easydata

Unknown Author

January 20, 2014

In [1]:

```
%pylab inline
import sys
import glob
import os
import random

import numpy as np
import numpy.linalg as linalg
import numpy.random as rnd
from mpl_toolkits.mplot3d.axes3d import Axes3D

import GPy
```

```
Populating the interactive namespace from numpy and matplotlib
```

Part I

Easydata GPLVM tests

1 Generating the data

In [2]:

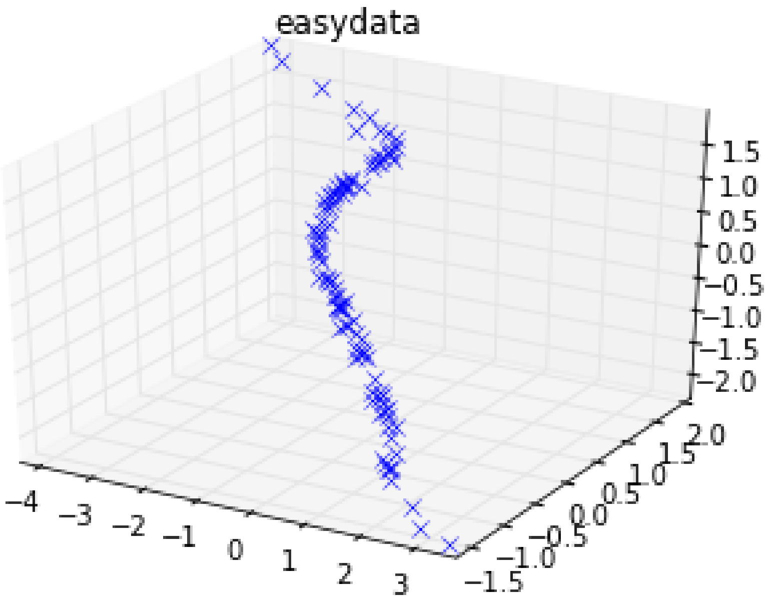
```
sys.path.append('../tools/')
sys.path.append('..')
import easy_dataset
reload(easy_dataset)

Y, Xt = easy_dataset.gen_easydata(100, 1, 3)
```

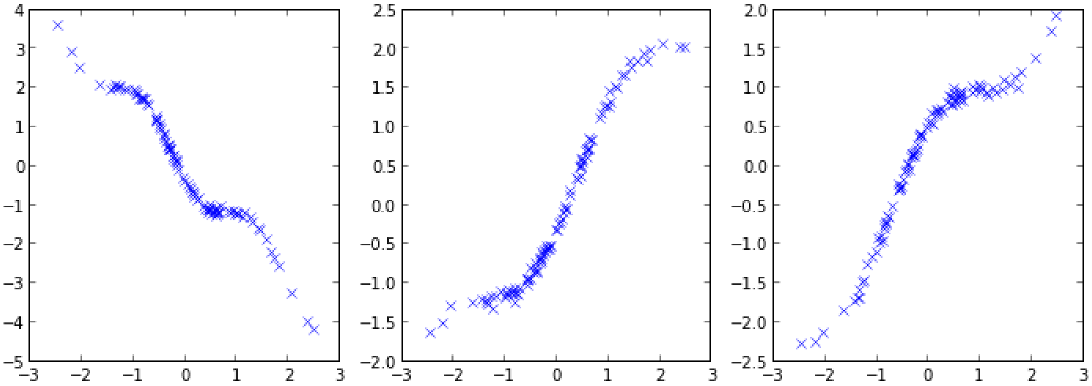
After generating the data, plot in 3D and then each dimension as a function of the latent variable X:

In [3]:

```
fig = plt.figure()
ax = fig.gca(projection='3d')
ax.plot(Y[:, 0], Y[:, 1], Y[:, 2], 'x')
#ax.view_init(elev=60, azim=300)
ax.set_title('easydata')

fig, ax = plt.subplots(1, 3, figsize=(12, 4), dpi=180)
ax[0].plot(Xt, Y[:, 0], 'x')
ax[1].plot(Xt, Y[:, 1], 'x')
ax[2].plot(Xt, Y[:, 2], 'x')
```

```
Out [3]: [<matplotlib.lines.Line2D at 0x52bef90>]
```





