Karhunen-Loève (KL) Expansions

- Assume stochastic process $F(x,\theta) : D \times \Theta \to \mathbb{R}$ an L^2 random field on D
- With covariance function $Cov(x,y)$
- F can be written as

$$F(x,\theta) = \langle F(x,\theta) \rangle_\theta + \sum_{k=1}^{\infty} \sqrt{\lambda_k} f_k(x) \xi_k \tag{1}$$

- $f_k(x)$: eigenfunctions of $Cov(x,y)$
- λ_k: corresponding eigenvalues, all positive
- ξ_k: uncorrelated random variables, unit variance
  - Samples are obtained by projecting realizations of F onto f_k
  - Generally not independent
    - Special case: for Gaussian F, ξ_k are i.i.d. normal random variables
- The KL Expansion is *optimal*: of all possible orthonormal bases for $L^2(\Theta \times D)$ the above $\{f(x)\}$ minimize the mean-square error in a finite linear representation of $F(\cdot)$.

- Covariance Matrix:
  - specified analytically, e.g.

$$Cov(x,y) = \exp\left(-\frac{|x-y|^2}{c_l^2}\right) \tag{2}$$

    where $|x-y|$ is the distance between x and y and c_l is the correlation length.
  - estimated from realizations, e.g.

$$Cov(x,y) = \frac{1}{N_\theta}\sum_\theta (F(x,\theta) - \langle F(x,\theta)\rangle_\theta)(F(y,\theta) - \langle F(y,\theta)\rangle_\theta) \tag{3}$$

    where N_θ is the number of samples, and $\langle F(x,\theta)\rangle_\theta$ is the mean over the random field realizations at x.

Note: *Equation (3) requires a large number of realizations N_θ to estimate the covariance accurately. Algorithms capable to deal with limited number of samples are currently under development.*

KL Expansions - Numerical Approach - 2

- Estimate eigenvalues and eigenvectors for the Fredholm equation of second kind:

$$\int Cov(x,y)f(y)dy = \lambda f(x) \tag{4}$$

  - The Nystrom algorithm employs a discretization of the integral in the left-hand side of eq. (4)

$$\sum_{i=1}^{N_p} w_i Cov(x,y_i)f(y_i) = \lambda f(x) \tag{5}$$

Here w_i are the weights for the quadrature rule that uses N_p points y_i where realizations are provided. In a 1D configuration, one can employ the weights corresponding to the trapezoidal rule:

$$w_i = \begin{cases} \frac{y_2 - y_1}{2} & \text{if } i = 1, \\ \frac{y_{i+1} - y_{i-1}}{2} & \text{if } 2 \leq i < N_p, \\ \frac{y_{N_p} - y_{N_p-1}}{2} & \text{if } i = N_p, \end{cases} \tag{6}$$

KL Expansions - Numerical Approach - 3

- After further manipulation, eq. (5) is written as

$$Ag = \lambda g$$

  where $A = WKW$ and $g = Wf$, with W being the diagonal matrix $W_{ii} = \sqrt{w_i}$ and $K_{ij} = Cov(x_i, y_j)$.

- Since matrix A is symmetric, one can employ efficient algorithms to compute its eigenvalues λ_k and eigenvectors g_k. Currently **UQTk** relies on the *dsyevx* function provided by the LAPACK library.

- The KL eigenvectors are computed as $f_k = W^{-1} g_k$.

KL Expansions - Numerical Approach - 4

- Samples of random variables ξ_k are obtained by projecting realizations of the random process F on the eigenmodes f_k

$$\xi_k|_{\theta_l} = \left\langle F(x,\theta_l) - \langle F(x,\theta)\rangle_\theta \, , f_k(x)\right\rangle_x / \sqrt{\lambda_k}$$

  - ... or numerically

$$\xi_k|_{\theta_l} = \sum_{i=1}^{N_p} w_i \left(F(x_i,\theta_l) - \langle F(x_i,\theta)\rangle_\theta\right) f_k(x_i)/\sqrt{\lambda_k} \tag{7}$$

- If Gaussian process, ξ_k are *i.i.d.* normal RVs, i.e. automatically have first order Wiener-Hermite Polynomial Chaos Expansions (PCE).
- If not, the KL RVs can be converted to PCEs (not shown in the current example).

KL Example

The KL Expansion example can be run in two modes.

- **Mode 1**: The eigenvalues and eigenvectors are computed using analytical values for the covariance given by eq. (2), evaluated on an uniform grid in $[0, 1]$. In the example command-line below, "corr_len" represents the numerical value for the correlation length.

  ```
  ./kl_sample.x klcov corr_len
  ```

- **Mode 2**: The following procedure is used to generate random field (RF) realizations realizations and compute the associated KL Expansion.
  - The covariance given by eq. (2) is first evaluated on an uniform grid in $[0, 1]$. The resulting covariance matrix is then used to generate multivariate normal (MVN) realizations on the same grid.
  - The analytical values are then discarded, and a new (numerical) covariance matrix is evaluated from the MVN RF realizations, see eq. (3).
  - In addition to KL eigenvalues and eigenvectors, realizations of RVs ξ_k are also computed, using eq. (7).
- The command-line options are shown below:

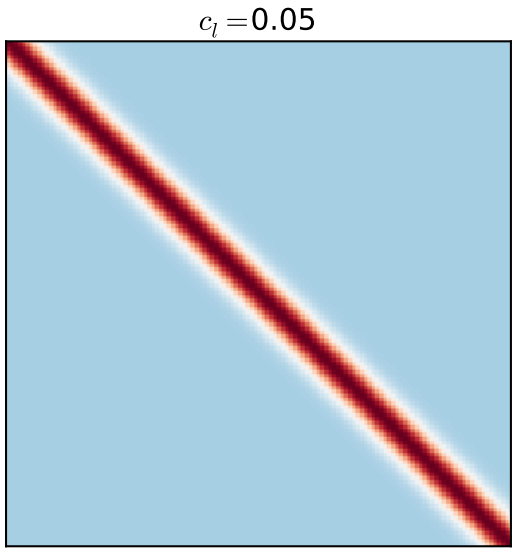
  ```
  ./kl_sample.x klsampl corr_len
  ```

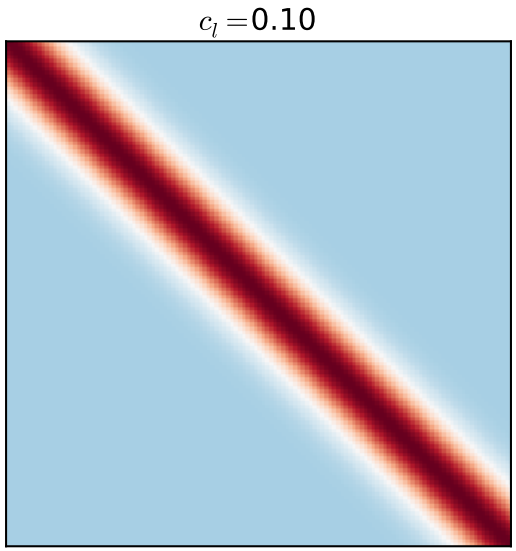
KL Example - Mode 1 - Analytical Covariance Matrix

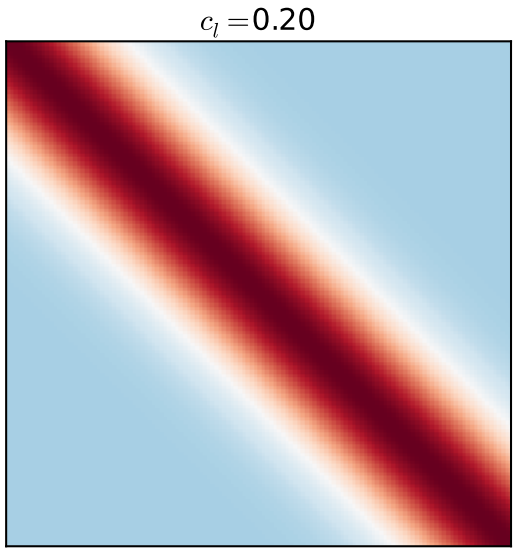
Three sets of results are generated for correlation lengths, $c_l = 0.05$, $c_l = 0.10$, and , $c_l = 0.20$:

```
./kl_sample.x klcov 0.05
./kl_sample.x klcov 0.10
./kl_sample.x klcov 0.20
```

The corresponding covariance matrices are shown below.



