L 2  main_scripts/
main_scripts/
├── bin/
│   ├── cdo
│   ├── eof_reconstr_v1
│   └── ncpdq
├── ChangeTime
├── ChangeTime.f90
├── cmod_date.f90
├── cmod_date.mod
├── cmod_date.o
├── do_boundary_cmemsnrt.sh
├── do_clean_rest_op
├── do_compile
├── do_cprest_op
├── do_cycle_yymmdd
├── do_dataassim.sh
├── do_dataassim_SUR.sh
├── do_meteo_ecmwf.sh
├── do_op_update
├── do_op_update_DA
├── do_runoff.sh
├── do_sendEmail
├── eof_reconstr_v1
├── eof_reconstr_v1.f90
├── include/
├── init_user_vars.sh
├── lib/
│   └── libnco-4.9.5.so
├── redo_meteo_ecmwf.sh
├── run_model_op_DA.sh
└── run_model_op.sh
```

```
[iljama@lohe1 nemo_op_sys]$ tree -R -L 2  setup/
setup/
├── bfr_roughness.nc
├── context_nemo.xml
├── coordinates.bdy.nc
├── do_cprest_op
├── do_extr_output_op
├── do_linkrest
├── do_linkrest_op
├── domain_def_nemo.xml
├── do_oceanlogERR
├── field_def_nemo-ergom.xml
├── field_def_nemo-ice.xml
├── field_def_nemo-oce.xml
├── file_def_nemo-ergom.xml
├── file_def_nemo-ice.xml
├── file_def_nemo-oce_oil.xml
├── file_def_nemo-oce.xml
├── grid_def_nemo-ergom2d.xml
├── grid_def_nemo.xml
├── iodef.xml
├── job_nemo.sh
├── namelist_cfg
├── namelist_ice_cfg
├── namelist_ice_ref_template
├── namelist_ref_template_V110_op
├── nemo.intel.exe
└── xios_server.intel.exe
```

```
[iljama@lohe1 nemo_op_sys]$ tree -R -L 2  setup_DA/
setup_DA/
├── assim_work/
│   ├── do_check_assim.m
│   ├── do_compile
│   ├── edit_netcdf
│   ├── edit_netcdf.f90
│   ├── edit_netcdf_SUR
│   └── edit_netcdf_SUR.f90
└── do_testassim_op
```

```
[iljama@lohe1 nemo_op_sys]$ tree -R -L 2  op_output/
op_output/
├── EST05nm_op_run1/
│   ├── dailymean
│   ├── stations
│   └── surface
└── EST05nm_op_run2/
    ├── dailymean
    ├── stations
    └── surface
```

```
[iljama@lohe1 nemo_op_sys]$ tree -R -L 2 -d   model/
model/
├── extractpoint
│   └── extract_point
├── nemo4_dev
│   ├── arch/
│   ├── cfgs/
│   ├── ext/
│   ├── src/
│   ├── tools/
│   └── ...
└── xios-2.5
    ├── arch/
    ├── bin/
    ├── src/
    └...
```

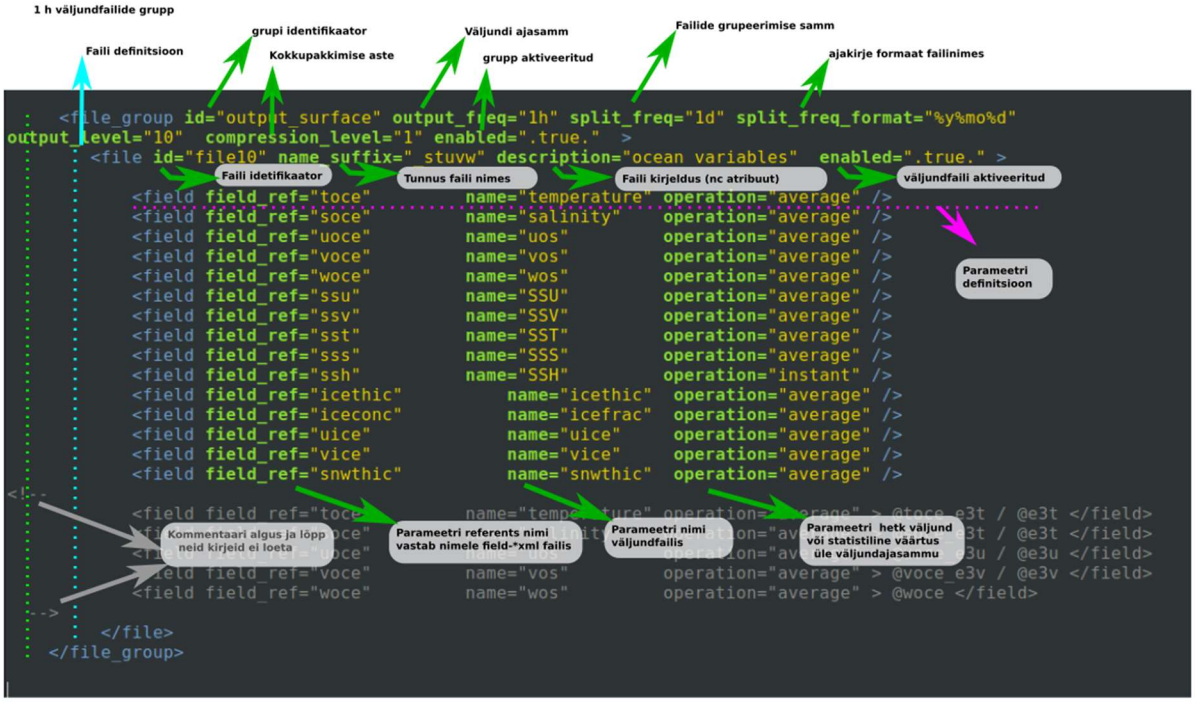
Lisa 2 Algväljade interpoleerimine

Järgnevaid käske saab kasutada NEMO sisend väljade genereerimiseks NetCDF failidest, milles olevad pikkus/laius võrk vastab CDF standarditele. Sarnaselt saab genereerida ka toitainet väljad TOP (ERGOM) mudelile. Näitena on kasutatud Copernicuse prognoosproduktisti välju, mis on käsureal allalaetud CMEMS FTP serverist, kasutades wget käsku. Kasutaja peab sisestama oma CMEMS kasutajanime ja salasõna (cmems_user, cmems_pasw).

```
# Copernicuse prognoosproduktist (BALTICSEA_ANALYSISFORECAST_PHY_003_006) päevakeskmiste väljade allalaadimine FTP serverist
yy=2021
mm=09
dd=24
tstr=$yy$mm$dd
idir="ftp://nrt.cmems-du.eu/Core/BALTICSEA_ANALYSISFORECAST_PHY_003_006/dataset-bal-analysis-forecast-phy-dailymeans/$yy/$mm/"
filename="BAL-NEMO_PHY-DailyMeans-$tstr.nc"
cmems_user=
cmems_pasw=
wget --user=$cmems_user --password=$cmems_pasw $idir/$filename

# Identifikaator
str='V110'
# NEMO T-sügavused . Pinnal ja põhjas on sügavused seatud nii, et tühjad võrgupunktid (0.5, 125, .. jne ) saaksid lähima väärtuse.
Zax="1.51,1.52,2.5,3.5,4.5,5.5,6.5,7.5,8.5,9.5,10.5,11.5,12.5,13.5,14.5,15.5,16.5,17.5,18.5,19.5,20.5,21.5,22.5,23.5,24.5,25.5,26.5,27.5,28.5,29.5,30.5,31.5,32.5,33.5,34.5,35.5,36.5,37.5,38.5,39.5,40.5,41.5,42.5,43.5,44.5,45.5,46.5,47.5,48.5,49.5,50.5,51.5,52.5,53.5,54.5,55.5,56.5,57.5,58.5,59.5,60.5,61.5,62.5,63.5,64.5,65.5,66.5,67.5,68.5,69.5,70.5,71.5,72.5,73.5,74.5,75.5,76.5,77.5,78.5,79.5,80.5,81.5,82.5,83.5,84.5,85.5,86.5,87.5,88.5,89.5,90.5,91.5,92.5,93.5,94.5,95.5,96.6,97.8,99.5,102,105.4,109.3,113.4,117.5,121.1,121.2,121.2,121.3,121.4,121.5"

# Sisendfail. Kopeeritud Copernicuse prognoos produktist

ifile="BAL-NEMO_PHY-DailyMeans-$tstr.nc"
ofile="phy_nrt_$str.20210924.nc"

 cdo -s -O -L  --reduce_dim intlevel,$zax -setmisstonn -remapbil,bathy_meter.nc -expr,"so=so;thetao=thetao" $ifile $ofile

