Modifying MED for Model Selection

Kristyn Pantoja

11/25/2019

MMED for 2 Slopes (p = 1)

GP: Test Function

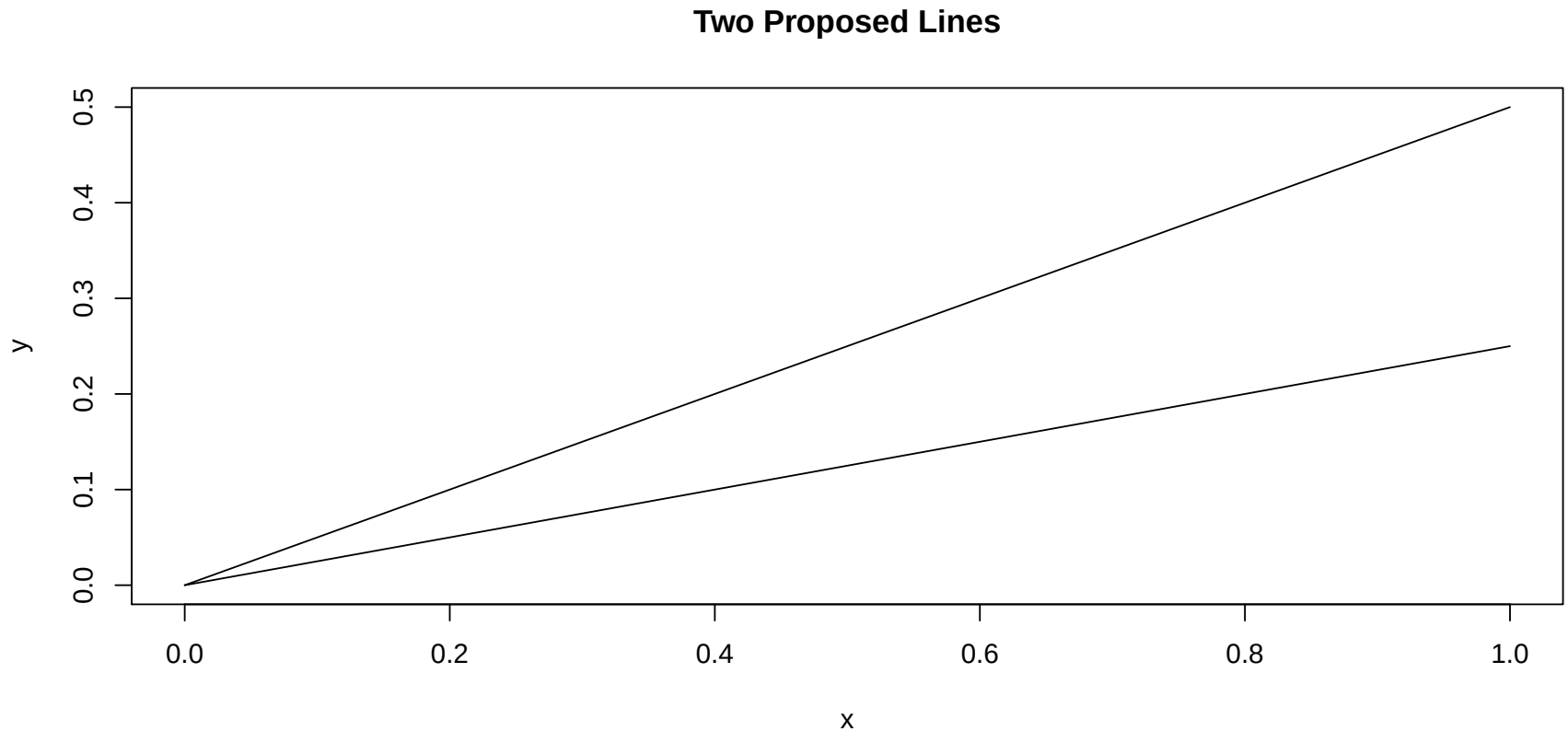
GP: Space-filling (N = 3)

GP: Space-filling (N = 6)

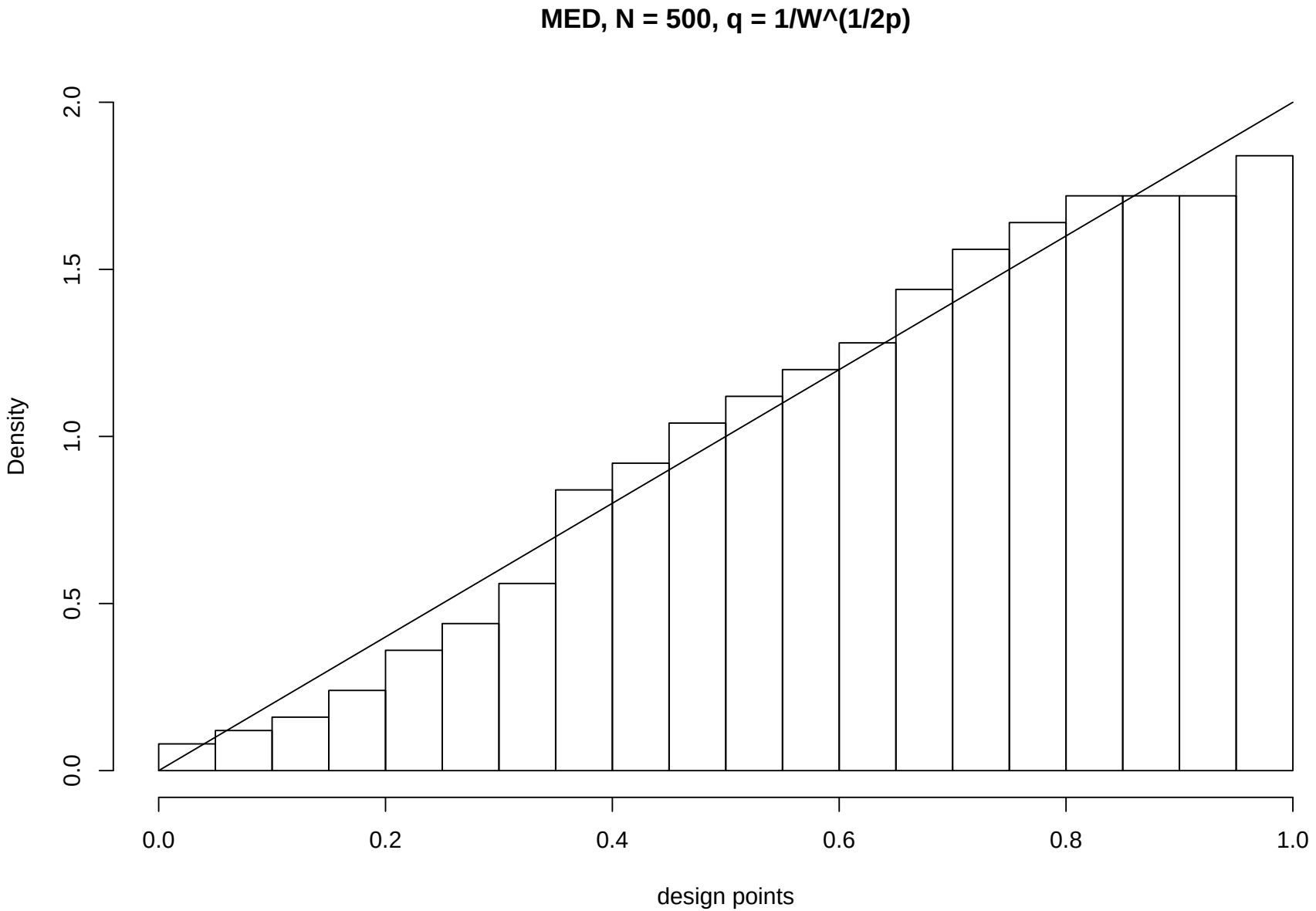
GP: MMED with Training Points in one region of support

MMED for 2 Slopes ($p = 1$)

Original Motivating Example



Greedy Algorithm, k = 2, p = 1, N = 500



Parameter correction:

- $p = 1$ instead of $p = 2$, since p is for number of *dimensions*, not number of *parameters*!
- This explains why the tuning parameter $\alpha = 2p$ almost fixed things - it partially corrected the mistake (the numerator was correct, the denominator was scaled up more than it should have been, hence why the MMED design was sharper)
  - this means we don't need to re-do any of the analyses.
- Also explains why designs with high density in the middle worked better (square root overemphasized those regions)
  - This just so happened to be the shape that showed the error most clearly.