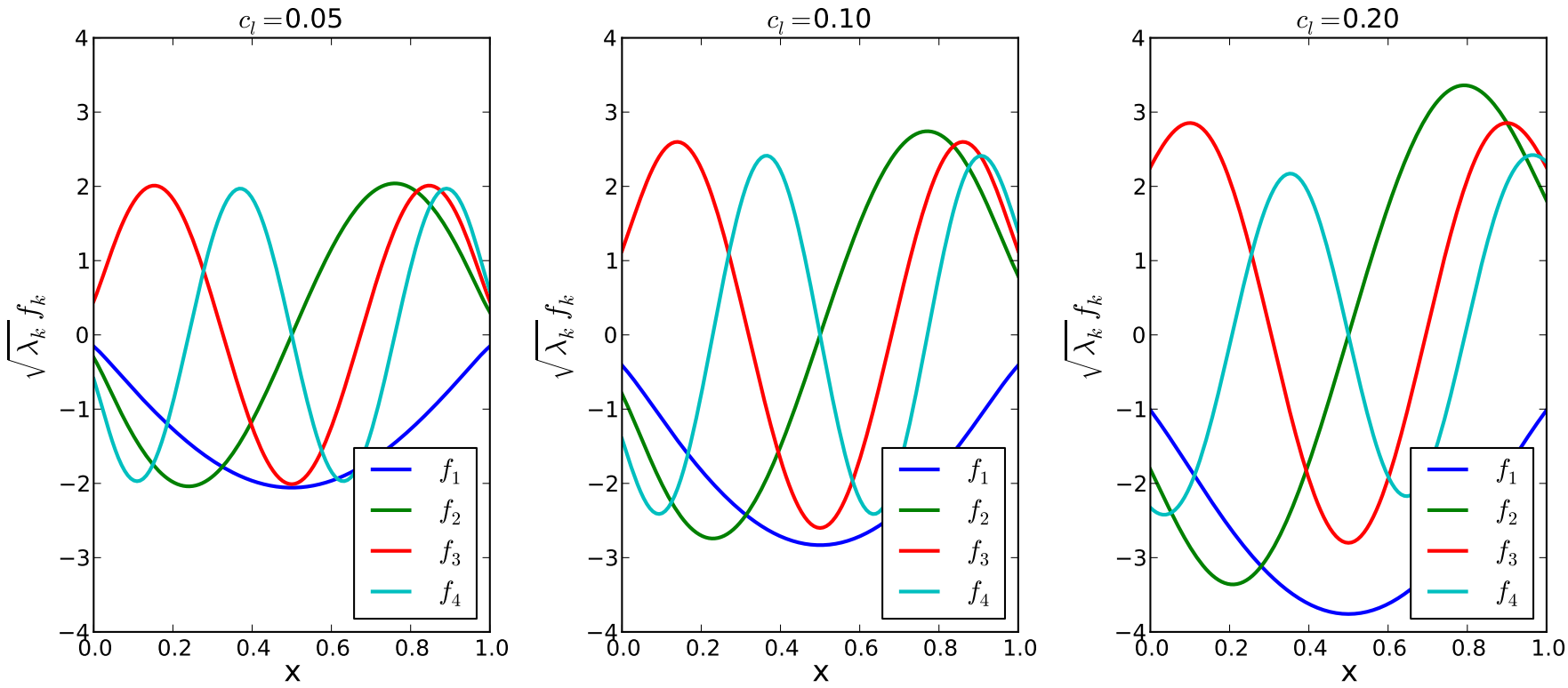




Plots were created using:

```
./mkplots.py anlcov SqExp 0.05
./mkplots.py anlcov SqExp 0.10
./mkplots.py anlcov SqExp 0.20
```

KL Example - Mode 1 - KL modes



Plots were created using:

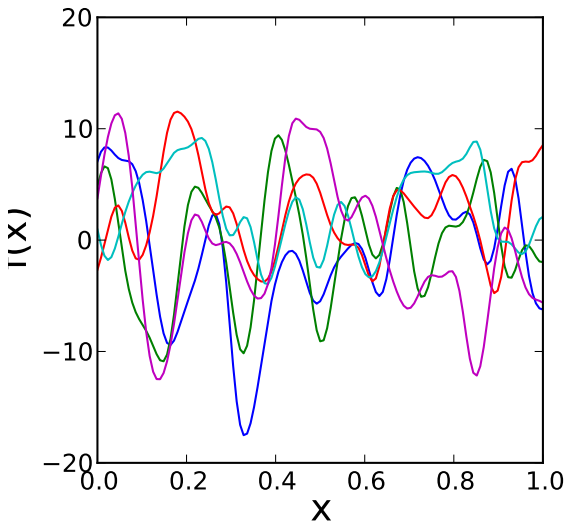
```
./mkplots.py anlKLevec SqExp 0.05
./mkplots.py anlKLevec SqExp 0.10
./mkplots.py anlKLevec SqExp 0.20
```

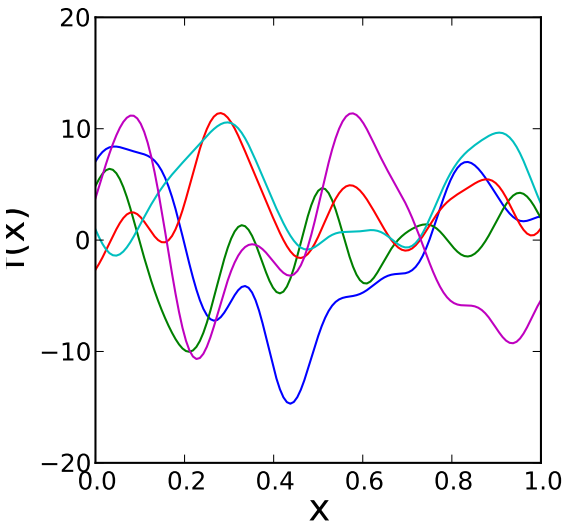
KL Example - Mode 2 - MVN realizations

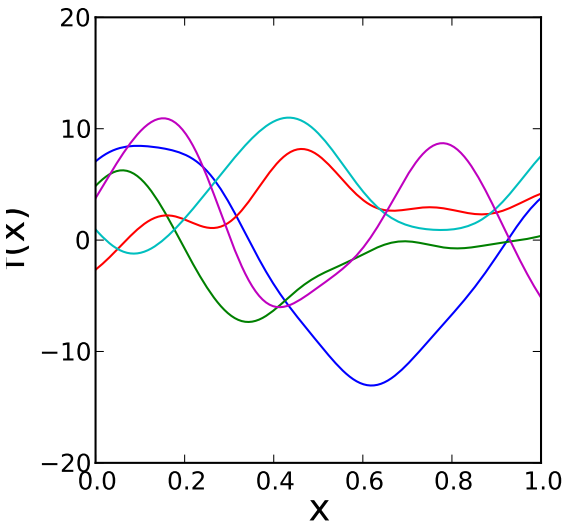
Three sets of results are generated for correlation lengths, $c_l = 0.05$, $c_l = 0.10$, and , $c_l = 0.20$:

```
./kl_sample.x klsampl 0.05
./kl_sample.x klsampl 0.10
./kl_sample.x klsampl 0.20
```

Sample realizations for each correlation length value are shown below:





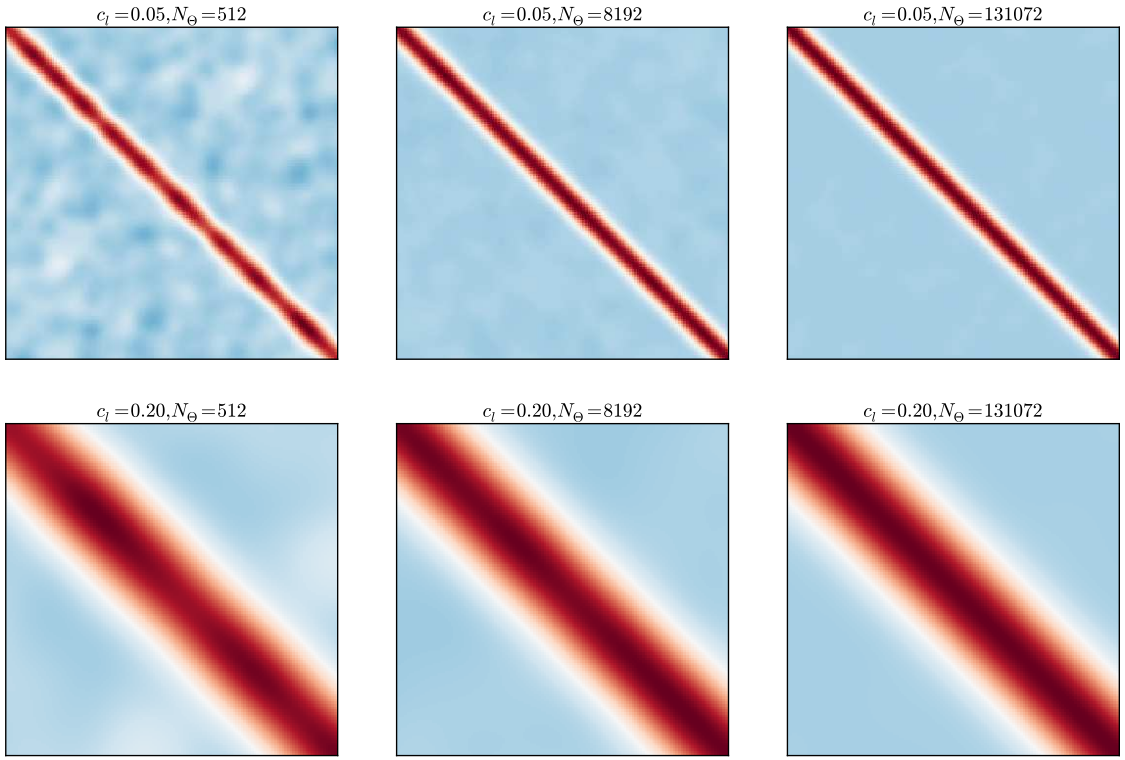


Plots were created using:

```
./mkplots.py samples 0.05
./mkplots.py samples 0.10
./mkplots.py samples 0.20
```

KL Example - Mode 2 - Covariance matrices

For each set of results, the covariance matrix is estimated incrementally based on $N_\Theta = (2^9, 2^{10}, \ldots, 2^{17})$ MVN realizations.