# expr,"so=so;thetao=thetao"  :  valib sisendfailist välja soolsuse ja temperatuuri
# remapbil,bathy_meter.nc      :  lineaarne interpolatsioon bathy_meter.nc olevale võrgule
# setmisstonn    :  seab numbrita punktidele horisontaalselt lähimad väärtused
#  intlevel,$zax    :  interpoleerib vertikaalselt eelpool defineeritud võrgule zax=
#   --reduce_dim   :  kustutab dimensioonid  pikkusega 1
# -L:   võimaldab pipe-funktisooni (mitu operaatorit püksteise järel), -O : ülekirjutus, -s: vaikselt.
```

Lisa 3 NEMO väljundite deklareerimine

Alloleval skeemil on toodud 1. 24-tunni komposiit väljund EST05nm_op_run1_1h_stuvw väljundfaili (NetCDF päis esitatud Elken et al. 2021 Lisas 9) defineerimise skeem koos kirjeldustega. Parameetrite nimetused on defineeritud failis `field_def_nemo-oce.xml`.



Lisa 4 Assimileerimise metodoloogia ja rakendus (inglise keeles)

By Jüri Elken, Ilja Maljutenko and Priidik Lagemaa

Notes on data assimilation in NEMO-Est using EOF method

1. The method

We use the relaxation method for the data assimilation (DA) of observed variables into the oceanographic model results, in order to make the model results better matching the observations. In this approach, observations reconstructed to the model grid are gradually merged with model output. Zujev et al. (2021) tested the values of coefficients 0.1 and 0.2 for the reconstruction and 0.9 and 0.8 for the model, which gave good results while the results were in most cases closer to the observations with the coefficient value of 0.2.

For the reconstruction of gridded fields, we can use full knowledge of global (in the sub-region) covariance fields, obtained via dense datasets like the model data. Global variability patterns can be determined by the EOF/PCA method as a limited number of dominating modes. Observational data sets that contain gaps, can still be used to estimate the time-dependent amplitudes of dominating "full" EOF modes (spatial eigenvectors). By this method, we can reconstruct global gap-free patterns of observational data. Within the forecasting process, the model data can be corrected in the analysis step in relation to mismatch in the seasonal and shorter period large-scale responses. Remaining shorter scale processes (as deviations from the large-scale patterns) that have local correlation scales, can be further assimilated using optimal interpolation or its extended methods.

Following Elken et al. (2019), we use the sequence of modeled state vectors $\mathbf{x}(n)$ of length M (deviations at time n=1…N from temporal mean) that are decomposed by space-dependent empirical orthogonal mode vectors $\mathbf{f}_m$ and time-dependent amplitude vectors $\mathbf{a}_m(n)$, since the modeled value is equal to $\mathbf{x}(n) = \mathbf{a}_m(n)\,\mathbf{f}_m$ which a scalar product over all the EOF modes. The modes $\mathbf{f}_m$ are calculated as eigenvectors of the covariance matrix $\mathbf{D} = \mathbf{X}^T\mathbf{X}/N$ where $\mathbf{X}$ is the model data matrix containing N vectors of $\mathbf{x}(n)$. Observations $\mathbf{y}$ as a vector of length K are taken from different (fewer) locations $K < M$ than $\mathbf{x}$. Extracting the values of mode vectors at observation points, equal to $\mathbf{g}_m = \mathbf{A}\mathbf{f}_m$, where $\mathbf{A}$ is the spatial mask (observation operator) of the size M by K, the products $\mathbf{g}_i^T\mathbf{g}_j$ are only approximately orthogonal. Therefore, the standard EOF procedure to find the amplitudes cannot be used. Instead, we find the "observational" amplitudes $\mathbf{b}_i(n)$ by least-square minimization of analysis errors over the first L (most energetic) modes compared to the observation vector: $E = (\mathbf{y} - \mathbf{b}_i\mathbf{g}_i)^2 \rightarrow$ min, where the scalar product is taken over L modes. This results in the system of linear equations for finding the amplitude vector $\mathbf{b}$, namely $\mathbf{C}\mathbf{b} = \mathbf{h}$, where $\mathbf{C} = \{\mathbf{g}_i^T\mathbf{g}_i\}$ and $\mathbf{h} = \{\mathbf{y}\ \mathbf{g}_i\}$. The reconstructed field is then found from the "original" model-based EOF patterns $\mathbf{x}_b(n) = \mathbf{b}_m(n)\mathbf{f}_m$.

Practical reconstruction has to include observations in the wider interval $n_1 < n < n_2$ (observation window), otherwise there might be too few observations included in the calculation. When keeping the EOF amplitudes constant in the observation window, Elken et al. (2019) named this approach as time-fixed amplitudes. The method has been also extended to time-dependent amplitudes, where linear change of the amplitude $\boldsymbol{\alpha}_i$ within the observation window has been assumed $\mathbf{g}_i = \mathbf{g}_i + \boldsymbol{\alpha}_i\,\Delta t_p$ for each time interval Δt_p .

The field is reconstructed by superposition of continuous model-based mode patterns multiplied by observational amplitudes that meet adopted statistical limits. If the observational amplitude exceeds the limits, gridded fields for this and higher modes are not produced.

2. Basic implementation of EOF reconstruction

The source data are daily mean NEMO reanalysis data from 01.01.1993 to 31.12.2019, accessed from the CMEMS data portal. Each reanalysis file covers the region from 9.013755 to

30.235765 E by longitude and from 48.491699 to 65.891418 N by latitude with a grid of 383 by 523 points with a spacing of 0.05555 E and 0.03333 N (about 2 miles). The coordinates are given for the centers of the grid cells. The data have 56 vertical levels, with a grid step from about 3 m in the upper layers and 10 m at around 100 m depth.

The target area is from 18.40254974 to 30.0690994262695 E and from 56.7915649414063 to 60.7915000915527 N, covering the grid points from 170 to 382 by E and from 250 to 372 by N. This area extends to the west from the fine-resolution model grid, in order to allow inclusion of observational data e.g. from the Gotland Deep and the Northern Baltic Proper. Horizontal grid is compressed by 3 times using `RA_makemasks_3D_NE.m`, which results in a horizontal regional coarse grid of 71 by 41 points with a grid step about 6 miles. Depth levels are kept the same as in the source data, only the number of levels is truncated at the level 35. For the calculation of mask file, on each depth level, wet and dry points are counted in a 3 by 3 fine grid cell box around the coarse grid cell; if the number of dry points is less or equal 4 the point is taken as a wet point, otherwise a dry point. The mask value is adopted as the value of deepest wet level in the given coarse grid point. Dry (land) points are taken as NaN. In order to blank specific fragmented regions, NaN is assigned in the Archipelago Sea area for lonC(kx) < 21.56922 & latC(ky) > 60.0249 and lonC(kx) < 22.56922 & latC(ky) > 60.1249. Also, the wet points (34,28), (33,4) and (33,5) are cleared as the land. The results are saved to maskC_bal_ra_3D_NE.mat containing the values of longitude, latitude, depth and mask of each coarse grid cell.