Showing alpha fixed it, in a way

$$\sum_{i=1}^{j}\left(\frac{q(\mathbf{x}_i)q(\mathbf{x})}{d(\mathbf{x}_i,\mathbf{x})}\right)^k = \sum_{i=1}^{j}\frac{\frac{1}{f(\mathbf{x}_i)^{k/2p}}\frac{1}{f(\mathbf{x})^{k/2p}}}{d^k(\mathbf{x}_i,\mathbf{x})}$$

$$= \sum_{i=1}^{j}\frac{\frac{1}{f(\mathbf{x}_i)^{k/2}}\frac{1}{f(\mathbf{x})^{k/2}}}{d^k(\mathbf{x}_i,\mathbf{x})} \text{ when } p=1$$

$$= \sum_{i=1}^{j}\frac{\frac{1}{f(\mathbf{x}_i)^{k/4}}\frac{1}{f(\mathbf{x})^{k/4}}}{d^k(\mathbf{x}_i,\mathbf{x})} \text{ when } p=2$$

with the α parameter,

$$\sum_{i=1}^{j}\frac{\frac{1}{f(\mathbf{x}_i)^{\alpha k/2p}}\frac{1}{f(\mathbf{x})^{\alpha k/2p}}}{d^k(\mathbf{x}_i,\mathbf{x})} = \sum_{i=1}^{j}\frac{\frac{1}{f(\mathbf{x}_i)^{k}}\frac{1}{f(\mathbf{x})^{k}}}{d^k(\mathbf{x}_i,\mathbf{x})} \text{ when } p=2, \alpha=2p=4$$

$$= \sum_{i=1}^{j}\left(\frac{\frac{1}{f(\mathbf{x}_i)^{k/2}}\frac{1}{f(\mathbf{x})^{k/2}}}{d^{k/2}(\mathbf{x}_i,\mathbf{x})}\right)^2$$

- The case with the α parameter is closer to the correct case ($p = 1$) in the original greedy algorithm (without α), except $d(\cdot, \cdot)$ has a smaller power, making the distances less influential, and leading to the decrease in space-filling tendency that we saw in the designs.

GP: Test Function

Last Time: length-scale parameter

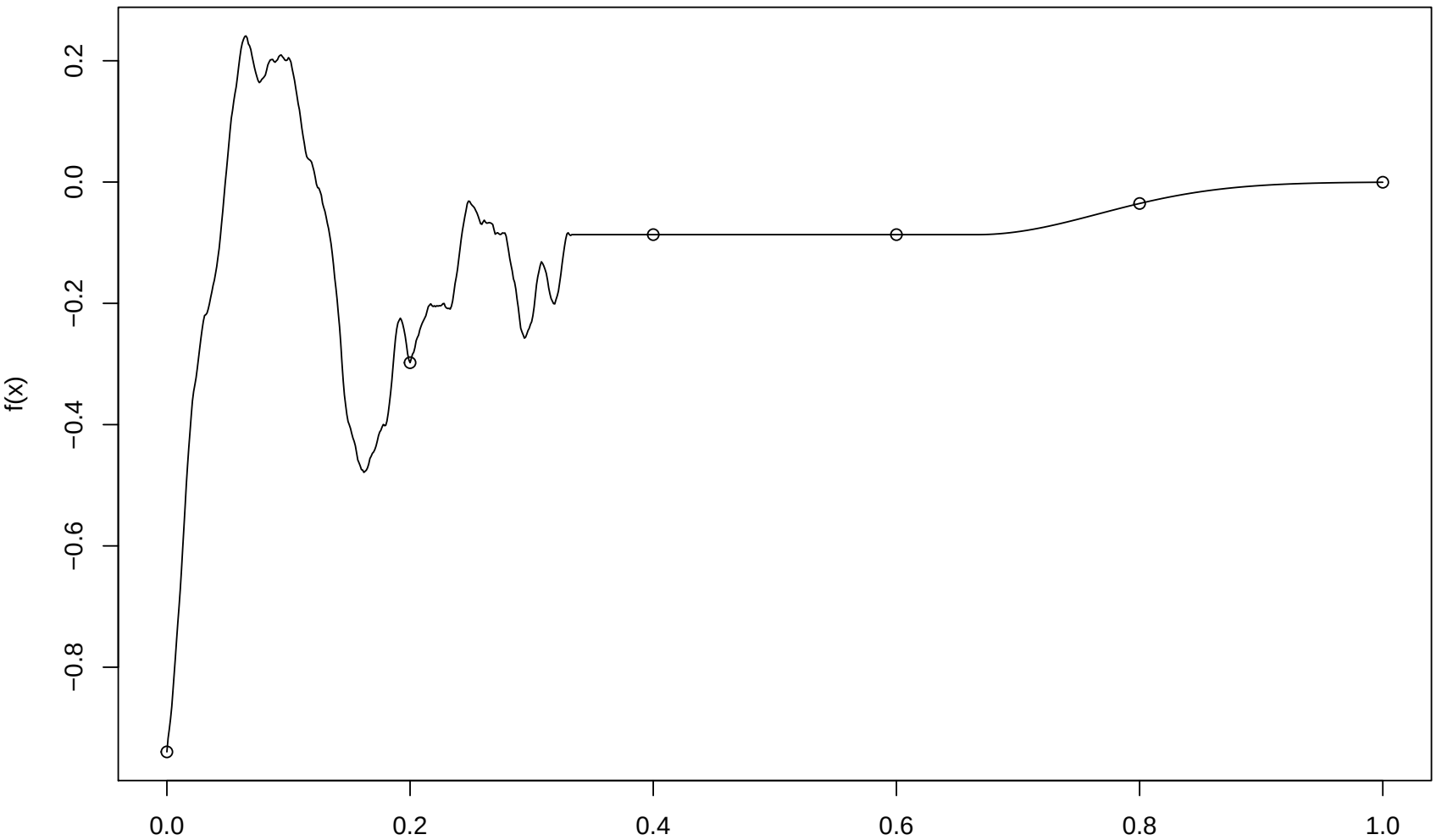
- The length-scale parameter ℓ is very important when it comes to how a GP with (misspecified) Gaussian kernel is applied to data generated from a matern kernel.
  - $\ell = 0.5$ looks a lot weirder at the edges than $\ell = 0.1$ when interpolating with the (misspecified) Gaussian kernel, likely because the kernel is trying to fit data under this strong assumption on its structure.
- When ℓ is a reasonable value, it looks like the GP will tend toward a space-filling design. I think this makes sense, since the Wasserstein distance between the points will be highest in those regions, as the variance is high in-between training points.
  - It might be more reasonable, then, to only apply MMED between two data points.

