# S7.3) Big data approximations with discrepancy

This notebook contains the code of the paper "Bayesian Calibration of Imperfect Computer Models using Physics-Informed Priors". The models are fitted in rstan and the code is available in the folder "STAN/Approximations".

### Load packages

```
# uncomment to install
# install.packages("rstan")
# install.packages("ggplot2")
# install.packages("tidyverse")
library(rstan)
library(ggplot2)
library(tidyverse)

rstan_options(auto_write = TRUE)
options(mc.cores = 3) # allocate 3 cores (for each model we run 3 chains in parallel)
# Numerical simulator of the WK3 model
source("functions/WK2and3_sim_fn.R")
# Load flow data
d = readRDS("Data/Inflow_time.rds")
```

### Reality and modelling choice

$$\mathcal{R}: \quad \frac{dP(t)}{dt} + \frac{P(t)}{R_2 C} = \frac{Q(t)}{C}\left(1 + \frac{R_1}{R_2}\right) + R_1 \frac{dQ(t)}{dt} \quad \text{(the misspesified model we use to fit the data)} \quad \text{[WK3]} \tag{1}$$

$$\eta: \quad Q(t) = \frac{1}{R}P(t) + C\frac{dP(t)}{dt} \quad \text{(the model we use to simulate data)} \quad \text{[WK2]} \tag{2}$$

```
Rtrue = 1; Ctrue = 1.1; Ztrue = 0.05
flow = d$inflow*0.95
time = d$time
nP = 90 # number of pressure data
nI = 100# number of inflow data
nc = 1  # number of cardiac cycles
nflow = length(flow)
# 1. simulate WK3 data (R=R_2, Z=R_1)
Psim = WK3_simulate(flow = flow, time = time, R = Rtrue, C = Ctrue, Z=Ztrue) # simulate WK3 data for a
P_true = Psim
# 2. choose pressure and inflow indices
indP = round(seq(1, nflow, length.out = nP)); indI = round(seq(1, nflow, length.out = nI))
yP_real = Psim[indP]; yI_real = flow[indI] # noise free fimulated pressure and flow
# 3. Add noise
# set.seed(0)
```

```r
set.seed(1)
Pnoise = rnorm(nP*nc, 0, 4) # sample pressure noise from N(0, 4^2)
Inoise = rnorm(nI*nc, 0, 10) # sample flow noise from N(0,10^2)
yP_real = rep(yP_real,nc)
yI_real = rep(yI_real,nc)
# 4. store individual data in the population matrices
yP = yP_real + Pnoise # add noise
yI = yI_real + Inoise # add noise
tP = time[indP] # corresponding time (synchronized for the two cycles)
tI = time[indI] # corresponding time (synchronized for the two cycles)

data_PI = list(nP=nc*nP, nI=nc*nI, tP=rep(tP,nc), tI=rep(tI,nc), yP=yP, yI=yI, mP=12, mI=10)
```

```r
WK2_VFE = stan_model('STAN/Approximations/VFE/WK2_delta_VFE.stan')
kp = kmeans(data.frame(x=data_PI$tP), centers = data_PI$mP)
ki = kmeans(data.frame(x=data_PI$tI), centers = data_PI$mI)

init = list("zP" = as.vector(kp$centers), "zI" = as.vector(ki$centers))
op_VFE=optimizing(WK2_VFE, data=data_PI, hessian=FALSE, init = init, verbose=TRUE, seed=0)
```

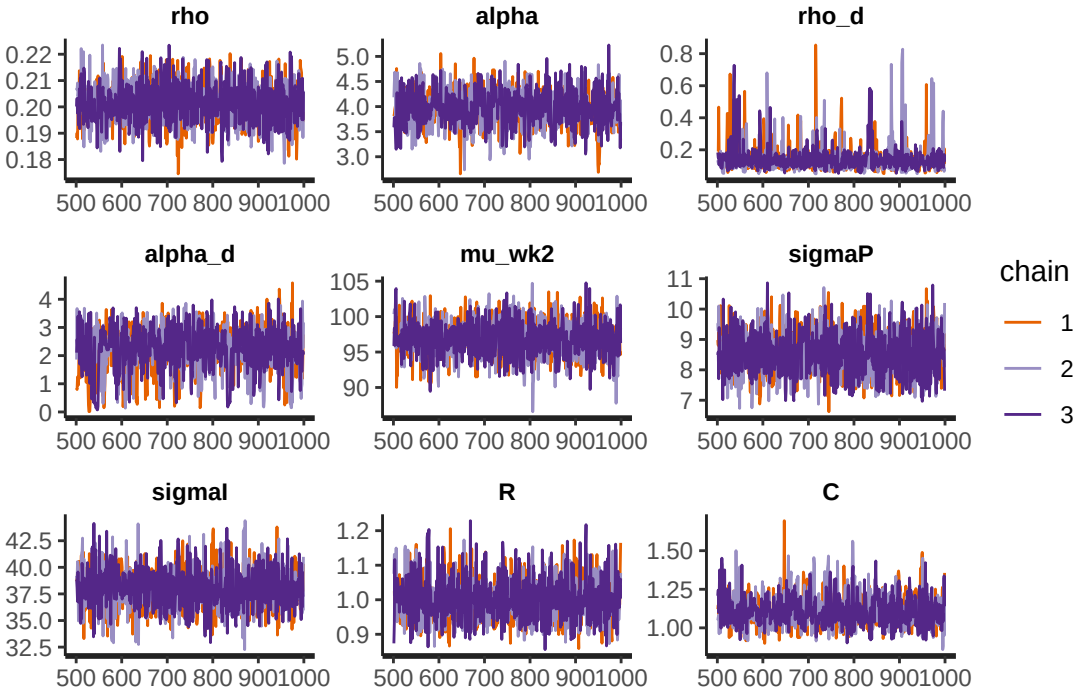
```
Chain 1: Initial log joint probability = -13400
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:       19      -962.859     0.0138579        2.3978           1           1       31
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:       39      -949.581     0.0272867       2.43968           1           1       56
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:       49      -949.485   0.000188747     0.0088334           1           1       68
Chain 1: Optimization terminated normally:
Chain 1:   Convergence detected: relative gradient magnitude is below tolerance
```

```r
op_VFE
```

```
$par
         rho        alpha        rho_d      alpha_d       mu_wk2       sigmaP
 0.195718978  4.897341220  0.057918221  0.001593782 97.398040198  9.760864357
      sigmaI            R            C        zP[1]        zP[2]        zP[3]
63.624564498  0.955552758  1.013489858  0.080072408  0.133867715  0.455149871
       zP[4]        zP[5]        zP[6]        zP[7]        zP[8]        zP[9]
 0.194511717  0.274922964  0.724878480  0.826984618  0.944283254  0.365017283
      zP[10]       zP[11]       zP[12]        zI[1]        zI[2]        zI[3]
 0.544772397  0.020043511  0.634880324  0.970363221  0.614334337  0.714632658
       zI[4]        zI[5]        zI[6]        zI[7]        zI[8]        zI[9]
 0.049797442  0.394214611  0.511302773  0.809613912  0.274381891  0.161115333
      zI[10]
 0.894987597

$value
[1] -949.4846

$return_code
[1] 0

$theta_tilde
