Registering the Daily Montreal Temperatures

Annotated Analyses in Matlab

Before you begin, don't forget to add the path to the functional data analysis functions. On my system, this is achieved by the command

```
addpath('../fdaM')
```

The Data

The data are 12410 daily mean temperatures at Montreal over the 34 years from 1961 to 1994 (in leap years temperatures for Feb. 28 and 29 were averaged). The following code inputs the data from a file, stripping off the year values, and reshaping them into a single vector. The data in the file are in tenths of a degree Celsius, and we also convert them to degrees Celsius.

```
fid     = fopen('montreal.txt');
mtltemp = fscanf(fid,'%d');
mtltemp = reshape(mtltemp,366,34);
mtltemp = mtltemp(2:366,:);
mtltemp = reshape(mtltemp,365*34,1)./10;
mtltempmean = mean(mtltemp);
```

Now we define the number of observations, and the time values with a unit of one year.

```
N = length(mtltemp);
daytime  = ((1:N)-0.5)./365;
```

Smoothing the Data

We want to replace the large number of discrete observations by a smooth function that captures as much of the variability as we need. We settled on an order 6 B-spline basis using 500 basis functions. This gives us a knot about every 25 days. We are undoubtably missing some detail, but the computations will be heavy even with this number of basis functions, and this looked like a reasonable compromise that would serve the goal of registration adequately.

```
nbasis = 500;
norder =   6;
basis  = create_bspline_basis([0,34], nbasis, norder);
```

```
mtlfd  = data2fd(mtltemp, daytime, basis);
```

We now set up a vector of smoothed temperature values for every day, and evaluate the standard deviation of the difference between the actual daily temperatures and their smoothed values. This turned out to be 4.3 degrees Celsius.

```
mtlsmthvec = eval(mtlfd, daytime);
residual   = mtltemp-mtlsmthvec;
stderr0    = sqrt((N/(N-nbasis)).*mean(residual.^2));
```

Estimating the Mean Annual Temperature

Now we need to get an estimate of the annual temperature variation by combining the data across the 34 years. This we achieve by smoothing the data again, but this time with a Fourier series basis with a period of one year. The use of the Fourier series basis ensures that the estimated annual variation will be strictly periodic, and the one year period forces the estimate to average across the years. Nine basis functions were judged sufficient.

```
basisF1 = create_fourier_basis([0,34],9,1);
mtlfd0  = data2fd(mtltemp,daytime,basisF1);
```

As above, we set up a vector of daily values of this strictly periodic trend, and estimate the standard deviation of the residuals, which came out to be 4.7 deg. C.

```
mtl0vec = eval(mtlfd0, daytime);
stderr1   = sqrt((N/(N-9)).*mean((mtltemp-mtl0vec).^2));
```

Here is the code to produce the plot of the temperatures for mid 1987 to mid 1989, along with the two types of smooth curves.

```
iyear = 89;
m1 = (iyear - 61)*365 + 1;
m2 = (iyear - 60)*365;
i  = iyear - 60;
index = max([m1-183,1]):min([m2+183,34*365]);
time1 = daytime(index)+1960;
xlo = min(time1);
xhi = max(time1);
plot(time1, mtltemp(index),                       'k.', ...
     time1, mtlsmthvec(index),                    'k-', ...
     time1, mtl0vec(index),                       'k--',...
     [xlo, xhi], [mtltempmean, mtltempmean],       'k:',...
     [daytime(m1)+1960,daytime(m1)+1960],[-35,35], 'k:',...
     [daytime(m2)+1960,daytime(m2)+1960],[-35,35], 'k:')
xlabel('\fontsize{16} Year')
```

```
ylabel('\fontsize{16} Degrees Celsius')
axis([xlo, xhi, -35, 35]);
```

Registering the Smooth Trend to the Periodic Trend

We now want to use function `registerfd` to aline the features of the smooth curve using 500 B-spline basis functions to the strictly periodic curve using 9 Fourier series. In order to use `registerfd`, however, we must first define another functional data object, that for function *W(t)* defining the warping function. If we use 140 basis functions for this purpose and order 5 B-spline basis functions, this effectively gives us four knots per year.
This should be sufficient to capture phase variation down to a few weeks at least.

```
nwbasis = 140;
nworder =   5;
wbasis  = create_bspline_basis([0,34], nwbasis, nworder);
Wfd0    = fd(zeros(nwbasis,1), wbasis);
```

Now we are ready to do the registration, which will define the registered smooth curve in functional data object `mtlregfd` and the warping function *h(t)* defined by functional data object `mtlwarpfd.`

```
regstr    = registerfd(mtlfd0, mtlfd, Wfd0);
mtlregfd  = regstr.regfd;
mtlwarpfd = regstr.Wfd;
```

We set up vectors of daily values for both the registered curve and the warping function.

```
mtlregvec  = eval(mtlregfd, daytime);
mtlwarpvec = eval_mon(daytime, mtlwarpfd);
mtlwarpvec = mtlwarpvec.*34./max(mtlwarpvec);
```

Here is the code for plotting the three types of smooth trend from mid 1987 to mid 1989.

```
plot(time1, mtlsmthvec(index),                     'k--',...
     time1, mtlregvec(index),                      'k-', ...
     time1, mtl0vec(indedex),                      'k-.',...
     [xlo, xhi], [mtltempmean, mtltempmean],       'k:', ...
     [daytime(m1)+1960,daytime(m1)+1960],[-20,25],'k:', ...
     [daytime(m2)+1960,daytime(m2)+1960],[-20,25],'k:')
xlabel('\fontsize{16} Year')
ylabel('\fontsize{16} Temperature (deg C)')
axis([xlo, xhi, -20, 25]);
```

We can plot the warping function for the same period.

```
plot(time1, (mtlwarpvec(index)+1960 - time1).*365,'k-', ...
     [xlo, xhi], [0,0],                             'k:', ...
     [daytime(m1)+1960,daytime(m1)+1960],[-30,20],'k:', ...
     [daytime(m2)+1960,daytime(m2)+1960],[-30,20],'k:')
xlabel('\fontsize{16} Year')
ylabel('\fontsize{16} Time Deformation (days)')
axis([xlo, xhi, -30, 20]);
```