Plots were created using:

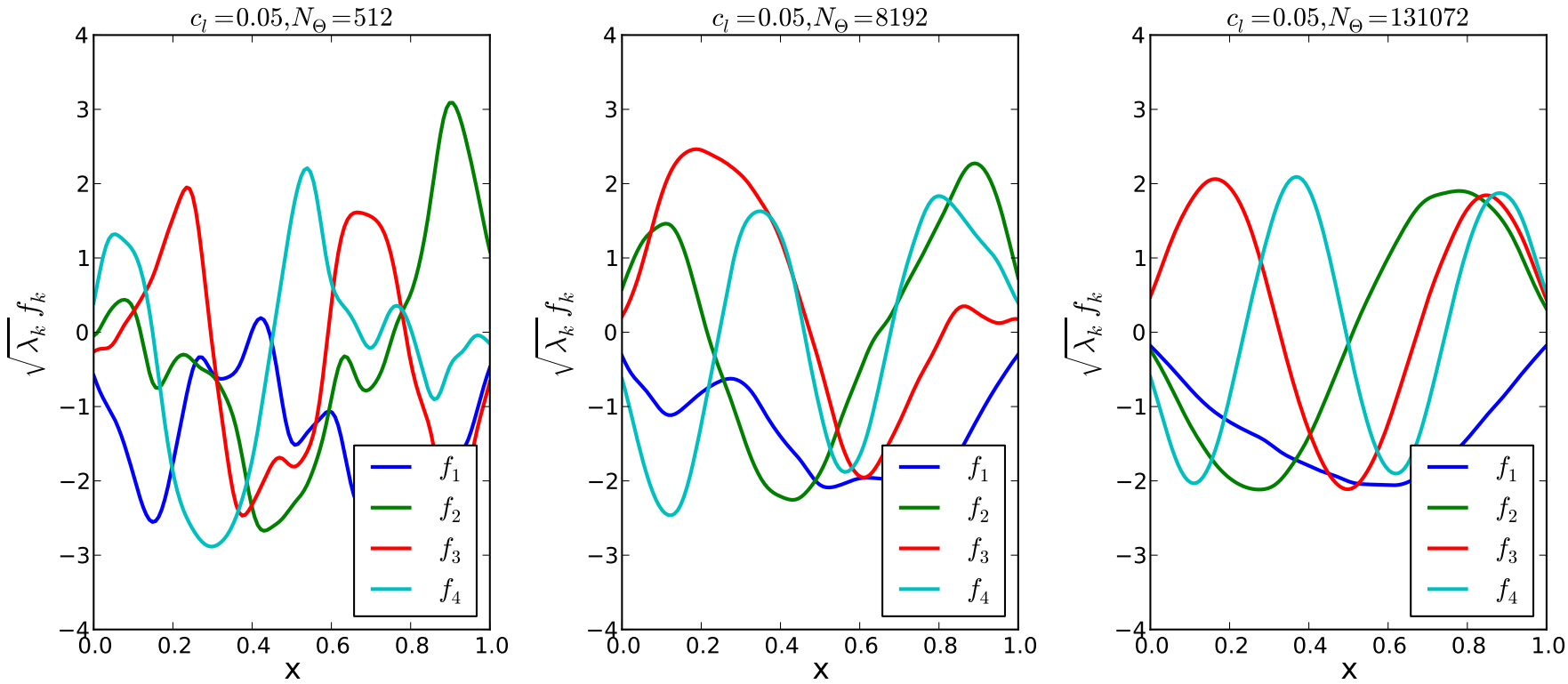
```
./mkplots.py numcov 0.05 nreal
```

and

```
./mkplots.py numcov 0.20 nreal
```

where “nreal” was set to $2^9 = 512$, $2^{13} = 8192$, and $2^{17} = 131072$

KL Example - Mode 2 - KL modes for $c_l = 0.05$

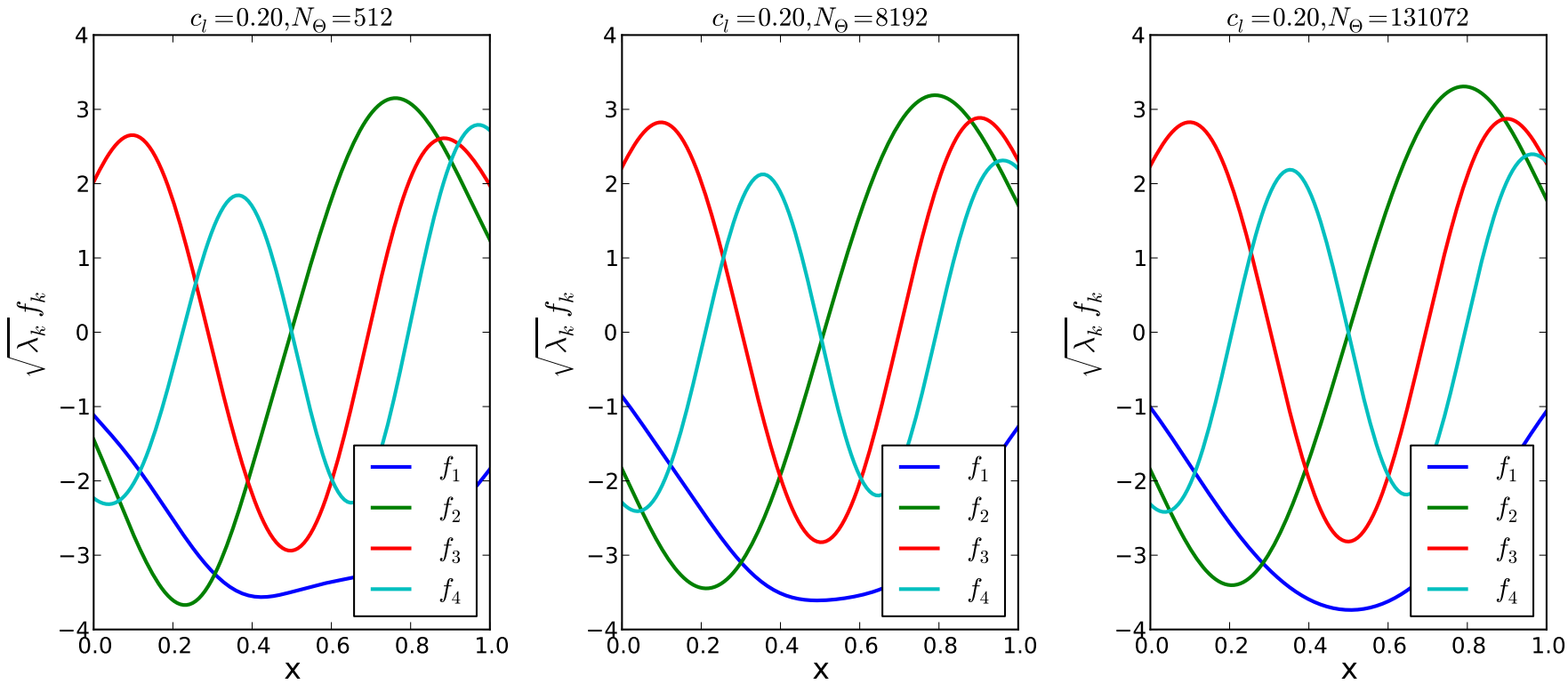


Plots were created using:

```
./mkplots.py numKLevec 0.05 nreal
```

where “nreal” was set to $2^9 = 512$, $2^{13} = 8192$, and $2^{17} = 131072$

KL Example - Mode 2 - KL modes for $c_l = 0.20$



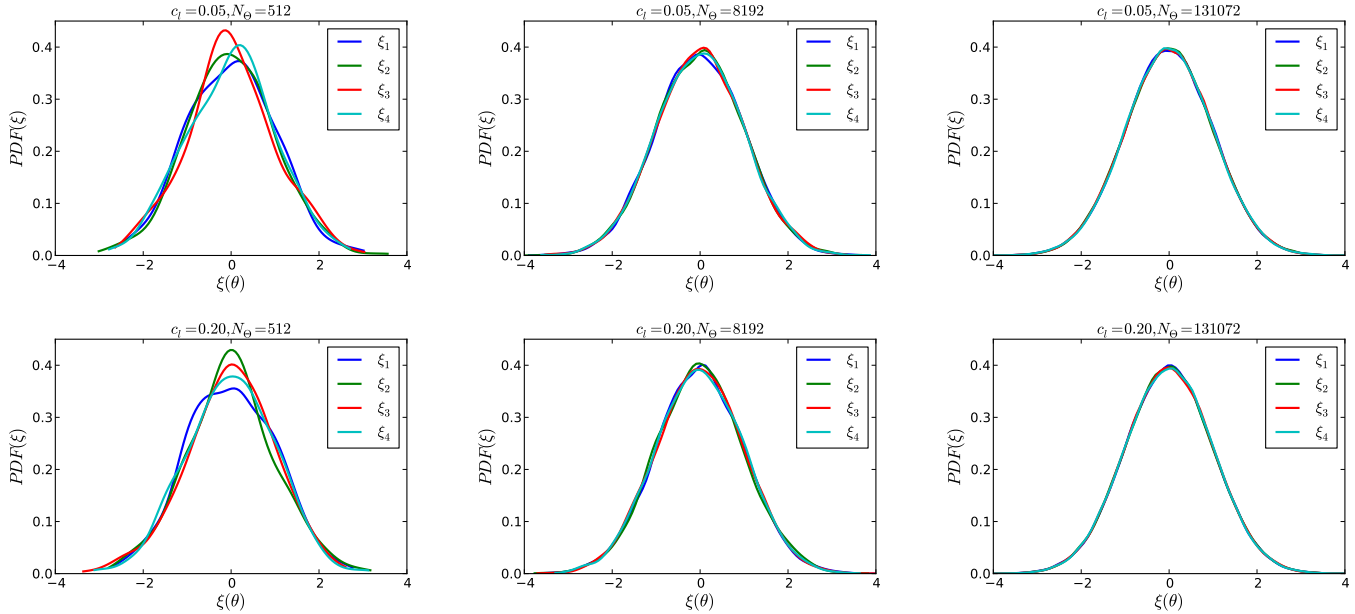
Plots were created using:

```
./mkplots.py numKLevec 0.20 nreal
```

where “nreal” was set to $2^9 = 512$, $2^{13} = 8192$, and $2^{17} = 131072$

KL Example - Mode 2 - PDF's of ξ_k

The PDF's for the random variables ξ_k are based on Kernel Density Estimations:



Plots were created using:

```
./mkplots.py xidata 0.05 nreal
```

and

```
./mkplots.py xidata 0.20 nreal
```

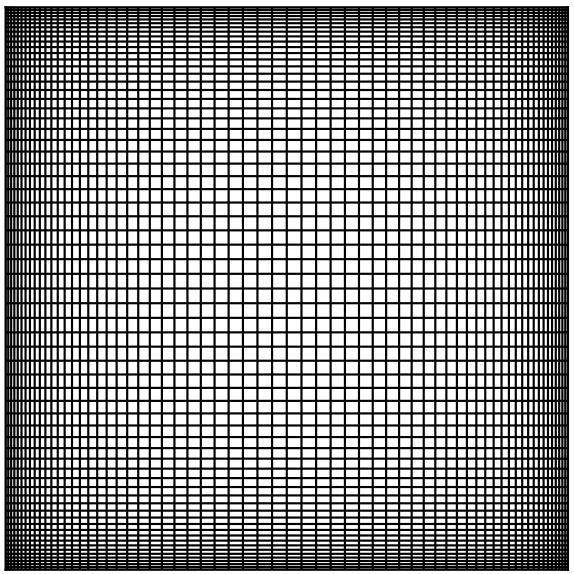
where "nreal" was set to $2^9 = 512$, $2^{13} = 8192$, and $2^{17} = 131072$

2D KL Example - Grid and Sample Realizations

A non-uniform grid can be constructed as

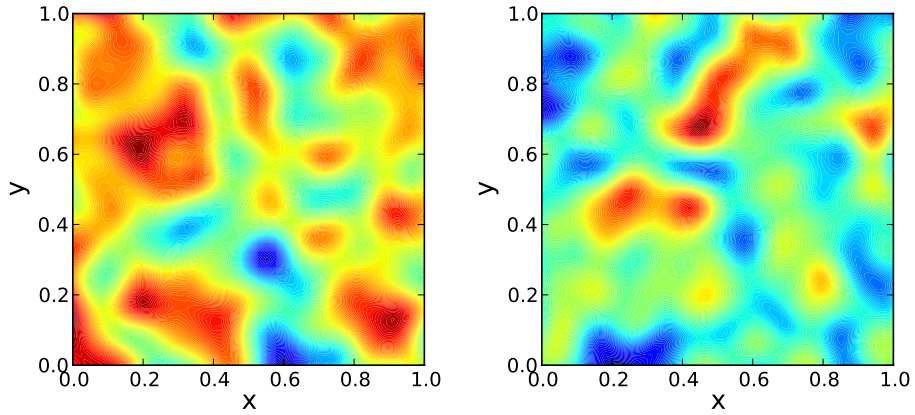
$$x = L_x \frac{(2\alpha+\beta)\frac{\beta+1}{\beta-1}^{\frac{\xi_i-\alpha}{1-\alpha}} + 2\alpha - \beta}{(2\alpha+1)\left(1+\frac{\beta+1}{\beta-1}^{\frac{\xi_i-\alpha}{1-\alpha}}\right)}$$

where $\xi_i = \frac{i-1}{N-1}$.

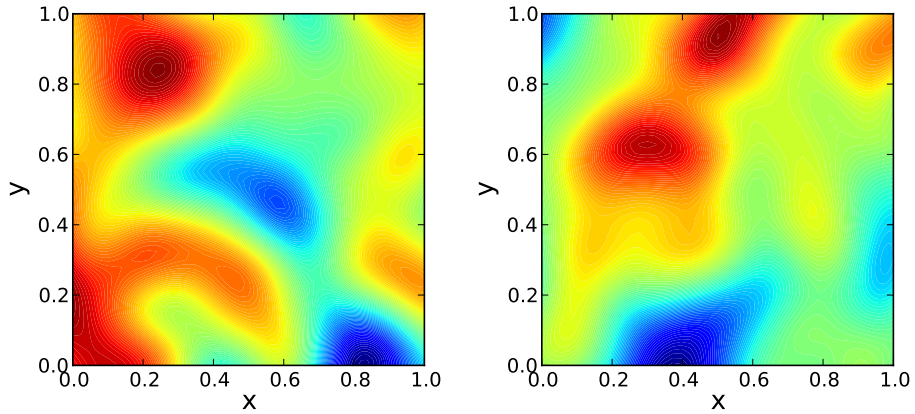


$\alpha = 1/2, \beta = 1.1$

$c_l = 0.1$



$c_l = 0.2$

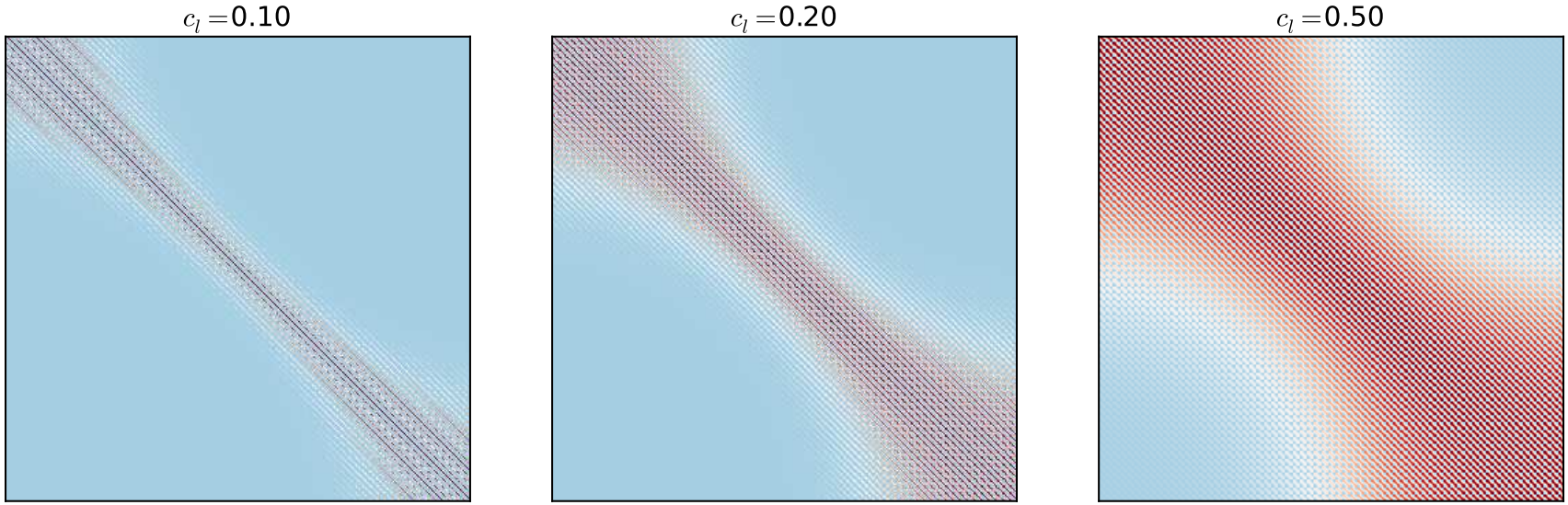


Plots were created using:

```
./mkplots.py samples2D 0.10 1024 2
./mkplots.py samples2D 0.20 1024 2
```