          rho    alpha      rho_d     alpha_d   mu_wk2   sigmaP   sigmaI
[1,] 0.195719 4.897341 0.05791822 0.001593782 97.39804 9.760864 63.62456
```

```
             R       C      zP[1]      zP[2]      zP[3]      zP[4]     zP[5]
[1,] 0.9555528 1.01349 0.08007241 0.1338677 0.4551499 0.1945117 0.274923
         zP[6]     zP[7]     zP[8]     zP[9]     zP[10]     zP[11]    zP[12]
[1,] 0.7248785 0.8269846 0.9442833 0.3650173 0.5447724 0.02004351 0.6348803
         zI[1]     zI[2]     zI[3]      zI[4]     zI[5]     zI[6]     zI[7]
[1,] 0.9703632 0.6143343 0.7146327 0.04979744 0.3942146 0.5113028 0.8096139
         zI[8]     zI[9]    zI[10]
[1,] 0.2743819 0.1611153 0.8949876
```

```
zP_opt_VFE=op_VFE$par[grep("zP",names(op_VFE$par))]
zI_opt_VFE=op_VFE$par[grep("zI",names(op_VFE$par))]
# plot(sort(zP_opt_VFE))
# plot(sort(zI_opt_VFE))
data_PI_Z_VFE= data_PI
data_PI_Z_VFE$zP = zP_opt_VFE
data_PI_Z_VFE$zI = zI_opt_VFE

fit_post_VFE=stan(file='STAN/Approximations/VFE/WK2_delta_VFE_fixed_Z.stan',
                  data=data_PI_Z_VFE,
                  chains=3,
                  iter=1000,
                  seed=0
)
# stan_hist(fit_post_VFE)
stan_trace(fit_post_VFE)
```




**FITC**

```
WK2_FITC = stan_model('STAN/Approximations/FITC/WK2_delta_FITC.stan')
op_FITC=optimizing(WK2_FITC, data=data_PI, hessian=FALSE, verbose=TRUE,init=init,seed=31)
```

```
Chain 1: Initial log joint probability = -2004.04
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:       19      -669.435      0.310303       41.9199           1           1       27
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:       39      -664.281     0.0230328       8.07629           1           1       48
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:       59      -661.441      0.054124       7.75956           1           1       69
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1: Exception: cholesky_decompose: A is not symmetric. A[1,2] = 3.52125e+08, but A[2,1] = 3.52125e


Chain 1:       79      -660.309    0.00699334       71.7644           1           1      107
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:       99      -659.913     0.0034056       36.0532           1           1      134
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      119      -659.502   0.000443434       17.5017      0.9052      0.9052      176
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      139      -659.349    0.00364397       133.066      0.1929           1      215
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      159      -659.341   0.000473607       40.2006           1           1      237
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      179       -659.25    0.00250924       189.871           1           1      267
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      199      -659.195     0.0111379       37.2306           1           1      297
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      219      -658.957    0.00852561       30.5831           1           1      324
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      239      -658.816     0.0197552       56.3979      0.1756           1      349
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      259      -658.715    0.00994337       6.95179           1           1      384
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      279      -658.655   0.000494335        8.9195      0.2607      0.2607      417
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      299      -658.493     0.0084704       70.9239           1           1      453
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      319      -658.271    0.00771376       44.4081           1           1      477
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      339      -658.024    0.00115201       57.4278           1           1      500
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      359      -657.898     0.0161518       53.2322           1           1      525
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      379      -657.642      0.065786       496.343           1           1      548