Wet point data from the original 2D coarse grid mask file are then converted to the linear mask file `masklinC_bal_ra_3D_NE.mat`, where the index is counted over wet points first over longitude (inner cycle), then over latitude (outer cycle). The linear mask file contains data of 17468 wet points; namely, each line presents for a given wet grid cell the indexes over longitude, latitude and depth, and then their physical values, all sorted over depth, then longitude and then latitude.

Temperature and salinity data are aggregated on the regional coarse grid as single linear files `flin_bal_ra_T3D_NE.mat` and `flin_bal_ra_S3D_NE.mat` for the whole 28 years (9 861 days) using `flinC_bal_ra_3D_NE.m`. Horizontal averaging is applied over the wet points of 3 by 3 point boxes at each depth level. In MATLAB, the data have dimension 9 861x17 468.

Because of the pure observational data coverage in recent years due to the coronavirus situation, the whole domain of EOF calculation was also split into the surface layer (1134 model points) and deeper layer (6092 points from vertical index 13 to 35, depths from 40.0705 m to 260.612 m). The corresponding scripts and data files have the suffix SUR or DEP, for the surface or deeper options.

3. Seasonal background for statistics and EOF calculations

The EOF amplitudes calculated from the deviations from temporally mean values revealed high seasonal variability, especially for the temperature. Therefore, the mean seasonal signal was removed prior to the calculations.

Seasonal harmonics were calculated as the best fit (Elken et al., 2019) to the CMEMS reanalysis data 1993-2019. Altogether 4 sine and cosine functions were used, of which 9 coefficients were determined to minimize the RMSD (root-mean-square difference) between the original reanalysis data and the fitted curve.

The seasonal-mean method uses `Calc_EOF_seas_bal_ra_3D_NE.m` and estimates at each grid point *k* an individual seasonal cycle $s_k(t)$ by least-squares fitting the harmonic coefficients $C_{1,k}$ in the formula

$$s_k(t)=C_{1,k}\sin 2\pi t + C_{2,k}\cos 2\pi t + C_{3,k}\sin 4\pi t + C_{4,k}\cos 4\pi t + C_{5,k}\sin 6\pi t + C_{6,k}\cos 6\pi t + C_{7,k}\cos 8\pi t + C_{8,k}\sin 8\pi t + C_{9,k} \quad (1)$$

where times t are counted as fractions of decimal year. In the further procedures, deviations are calculated from the mean seasonal cycle, the rest of the EOF calculations is the same as for the constant-mean method. Example maps of seasonal surface temperature and salinity are presented in Figs. A4.1 and A4.2, respectively.

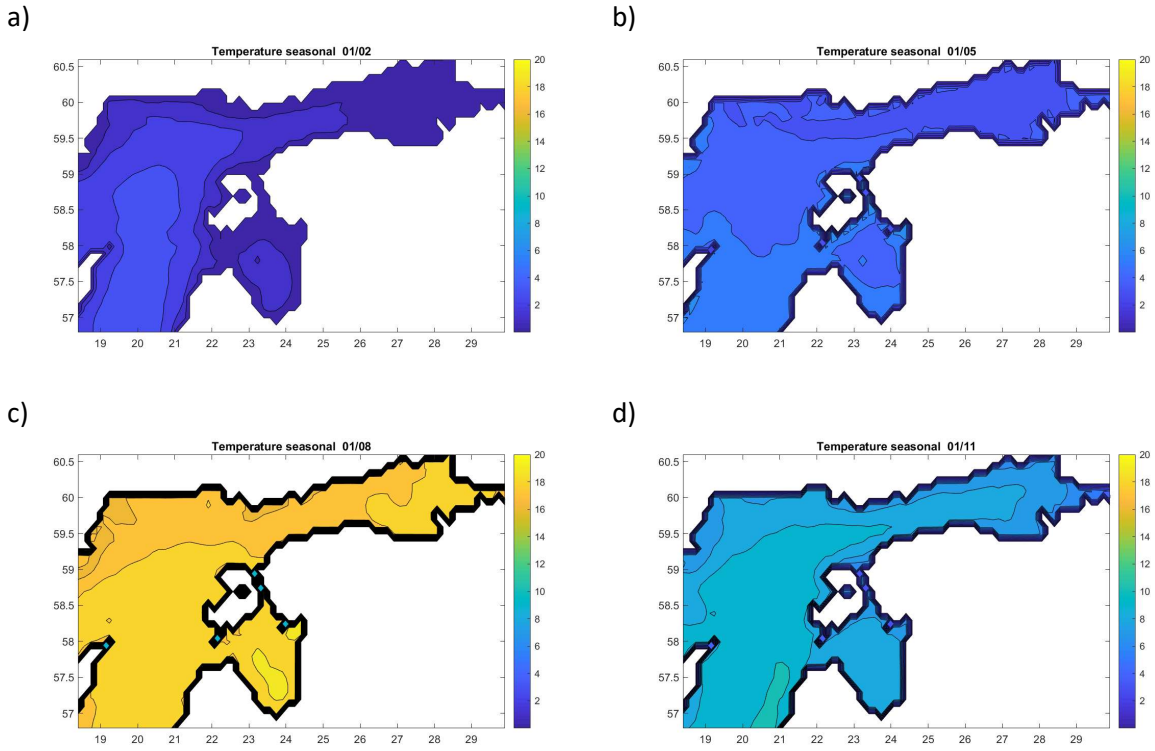


Figure A4.1. Maps of mean seasonal surface temperature on 1 February (a), 1 May (b), 1 August (c) and 1 November (d). Calculated using equation (1) from the CMEMS reanalysis data 1993-2019.

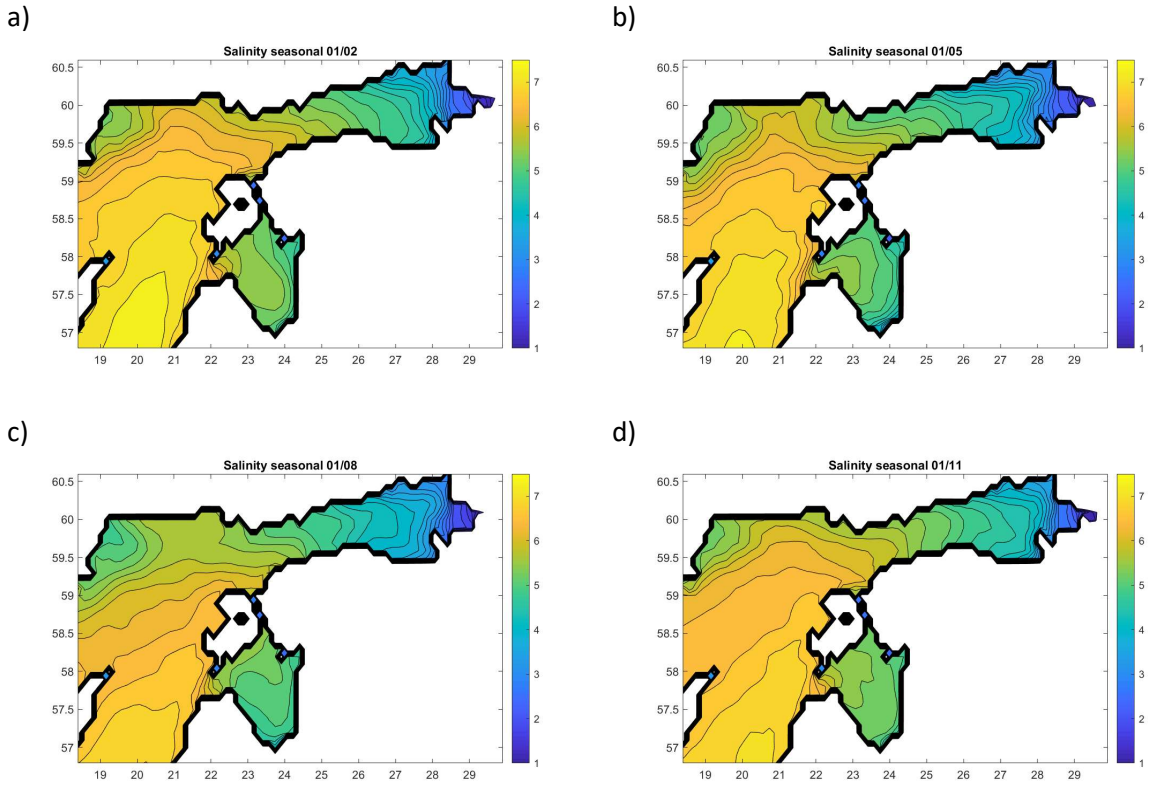


Figure A4.2. Same as Fig. A4.1, but for salinity.

4. Issues with observational data. Check and processing of EOF amplitudes

In the operational mode, initial DA run used the aggregated geographically bounded file `ts_obs_20210101-20220131.dat` that used `Extract_FB_CMEMS.m` to read and process monthly netcdf FerryBox (FB) files from 20150101 to date. Each monthly set of different FB lines is a direct download from CMEMS long time series FB product `/Core/INSITU_BAL_NRT_OBSERVATIONS_013_032/bal_multiparameter_nrt/monthly/FB`.

These raw FB data contained errors:

1) Sensors have been blocked over certain times, resulting in practically constant salinity and unrealistic temperature
2) In and near some harbors (Helsinki) observations have been taken from slow and/or standing ship, resulting in too small salinity and unrealistic temperature

The errors have been already corrected manually by visual check of salinity-longitude and temperature-longitude. Therefore new data were added to the already corrected earlier data. The updated dataset of observations is `ts_obs_20210101-20220430.dat`

Due to the gaps in data coverage, existing DA versions may generate unrealistic amplitude spikes. Otherwise the FORTRAN engine works well, therefore it would be better to keep the main code the same and adjust parameters if necessary, and/or process the DA output as EOF amplitudes.

Strong outliers of EOF amplitudes occurred when the number of “good” amplitudes was small. Handling of good amplitudes is defined by the selection of namelist parameter values of `eof_reconstr_v1a.f90`, shown later in section 6.

In the operational mode, the days with less than 3 good amplitudes were omitted. “Bad” amplitudes were replaced by the missing value codes. Further, the amplitudes were filtered by an 11-day moving average, excluding the operations with missing amplitudes.

Since in the operational mode there are only past observations and amplitudes available, an experiment was made where the reference time `tref` was changed to a shifted residence time `tref+5` days. Remember, that time-dependent EOF reconstruction allows specifying the reconstruction target time, based on the estimated linear rates of change of the amplitudes. Setting the target time 5 days forward allows us to correct the time shift introduced by the backward moving average filter. However, the shift of target time was not included in the main version of the reconstruction program, since we consider that poor data coverage is a temporary feature because of closing a number of ferry lines due to the coronavirus situation.

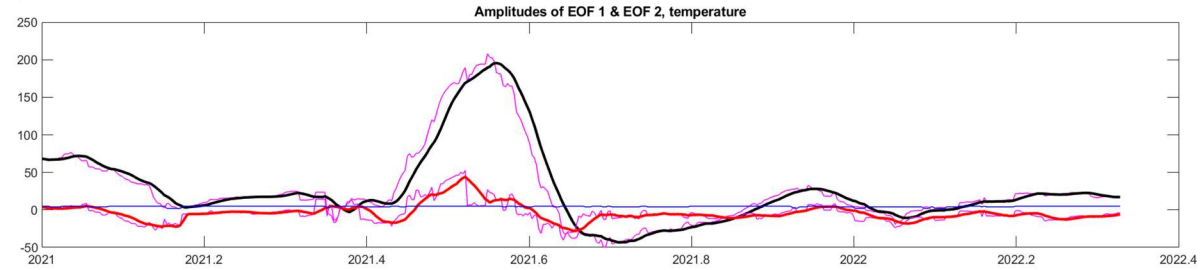
Calculation of EOF amplitudes as well as other daily output files were made using `Reconstr_FB_CMEMS.m` that is a script to prepare the observation files prior to call the compiled version of `eof_reconstr_v1a.f90`, and to perform the calculation for all the available observation files. Each observation file contains time, longitude, latitude, depth, temperature, salinity and name of observation platform for 30 days backward.

For the robustness of postprocessing of EOF amplitudes, this is done separately after the amplitudes are calculated. The post processing routine could include various filters. As a first step, only the backward moving average was applied in the shell script since during 2021-2022 there was a low number of observations. Missing values were omitted from the filtered data set. The processed amplitudes, with low-pass filter applied, are named `filt_s_ampl_20210101_SUR.txt` and `filt_t_ampl_20210101_SUR.txt`. The results shown in Fig. A4.3 reveals that noisy and spiky values of initial EOF amplitudes were effectively suppressed.

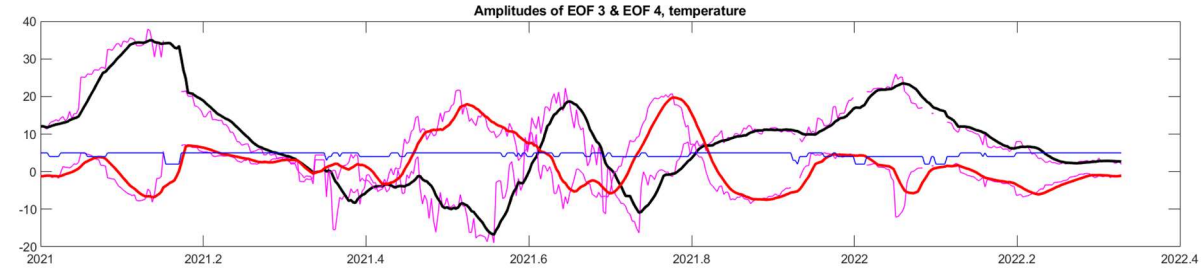