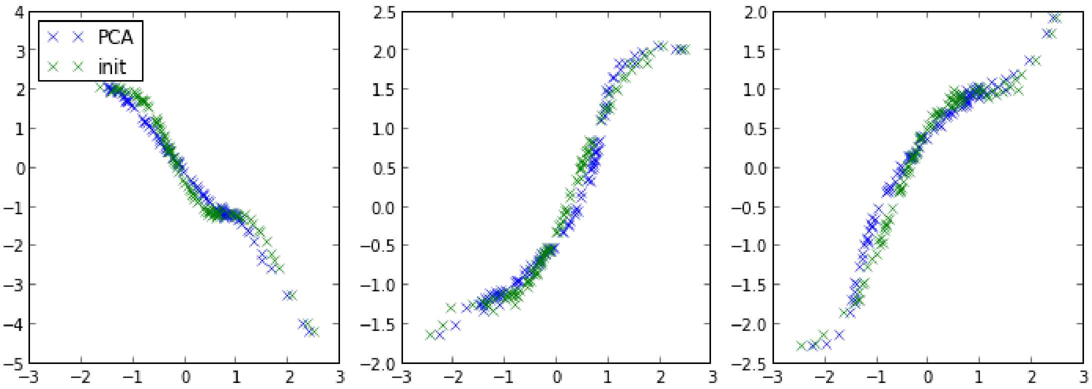
2 Initialisation with PCA

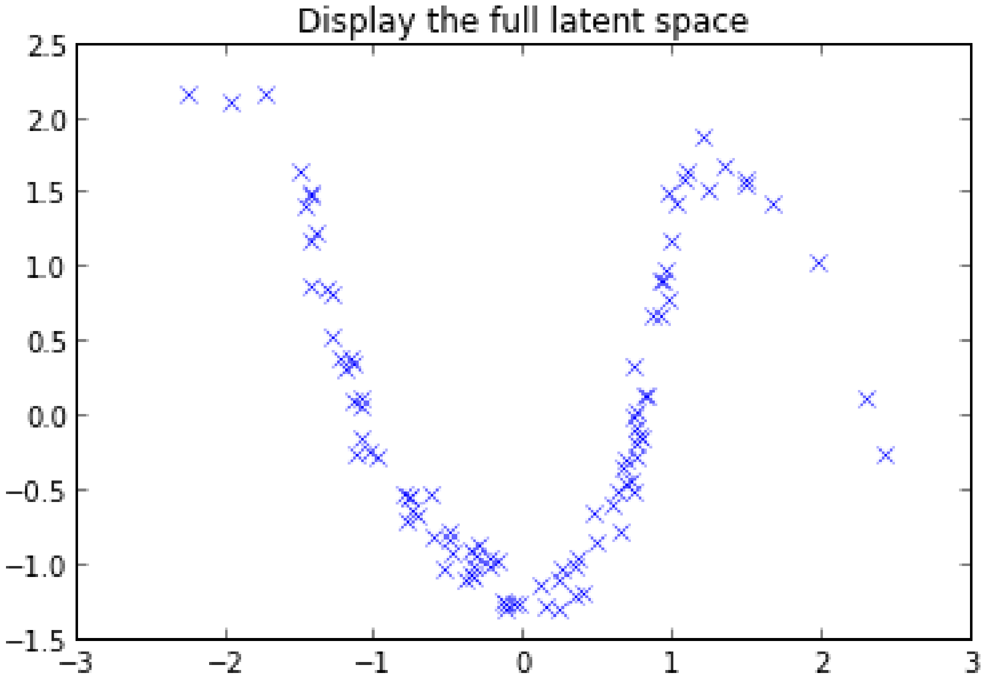
```
In [4]: def PCA(Y, input_dim):
        Z = numpy.linalg.svd(Y - Y.mean(axis=0), full_matrices=False)
        [X, W] = [Z[0][:, 0:input_dim], numpy.dot(numpy.diag(Z[1]), Z[2]).T[:, 0:
        v = X.std(axis=0)
        X /= v;
        W *= v;
        return X, W

        X, W = PCA(Y, 2)
```

The main principle component of PCA manages to recover the correct latent coordinates (with perhaps a horizontal flip). In any case, the ordering is correctly found, which should give a very good initialisation for the GPLVM.

```
In [5]: fig, ax = plt.subplots(1, 3, figsize=(12, 4), dpi=180)
        ax[0].plot(X[:, 0], Y[:, 0], 'x', label='PCA')
        ax[0].plot(Xt, Y[:, 0], 'x', label='init')
        ax[0].legend(loc=2)
        ax[1].plot(X[:, 0], Y[:, 1], 'x')
        ax[1].plot(Xt, Y[:, 1], 'x')
        ax[2].plot(X[:, 0], Y[:, 2], 'x')
        ax[2].plot(Xt, Y[:, 2], 'x')

        plt.figure()
        plt.plot(X[:, 0], X[:, 1], 'x')
        plt.title('Display the full latent space')
```

```
Out [5]: <matplotlib.text.Text at 0x58c0cd0>
```





We can also find the marginal likelihood of the model:

In [6]:

```
import MLtools
# Requires optimisation over beta, so maybe later.
```

3 GPy GPLVM results

We now run the GPy Bayesian GPLVM to see if sensible results are obtained. Sheffield seem to fix the noise while performing optimisation over (presumably) X, ard, Z etc.