Last time: questions

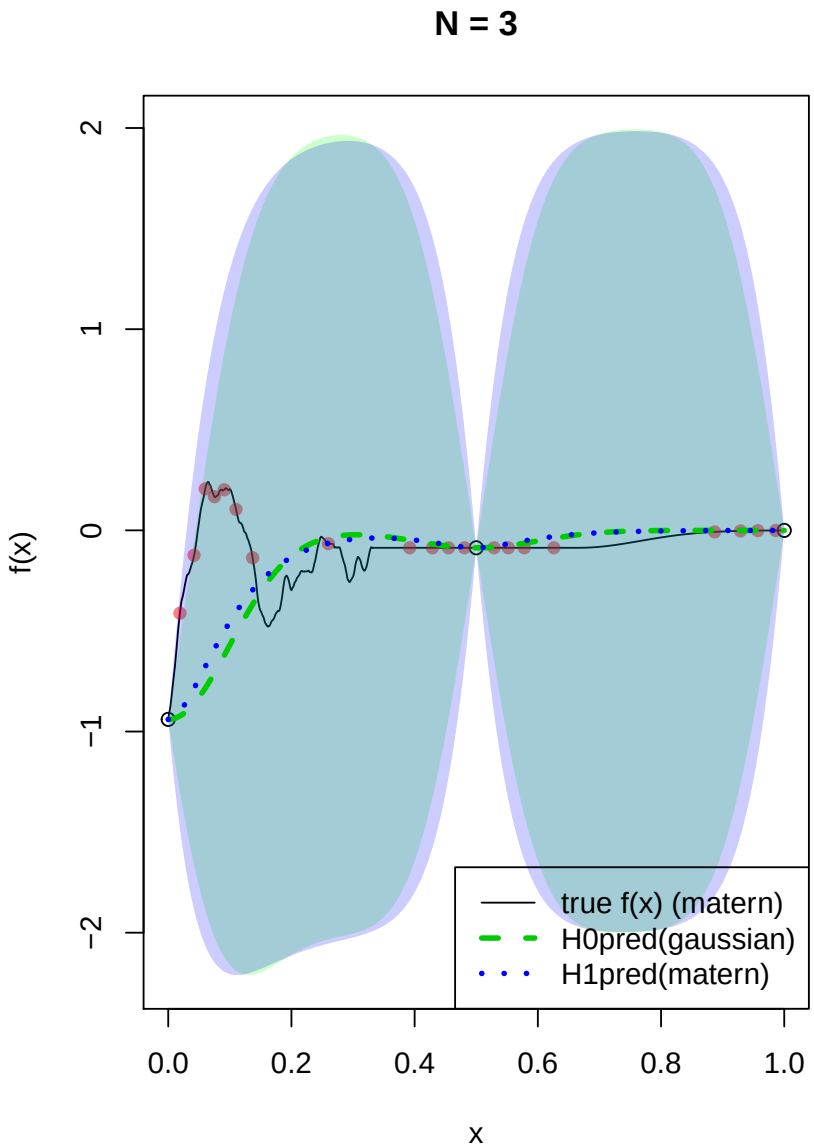
1. Exploring the behavior of MMED applied to GP? Specifically for Gaussian vs. Matern kernel. The idea of looking at how points are selected when the function is erratic in the first third of its support, constant in the second part, and smoother in the last part.
2. If the training points are not from a space-filling design? e.g. from a uniform distribution, or if the points are concentrated in the first part of its support.

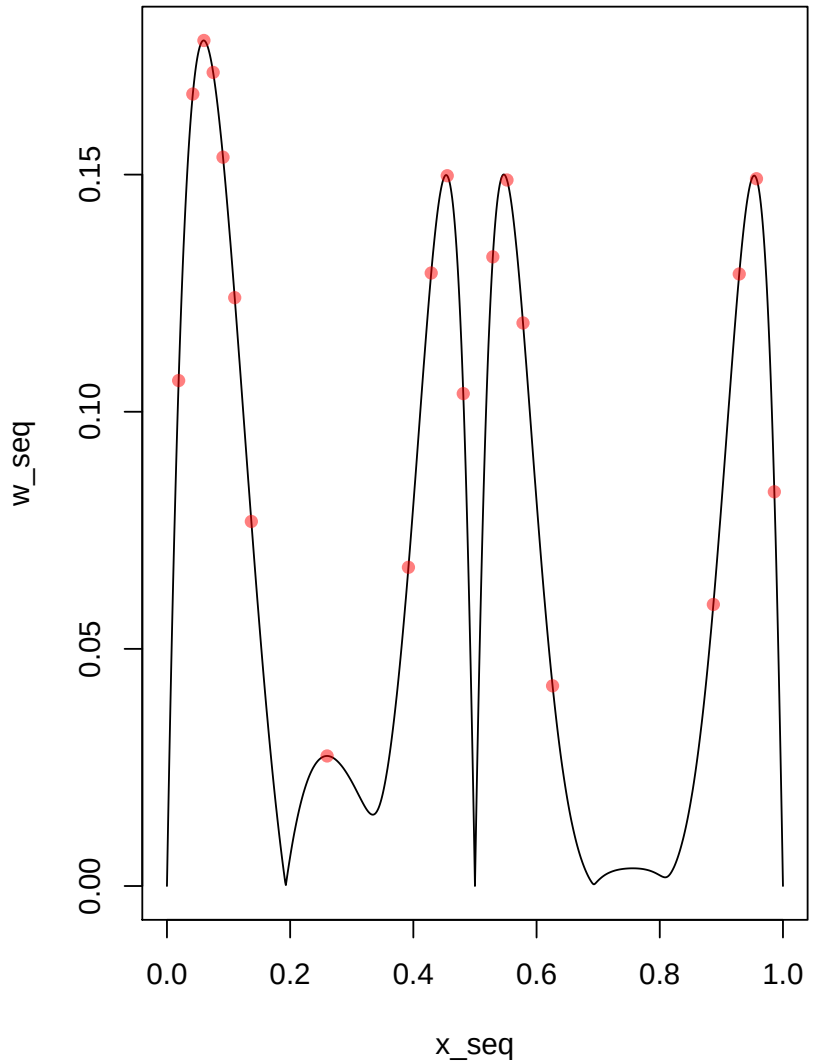
Test Function to compare Gaussian (H0) to Matern (H1) kernel

Here is a function where the first third is generated from a Matern kernel (with $\ell = 0.1$), the second part is a constant, and the third part is generated from a Gaussian kernel (also with $\ell = 0.1$).