For the analysis of DA in the long-term hindcast mode, the observations were collected in 2009-2018 similar to the paper by Zujev et al. (2021). Reprocessing was done in order to include the westward-extended area of EOF calculation, as compared to the NEMO-Est model area. Because both the past and future observations are available for the particular hindcast time, a time-centered window of -20/+20 days was used in the time-dependent method for calculation of EOF amplitudes in the surface layer and -60/+60 days in the deeper layers below 40 m. Note that in the deeper layers only fragmentary vertical casting data were available, while the surface data also included dense FB observations. Experiments with shorter calculation windows revealed too spiky results. Visual inspection of outliers of the amplitudes identified six time intervals where manual deletion of spiky amplitudes were made. Further, the remaining amplitudes were filtered by an 11 days moving average as for the calculations in the operational mode, only the filter was applied as a time-centered filter.

Fig. A4.4 shows the time series of hindcasted EOF amplitudes for the sea surface temperature and salinity with a filter of 11 days. Amplitudes of deeper layers from 40 m to the bottom were filtered using the filter length of 61 days (Fig. A4.5).

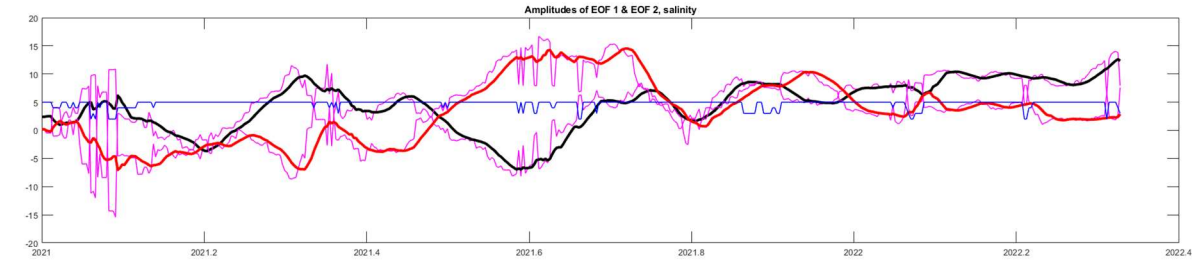
a)



b)



c)



d)

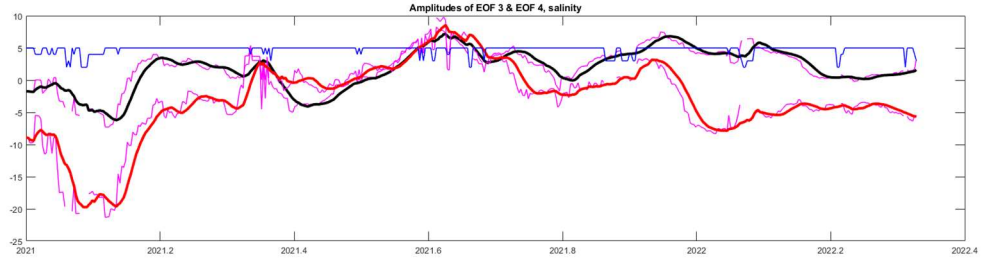


Figure A4.3. Reconstructed EOF amplitudes of surface temperature (a, b) and salinity (c, d) from the observations collected from 01.01.2021 to 30.04.2022. Shown are the filtered amplitudes of the mode 1 (black, a, c), mode 2 (red, a, c), mode 3 (black, b, d) and mode 4 (red, b, d). Unfiltered data are shown by thin magenta curves. Blue curve shows the number of "good" modes (default value is 5).

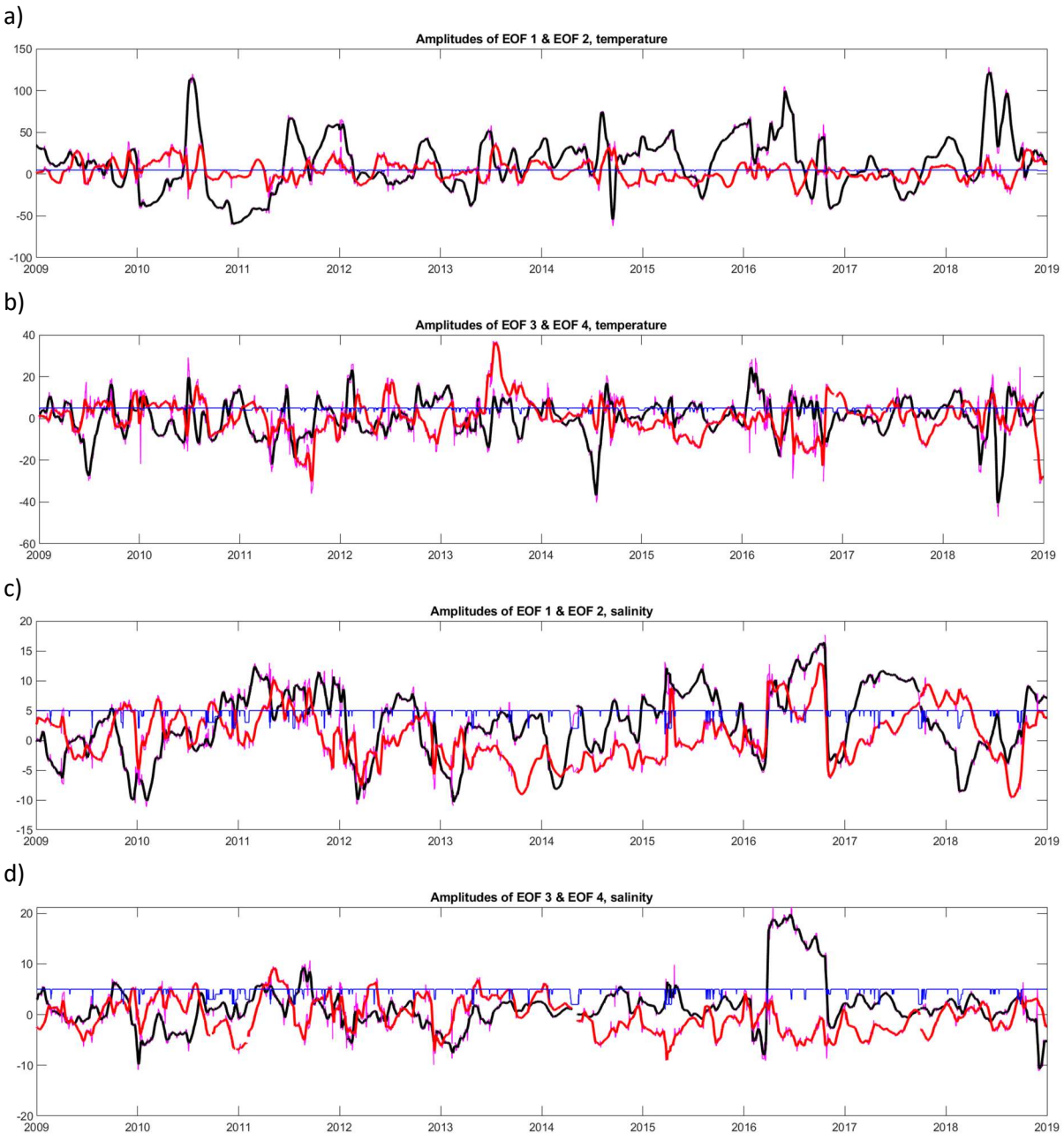


Figure A4.4. Same as Fig. A4.3, but for the period from 01.01.2009 to 31.12.2018.

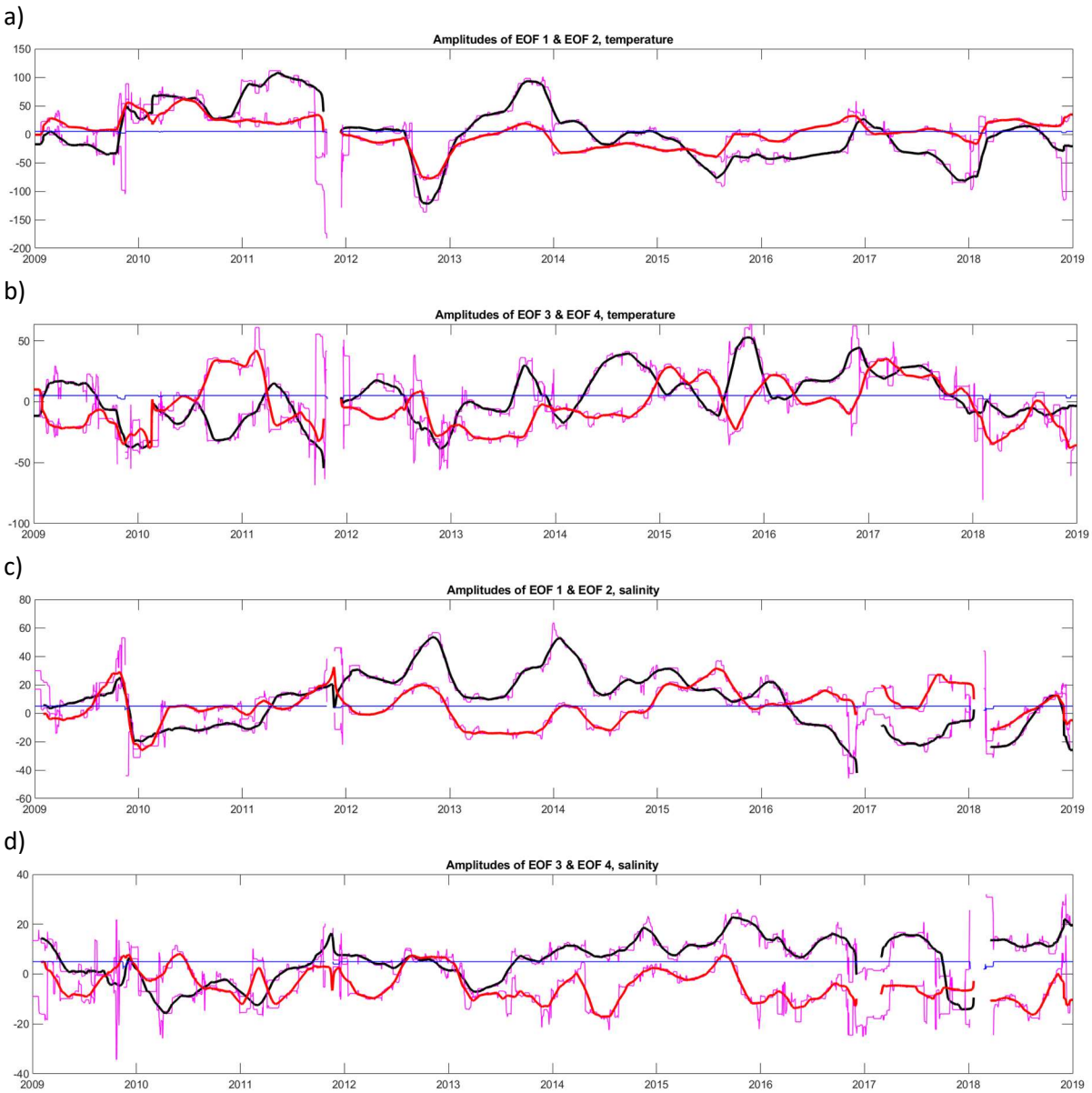


Figure A4.5. Same as Fig. A4.4, but for the deep layer from 40 m to bottom.

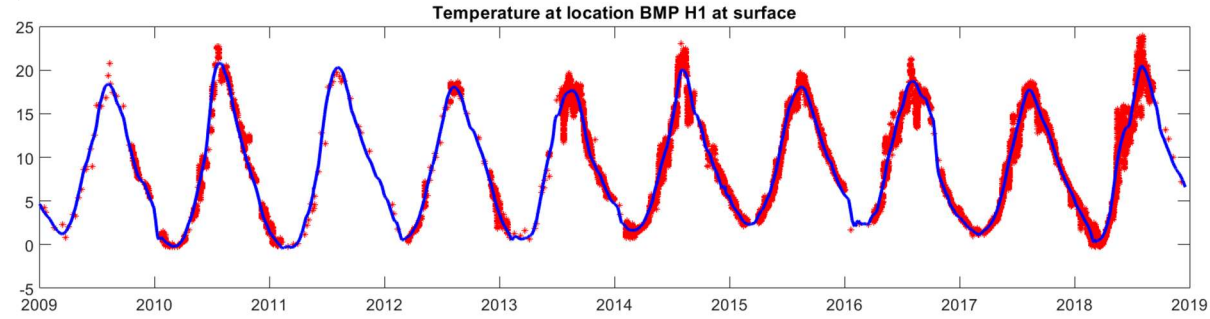
Although EOF reconstruction allows building of the observational grid data over the entire region, we found it necessary to skip the reconstruction and associated DA in the areas and the cases:

1. For temperature: time-dependent areas of cold surface waters with temperature below 1 °C. Correction by DA may lead in the data-poor areas to temperatures below freezing point and artificial ice formation.
2. For salinity: areas near the mouth of larger rivers: Daugava, Pärnu, Kymijoki, Narva and some others. On several occasions, DA may increase the salinity of river bulges too much and suppress the river mouth dynamics.

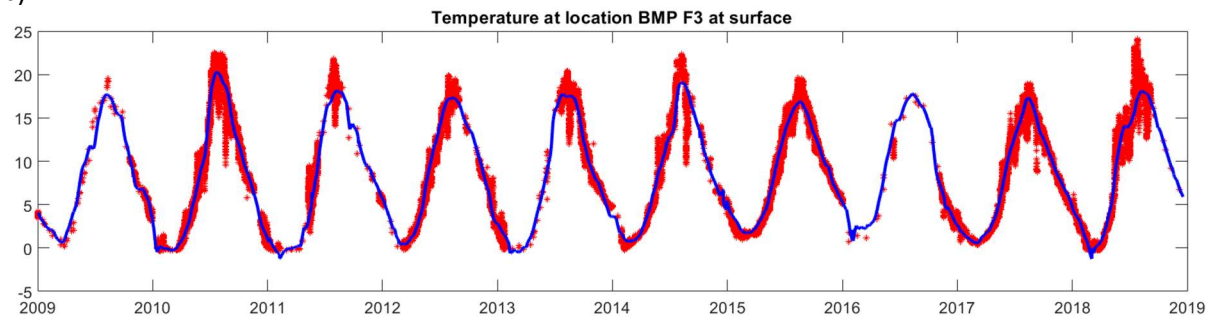
5. Comparison of reconstruction with original observations

Comparison of reconstructions of temperature (Fig. A4.6 and A4.8) and salinity (Fig. A4.7 and A4.9) in the surface layer (Fig. A4.6 and A4.7) and in the deeper layer (Fig. A4.8 and A4.9) demonstrate reasonable match over the large scales of the basins.

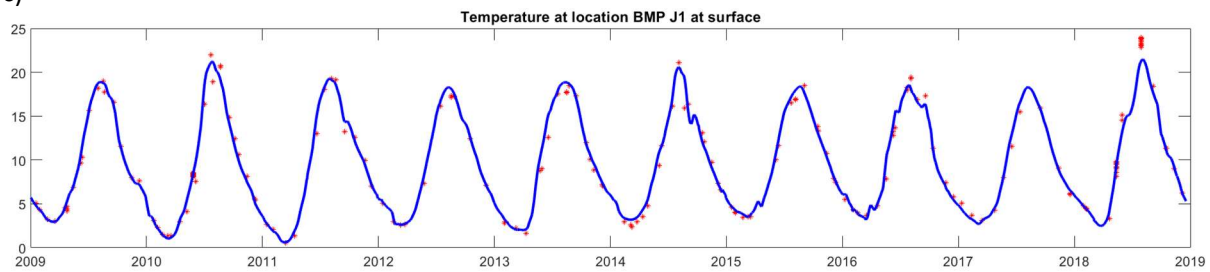
a)



b)



c)



d)

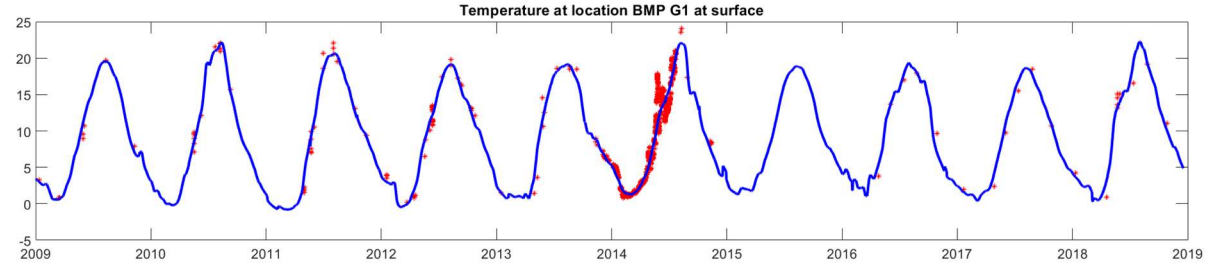
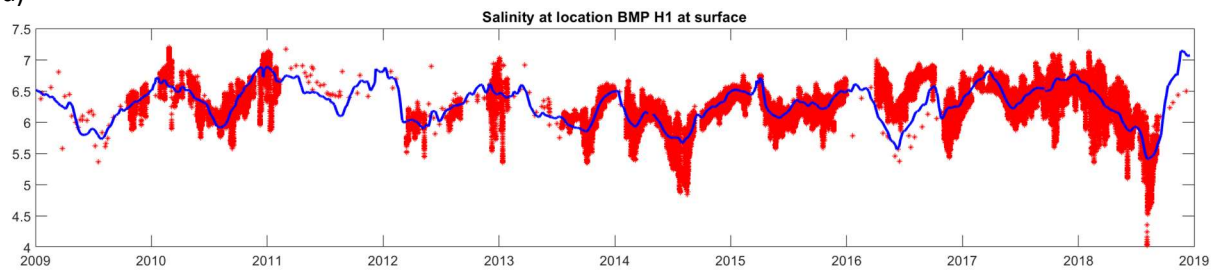


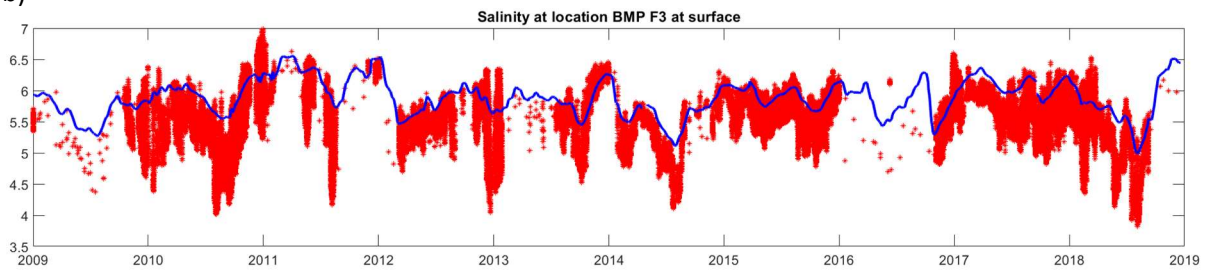
Figure A4.6. Reconstructions of sea surface temperature time series in the western Gulf of Finland (a), the central Gulf of Finland (b), the Gotland Deep (c) and the Gulf of Riga (d). Solid blue line presents reconstructions and red stars represent fragmentary observations in the ICES database.

The surface layers have large FB data sets in the Gulf of Finland, shown by accumulation of red observational markers. We note that density of observations is not present in our simple graphs, therefore the average of observations does not need to be in the center of the red vertical “stripe” of observational range.

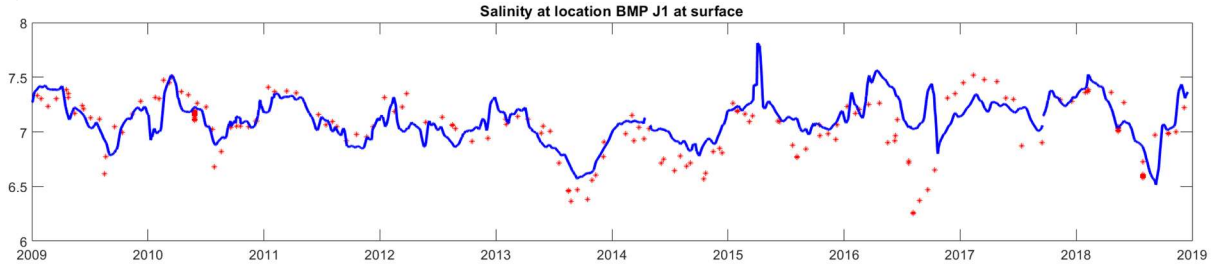
a)



b)



c)



d)

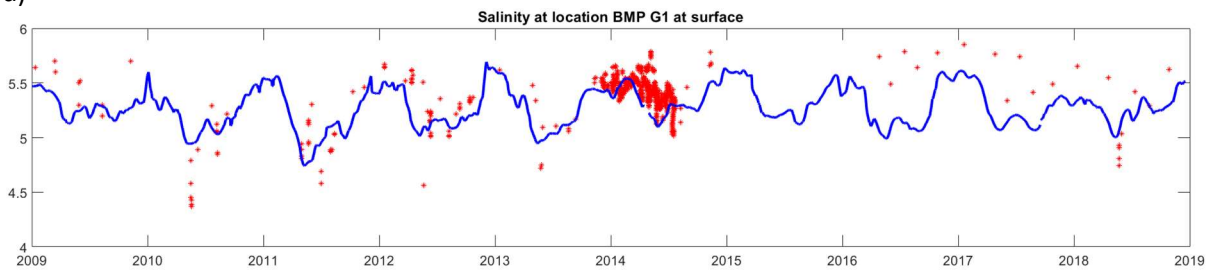


Figure A4.7. Same as Fig. A4.6 but for sea surface salinity.

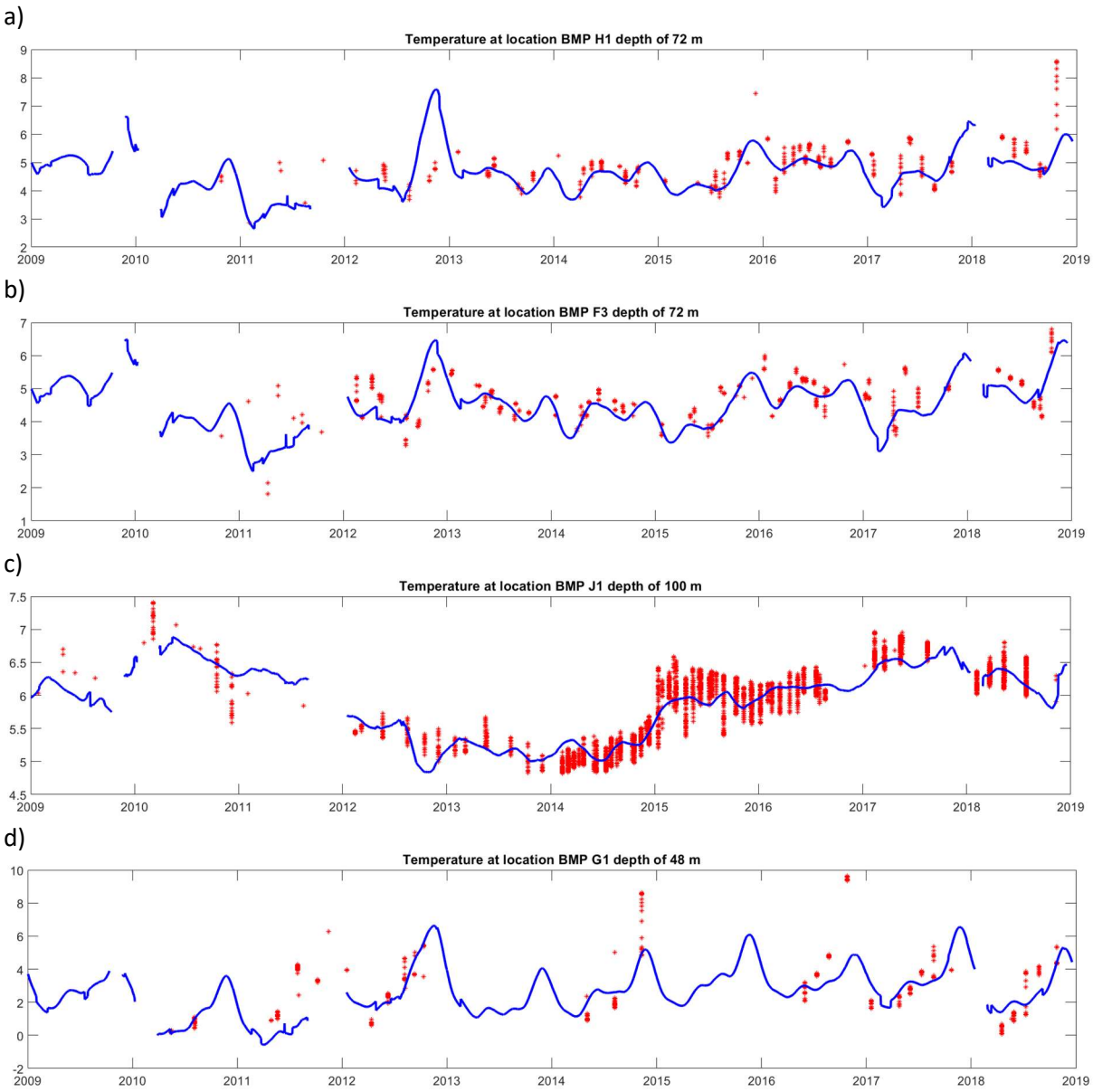
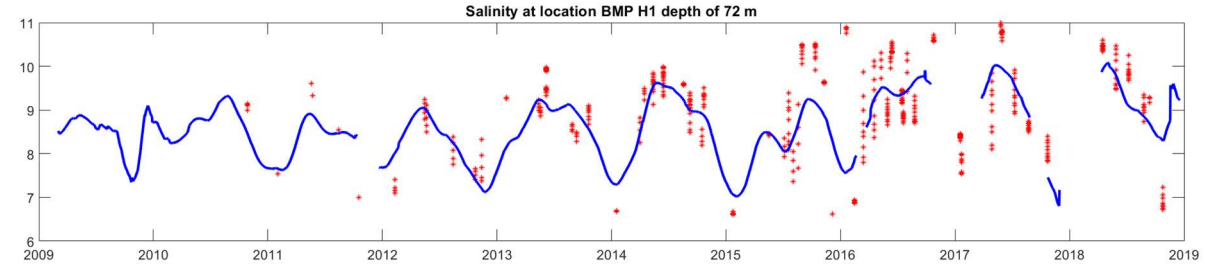
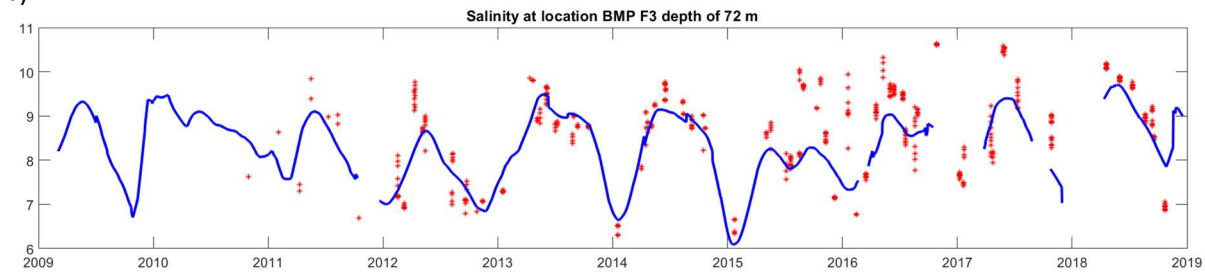


Figure A4.8. Reconstructions of temperature time series in the western Gulf of Finland (a), the central Gulf of Finland (b), the Gotland Deep (c) and the Gulf of Riga (d) at depths of 72, 72, 100 and 48 m, respectively. Solid blue line presents reconstructions and red stars represent fragmentary observations in the ICES database.

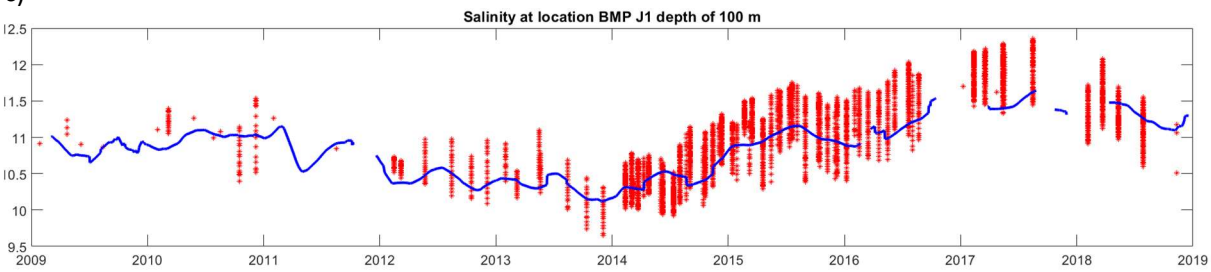
a)



b)



c)



d)

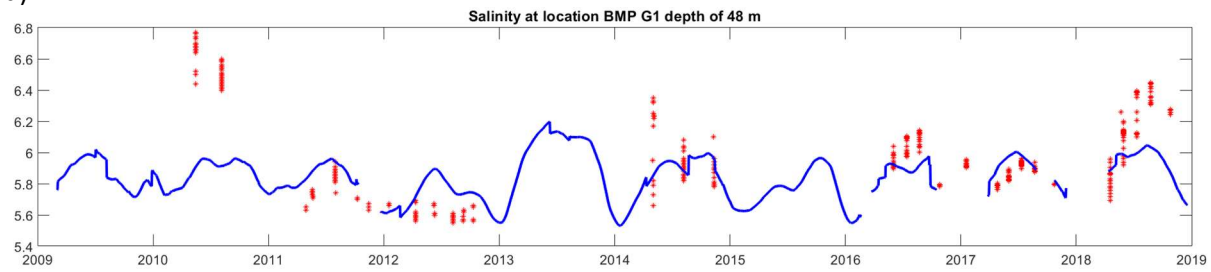


Figure A4.9. Same as Fig. A.4.8 but for salinity.

6. Technical implementation

Although the first 3D implementation of EOF analysis included the whole water column, reaching 17 468 wet points of the grid, drastic difference in data availability at the surface (that includes dense FB observations) and in the deeper layers (that contains rare shipborne profiling) called for separate EOF analysis as shown in Table A4.1. As a result, in addition to the basic files we have a 2D surface option with a suffix SUR and a deeper 3D option with DEP that includes all the layers below 40 m. The precalculated EOF analysis files are listed in the Table A4.2.

Table A4.1. Grids of DA calculated from the CMEMS reanalysis. Abbreviations: M - number of wet points.

	Whole water column	Surface layer	Deeper layer
Grid file	`grid_coarse_DA.txt`	`grid_coarse_SUR_DA.txt`	`grid_coarse_DEP_DA.txt`
Number of wet grid points	M = 17 468	M = 1 134	M = 6 092

Table A4.2. Precalculated EOF and seasonal files. Abbreviations: np - sequential number of wet point, i, j, k, x, y, z - indexes and physical coordinates along the axis, nmode - number of EOF mode, fmode - EOF mode value in point np for the mode nmode, modemax - maximum number of EOF modes, eigen - eigenvalue of the mode nmode, nseas - day 1…365 within a year, temp - temperature, sal - salinity.

File	File names	ASCII content

Grid	`grid_coarse_DA.txt`
	`grid_coarse_SUR_DA.txt`
	`grid_coarse_DEP_DA.txt`
		np, i, j, k, x, y, z
		np = 1…M

EOF modes of temperature	`modes_coarse_T3D_DA.txt`
	`modes_coarse_T3D_SUR_DA.txt`
	`modes_coarse_T3D_DEP_DA.txt`
		(nmode, (fmode 1…np) 1… modemax)

EOF modes of salinity	`modes_coarse_S3D_DA.txt`
	`modes_coarse_S3D_SUR_DA.txt`
	`modes_coarse_S3D_DEP_DA.txt`
		(nmode, (fmode 1…np) 1… modemax)

EOF eigenvalues of temperature	`eigenvalues_coarse_T3D_DA.txt`
	`eigenvalues_coarse_T3D_SUR_DA.txt`
	`eigenvalues_coarse_T3D_DEP_DA.txt`
		nmode, eigen
		nmode = 1…modemax

EOF eigenvalues of salinity	`eigenvalues_coarse_S3D_DA.txt`
	`eigenvalues_coarse_S3D_SUR_DA.txt`
	`eigenvalues_coarse_S3D_DEP_DA.txt`
		nmode, eigen
		nmode = 1…modemax

Seasonal background of temperature	`seasonal_coarse_T3D_DA.txt`
	`seasonal_coarse_T3D_SUR_DA.txt`
	`seasonal_coarse_T3D_DEP_DA.txt`
		np, nseas, temp

Seasonal background of salinity	`seasonal_coarse_S3D_DA.txt`
	`seasonal_coarse_S3D_SUR_DA.txt`
	`seasonal_coarse_S3D_DEP_DA.txt`
		np, nseas, sal

Data assimilation is done using a separate FORTRAN program (the text is `eof_reconstr_v1_1.f90`), where the options are specified in the NAMELIST files

DA_Config.nml (specifies the basic DA parameters) and DA_Config.nml (specifies the date and the type of DA). The program requires the latest file of observations. The file name format is ts_obs_yyyymmdd.dat, where yyyy is the year of current date, mm is the month number and dd is the day number.

Running of the eof_reconstr is controlled by two namelist files as follows.

DA_RunConfig.nml includes