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      399      -657.387     0.0523804       350.631           1           1      577
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      419      -657.126     0.0082776       58.5751      0.6714      0.6714      601
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      439       -655.99   0.000935883       20.6301           1           1      640
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      459       -655.89   0.000435998       675.223    0.009165           1      670
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      479       -655.65    0.00597702       77.9018           1           1      693
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      499      -655.583    0.00367783       87.4731           1           1      715
```

```
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      519      -655.514    0.00377683       20.8182           1           1      743
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      539      -655.488    0.00154758       27.3801           1           1      769
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      559       -655.26     0.0141559       239.067           1           1      793
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      579      -655.155    0.00275471       21.9806           1           1      815
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      599      -654.969     0.0129091       35.6659           1           1      846
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      619      -654.875    0.00488429       18.1155           1           1      877
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      639        -654.8   0.000761773       18.5667           1           1      900
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      659      -654.794    0.00141579       14.6775           1           1      924
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      679      -654.737    0.00182673        48.432           1           1      947
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      699      -654.643     0.0468412       173.352           1           1      977
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      719      -654.614    0.00123389       76.5774      0.6621      0.6621     1000
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      739      -654.603   0.000575251       25.0344           1           1     1020
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      759      -654.595   0.000808133        183.98           1           1     1043
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      779      -654.578   0.000186752       45.4112           1           1     1067
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      799      -654.564    0.00137559       92.0919           1           1     1088
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      819      -654.529    0.00348998       45.4762           1           1     1115
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      839       -654.51   9.87255e-05       8.01886           1           1     1142
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      859      -654.507   0.000122429       59.7789      0.5742      0.5742     1164
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      879      -654.493   0.000481615       157.102      0.6711      0.6711     1185
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      899      -654.487   6.58802e-06       3.23795           1           1     1206
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      919      -654.481   0.000268013        9.3895      0.4215           1     1230
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      939      -654.479    0.00032592       4.87586           1           1     1256
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      959      -654.468   0.000689491       7.34697           1           1     1276
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      979      -654.462   0.000413834       16.3907           1           1     1301
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:      999      -654.455    0.00190993       12.2784      0.4381           1     1324