2D KL Example - Mode 1 - Covariance matrices

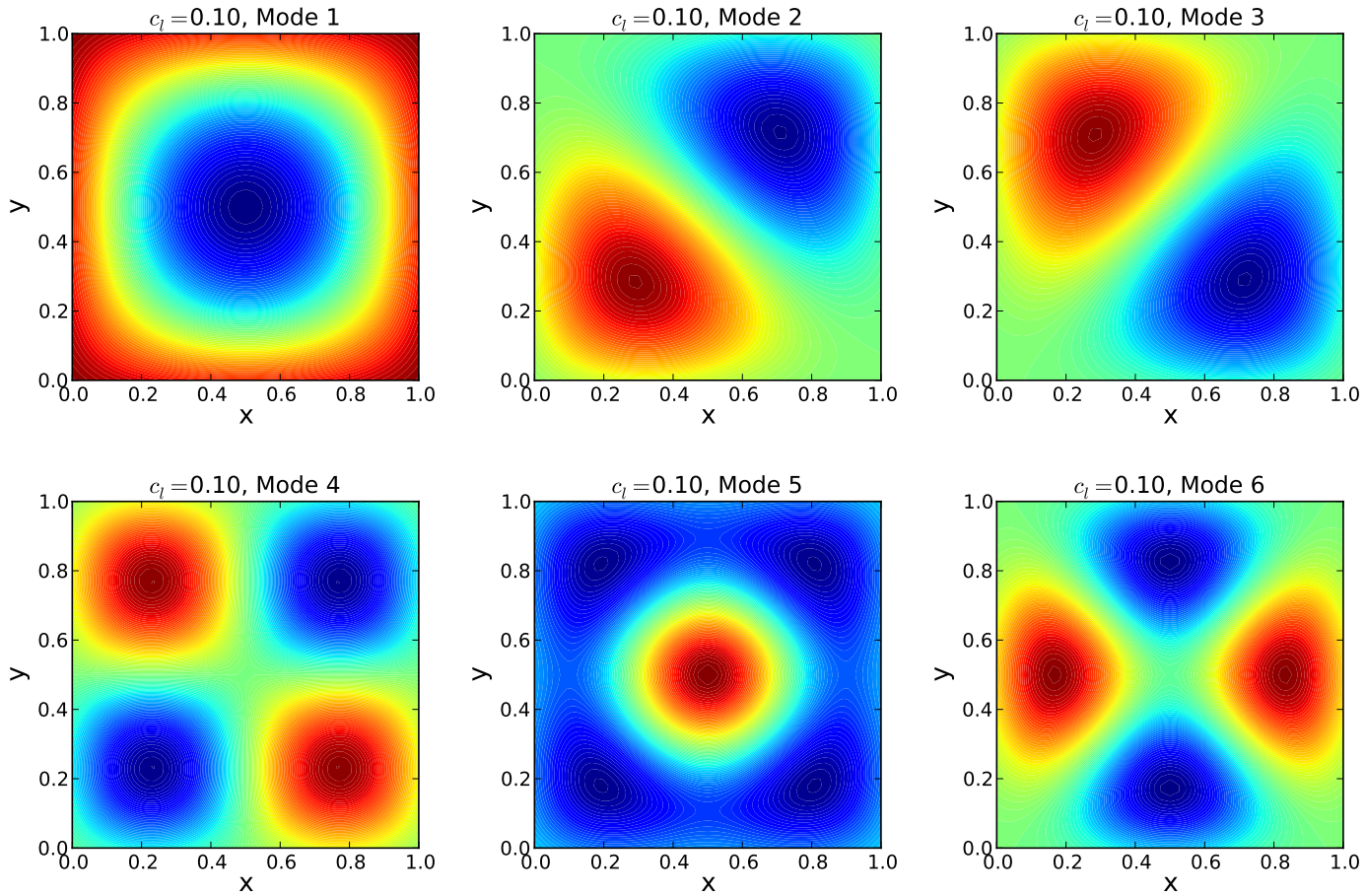
Analytical covariance matrices for $c_l = 0.10$, $c_l = 0.20$, and $c_l = 0.50$.



Plots were created using:

```
./mkplots.py anlcov2D SqExp 0.10
./mkplots.py anlcov2D SqExp 0.20
./mkplots.py anlcov2D SqExp 0.50
```

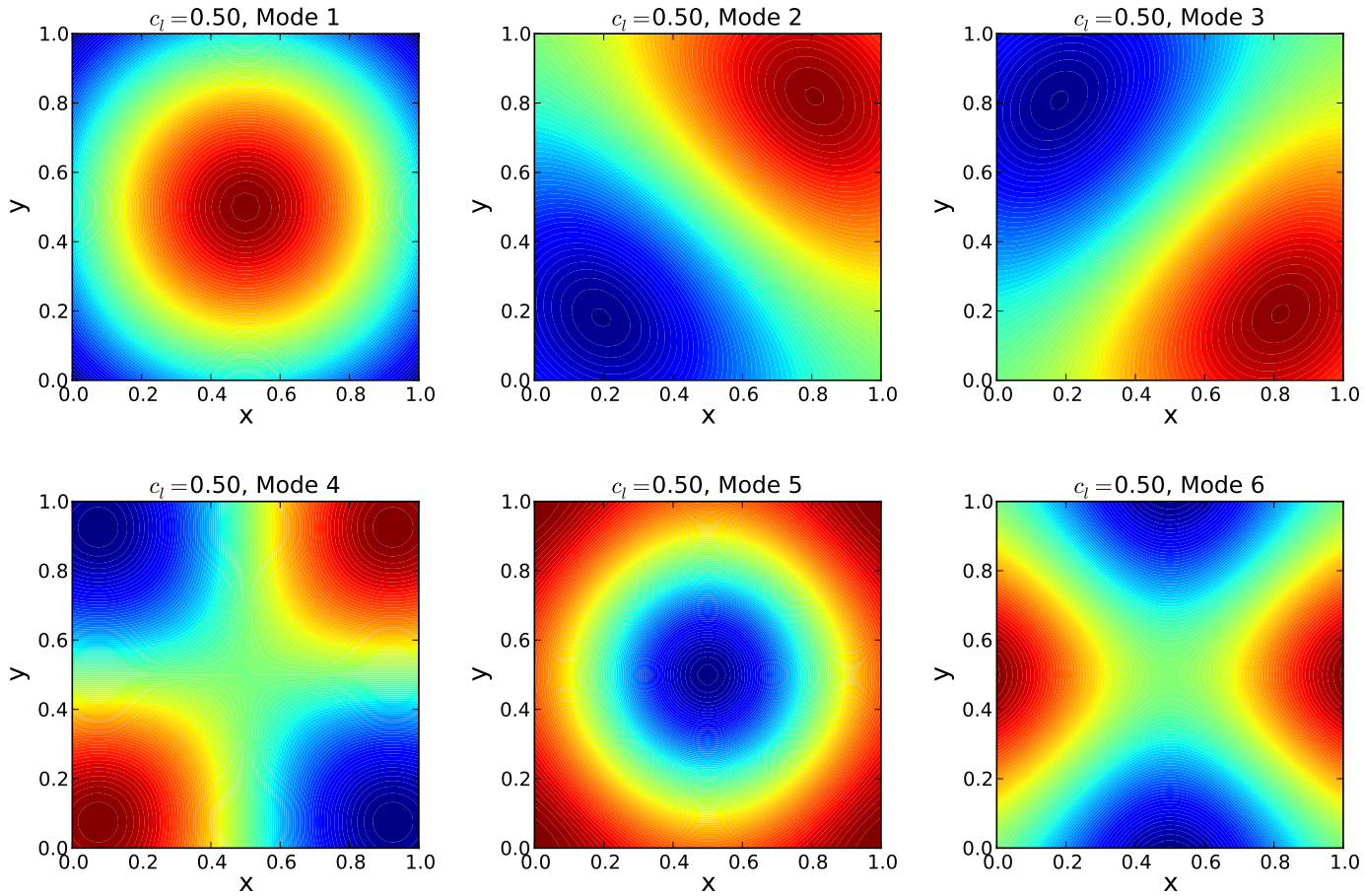
2D KL Example - Mode 1 - KL modes for $c_l = 0.10$