```
$Oper
ndate= 20220430
nda=2
/
```

The date, given as integer by ndate is updated by the observations preprocessing script. The script also specifies the layers and grids using the update to nda: 1 - full 3D, 2 - surface only, 3 - deep layer only, 4 - surface and deeper layers separately (2 sets of DA files)

DA_Config.nml is manually prescribed and contains:

```
$General
path='C:\Users\juri.elken\Doc-Arhiiv\FORT_Projects_CMEMS\Nemo-
Est\Hindcast_2009+\Observations\ICES\'
obsfile='ts_obs_yyyymmdd.dat'
tyhi=9999.
nmode=50
/
```

and grid-specific subsets depending on the value nda. An example of surface subset for salinity during the hindcast is

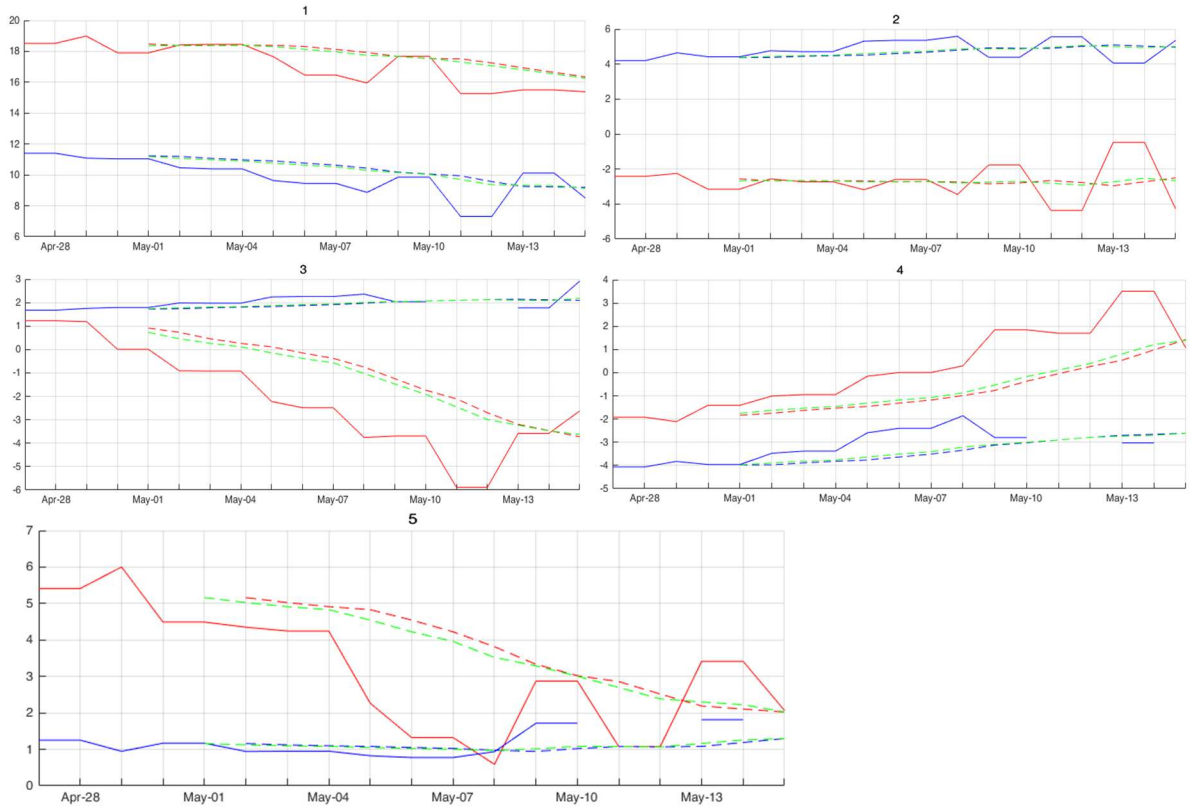
```
$Grid_2_2
gridfile='grid_coarse_SUR_DA.txt'
seasfile='seasonal_coarse_S3D_SUR_DA.txt'
modefile='modes_coarse_S3D_SUR_DA.txt'
eigenfile='eigenvalues_coarse_S3D_SUR_DA.txt'
npoint=1134
modemax=5
tback=20.
tforw=20.
amcoef=3.
am1=3.
am2=5.
/
```

The control values are explained in the FORTRAN code as