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1019      -654.451    0.00107279       27.6408           1           1     1348
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1039      -654.449   0.000142376       6.11789           1           1     1372
```

```
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1059      -654.447   3.42454e-05       3.62534           1           1     1395
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1079      -654.444   3.23156e-05       15.3321      0.2697      0.2697     1418
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1099      -654.442   0.000162787       3.02441       0.804      0.0804     1446
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1119       -654.44   3.95867e-05       3.20506      0.3881      0.3881     1479
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1139      -654.439   0.000185754       3.72609           1           1     1504
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1159      -654.434   4.70615e-05       7.00515           1           1     1530
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1179       -654.43     0.0020549       48.4867       2.265     0.03384     1558
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1199      -654.425   0.000770176       18.1637           1           1     1585
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1219      -654.423   0.000791161       10.1872           1           1     1609
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1239      -654.416    0.00211603       27.0056           1           1     1635
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1259      -654.401    0.00024892        6.1026      0.3063      0.3063     1667
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1279      -654.389    0.00123802       14.2112           1           1     1704
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1299      -654.388    0.00154492       9.09395           1           1     1728
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1319      -654.381   0.000328318        21.443           1           1     1751
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1339      -654.378   7.98946e-05       2.52148           1           1     1774
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1359      -654.374   0.000164354       2.97219           1           1     1802
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1379      -654.371   0.000898668       17.5698           1           1     1828
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1399      -654.368   0.000107954       4.36249           1           1     1852
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1419      -654.368   0.000246819        27.818      0.3969           1     1874
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:     1421      -654.368    0.00013028       2.33613           1           1     1876
Chain 1: Optimization terminated normally:
Chain 1:   Convergence detected: relative gradient magnitude is below tolerance
```

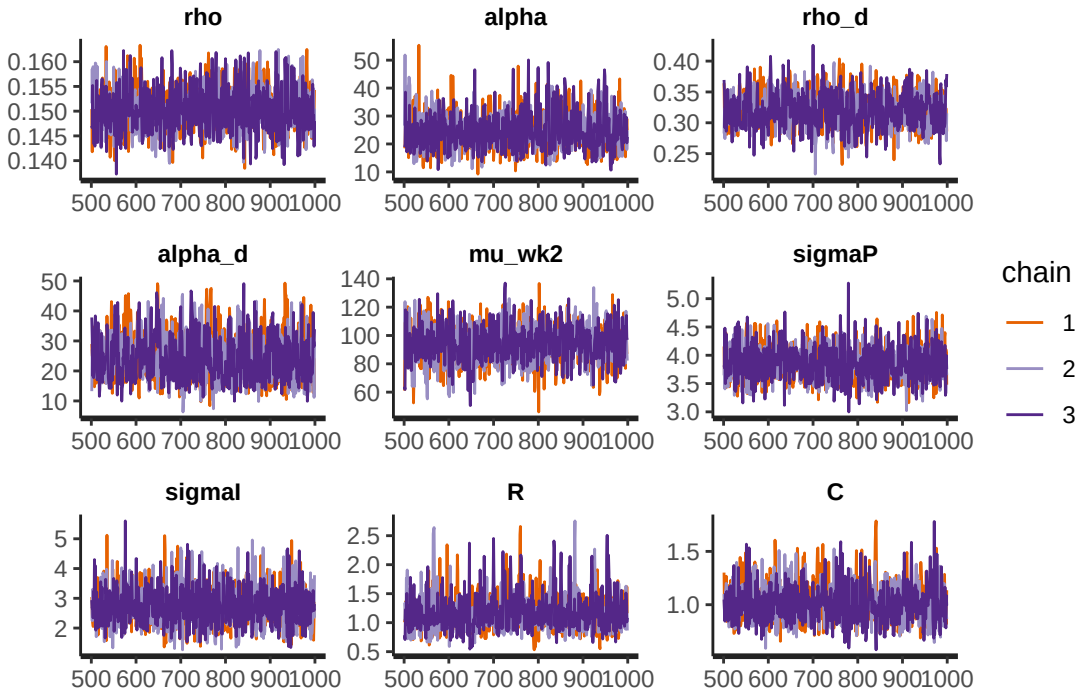
```
op_FITC
```

```
$par
       rho       alpha       rho_d     alpha_d      mu_wk2      sigmaP
0.14840932 21.92549211  0.29910806 17.76112432 91.71812997  3.76751934
    sigmaI           R           C       zP[1]       zP[2]       zP[3]
2.52790544  1.03745494  0.94479338  0.04139094  0.09266307  0.35804464
     zP[4]       zP[5]       zP[6]       zP[7]       zP[8]       zP[9]
0.19328431  0.24632607  0.76518413  0.86656296  0.89083452  0.34639391
    zP[10]      zP[11]      zP[12]       zI[1]       zI[2]       zI[3]
0.59069336  0.01498776  0.50945618  0.99295215  0.62821519  0.72233306
     zI[4]       zI[5]       zI[6]       zI[7]       zI[8]       zI[9]
```

```
 0.10422541  0.25617667  0.47110524  0.84384255  0.10350617  0.10350590
     zI[10]
 0.99294769

$value
[1] -654.3675

$return_code
[1] 0

$theta_tilde
           rho    alpha     rho_d  alpha_d   mu_wk2   sigmaP   sigmaI        R
[1,] 0.1484093 21.92549 0.2991081 17.76112 91.71813 3.767519 2.527905 1.037455
             C      zP[1]      zP[2]     zP[3]     zP[4]     zP[5]     zP[6]
