USING COMPILED CODE IN `POMP`

AARON A. KING

Contents

1. A two-dimensional Ornstein-Uhlenbeck process.

Let's look again at our example of the discrete-time 2-D Ornstein-Uhlenbeck process with normal measurement error. Recall that the unobserved Ornstein-Uhlenbeck (OU) process $X_t \in \mathbb{R}^2$ satisfies

$$X_t = A\,X_{t-1} + \xi_t.$$

The observation process is

$$Y_t = B\,X_t + \varepsilon_t.$$

In these equations, A and and B are 2×2 constant matrices; ξ_t and ε_t are mutually-independent families of i.i.d. bivariate normal random variables. We let $\sigma\sigma^T$ be the variance-covariance matrix of ξ_t, where σ is lower-triangular; likewise, we let $\tau\tau^T$ be that of ε_t.

Since many of the methods we will use require us to simulate the process and/or measurement models many times, it is a good idea to use native (compiled) codes for the computational heavy lifting. The package includes some C codes that were written to implement the OU example. Read the source (file 'ou2.c') for details.

```
> ou2.rprocess <- function(xstart, times, params,
+     paramnames, ...) {
+     nvar <- nrow(xstart)
+     npar <- nrow(params)
+     nrep <- ncol(xstart)
+     ntimes <- length(times)
+     parindex <- match(paramnames, rownames(params)) -
+         1
+     array(.C("ou2_adv", X = double(nvar * nrep *
+         ntimes), xstart = as.double(xstart), par = as.double(params),
+         times = as.double(times), n = as.integer(c(nvar,
+             npar, nrep, ntimes)), parindex = as.integer(parindex),
+         DUP = FALSE, NAOK = TRUE, PACKAGE = "pomp")$X,
+         dim = c(nvar, nrep, ntimes), dimnames = list(rownames(xstart),
+             NULL, NULL))
+ }
> ou2.dprocess <- function(x, times, params, log,
+     ...) {
+     nvar <- nrow(x)
+     npar <- nrow(params)
```

```
+     nrep <- ncol(x)
+     ntimes <- length(times)
+     parindex <- match(paramnames, rownames(params)) -
+         1
+     array(.C("ou2_pdf", d = double(nrep * (ntimes -
+         1)), X = as.double(x), par = as.double(params),
+         times = as.double(times), n = as.integer(c(nvar,
+             npar, nrep, ntimes)), parindex = as.integer(parindex),
+         give_log = as.integer(log), DUP = FALSE,
+         NAOK = TRUE, PACKAGE = "pomp")$d, dim = c(nrep,
+         ntimes - 1))
+ }
> ou2 <- pomp(times = seq(1, 100), data = rbind(y1 = rep(0,
+     100), y2 = rep(0, 100)), t0 = 0, rprocess = ou2.rprocess,
+     dprocess = ou2.dprocess, dmeasure = "normal_dmeasure",
+     rmeasure = "normal_rmeasure", obsnames = c("y1",
+         "y2"), paramnames = c("alpha.1", "alpha.2",
+         "alpha.3", "alpha.4", "sigma.1", "sigma.2",
+         "sigma.3", "tau"), statenames = c("x1",
+         "x2"))
```

We'll specify some parameters:

```
> p <- c(alpha.1 = 0.9, alpha.2 = 0, alpha.3 = 0,
+     alpha.4 = 0.99, sigma.1 = 1, sigma.2 = 0,
+     sigma.3 = 2, tau = 1, x1.0 = 50, x2.0 = -50)
```

```
> tic <- Sys.time()
> ou2 <- simulate(ou2, params = p, nsim = 1000,
+     seed = 800733088)[[1]]
> toc <- Sys.time()
> print(toc - tic)
```

```
Time difference of 3.976824 secs
```

Fig. 1 plots the data.

The `pomp` object we just created is included in the package: use `data(ou2)` to retrieve it.

A. A. King, Departments of Ecology & Evolutionary Biology and Mathematics, University of Michigan, Ann Arbor, Michigan 48109-1048 USA

E-mail address: `kingaa at umich dot edu`

URL: `http://www.umich.edu/~kingaa`

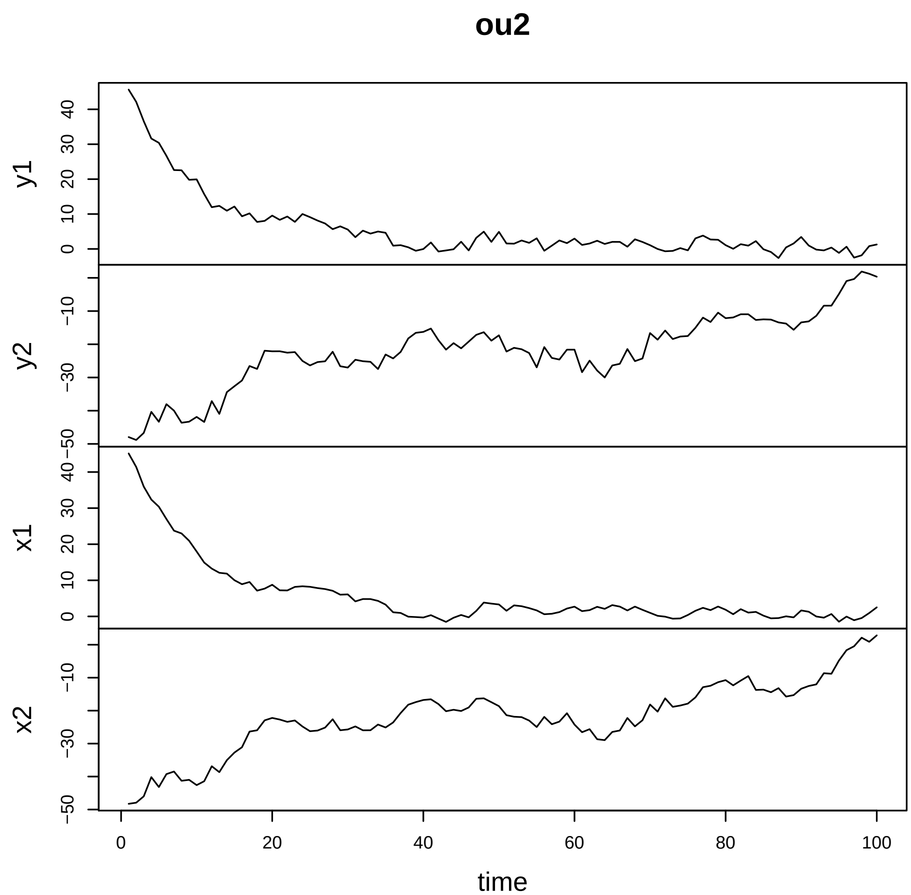

FIGURE 1. One realization of the two-dimensional OU process.