In [7]:

```
# Parameters to adjust
Q = 2
num_inducing = 10
```

In [8]:

```
np.random.seed(0)

# Normalise data
Yn = Y - Y.mean(0)
Yn /= Yn.std(0)

# Set up model
rbf_comp = GPy.kern.rbf(Q, ARD=True)
kern = rbf_comp + GPy.kern.bias(Q, np.exp(-2)) + GPy.kern.white(Q, np.exp
m = GPy.models.BayesianGPLVM(Yn, Q, kernel=kern, num_inducing=10)
m['.*lengt'] = 1.               # ???
# m['noise'] = Yn.var() / 100
m['noise'] = 1
m.ensure_default_constraints()

m.constrain_positive('noise')
m.optimize('scg', messages=1, max_iters=100, gtol=.05)
print m.log_likelihood()
```

```
Warning: re-constraining these parameters
noise_variance
 I     F                Scale           |g|
 001   5.168420e+02   1.000000e+00   5.924867e+03  002   5.168420e+02
5.000000e-01   2.530909e+03  003   5.168420e+02   2.000000e+00
2.530909e+03  004   5.168420e+02   8.000000e+00   2.530909e+03  005
4.522405e+02   3.200000e+01   2.530909e+03
 006   4.292024e+02   1.600000e+01   1.710337e+03  007   4.171313e+02
8.000000e+00   2.039097e+03  008   3.987877e+02   4.000000e+00
3.337231e+03  009   3.604956e+02   2.000000e+00   2.411222e+02  010
3.582773e+02   1.000000e+00   9.596617e+01  011   3.468853e+02
5.000000e-01   2.577887e+02
 012   3.357225e+02   2.500000e-01   7.608609e+02  013   3.276896e+02
1.250000e-01   1.362646e+02  014   3.264190e+02   6.250000e-02
1.328279e+02  015   3.168646e+02   3.125000e-02   1.346917e+02  016
3.125993e+02   1.562500e-02   1.139487e+03  017   3.062507e+02
7.812500e-03   2.207097e+02  018   3.043641e+02   3.906250e-03
2.439367e+02  019   3.028114e+02   1.953125e-03   1.008206e+02  020
3.021098e+02   9.765625e-04   6.597391e+01
 021   2.975714e+02   4.882812e-04   9.624438e+01  022   2.919464e+02
4.882812e-04   1.286619e+03  023   2.823386e+02   2.441406e-04
2.832721e+02  024   2.817147e+02   2.441406e-04   8.571100e+02  025
2.788369e+02   1.220703e-04   4.961585e+02  026   2.743039e+02
6.103516e-05   4.774420e+01  027   2.730845e+02   6.103516e-05
```

```
1.237038e+03  028   2.681082e+02   3.051758e-05   1.154834e+03  029
2.617454e+02   1.525879e-05   2.340344e+02  030   2.610417e+02
7.629395e-06   4.153035e+02  031   2.593784e+02   3.814697e-06
1.866055e+02  032   2.581441e+02   1.907349e-06   7.149518e+01  033
2.563949e+02   9.536743e-07   1.567828e+02  034   2.497879e+02
4.768372e-07   1.427305e+02  035   2.497879e+02   2.384186e-07
9.723411e+02  036   2.497879e+02   9.536743e-07   9.723411e+02  037
2.497879e+02   3.814697e-06   9.723411e+02  038   2.497879e+02
1.525879e-05   9.723411e+02  039   2.497879e+02   6.103516e-05
9.723411e+02  040   2.497879e+02   2.441406e-04   9.723411e+02  041
2.497879e+02   9.765625e-04   9.723411e+02  042   2.497879e+02
3.906250e-03   9.723411e+02  043   2.497584e+02   1.562500e-02
9.723411e+02  044   2.288152e+02   6.250000e-02   6.444170e+03  045
2.193732e+02   3.125000e-02   9.650134e+02  046   2.121900e+02
3.125000e-02   2.224503e+03  047   2.098093e+02   1.562500e-02
1.364856e+02  048   2.087344e+02   7.812500e-03   1.286572e+02  049
2.012390e+02   3.906250e-03   1.557738e+02  050   1.785972e+02
3.906250e-03   5.130277e+03  051   1.710893e+02   1.953125e-03
1.172611e+03  052   1.710869e+02   1.953125e-03   8.618138e+02  053
1.674303e+02   9.765625e-04   8.764041e+02  054   1.651065e+02
4.882812e-04   1.573740e+02  055   1.601509e+02   2.441406e-04
1.757272e+02  056   1.599778e+02   2.441406e-04   1.895366e+03
 057   1.530850e+02   1.220703e-04   2.082296e+03  058   1.478566e+02
6.103516e-05   4.198236e+02  059   1.446052e+02   3.051758e-05
3.848970e+02  060   1.411180e+02   1.525879e-05   2.919758e+02  061
1.385930e+02   7.629395e-06   1.772371e+02  062   1.360729e+02
3.814697e-06   3.884955e+02  063   1.307891e+02   1.907349e-06
4.407092e+02  064   1.252920e+02   9.536743e-07   5.334439e+02  065
1.200228e+02   4.768372e-07   1.283334e+03  066   1.083619e+02
2.384186e-07   2.613571e+03  067   9.759192e+01   1.192093e-07
1.357414e+03
 068   9.303668e+01   5.960464e-08   1.226466e+03  069   8.909969e+01
5.960464e-08   1.515157e+03  070   8.239233e+01   2.980232e-08
1.645566e+03  071   7.877575e+01   1.490116e-08   1.054361e+03  072
7.603134e+01   7.450581e-09   4.696288e+02  073   7.447702e+01
3.725290e-09   5.020106e+02  074   7.291693e+01   1.862645e-09
4.153230e+02  075   7.143683e+01   9.313226e-10   3.062635e+02  076
7.001693e+01   4.656613e-10   2.463829e+02
 077   6.817430e+01   2.328306e-10   3.598164e+02  078   6.655523e+01
1.164153e-10   4.555440e+02  079   6.291735e+01   5.820766e-11
6.890208e+02  080   5.762537e+01   2.910383e-11   1.365065e+03  081
5.339348e+01   1.455192e-11   1.225238e+03  082   5.039808e+01
7.275958e-12   9.291512e+02  083   4.735987e+01   3.637979e-12
1.318892e+03  084   4.380208e+01   1.818989e-12   2.022544e+03  085
3.894505e+01   9.094947e-13   2.496651e+03  086   3.337923e+01
4.547474e-13   3.572837e+03  087   2.983963e+01   2.273737e-13
2.343169e+03  088   2.798167e+01   1.136868e-13   7.971445e+02  089
2.634055e+01   5.684342e-14   9.367525e+02  090   2.503426e+01
2.842171e-14   8.354655e+02  091   2.395210e+01   1.421085e-14
6.688496e+02  092   2.296175e+01   7.105427e-15   5.426551e+02  093
2.183648e+01   3.552714e-15   4.936033e+02  094   2.105883e+01
1.776357e-15   5.440224e+02  095   2.038260e+01   8.881784e-16
3.196197e+02  096   1.975690e+01   4.440892e-16   2.794310e+02  097
1.892264e+01   2.220446e-16   3.787131e+02  098   1.822298e+01
1.110223e-16   4.785638e+02  099   1.751613e+01   5.551115e-17
```