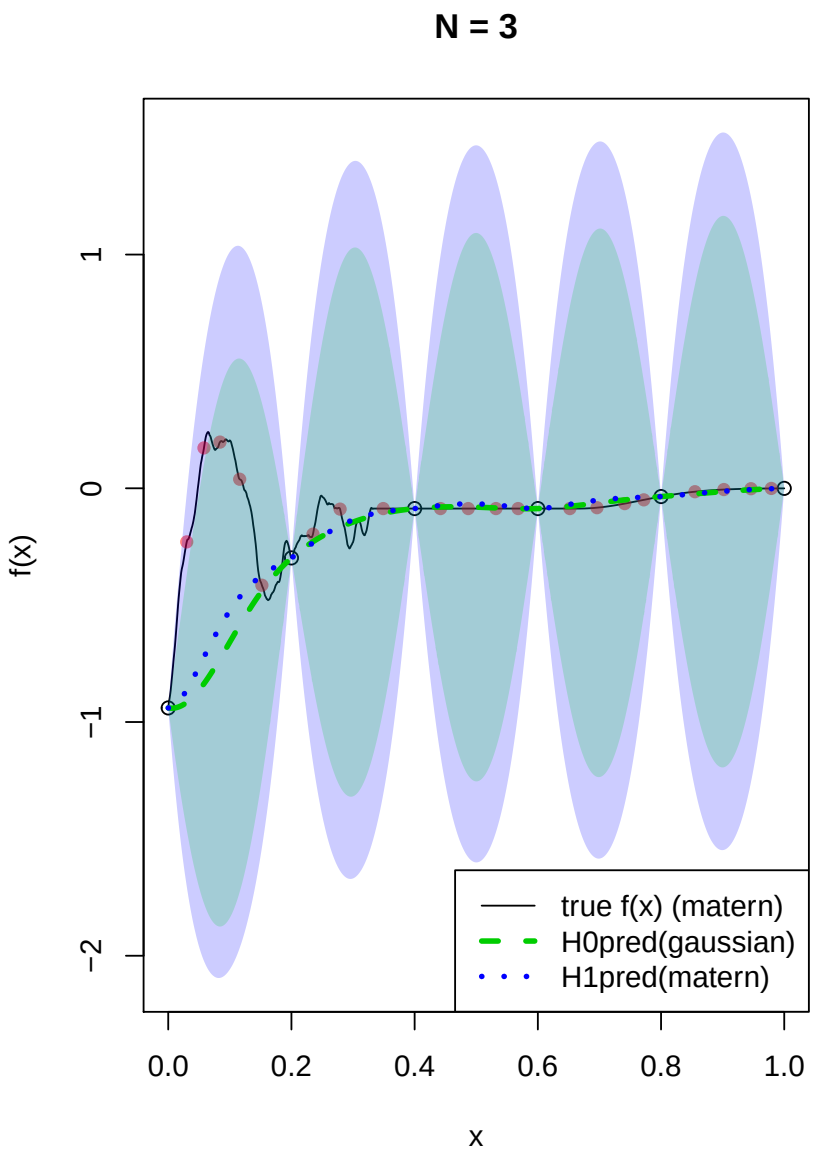


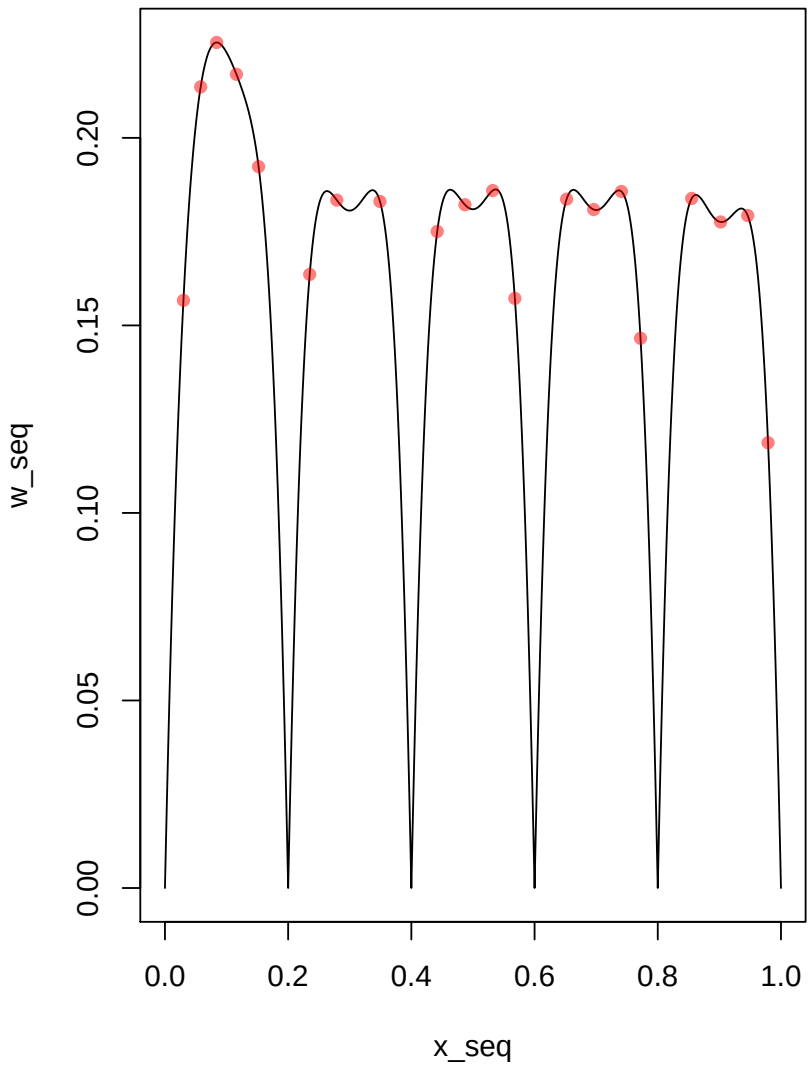
MMED, N = 3 Training Points



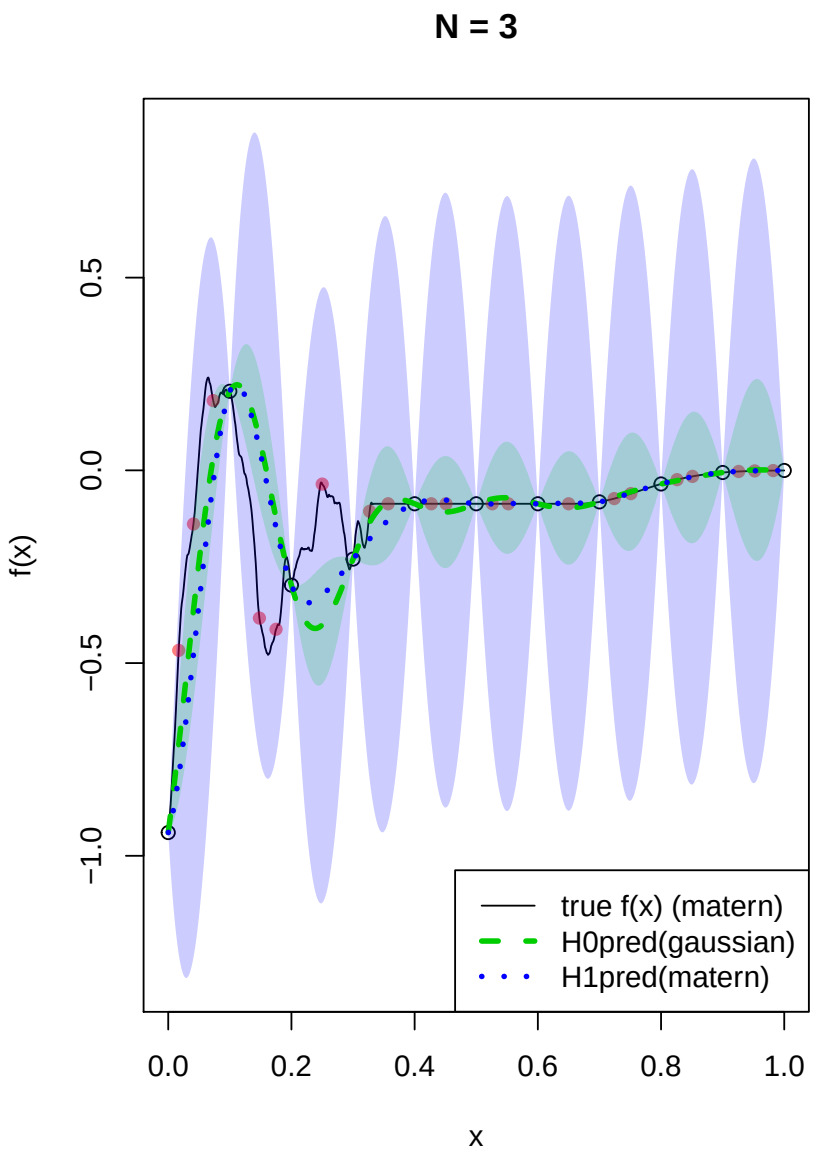


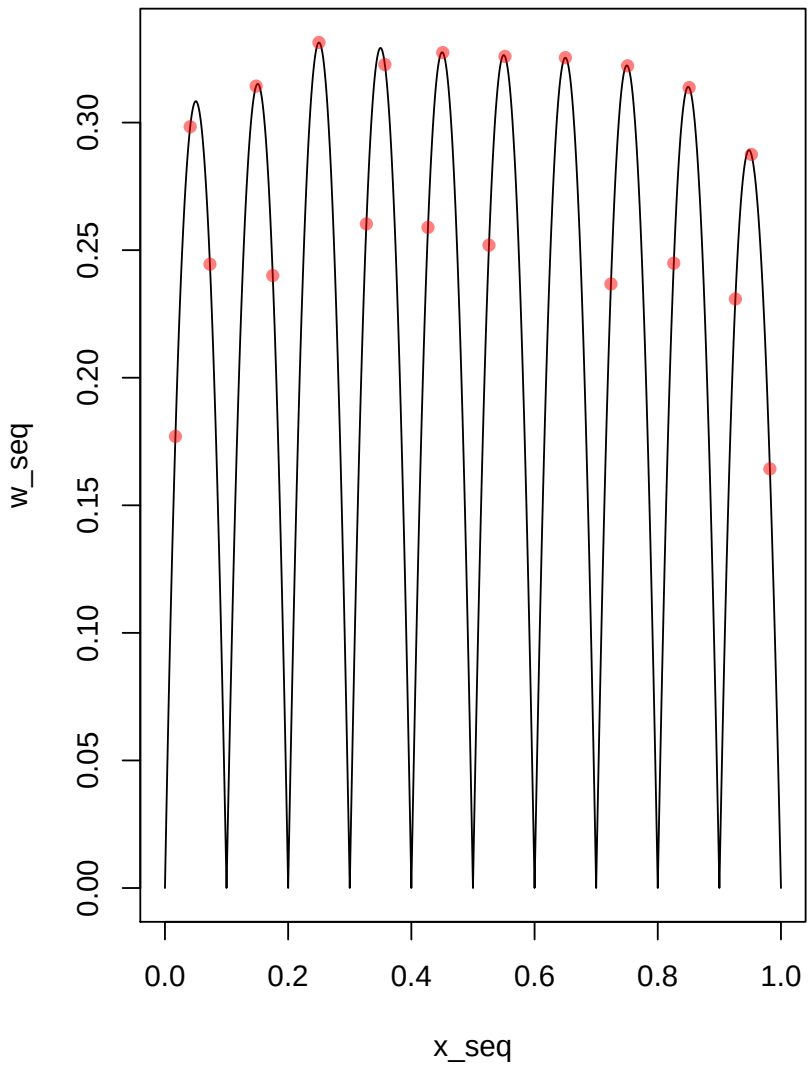
MMED, N = 6 Training Points





MMED, N = 11 Training Points





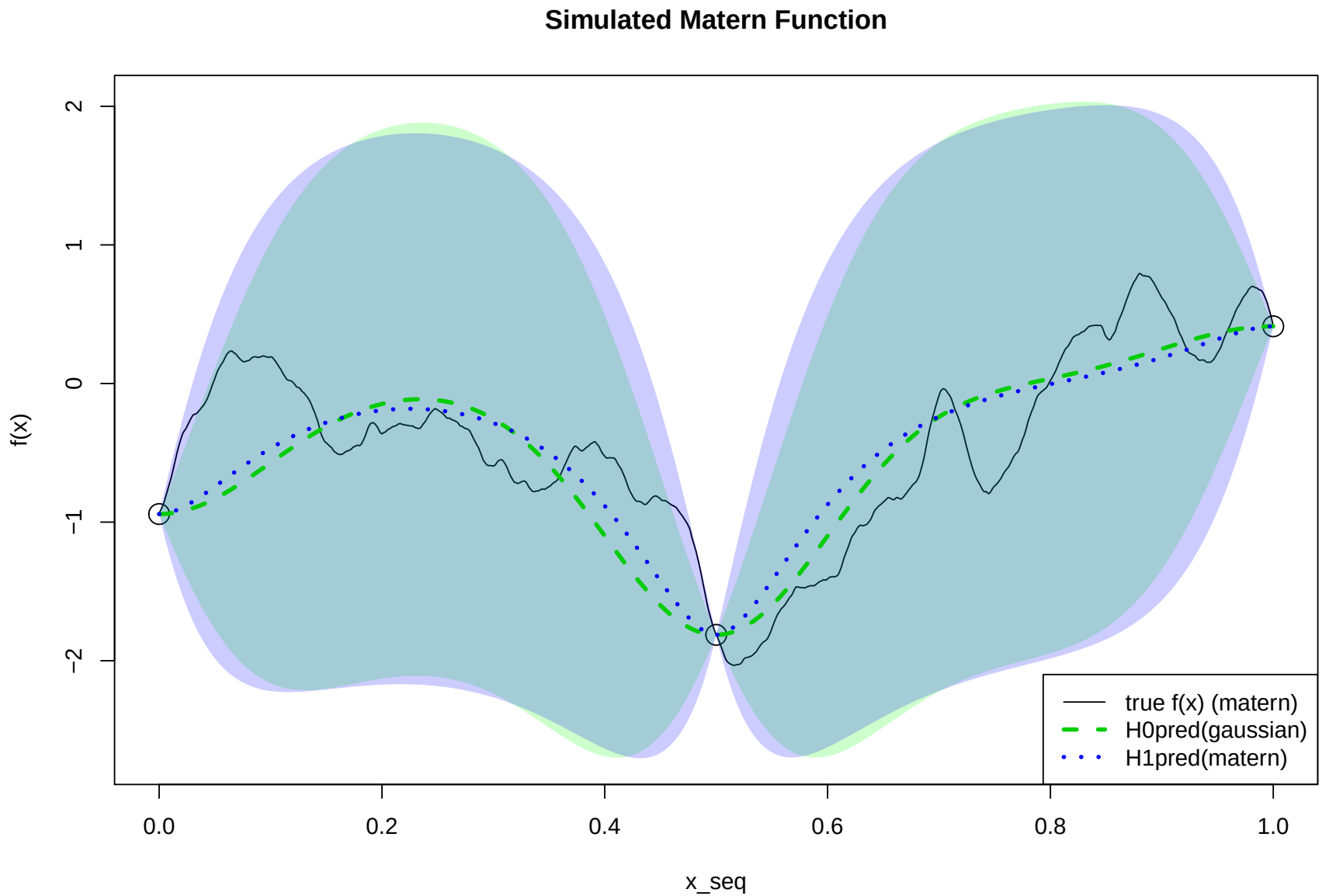
Space-filling by necessity

- The above shapes for $N = 6$ and $N = 11$ (new) designg points make sense if we consider that the uncertainty is highest in-between points, and so if we have a sufficient amount of data (from space-filling design for training data) already, the design for the new points will tend toward a space-filling design as well.
- $N = 3$ has dips in-between data points, which suggests that both are inadequate at determining the shape for areas of the support that are too far from training data.
  - there is that strange bump in the first half of the Wasserstein curve, though.
- If we consider the entire support, then the design will be space-filling by necessity, as we see above.
- The following examples show what happens when we look at space-filling training points as well as when we restrict the areas of the training points.