Plots were created using:

```
./mkplots.py anlKLevec2D SqExp 0.10
```

2D KL Example - Mode 1 - KL modes for $c_l = 0.50$

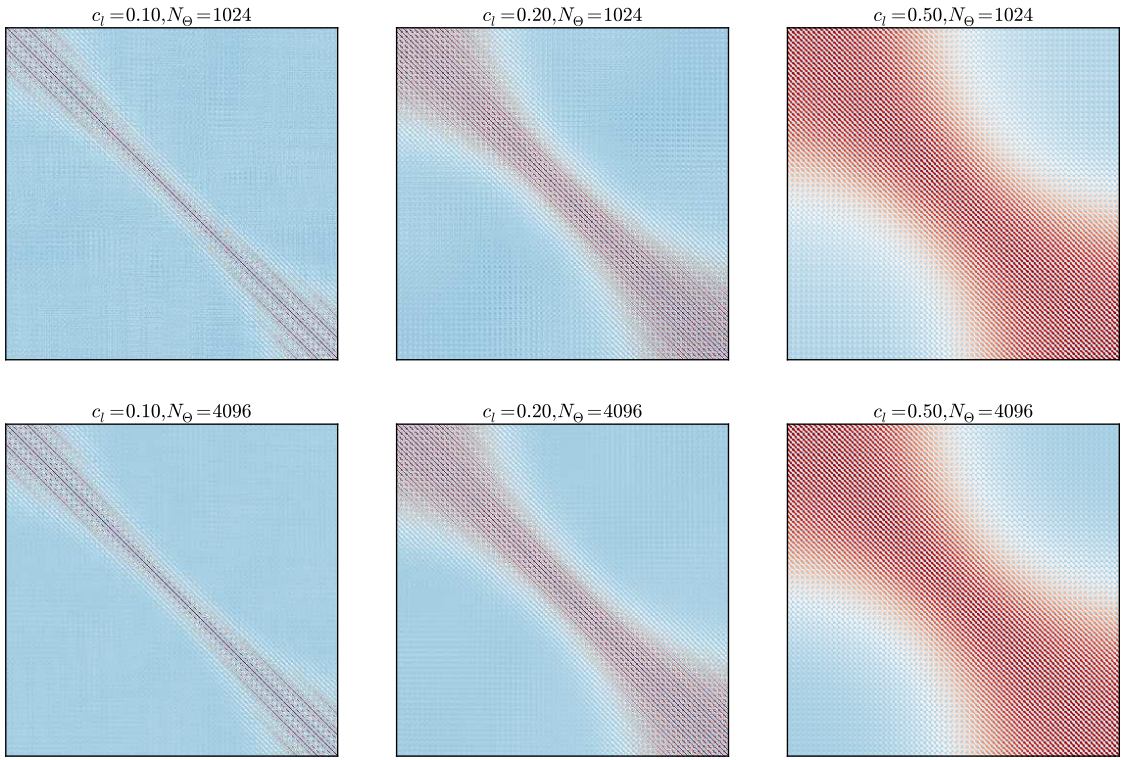


Plots were created using:

```
./mkplots.py anlKLevec2D SqExp 0.50
```

2D KL Example - Mode 2 - Covariance matrices

For each correlation length, c_l, the covariance matrix is estimated based on 1024 (top row) or 4096 (bottom row) MVN realizations.



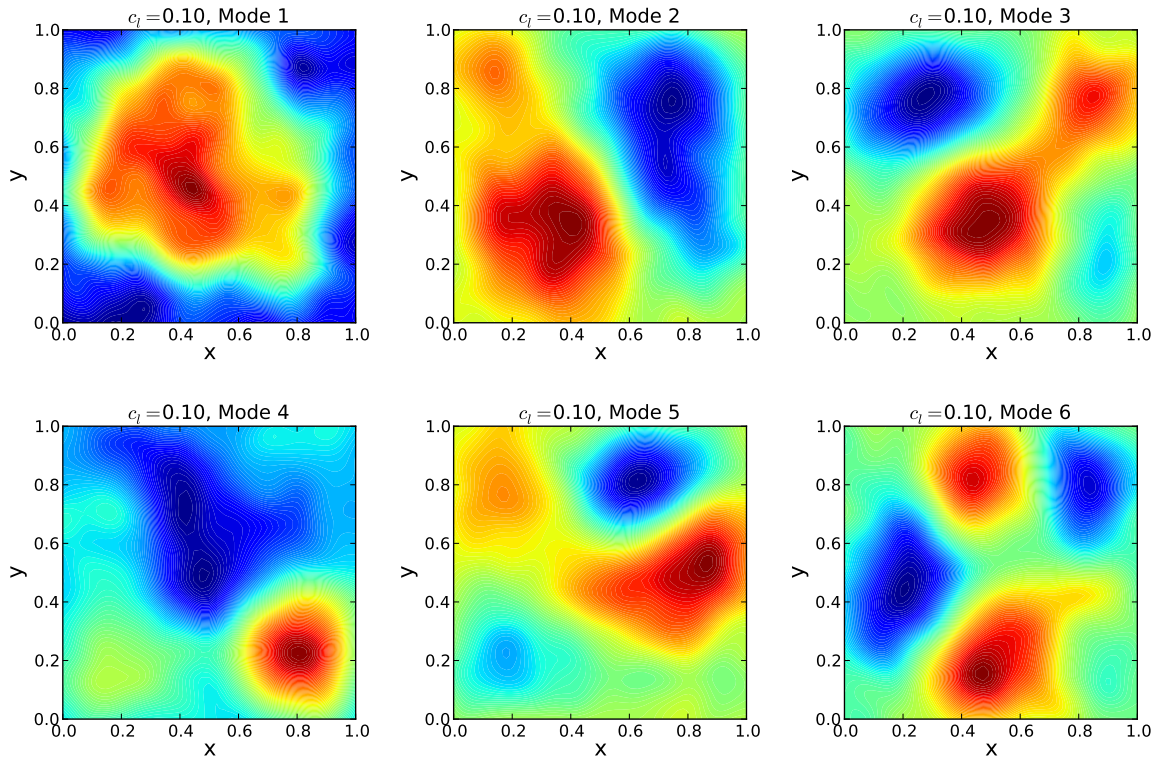
Plots were created using:

```
./mkplots.py numcov2D 0.10 nreal
./mkplots.py numcov2D 0.20 nreal
./mkplots.py numcov2D 0.50 nreal
```

where “nreal” was set to $2^{10} = 1024$, and $2^{14} = 4096$

2D KL Example - Mode 2 - KL modes for $c_l = 0.10$ - Numerical Covariance

The KL modes are estimated based on covariance matrices obtained from 4096 MVN realizations.

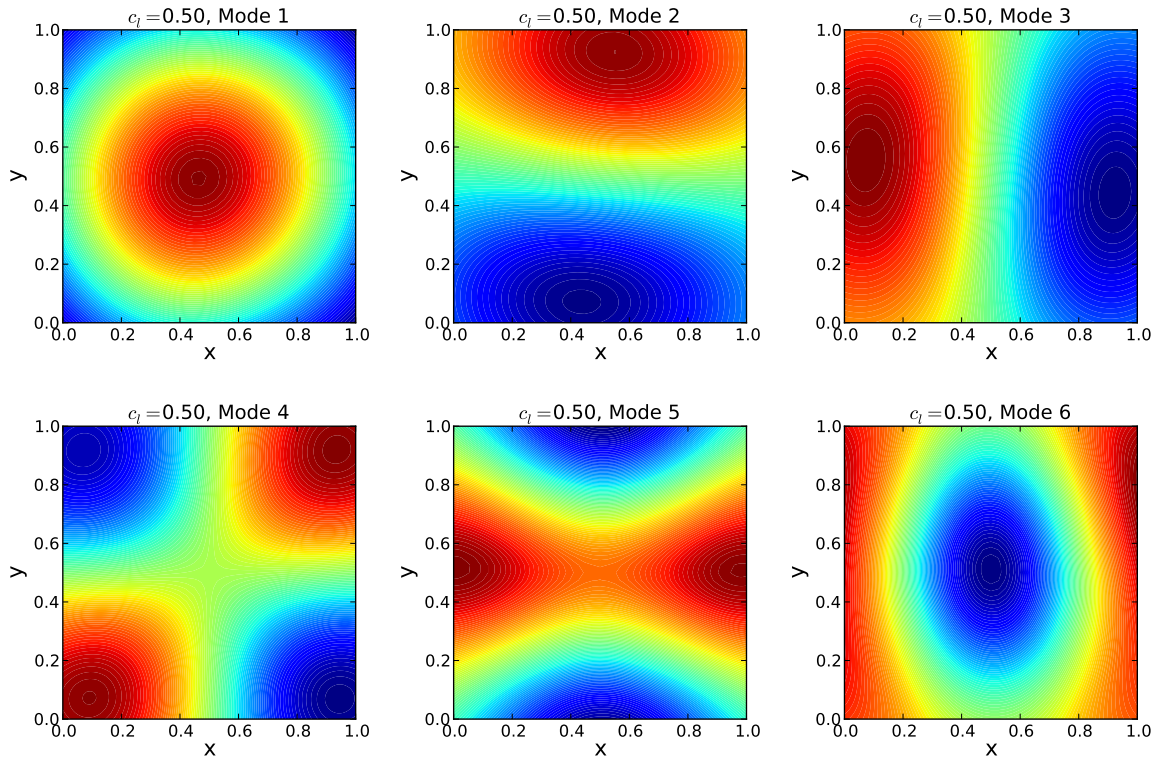


Plots were created using:

```
./mkplots.py numKLevec2D 0.10 4096
```

2D KL Example - Mode 2 - KL modes for $c_l = 0.50$ - Numerical Covariance

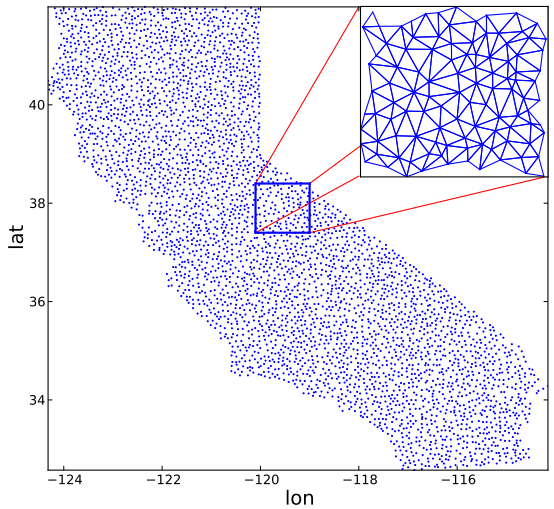
The KL modes are estimated based on covariance matrices obtained from 4096 MVN realizations.