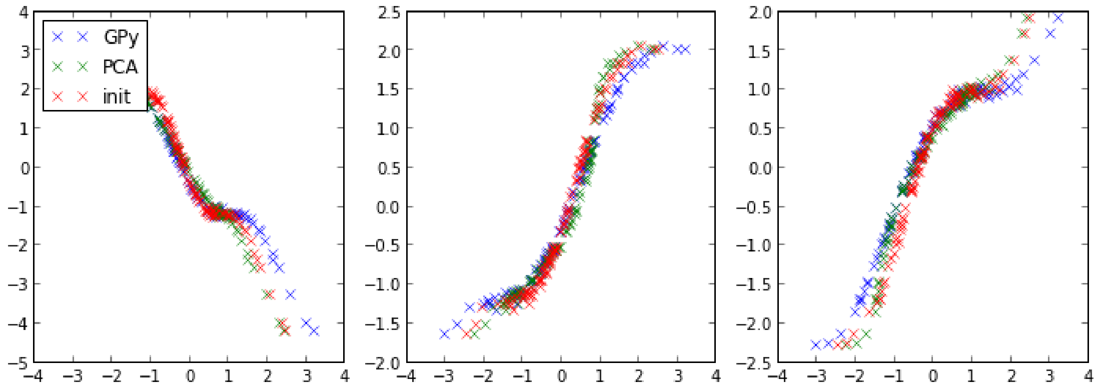
```
3.938918e+02  100   1.677444e+01   2.775558e-17   3.614221e+02  100
1.677444e+01   1.387779e-17   3.148492e+02
maxiter exceeded
-16.7744368322
```

In [9]:

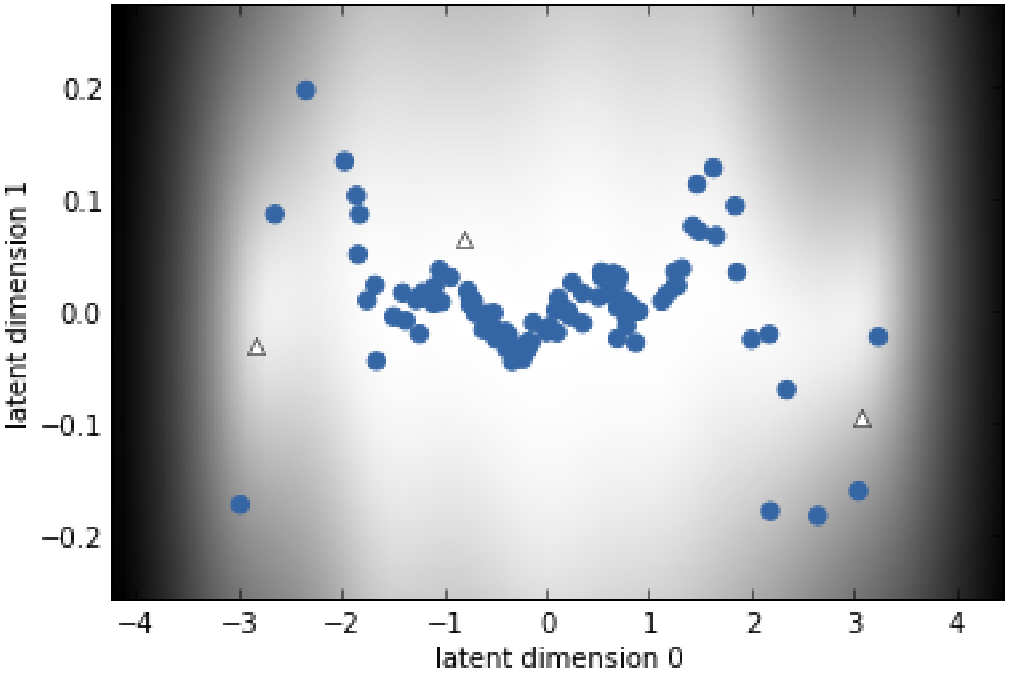
```
fig, ax = plt.subplots(1, 3, figsize=(12, 4), dpi=180)
ax[0].plot(m.X[:, 0], Y[:, 0], 'x', label='GPy')
ax[0].plot(X[:, 0], Y[:, 0], 'x', label='PCA')
ax[0].plot(Xt, Y[:, 0], 'x', label='init')
ax[0].legend(loc=2)

ax[1].plot(m.X[:, 0], Y[:, 1], 'x', label='GPy')
ax[1].plot(X[:, 0], Y[:, 1], 'x', label='PCA')
ax[1].plot(Xt, Y[:, 1], 'x', label='init')

ax[2].plot(m.X[:, 0], Y[:, 2], 'x', label='GPy')
ax[2].plot(X[:, 0], Y[:, 2], 'x', label='PCA')
ax[2].plot(Xt, Y[:, 2], 'x', label='init')

#fig, (latent_axes, sense_axes) = plt.subplots(1, 2)
#plt.sca(latent_axes)
plt.figure()
m.plot_latent()

plt.figure()

kern.plot_ARD()
```