[1,] 0.9447934 0.04139094 0.09266307 0.3580446 0.1932843 0.2463261 0.7651841
        zP[7]     zP[8]     zP[9]    zP[10]     zP[11]    zP[12]     zI[1]
[1,] 0.866563 0.8908345 0.3463939 0.5906934 0.01498776 0.5094562 0.9929521
         zI[2]     zI[3]     zI[4]     zI[5]     zI[6]     zI[7]     zI[8]
[1,] 0.6282152 0.7223331 0.1042254 0.2561767 0.4711052 0.8438425 0.1035062
         zI[9]    zI[10]
[1,] 0.1035059 0.9929477
```

```
zP_opt=op_FITC$par[grep("zP",names(op_FITC$par))]
zI_opt=op_FITC$par[grep("zI",names(op_FITC$par))]
data_PI_Z_FITC = data_PI
data_PI_Z_FITC$zP = zP_opt
data_PI_Z_FITC$zI = zI_opt
# plot(sort(zP_opt))
# plot(sort(zI_opt))
```

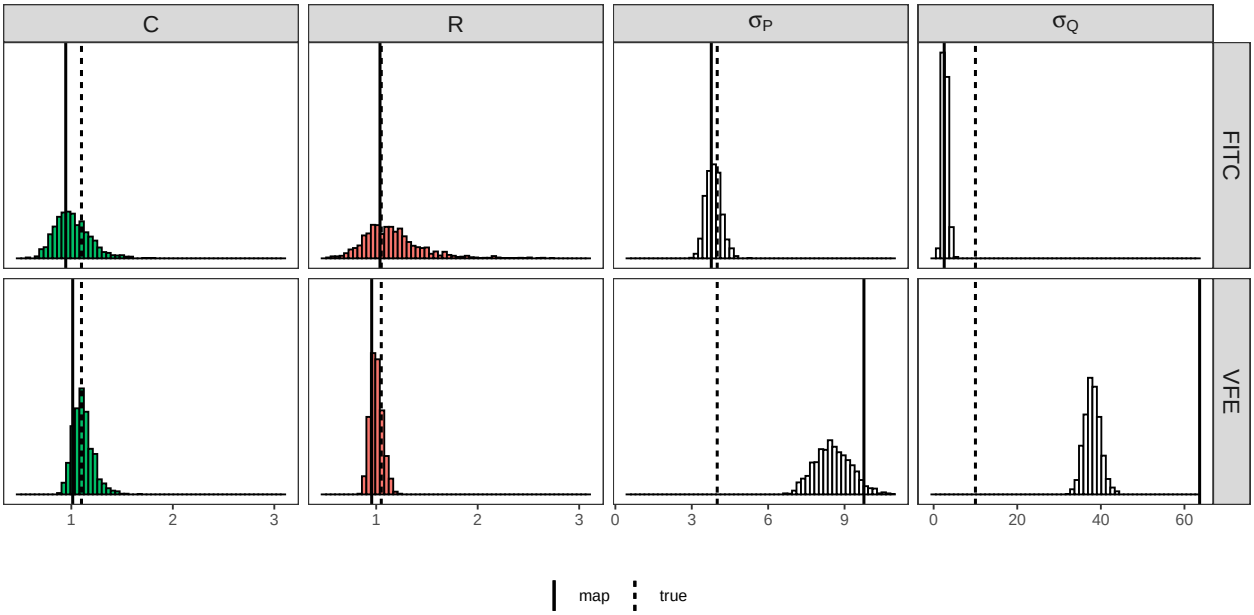
```
fit_post_FITC=stan(file='STAN/Approximations/FITC/WK2_delta_FITC_fixed_Z.stan',
                   data=data_PI_Z_FITC,
                   chains=3,
                   iter=1000,
                   seed=0
)

# stan_hist(fit_post_FITC)
stan_trace(fit_post_FITC)
```




```
post_VFE = data.frame(rstan::extract(fit_post_VFE))
post_FITC = data.frame(rstan::extract(fit_post_FITC))

pr = c("R", "C", "sigmaP", "sigmaI")
pVFE = as.vector(as.matrix(post_VFE[,pr]))
pFITC = as.vector(as.matrix(post_FITC[,pr]))
Ns = nrow(post_VFE)

df_plot_post = data.frame(post = c(pVFE, pFITC), par = rep(rep(pr, each = Ns),2), model = c(rep("VFE",l
```

```
mod_dat = df_plot_post%>%
  mutate(par = recode(par,
    "R" = "R",
    "C" = "C",
    "sigmaP" = "sigma[P]",
    "sigmaI" = "sigma[Q]"
  ))
df_true_par = data.frame(post=c(Rtrue+Ztrue, Ctrue, 4, 10),par = c("R", "C", "sigma[P]", "sigma[Q]"))
set_lim = data.frame(x=c(0.5,3.1),y=c(600, 600))
df_point_est = data.frame(val=c(op_VFE$par[pr],op_FITC$par[pr]),par = rep(df_true_par$par,2), model = r
pl_post = ggplot()+
  geom_histogram(data = mod_dat, aes(x=post, fill = par), color="black",bins = 60)+
  facet_grid(model~par, scales = "free", labeller = labeller(par = label_parsed))+
  geom_point(data = set_lim, aes(x=x,y=y), color = "white",alpha=0.000001, size=0.000001)+ # set limits
  geom_vline(aes(xintercept = post, linetype = "true"), data=df_true_par, size=0.8)+
  geom_vline(aes(xintercept = val, linetype = "map"), data=df_point_est, size=0.8)+
  theme_bw()+
  theme(#legend.position = "none",
        legend.title = element_blank(),
        legend.position="bottom",
        axis.title.y=element_blank(),
        axis.text.y=element_blank(),
```

```
        axis.ticks.y=element_blank(),
        strip.text.x = element_text(size = 13),
        strip.text.y = element_text(size = 13))+
  xlab("") + ylab("")+
  scale_fill_manual(
  breaks=c("R", "C", "sigma[P]", "sigma[Q]"),
  values=c("#F8766D","#00BE67","white", "white"),guide = "none")
(pl_post=pl_post + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()))
```




```
ggsave("figures/Appr_delta_post.pdf", plot = pl_post, width = 20, height = 12, units = "cm")
```

```
nP_pred = 40
ind_P_pred = round(seq(1,length(time),length.out = nP_pred))
tP_pred = time[ind_P_pred]
data_PI_Z_FITC$nP_pred = nP_pred
data_PI_Z_FITC$tP_pred = tP_pred
data_PI_Z_FITC$nI_pred = nP_pred
data_PI_Z_FITC$tI_pred = tP_pred

N_samples = nrow(post_FITC)
data_post_FITC = list(alpha=post_FITC$alpha, rho=post_FITC$rho, alpha_d=post_FITC$alpha_d
                      , rho_d=post_FITC$rho_d, sigmaP=post_FITC$sigmaP, sigmaI=post_FITC$sigmaI
                      , R=post_FITC$R, C=post_FITC$C, N_samples=N_samples
  )

data_pred_FITC = c(data_PI_Z_FITC, data_post_FITC)
```

```
pred_FITC = stan(file = 'STAN/Approximations/FITC/FITC_delta_predictions.stan',
                 data = data_pred_FITC,
                 chains = 1, iter = 1, seed=123,
                 algorithm = "Fixed_param")
```

```
SAMPLING FOR MODEL 'FITC_delta_predictions' NOW (CHAIN 1).
```

```
Chain 1: Iteration: 1 / 1 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0 seconds (Warm-up)
Chain 1:                3.00945 seconds (Sampling)
Chain 1:                3.00945 seconds (Total)
Chain 1:
```

```
post_mu_CIs.fn = function(post_pred, ci = c(0.05,0.95), time = tP_pred){
  pp = rstan::extract(post_pred)
  dfP = pp$y_P[1,,]
  Psmr = data.frame(mean = colMeans(dfP),