Plots were created using:

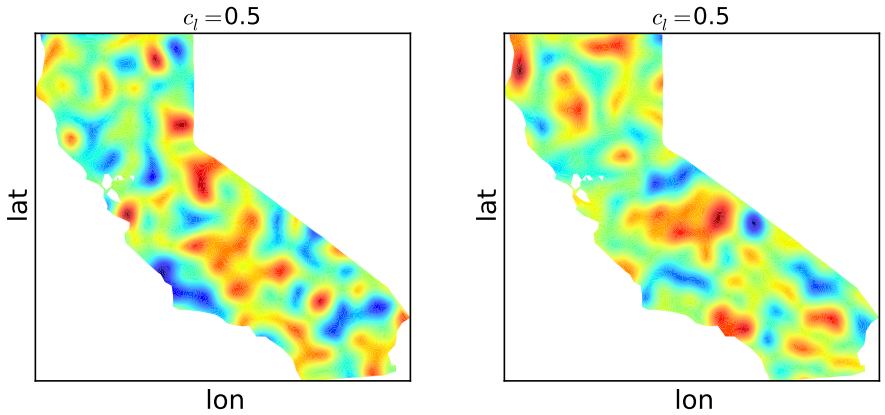
```
./mkplots.py numKLevec2D 0.50 4096
```

2D KL Example - Unstructured Grid

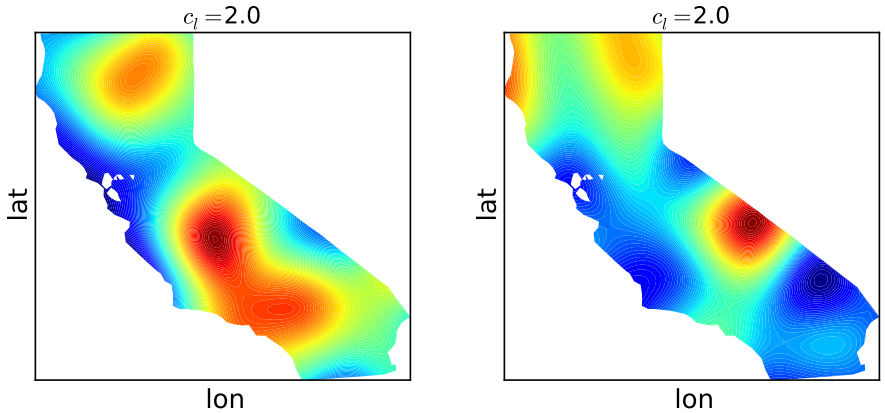


- 4096 grid points (*data/cali_grid.dat*)
- 8063 triangles (*data/cali_tria.dat*)

$c_l = 0.5^o$



$c_l = 2.0^o$

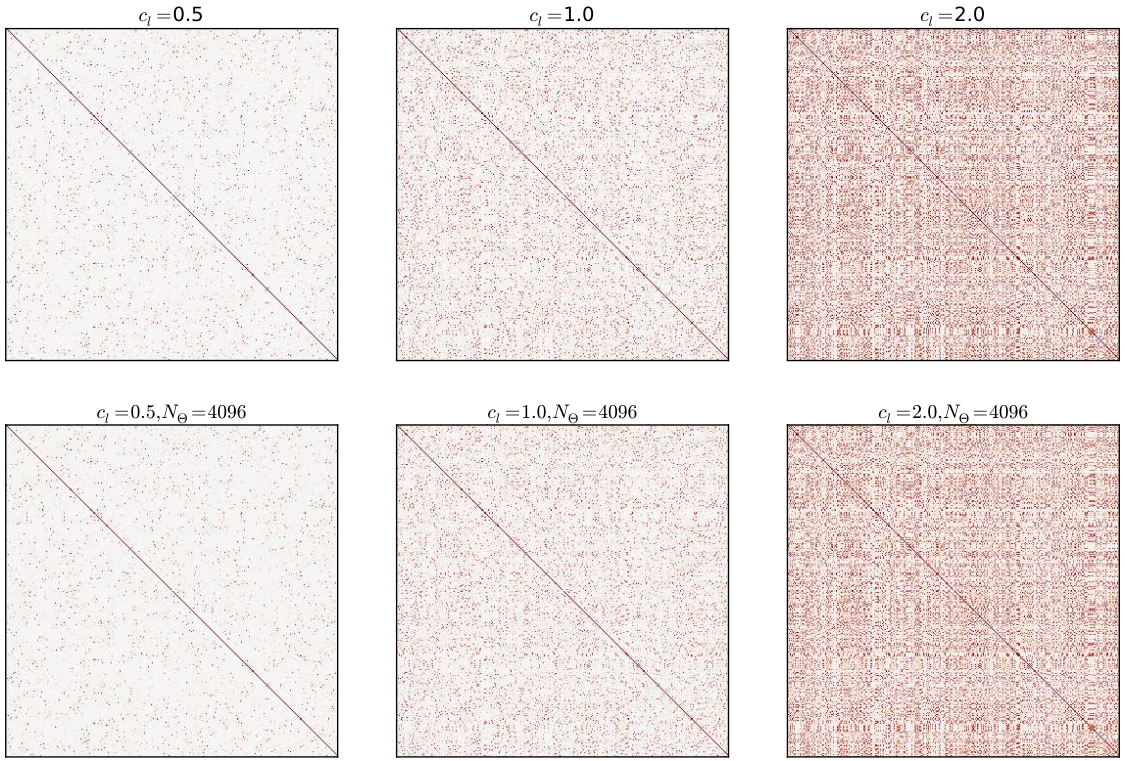


Plots were created using:

```
./mkplots.py samples2Du 0.5 4096 2
./mkplots.py samples2Du 2.0 4096 2
```

2D KL Example - Unstructured Grid - Covariance matrices

Analytical (top row) and numerical (bottom row) covariance matrices.
Numerical values are estimated based on 4096 MVN realizations.



Plots were created using:

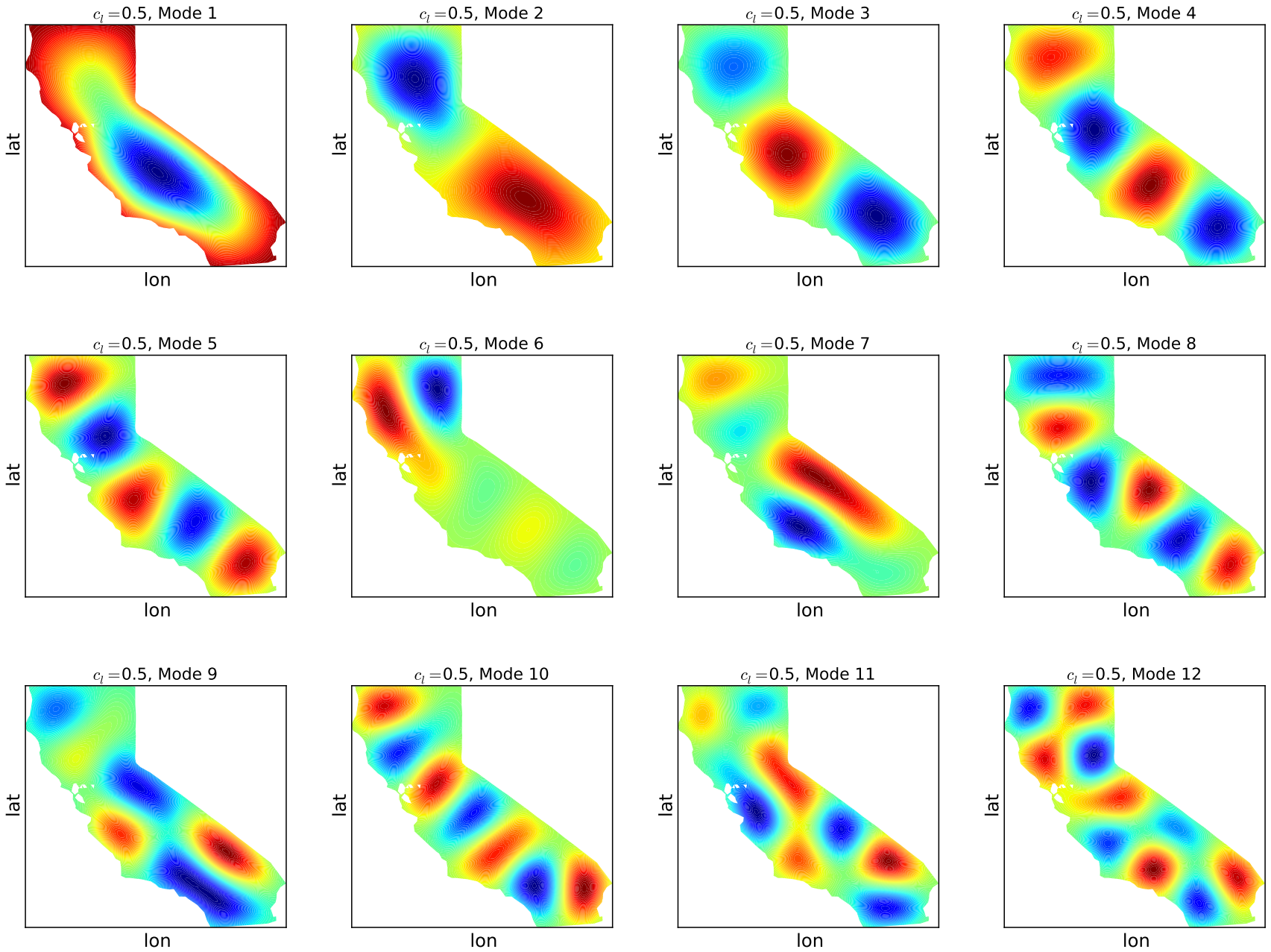
```
./mkplots.py numcov2Du SqExp cl
```

and

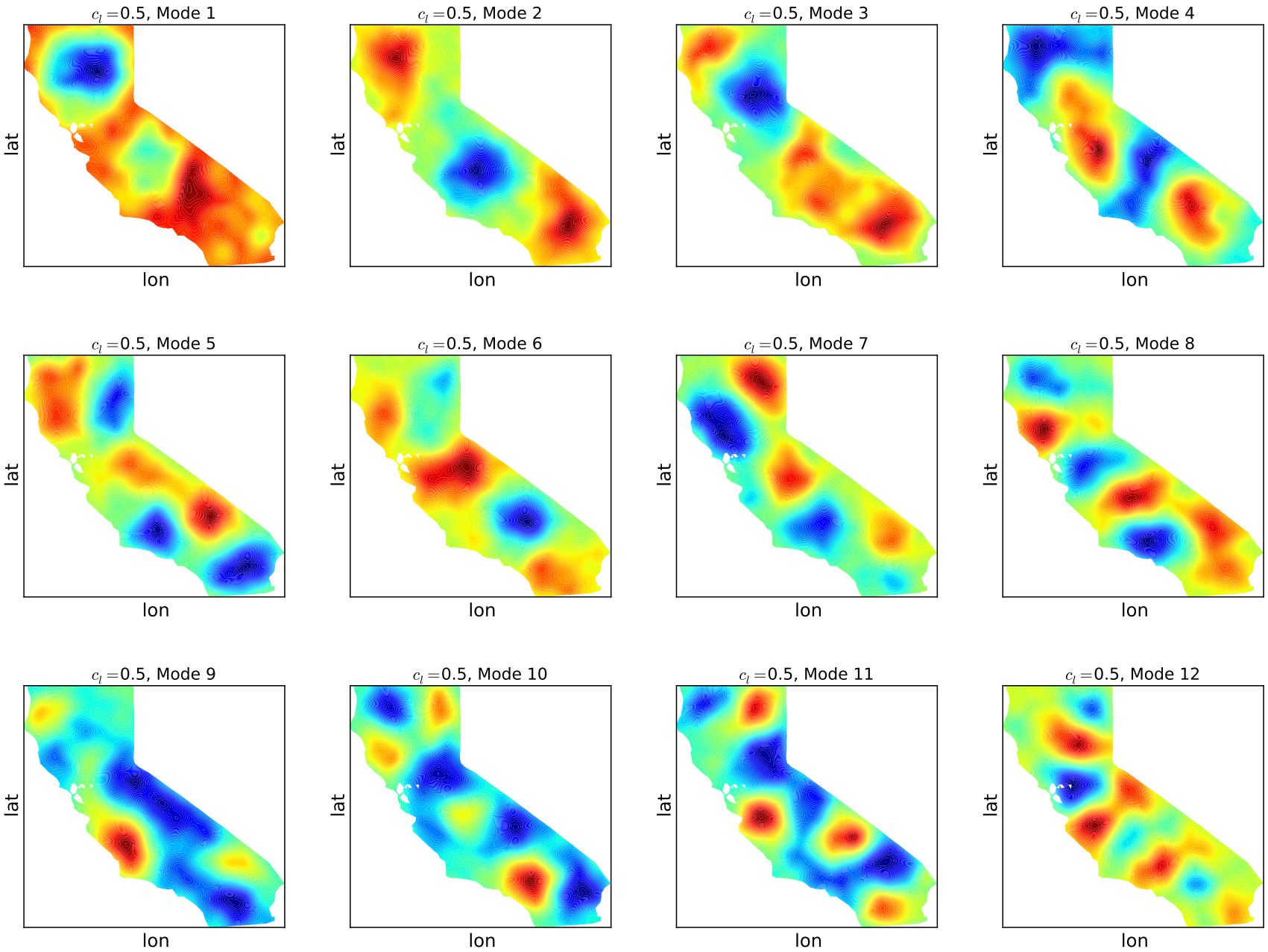
```
./mkplots.py numcov2Du cl 4096
```

where “cl” was set to 0.50, 1.0 and 2.0

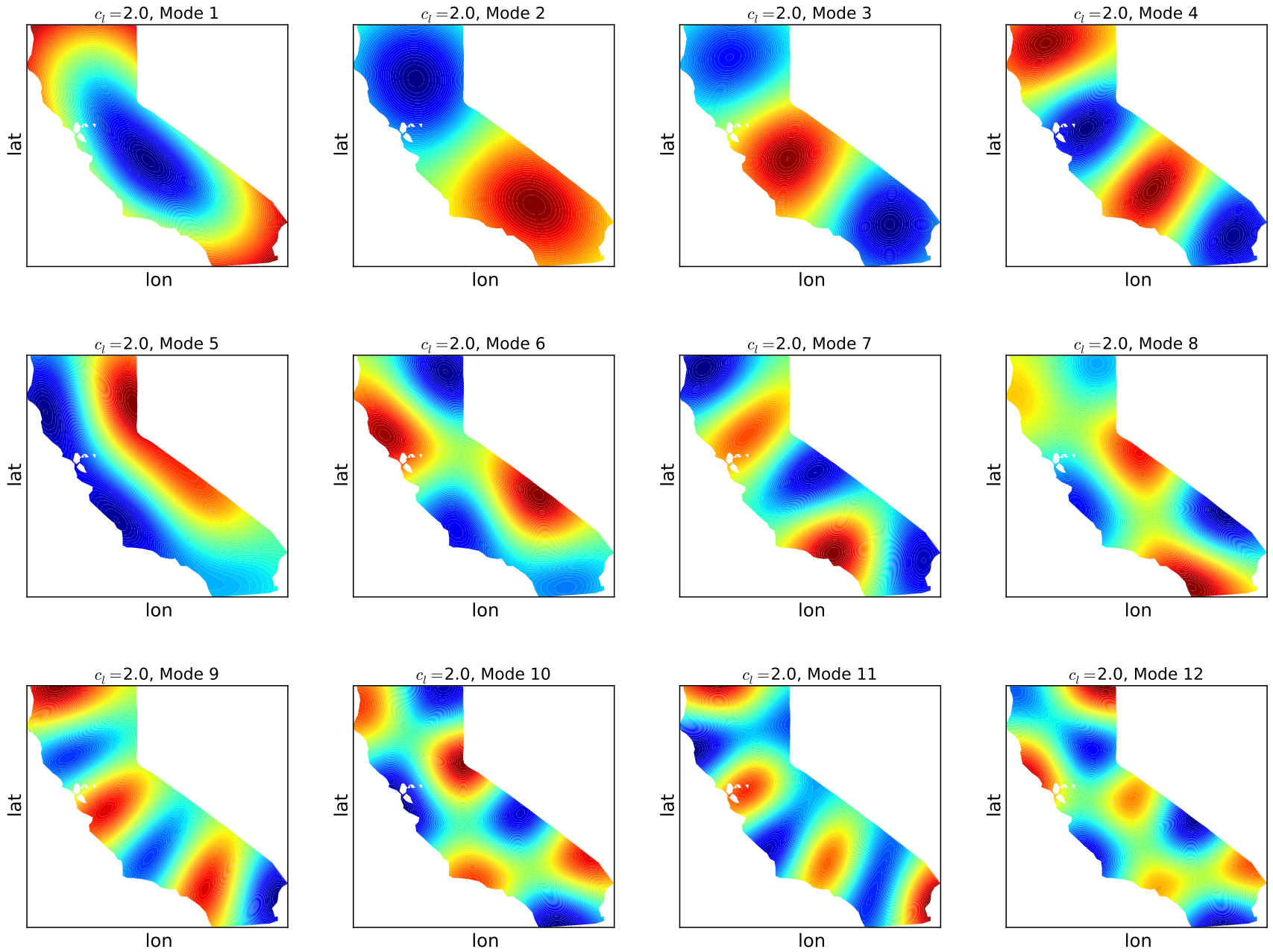
2D KL modes over California for $c_l = 0.5^o$ - Analytical Covariance



2D KL modes over California for $c_l = 0.5^o$ - Numerical Covariance



2D KL modes over California for $c_l = 2.0^o$ - Analytical Covariance



2D KL modes over California for $c_l = 2.0^o$ - Numerical Covariance