Out [9]:

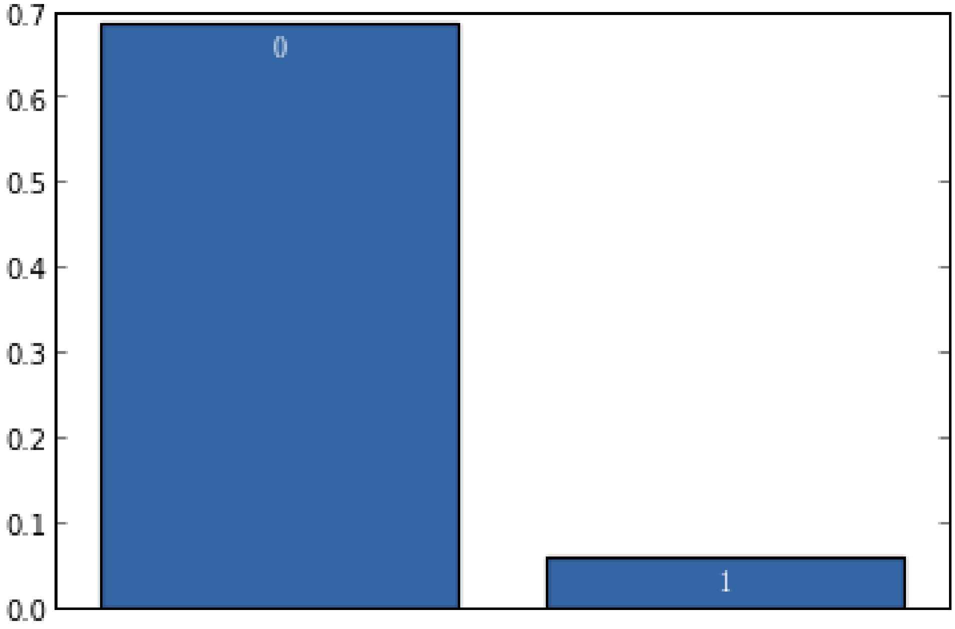
```
<matplotlib.axes.AxesSubplot at 0x62d9250>
```



```
<matplotlib.figure.Figure at 0x6135f10>
```


0.2
0.1
0.0
−0.1
−0.2
latent dimension 1
−4
−3
−2
−1
0
1
2
3
4
latent dimension 0

```
<matplotlib.figure.Figure at 0x6138a50>
```



4 Parallel GPLVM results

In [10]:

```
execfile('easydata-pargplvm.py')
```

```
Creating ../easydata//embeddings//easy_4.embedding.npy with 28 points
Creating ../easydata//embeddings//easy_4.variance.npy with 28 points
Creating ../easydata//embeddings//easy_2.embedding.npy with 21 points
Creating ../easydata//embeddings//easy_2.variance.npy with 21 points
Creating ../easydata//embeddings//easy_3.embedding.npy with 24 points
Creating ../easydata//embeddings//easy_3.variance.npy with 24 points
Creating ../easydata//embeddings//easy_1.embedding.npy with 27 points
Creating ../easydata//embeddings//easy_1.variance.npy with 27 points
Done! statistics Map-Reduce took  3  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds
```

```
 I     F              Scale          |g|
Starting optimisation for 100 iterations
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 001   5.412534e+02   1.000000e+00   1.747951e+03
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 002   4.735020e+02   4.000000e+00   1.747951e+03
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 003   4.700981e+02   2.000000e+00   1.585280e+03
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 004   4.330054e+02   1.000000e+00   1.876492e+03
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 005   4.228699e+02   5.000000e-01   2.327013e+02
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 006   4.028145e+02   2.500000e-01   2.497727e+02
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds
Done! statistics Map-Reduce took  1  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds
```

```
 007   3.933679e+02   1.250000e-01   1.614742e+02
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 008   3.820146e+02   6.250000e-02   1.003485e+02
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 009   3.623116e+02   3.125000e-02   5.270867e+01
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 010  -1.793316e+04   2.437172e-01   6.269914e+02
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 011  -1.793316e+04   1.218586e-01   5.710018e+27
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 012  -1.793316e+04   4.874344e-01   5.710018e+27
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 013  -1.793316e+04   1.949738e+00   5.710018e+27
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 014  -1.793316e+04   7.798950e+00   5.710018e+27
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 015  -1.793316e+04   3.119580e+01   5.710018e+27
Done! statistics Map-Reduce took  0  seconds
```

```
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 016  -1.793316e+04   1.247832e+02   5.710018e+27
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 017  -1.793316e+04   4.991328e+02   5.710018e+27
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 018  -1.793316e+04   1.996531e+03   5.710018e+27
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 019  -1.793316e+04   7.986125e+03   5.710018e+27
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 020  -1.793316e+04   3.194450e+04   5.710018e+27
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 021  -1.793316e+04   1.277780e+05   5.710018e+27
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 022  -1.793316e+04   5.111120e+05   5.710018e+27
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 023  -1.793316e+04   2.044448e+06   5.710018e+27
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 024  -1.793316e+04   8.177792e+06   5.710018e+27
Done! statistics Map-Reduce took  0  seconds

        Increasing failed count: 1


 025  -1.793316e+04   3.271117e+07   5.710018e+27
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 026  -1.793316e+04   1.308447e+08   5.710018e+27
```

```
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 027  -4.422451e+04   5.233787e+08   5.710018e+27
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds
Done! statistics Map-Reduce took  0  seconds

        Increasing failed count: 1


 028  -4.422451e+04   2.093515e+09   1.676112e+28
Done! statistics Map-Reduce took  0  seconds

        Increasing failed count: 2


 029  -4.422451e+04   8.374059e+09   1.676112e+28
Done! statistics Map-Reduce took  0  seconds

        Increasing failed count: 3


 030  -4.422451e+04   3.349624e+10   1.676112e+28
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 031  -4.422451e+04   1.339849e+11   1.676112e+28
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 032  -4.422451e+04   5.359398e+11   1.676112e+28
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 033  -4.422451e+04   2.143759e+12   1.676112e+28
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 034  -4.904575e+04   8.575036e+12   1.676112e+28
Done! statistics Map-Reduce took  0  seconds

        Increasing failed count: 1

Done! statistics Map-Reduce took  0  seconds

        Increasing failed count: 2
 035  -4.904575e+04   3.430015e+13   1.675603e+28
Done! statistics Map-Reduce took  0  seconds

        Increasing failed count: 3


 036  -4.904575e+04   1.372006e+14   1.675603e+28
Done! statistics Map-Reduce took  0  seconds

        Increasing failed count: 4


 037  -4.904575e+04   5.488023e+14   1.675603e+28
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 038  -4.904575e+04   2.195209e+15   1.675603e+28
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 039  -4.904575e+04   8.780837e+15   1.675603e+28
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 040  -4.904575e+04   3.512335e+16   1.675603e+28
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 041  -4.904575e+04   1.404934e+17   1.675603e+28
Done! statistics Map-Reduce took  0  seconds

        Increasing failed count: 1


 042  -4.904575e+04   5.619736e+17   1.675603e+28
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 043  -4.904575e+04   2.247894e+18   1.675603e+28
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 044  -4.904575e+04   8.991577e+18   1.675603e+28
Done! statistics Map-Reduce took  0  seconds

        Increasing failed count: 1


 045  -4.904575e+04   3.596631e+19   1.675603e+28
```