                    lower = apply(dfP, 2, quantile, probs = ci[1]),
                    upper = apply(dfP, 2, quantile, probs = ci[2]),
                    time=time)
  dfI = pp$y_I[1,,]
  Ismr = data.frame(mean = colMeans(dfI),
                    lower = apply(dfI, 2, quantile, probs = ci[1]),
                    upper = apply(dfI, 2, quantile, probs = ci[2]),
                    time=time)
  return(list(Psmr=Psmr, Ismr=Ismr))
}
```

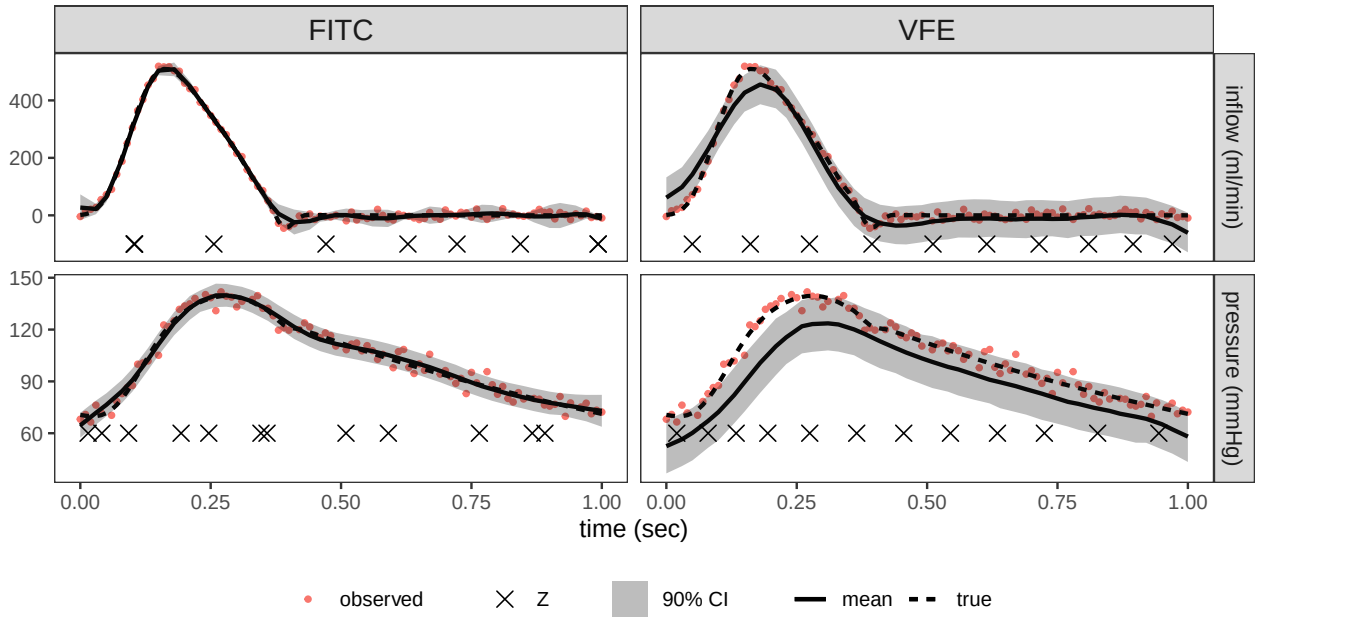
```
data_post_VFE = list(alpha=post_VFE$alpha, rho=post_VFE$rho, alpha_d=post_VFE$alpha_d
                     , rho_d=post_VFE$rho_d, sigmaP=post_VFE$sigmaP, sigmaI=post_VFE$sigmaI
                     , R=post_VFE$R, C=post_VFE$C, N_samples=N_samples
)


data_pred_VFE = c(data_PI_Z_VFE, data_post_VFE)
data_pred_VFE$nP_pred = nP_pred
data_pred_VFE$tP_pred = tP_pred
data_pred_VFE$nI_pred = nP_pred
data_pred_VFE$tI_pred = tP_pred
```

```
pred_VFE = stan(file = 'STAN/Approximations/VFE/WK2_delta_VFE_predictions.stan',
                data = data_pred_VFE,
                chains = 1, iter = 1, seed=123,
                algorithm = "Fixed_param")
```

```
SAMPLING FOR MODEL 'WK2_delta_VFE_predictions' NOW (CHAIN 1).
Chain 1: Iteration: 1 / 1 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0 seconds (Warm-up)
Chain 1:                2.52732 seconds (Sampling)
Chain 1:                2.52732 seconds (Total)
Chain 1:
```

```
pp_VFE = rbind(post_mu_CIs.fn(post_pred=pred_VFE)$Psmr, post_mu_CIs.fn(post_pred=pred_VFE)$Ismr)
pp_VFE$output = c(rep("pressure (mmHg)", nP_pred),rep("inflow (ml/min)", nP_pred))
pp_VFE$model = "VFE"
pp_FITC = rbind(post_mu_CIs.fn(post_pred=pred_FITC)$Psmr,post_mu_CIs.fn(post_pred=pred_FITC)$Ismr)
pp_FITC$output = c(rep("pressure (mmHg)", nP_pred),rep("inflow (ml/min)", nP_pred))
pp_FITC$model = "FITC"
pred_df = rbind(pp_VFE, pp_FITC)
# head(pred_df)
```

```
df_zP_VFE = data.frame(z=data_PI_Z_VFE$zP, y = rep(60, data_PI_Z_VFE$mP), model = "VFE", output = "pres
df_zI_VFE = data.frame(z=data_PI_Z_VFE$zI, y = rep(-100, data_PI_Z_VFE$mI), model = "VFE", output = "in
df_zP_FITC = data.frame(z=data_PI_Z_FITC$zP, y = rep(60, data_PI_Z_FITC$mP), model = "FITC", output = "
df_zI_FITC = data.frame(z=data_PI_Z_FITC$zI, y = rep(-100, data_PI_Z_FITC$mI), model = "FITC", output =
df_z = rbind(df_zP_VFE, df_zI_VFE, df_zP_FITC, df_zI_FITC)
P_true = data.frame(val=Psim, time=time)
P_true$output = "pressure (mmHg)"
I_true = data.frame(val=flow, time=time)
I_true$output = "inflow (ml/min)"
true_out = rbind(P_true, I_true)

obsP = data.frame(value=data_PI$yP, time = data_PI$tP, output = "pressure (mmHg)")
obsI = data.frame(value=data_PI$yI, time = data_PI$tI, output = "inflow (ml/min)")
obs = rbind(obsP,obsI)

pl_pred=ggplot()+
  geom_point(data = obs, aes(y=value, x=time, colour = "observed"), shape = 20)+
  geom_line(data = pred_df, aes(y=mean, x=time, linetype = "mean"), size=0.9)+
  geom_line(data = true_out, aes(y=val, x=time, linetype="true"), size=0.9)+
  geom_ribbon(data = pred_df,aes(ymin=lower, ymax=upper, x=time, fill = "90% CI"), alpha = 0.3)+
  facet_grid(output~model,scales = "free")+
  geom_point(data = df_z, aes(x=z, y=y, shape="Z"), size=3)+
  scale_fill_manual("",values=c("90% CI" = "grey12"))+
  theme_bw()+xlab("time (sec)")+ylab("")+
  scale_shape_manual("", values = c("Z" = 4))+
  theme(#legend.position = "none",
        legend.title = element_blank(),
        legend.position="bottom",
        strip.text.x = element_text(size = 13),
        strip.text.y = element_text(size = 10))
(pl_pred=pl_pred + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()))
```



```
ggsave("figures/Appr_delta_pred.pdf", plot = pl_pred, width = 16, height = 10, units = "cm")
```

### Plug in noise estimates

We observe that the VFE model can severely overestimate the noise and therefore result in underfitting. A remedy to this problem is to fix the noise parameter of the functions $P(t)$ and $Q(t)$, ($\sigma_P$ and $\sigma_I$). A possible solution for obtaining estimates for the noise parameters is to fit a standard GP model for each dataset $(y_P, t_P)$ and $y_Q, t_Q$ indepentently and obtain MLE estimates via maximizing the marginal log-likelihood.