```
!file names, npoint - number of wet points on a specific precalculated
grid, modemax - maximum number of EOF modes to be considered
!tback, tforw - backward and forward data time intervals from tref,
calculated from tref (determined from ndate in /Oper/)
!amcoef - check for amplitude limit, amplitude < amcoef*sqrt(eigen)
!am1, am2 - check for amplitude stability: compare relative change if
amplitudes 1 and 2 between the modes k and k-1: it is OK if:
!IF(abs(ampli(1,kmode)-ampli(1,kmode-
1))/abs(ampli(1,kmode)+ampli(1,kmode-1)).LE.am1)
!IF(abs(ampli(2,kmode)-ampli(2,kmode-
1))/abs(ampli(2,kmode)+ampli(2,kmode-1)).LE.am2)
```

Lisa 5 Amplituudide filtreerimine

Amplituutide filtreerimine on vajalik, et vältida suuri hüppeid järjestikuste päevade soolsus- ja temperatuuriväljade EOF rekonstruktsioonides. Suured hüpped (või sootuks puudulikud) amplituutides on tingitud vaatlusandmete hulga muutustes, millest sõltuvad oluliselt lähendusstatistikud. Filtreerimise algoritm on kirjutatud shell-koodis ning leiab 10 viimase päeva keskmised väärtused vastava päeva amplituudidele. Testperioodil 28. aprillist kuni 15. maini 2022 silus skript esinenud fluktuatsioonid ning andis väärtuse ka amplituudidele, mille väärtust antud päeval EOF rekonstruktsioonis ei leitud.