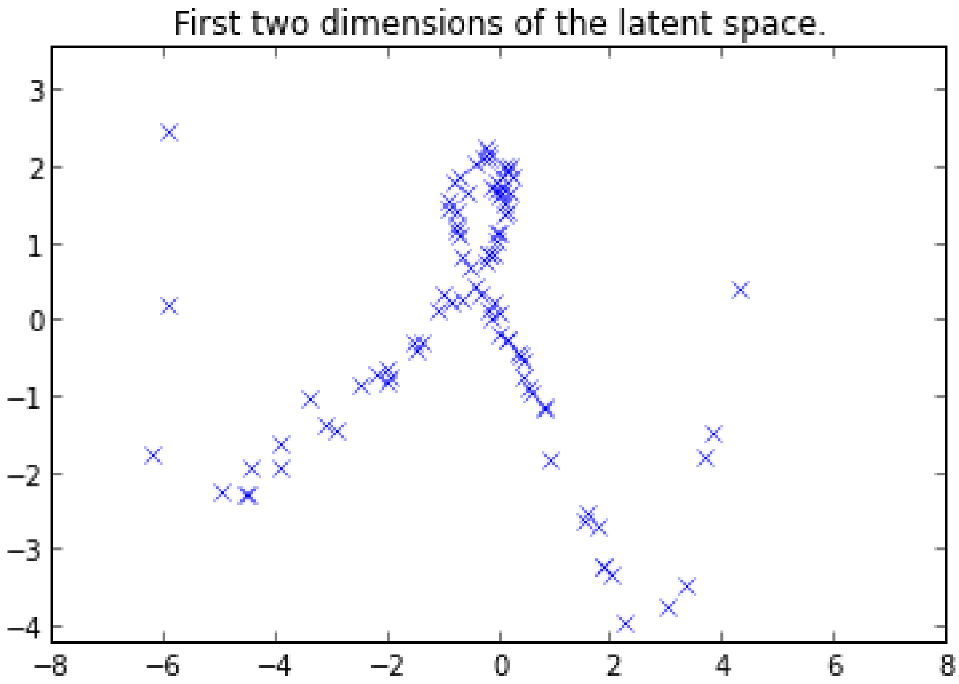
```
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 046  -4.904575e+04   1.438652e+20   1.675603e+28
Done! statistics Map-Reduce took  0  seconds

        Increasing failed count: 1


 047  -4.904575e+04   5.754609e+20   1.675603e+28
Done! statistics Map-Reduce took  0  seconds

        Increasing failed count: 2


 048  -4.904575e+04   2.301844e+21   1.675603e+28
Done! statistics Map-Reduce took  0  seconds

        Increasing failed count: 3


 049  -4.904575e+04   9.207375e+21   1.675603e+28
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 050  -4.904575e+04   3.682950e+22   1.675603e+28
Done! statistics Map-Reduce took  0  seconds

        Increasing failed count: 1


 051  -4.904575e+04   1.473180e+23   1.675603e+28
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 052  -4.904575e+04   5.892720e+23   1.675603e+28
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 053  -4.904575e+04   2.357088e+24   1.675603e+28
Done! statistics Map-Reduce took  0  seconds

        Increasing failed count: 1


 054  -4.904575e+04   9.428352e+24   1.675603e+28
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 055  -4.904575e+04   3.771341e+25   1.675603e+28
```