```
nsP = 25
indP = seq(1,data_PI$nP,length.out = nsP)
data_sample_P = list(N=nsP, x = data_PI$tP[indP], y = data_PI$yP[indP])
data_sample_I = list(N=nsP, x = data_PI$tI[indP], y = data_PI$yI[indP])

GP = stan_model('STAN/Approximations/GP_full/GP.stan')

GP_MLE_P=optimizing(GP, data=data_sample_P, hessian=FALSE, verbose=TRUE,seed=0)
```

```
Chain 1: Initial log joint probability = -84.3916
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:       18       -65.767    0.00556334   7.66548e-05           1           1       30
Chain 1: Optimization terminated normally:
Chain 1:   Convergence detected: relative gradient magnitude is below tolerance
```

```
GP_MLE_P
```

```
$par
       rho      alpha      sigma
 0.2147435 76.0190356  3.3560127

$value
[1] -65.767

$return_code
[1] 0

$theta_tilde
           rho    alpha    sigma
[1,] 0.2147435 76.01904 3.356013
```

```
sigma_P_MLE = GP_MLE_P$par["sigma"]

GP_MLE_I=optimizing(GP, data=data_sample_I, hessian=FALSE, verbose=TRUE,seed=0)
```

```
Chain 1: Initial log joint probability = -656.901
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:       19      -111.136      0.558199      0.622198           1           1       26
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:       39      -111.126      0.321616   0.000908148      0.4931     0.04931       55
Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
Chain 1:       41      -111.126      0.373032    0.00159772      0.1782           1       58
Chain 1: Optimization terminated normally:
Chain 1:   Convergence detected: relative gradient magnitude is below tolerance
```

```
GP_MLE_I
```

```
$par
         rho        alpha        sigma
  0.07883712  99.99999925  11.38052032

$value
[1] -111.1259

$return_code
[1] 0

$theta_tilde
            rho alpha    sigma
[1,] 0.07883712   100 11.38052
```

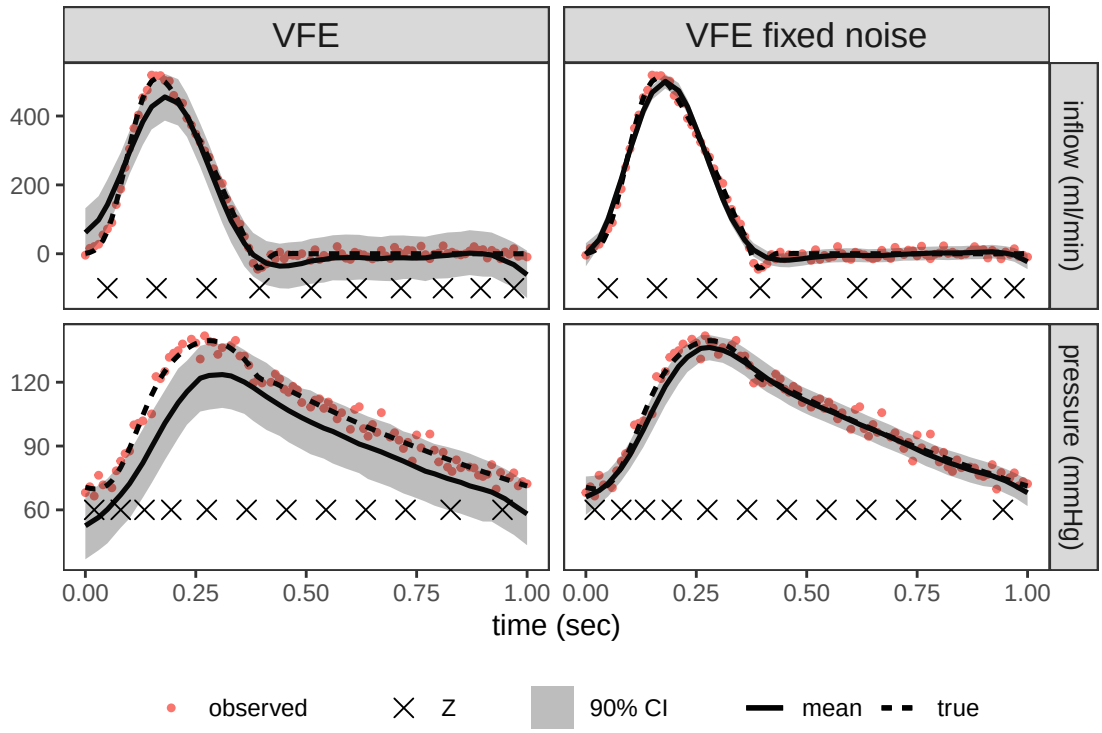
```
sigma_I_MLE = GP_MLE_I$par["sigma"]
```

```
data_pred_VFE_MLE = data_pred_VFE
data_pred_VFE_MLE$sigmaP = sigma_P_MLE
data_pred_VFE_MLE$sigmaI = sigma_I_MLE
pred_VFE_MLE = stan(file = 'STAN/Approximations/VFE/MLE_sigma/WK2_delta_VFE_predictions.stan',
                 data = data_pred_VFE_MLE ,
                 chains = 1, iter = 1, seed=123,
                 algorithm = "Fixed_param")
```

```
SAMPLING FOR MODEL 'WK2_delta_VFE_predictions' NOW (CHAIN 1).