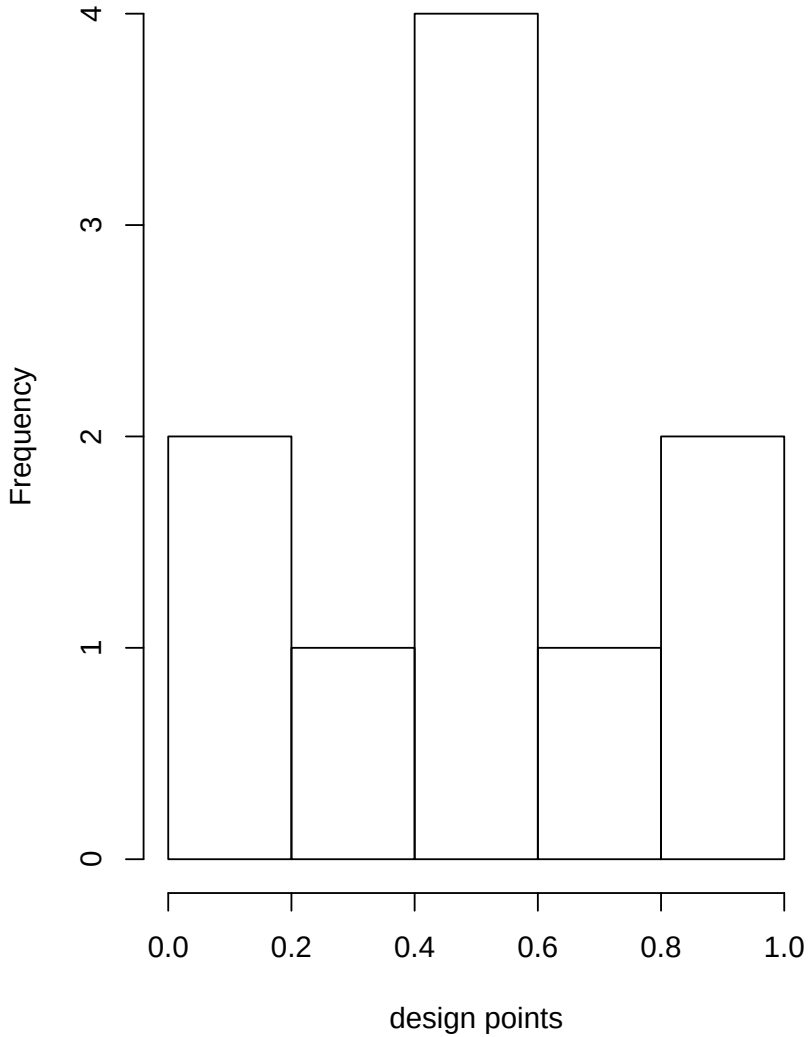
GP: Space-filling (N = 3)

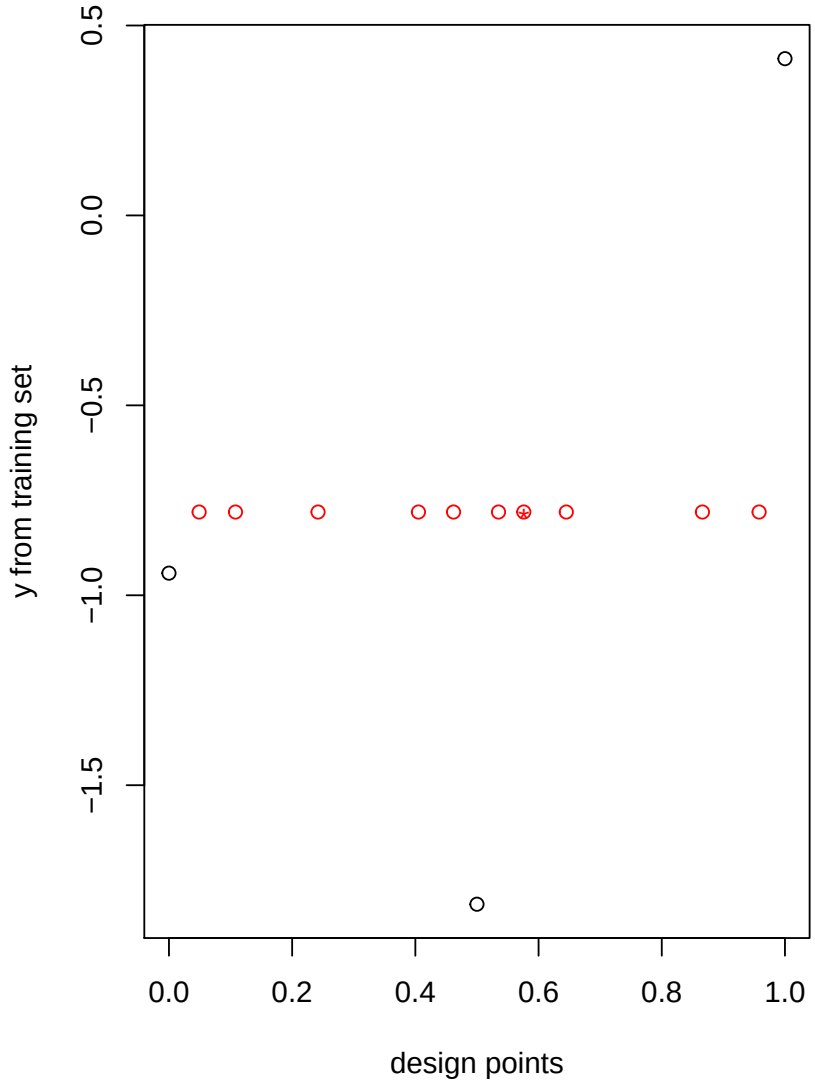
Function: Matern (N = 3)

For this example, I consider a function generated from the matern. H_0 is Gaussian, and H_1 is Matern, both with $\ell = 0.1$