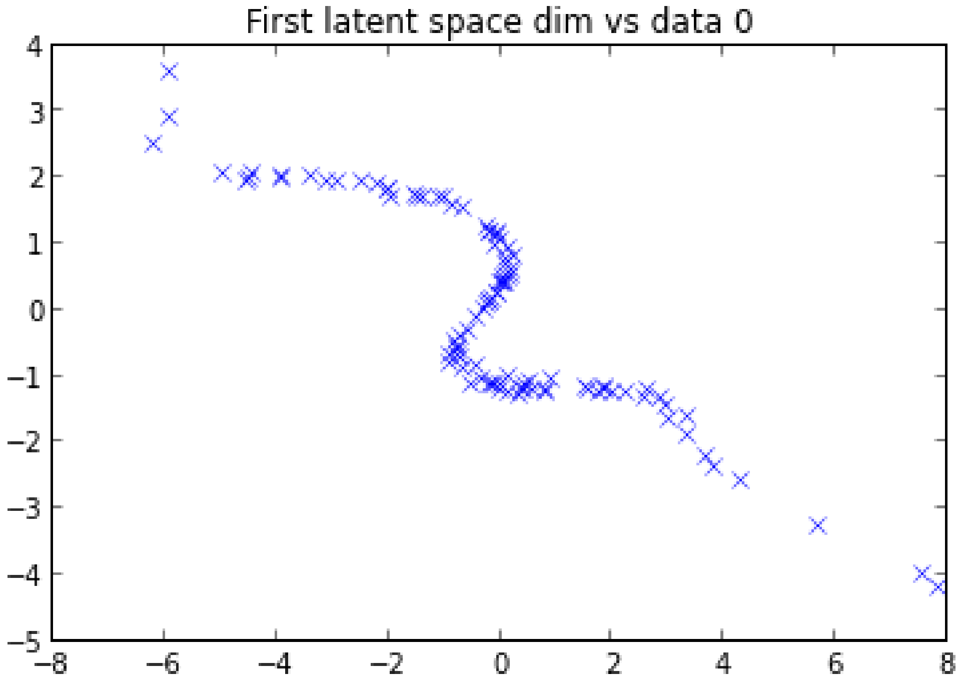
```
Done! statistics Map-Reduce took  0  seconds

        Increasing failed count: 1


 056  -4.904575e+04   1.508536e+26   1.675603e+28
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds

 057  -4.904575e+04   6.034145e+26   1.675603e+28

 057  -4.904575e+04   6.034145e+26   1.675603e+28


converged
Final global_statistics
{'alpha': array([[  2.58097155e-04,   5.04532613e-07]]), 'beta':
array([[ 19.65677885]]), 'Z': array([[-4.73064942,  1.05181189],
       [-0.63433125,  0.84981177],
       [-3.80424386,  0.09203524],
       [-1.20358076,  0.14469001],
       [ 3.04169981,  1.54617871],
       [ 6.30311326,  2.73916229],
       [ 6.07555499, -1.34316839],
       [ 1.22071946, -0.7732137 ],
       [ 1.24768093, -0.63532014],
       [ 0.41727327, -1.15034778]]), 'sf2': array([[ 20.54587036]])}
Done! statistics Map-Reduce took  0  seconds
Done! global statistics took  0  seconds
Done! embeddings Map-Reduce took  0  seconds
final F=49045.7514202
```

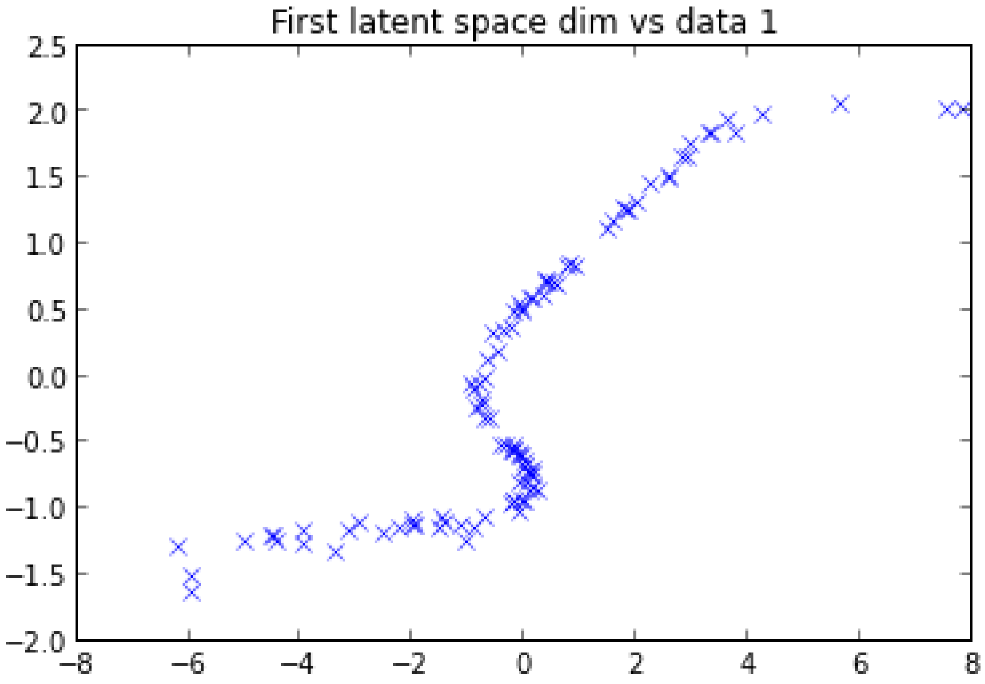
In [11]:

```python
import show_embeddings
reload(show_embeddings)
class empty:
    pass
disp_opt = empty()
disp_opt.verbose = True
disp_opt.dimension = [0, 1]
disp_opt.output_dimension = [0, 1, 2]
disp_opt.plot2d = True
disp_opt.plot3d = False
args = [path]
pX = show_embeddings.run(disp_opt, args)

plt.figure()
plt.plot(m.X[:, 0], m.X[:, 1], 'x', label='GPy')
plt.plot(pX[:, 0], pX[:, 1], 'x', label='par-gplvm')
plt.legend()
plt.title('GPy result vs. our implementation')
```