Chain 1: Iteration: 1 / 1 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0 seconds (Warm-up)
Chain 1:                2.55029 seconds (Sampling)
Chain 1:                2.55029 seconds (Total)
Chain 1:
```

```
pp_VFE_MLE = rbind(post_mu_CIs.fn(post_pred=pred_VFE_MLE)$Psmr, post_mu_CIs.fn(post_pred=pred_VFE_MLE)$I
pp_VFE_MLE$output = c(rep("pressure (mmHg)", nP_pred),rep("inflow (ml/min)", nP_pred))
pp_VFE_MLE$model = "VFE fixed noise"
pp_VFE = rbind(post_mu_CIs.fn(post_pred=pred_VFE)$Psmr, post_mu_CIs.fn(post_pred=pred_VFE)$Ismr)
pp_VFE$output = c(rep("pressure (mmHg)", nP_pred),rep("inflow (ml/min)", nP_pred))
pp_VFE$model = "VFE"
pred_df = rbind(pp_VFE_MLE,pp_VFE)
df_zP_VFE = data.frame(z=data_PI_Z_VFE$zP, y = rep(60, data_PI_Z_VFE$mP), model = "VFE", output = "press
df_zI_VFE = data.frame(z=data_PI_Z_VFE$zI, y = rep(-100, data_PI_Z_VFE$mI), model = "VFE", output = "in
df_zP_VFE_MLE = data.frame(z=data_pred_VFE_MLE$zP, y = rep(60, data_pred_VFE_MLE$mP), model = "VFE fixe
df_zI_VFE_MLE = data.frame(z=data_pred_VFE_MLE$zI, y = rep(-100, data_pred_VFE_MLE$mI), model = "VFE fi
df_z = rbind(df_zP_VFE, df_zI_VFE,df_zP_VFE_MLE,df_zI_VFE_MLE)
P_true = data.frame(val=Psim, time=time)
P_true$output = "pressure (mmHg)"
I_true = data.frame(val=flow, time=time)
I_true$output = "inflow (ml/min)"
true_out = rbind(P_true, I_true)

obsP = data.frame(value=data_PI$yP, time = data_PI$tP, output = "pressure (mmHg)")
obsI = data.frame(value=data_PI$yI, time = data_PI$tI, output = "inflow (ml/min)")
```

```
obs = rbind(obsP,obsI)

pl_pred=ggplot()+
  geom_point(data = obs, aes(y=value, x=time, colour = "observed"), shape = 20)+
  geom_line(data = pred_df, aes(y=mean, x=time, linetype = "mean"), size=0.9)+
  geom_line(data = true_out, aes(y=val, x=time, linetype="true"), size=0.9)+
  geom_ribbon(data = pred_df,aes(ymin=lower, ymax=upper, x=time, fill = "90% CI"), alpha = 0.3)+
  facet_grid(output~model,scales = "free")+
  geom_point(data = df_z, aes(x=z, y=y, shape="Z"), size=3)+
  scale_fill_manual("",values=c("90% CI" = "grey12"))+
  theme_bw()+xlab("time (sec)")+ylab("")+
  scale_shape_manual("", values = c("Z" = 4))+
  theme(#legend.position = "none",
        legend.title = element_blank(),
        legend.position="bottom",
        strip.text.x = element_text(size = 13),
        strip.text.y = element_text(size = 10))
(pl_pred=pl_pred + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()))
```




```
ggsave("figures/Appr_delta_pred_noise.pdf", plot = pl_pred, width = 16, height = 10, units = "cm")
```

```
sessionInfo()
```

```
R version 4.0.3 (2020-10-10)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS Big Sur 10.16

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
```

```
locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] forcats_0.5.1        stringr_1.4.0        dplyr_1.0.7
 [4] purrr_0.3.4          readr_2.1.2          tidyr_1.2.0
 [7] tibble_3.1.6         tidyverse_1.3.1      rstan_2.21.3
[10] ggplot2_3.3.5        StanHeaders_2.21.0-7

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.8          lubridate_1.8.0    prettyunits_1.1.1  ps_1.6.0
 [5] assertthat_0.2.1    digest_0.6.29      utf8_1.2.2         cellranger_1.1.0
 [9] R6_2.5.1            backports_1.4.1    reprex_2.0.1       stats4_4.0.3
[13] evaluate_0.14       httr_1.4.2         pillar_1.7.0       rlang_1.0.0
[17] readxl_1.3.1        rstudioapi_0.13    callr_3.7.0        rmarkdown_2.11
[21] labeling_0.4.2      loo_2.4.1          munsell_0.5.0      broom_0.7.12
[25] compiler_4.0.3      modelr_0.1.8       xfun_0.29          pkgconfig_2.0.3
[29] pkgbuild_1.3.1      htmltools_0.5.2    tidyselect_1.1.1   gridExtra_2.3
[33] codetools_0.2-18    matrixStats_0.61.0 fansi_1.0.2        crayon_1.4.2
[37] tzdb_0.2.0          dbplyr_2.1.1       withr_2.4.3        grid_4.0.3
[41] jsonlite_1.7.3      gtable_0.3.0       lifecycle_1.0.1    DBI_1.1.2
[45] magrittr_2.0.2      scales_1.1.1       RcppParallel_5.1.5 cli_3.1.1
[49] stringi_1.7.6       farver_2.1.0       fs_1.5.2           xml2_1.3.3
[53] ellipsis_0.3.2      generics_0.1.2     vctrs_0.3.8        tools_4.0.3
[57] glue_1.6.1          hms_1.1.1          processx_3.5.2     parallel_4.0.3
[61] fastmap_1.1.0       yaml_2.2.2         inline_0.3.19      colorspace_2.0-2
[65] rvest_1.0.2         knitr_1.37         haven_2.4.3
```