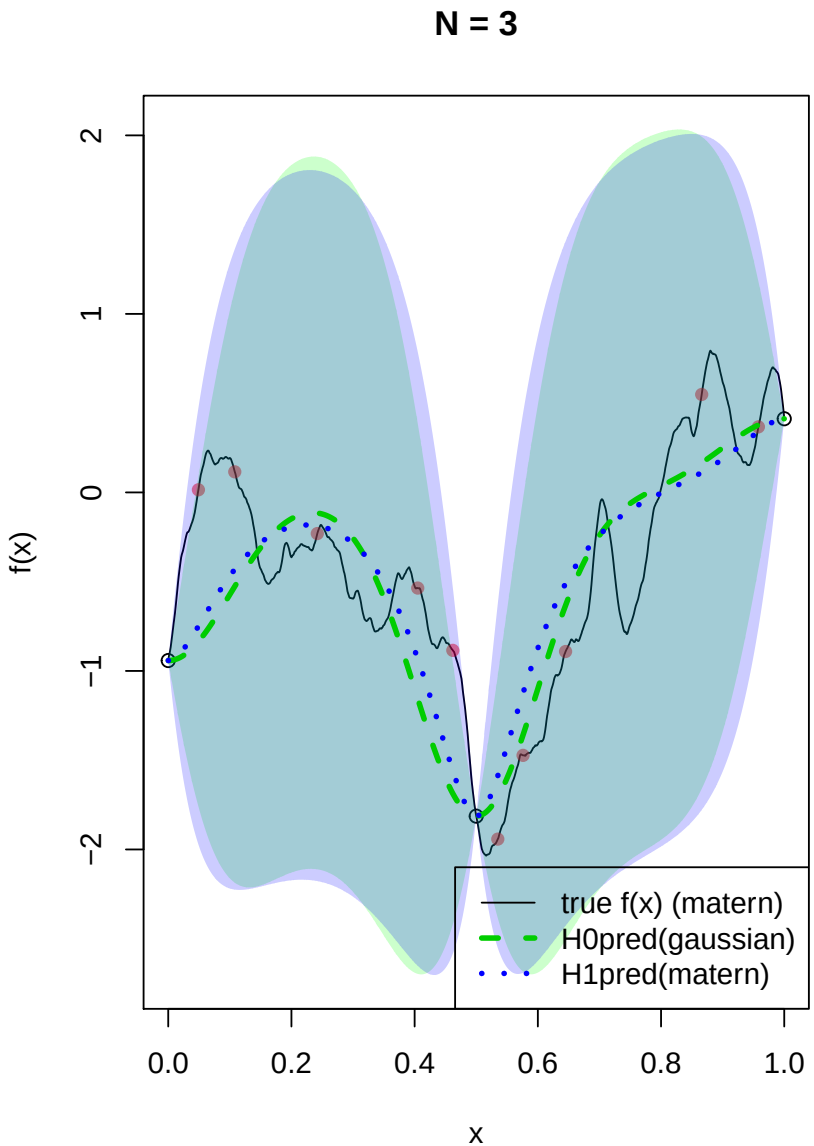


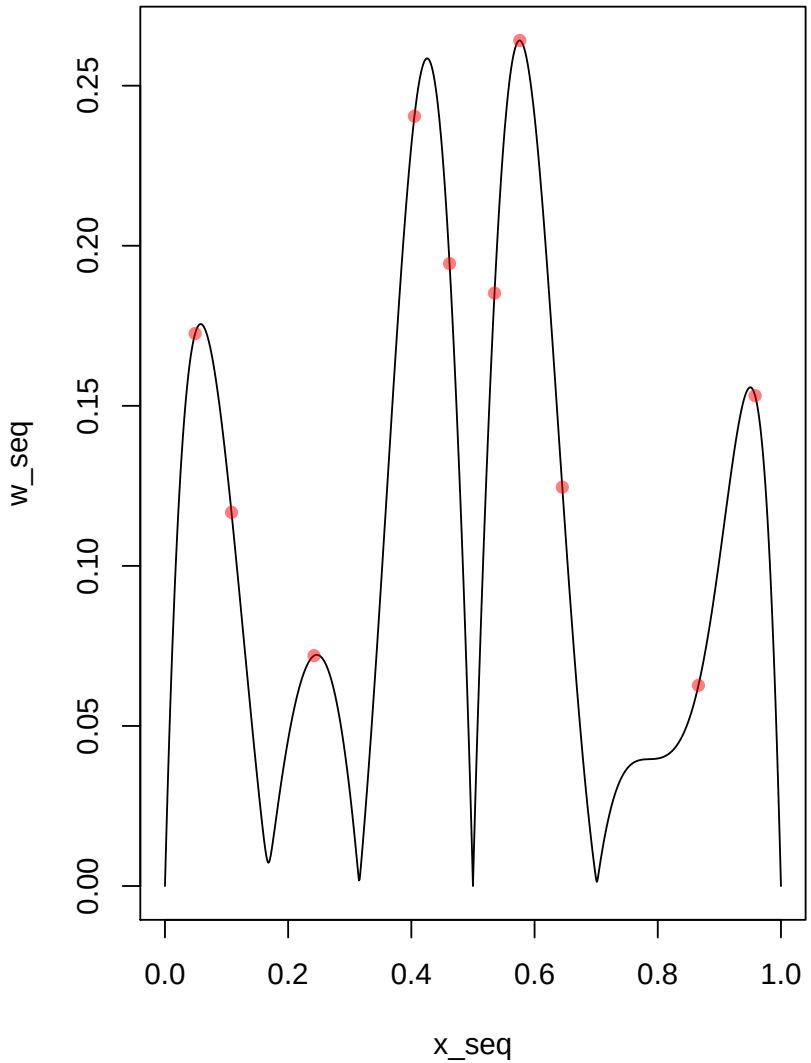
MMED


Frequency
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
1.0
design points


y from training set
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0
design points