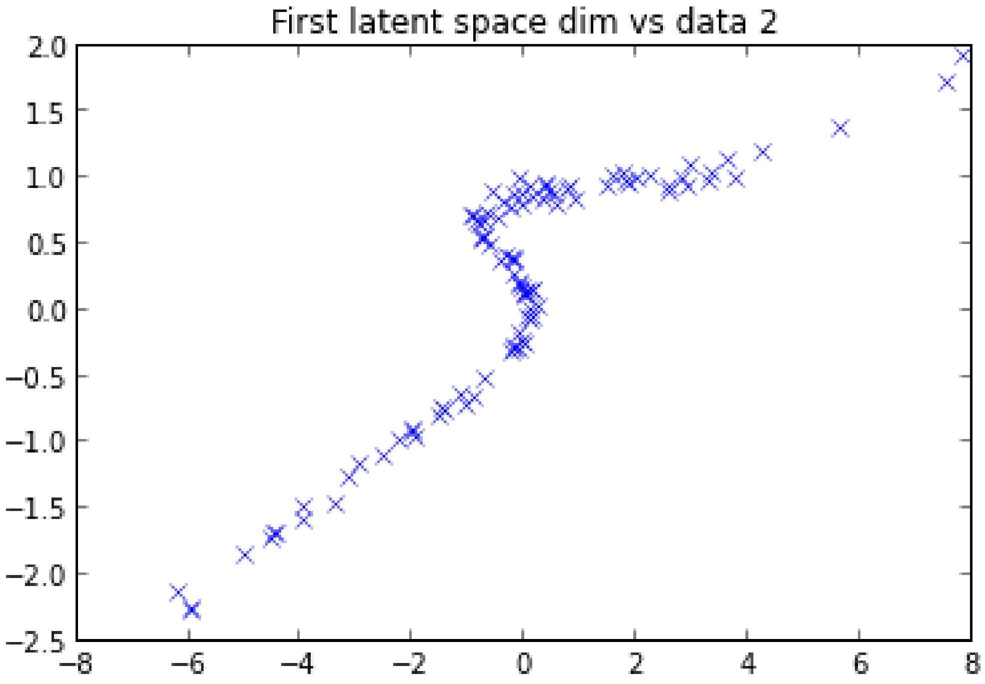
```
Displaying X in '../easydata/'...
alpha: [  2.58097155e-04   5.04532613e-07]
beta : 19.656778854
sf2  : 20.5458703551
```





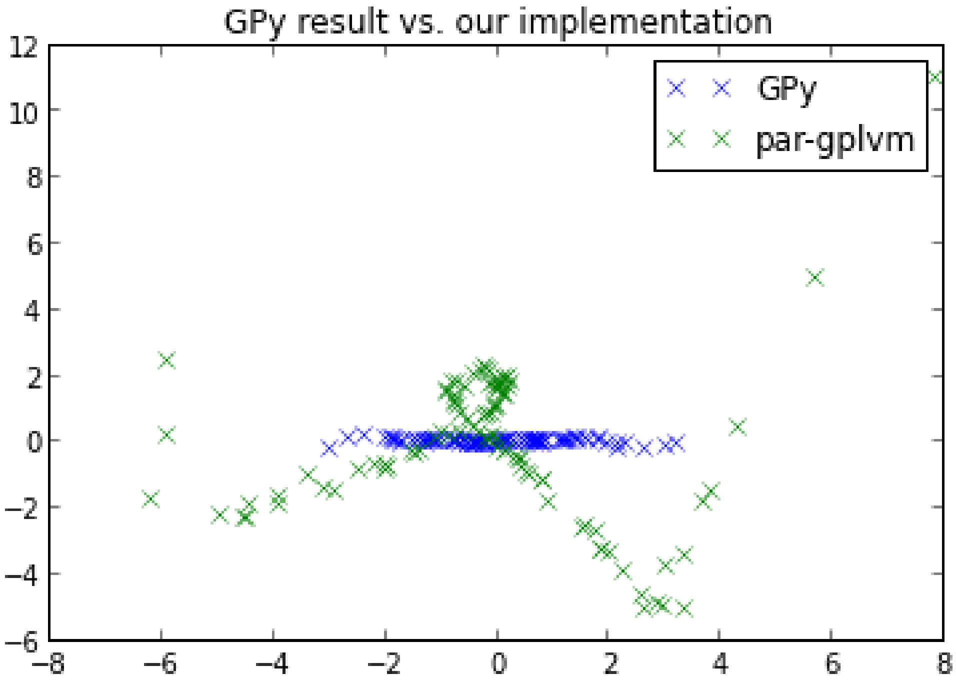


First latent space dim vs data 1



First latent space dim vs data 2

```
Out [11]: <matplotlib.text.Text at 0x7845bd0>
```


GPy result vs. our implementation
12
10
8
6
4
2
0
−2
−4
−6
−8
−6
−4
−2
0
2
4
6
8
GPy
par-gplvm