```
echo "FILTER"
do_filt=1
## GET DATA
if [ $do_filt == 1 ]; then
rm filelist_tamp_SUR.txt  filelist_samp_SUR.txt
nf=0
nd_ago=10
declare -A Ta
declare -A Sa
for d in `seq 1 $nd_ago`;do
 datetimeb=`date --date "$date_f $d day ago" +"%Y-%m-%d"`    # new
date
 yyb=`date +'%Y' -d$datetimeb`                 #  formating
 mmb=`date +'%m' -d$datetimeb`
 ddb=`date +'%d' -d$datetimeb`
tstrb="$yyb$mmb$ddb"
echo " -  try read  $tstrb"
### Reading data
if [ -f  "$sur_odir/amp/t_ampl_${tstrb}_SUR.txt" ];then
let nf++
ta_file=$sur_odir/amp/t_ampl_${tstrb}_SUR.txt
ni=0
Ta=""
  while read n f ; do
  let ni++;
  Ta[$ni,$nf]="$f"
  echo "Ta[$ni,$nf]",${Ta[$ni,$nf]}
  done < $ta_file


sa_file=$sur_odir/amp/s_ampl_${tstrb}_SUR.txt
ni=0
Sa=""
  while read n f ; do
  let ni++;
  Sa[$ni,$nf]="$f"
  echo "Sa[$ni,$nf]",${Sa[$ni,$nf]}
  done < $sa_file
fi
done

