- Create, evaluate, plot basis systems: $\{\phi_k\}_{k=1}^B$,
- in FDA toolbox: basis object.
Basis system with coefficients: $\{\phi_k\}_{k=1}^{B}, \mathbf{c}$
Create:

```plaintext
>>basis_Fourier = create_fourier_basis([0,2*pi],5);
>>coeffs = rand(5,2); %coefficients
    %B x N x m; B = |basis|, N = |repetitions|, m = dim(x_i)
>>fd_Fourier = fd(coeffs,basis_Fourier);
>>plot(fd_Fourier);
```
Labels to an fd object: \texttt{fdnames} object

\begin{verbatim}
>> fdnames_Fourier = cell(1,3);
>> fdnames_Fourier{1} = 'Time (t)';
>> fdnames_Fourier{2} = ['Trial1'; 'Trial2'];
>> fdnames_Fourier{3} = 'Approximation (y)';
>> fd_Fourier = fd(coeffs,basis_Fourier,fdnames_Fourier);
>> plot(fd_Fourier);
\end{verbatim}
Evaluate fd object, its derivatives; Lfd object

```matlab
>> tvec = linspace(0,pi,20);
>> xhat = eval_fd(tvec,fd_Fourier);
>> figure; plot(tvec,xhat);

>> xhat_D = eval_fd(tvec,fd_Fourier,1);
>> figure; plot(tvec,xhat_D);

% linear differential operators (Lfd object):
>> Lfd_D2 = int2Lfd(2); % L = D^2:
>> xhat_D2 = eval_fd(tvec,fd_Fourier,Lfd_D2);
>> figure; plot(tvec,xhat_D2);

>> omega=1; Lfd_harmonic = vec2Lfd([0,omega^2,0],[0,2*pi]);
>> xhat_harmonic = eval_fd(tvec,fd_Fourier,Lfd_harmonic);
>> figure; plot(tvec,xhat_harmonic);
```

Possible (see help Lfd): \( Lx(t) = \sum_{j=0}^{r} \beta_j(t) D^j x(t) \).
Objective function: \texttt{fdPar} object

\[ L = D^m: \]

\[
\texttt{>>fdPar\_my} = \texttt{fdPar(basis\_or\_fd\_my,m\_my,lambda\_my)};
\]

General \( L: \)

\[
\texttt{>>fdPar\_my} = \texttt{fdPar(basis\_or\_fd\_my,L\_my,lambda\_my)};
\]
Smoothing
Dataset description:

- heights of 54 subjects,
- measurements (31) for age=1 − 18.

Load data:

```matlab
>> N = 54; %# of subjects
>> n = 31; %# of measurements for each subject
>> fid = fopen('hgtf.dat','rt');
>> height = reshape(fscanf(fid,'%f'),[n,N]);
>> fclose(fid);
>> age = [1:0.25:2, 3:8, 8.5:0.5:18];
```
Smoothing with

1. least squares (small $B$).
2. regularization ($L$, $\lambda$).
3. regularization ($L$, $\hat{\lambda}$): $\hat{\lambda} = \arg \min_{\lambda} GCV(\lambda)$. 
Create spline basis ($B = 12$):

```matlab
>> rng_age = [1,18];
>> m = 6;
>> B = 12;  %massive regularization: B < 35
>> basis_spline12 = create_bspline_basis(rng_age,B,m);
   %uniformly placed knots
```

Smooth the data, plot:

```matlab
>> fd_smoothed12 = smooth_basis(age,height,basis_spline12);
>> plotfit_fd(height,age,fd_smoothed12);  %use the arrows!
```

Note: `basis` was used as `fdPar`; it makes sense (no $\lambda, L$).
Smoothing with regularization ($L = D^4$, $\lambda = 0.1$)

Create spline basis:

```matlab
>> rng_age = [1,18];
>> knots = age;
>> m = 6; %order
>> B = length(knots) + m - 2; %# of basis functions
>> basis_spline = create_bspline_basis(rng_age,B,m,knots);
```

Regularization:

```matlab
>> Lfd_spline = int2Lfd(4); %L=D^4
>> lambda = 1/10; %regularization parameter
>> fdPar_spline = fdPar(basis_spline,Lfd_spline,lambda);
```

Smooth the data, plot:

```matlab
>> fd_smoothed = smooth_basis(age,height,fdPar_spline);
>> plotfit_fd(height,age,fd_smoothed); %use the arrows!
```
Smoothing with regularization: \( \hat{\lambda} = \arg \min_{\lambda} GCV(\lambda) \)

Load data, create spline basis and L: done. The rest:

```matlab
>> log10lam = [-6:0.25:0];
>> gcv_save = zeros(length(log10lam),1);
>> for i = 1 : length(log10lam)
    >> fdPar_i = fdPar(basis_spline, Lfd_spline, 10^log10lam(i));
    >> [fd_smoothed_i,df_i,gcv_i]=smooth_basis(age,height,fdPar_i);
    >> gcv_save(i) = sum(gcv_i); %multiple subjects
>> end

Plot:
>> plot(log10lam, gcv_save);
>> xlabel('log_{10}(\lambda)');
>> ylabel('GCV(\lambda)');
```
Smoothing with regularization: $\log_{10}(\lambda) \leftrightarrow GCV(\lambda)$

Estimation: $\hat{\lambda} \approx 10^{-4}$. 
Smoothing with constraints
Monotone smoothing

\[ x(t) = \beta_0 + \beta_1 \int_{t_0}^{t} e^{W(u)} du. \]

- Relevant function: `smooth_monotone`.
- Other constraints:
  1. positivity: `smooth_pos`,
  2. pdf: `density_fd`.
Berkeley growth data: monotone smoothing

load data:

```matlab
>> N = 54; n = 31;
>> fid = fopen('hgtf.dat','rt');
>> height = reshape(fscanf(fid,'%f'),[n,N]);
>> fclose(fid);
>> age = [1:0.25:2, 3:8, 8.5:0.5:18]'; %n = |age|
```

basis, $L$, $\lambda$:

```matlab
>> rng_age = [1,18]; m = 6; B = length(age) + m - 2;
>> basis_spline = create_bspline_basis(rng_age,B,m,age);
>> fd_spline = fd(zeros(B,N),basis_spline); %J + init.
>> fdPar_spline = fdPar(fd_spline,3,10^(-0.5)); %L=D^3
```

monotone smoothing:

```matlab
[W_hat,beta_hat,height_hat] = smooth_monotone(age,height,fdPar_spline);
```
Objects:
- \textbf{fd}: $c^T \phi$.
- \textbf{fdnames}: labels for \textbf{fd}.
- \textbf{Lfd}: linear differential operator ($L$).
- \textbf{fdPar}: objective function ($J$).

We covered Chapter 4-5 in [2].