# Filtering
rm $sur_odir/amp_filt/s_ampl_${tstro}_SUR.txt
$sur_odir/amp_filt/t_ampl_${tstro}_SUR.txt

 #  TEMPERATURE
echo "  ######## TEMPERATURE "
declare -A Taf
f1="%$((${#ni}+2))s"
f2=" %9s"
printf "$f1" ''
for ((i=1;i<=ni;i++)) do
   printf "$f2" $i
done
```

```
echo

for ((j=1;j<=ni;j++)) do
#    printf "$f1" $j
   csum=0
   nfc=0
   for ((i=1;i<=nf;i++)) do
#       printf "$f2" ${Ta[$i,$j]}

echo " test :${Ta[$j,$i]}  ..."
re='^[+-]?[0-9]+([.][0-9]+)?$'
if ! [[ ${Ta[$j,$i]} =~ $re ]] ; then
  echo "error: Not a number ..skip"
else
       let nfc++
       csum=`echo "$csum + ${Ta[$j,$i]}" | bc -l | awk '{printf "%.4f\n",
$0}'`
  echo "cal sum  $nfc..."
fi

   done
   Taf[$j]=`echo "$csum/$nfc" | bc -l  | awk '{printf "%.4f\n", $0}'`
   echo
done
#echo "${Taf[@]}"
for ((i=1;i<=nf;i++)) do
echo  $i    ", " ${Taf[$i]} >> $sur_odir/amp_filt/t_ampl_${tstro}_SUR.txt
done
#exit


 #  SALINITY
echo "  ######## SALINITY"
declare -A Saf
printf "$f1" ''
for ((i=1;i<=ni;i++)) do
   printf "$f2" $i
done
echo

for ((j=1;j<=ni;j++)) do
   printf "$f1" $j
   csum=0
   nfc=0
   for ((i=1;i<=nf;i++)) do
      printf "$f2" ${Sa[$i,$j]}
#       let nfc++
#       csum=`echo "$csum + ${Sa[$j,$i]}" | bc -l | awk '{printf "%.4f\n",
$0}'`
echo " test :${Sa[$j,$i]}  ..."
re='^[+-]?[0-9]+([.][0-9]+)?$'
if ! [[ ${Sa[$j,$i]} =~ $re ]] ; then
  echo "error: Not a number ..skip"
else
       let nfc++
       csum=`echo "$csum + ${Sa[$j,$i]}" | bc -l | awk '{printf "%.4f\n",
$0}'`
     echo "cal sum  $nfc..."
fi

   done
   Saf[$j]=`echo "$csum/$nfc" | bc -l  | awk '{printf "%.4f\n", $0}'`
   echo
done
#echo "${Saf[@]}"
for ((i=1;i<=nf;i++)) do
echo  $i    ", " ${Saf[$i]} >> $sur_odir/amp_filt/s_ampl_${tstro}_SUR.txt
done

echo "Filtered amplitudes are"
echo " Temp:  "${Taf[@]}""
echo " Salt:  "${Saf[@]}""

else
```