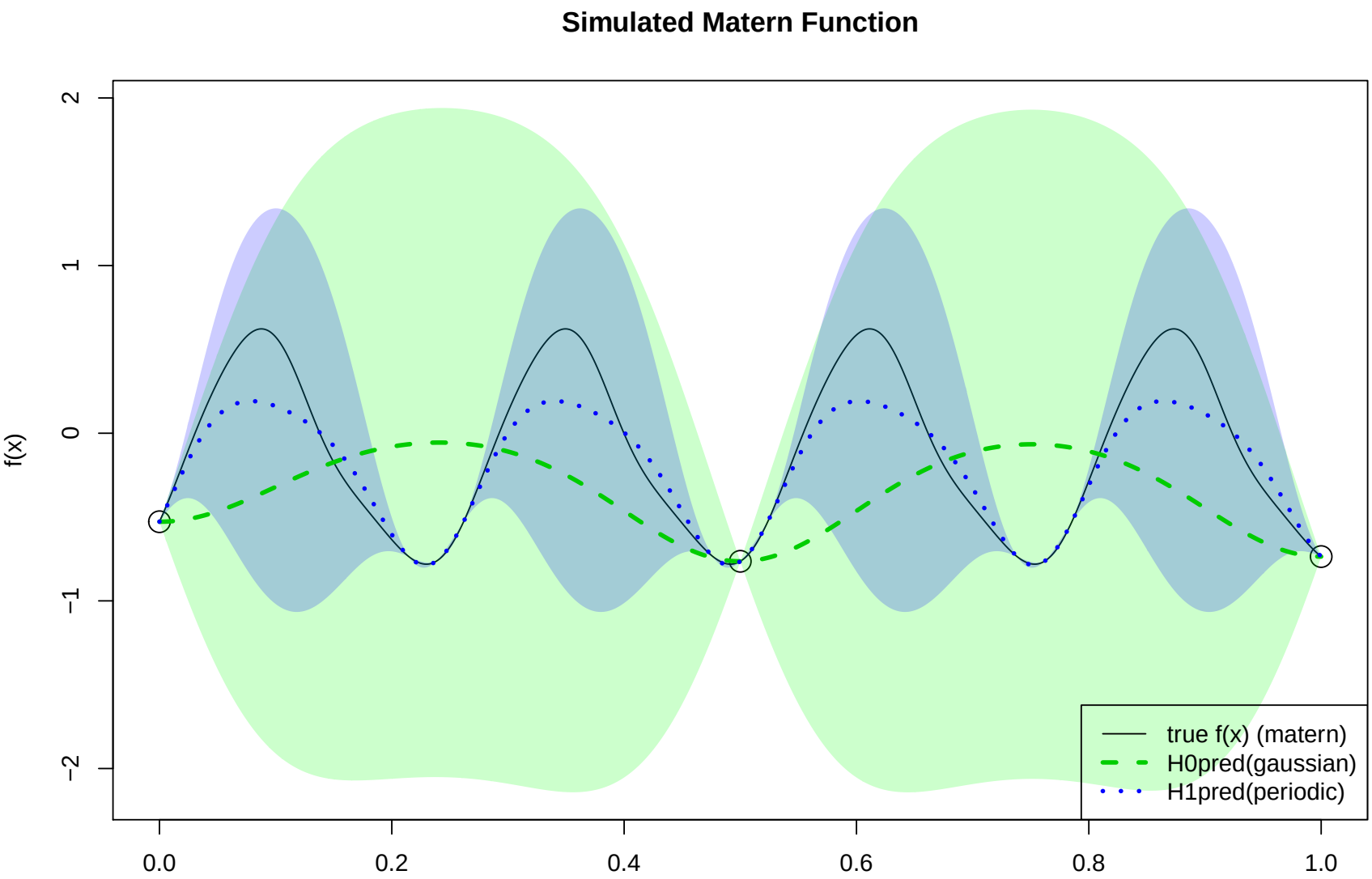
Wasserstein distance



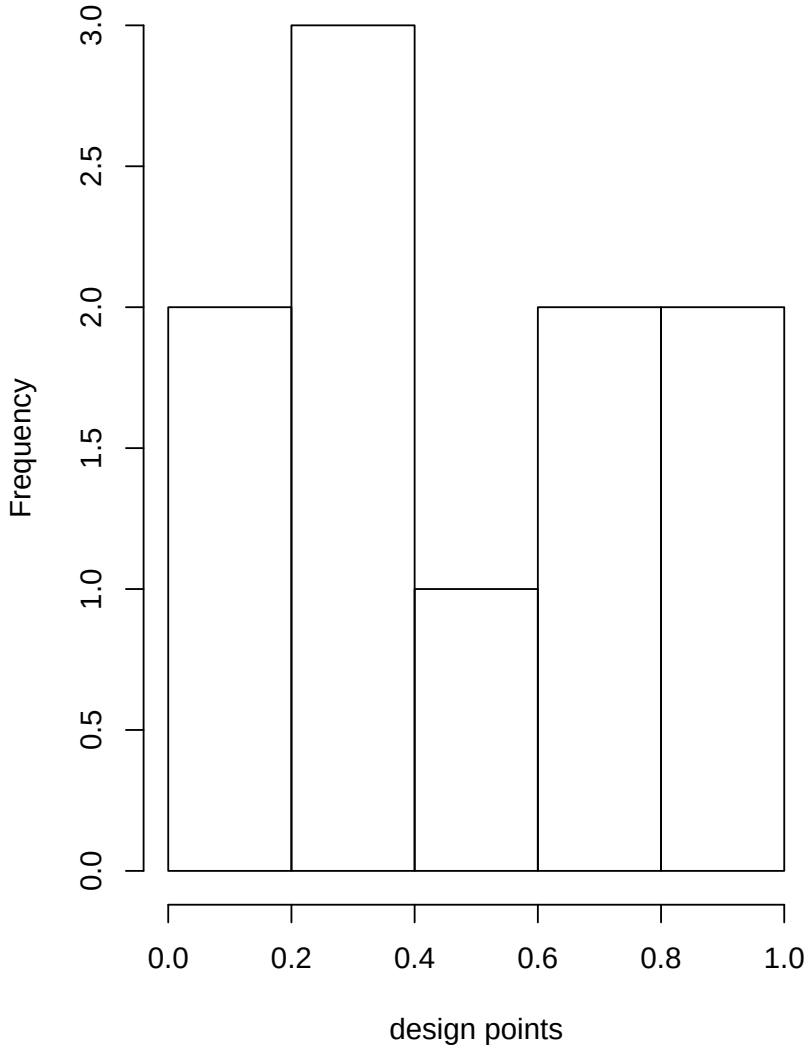


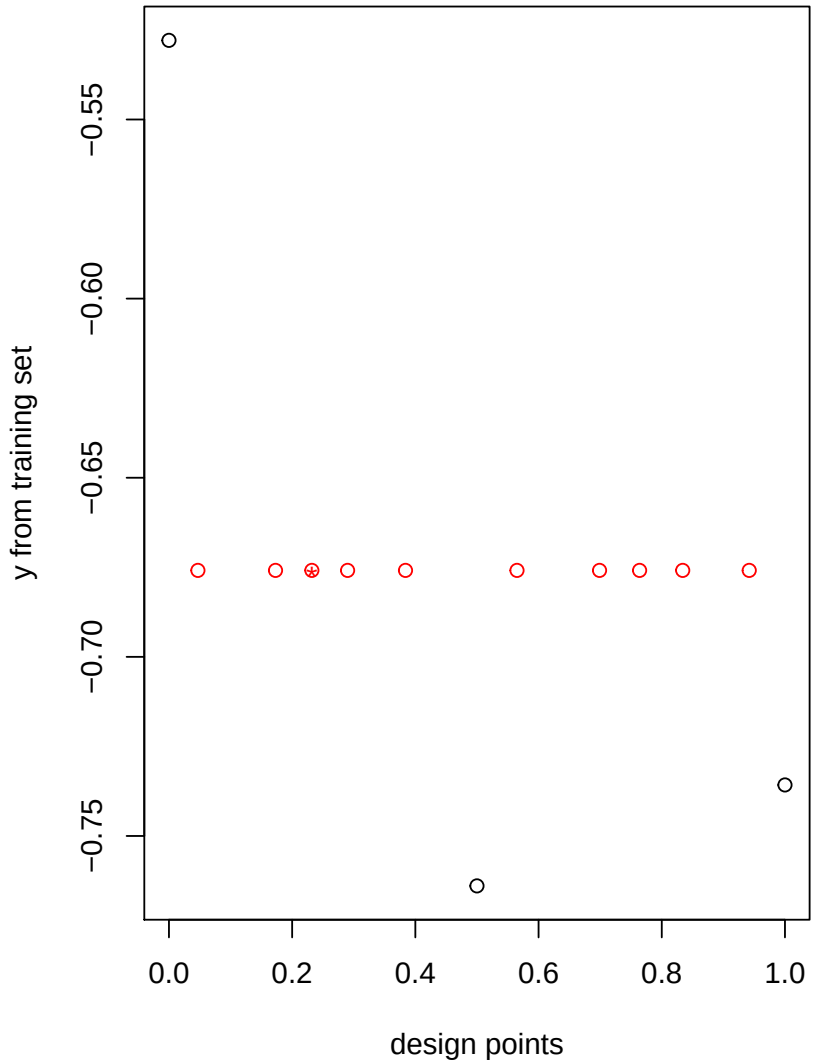
Function: Periodic

For this example, I consider a function generated from the periodic kernel. H_0 is Gaussian, and H_1 is Periodic, both with $\ell_G = 0.1$ and $\ell_P = 2$ respectively.

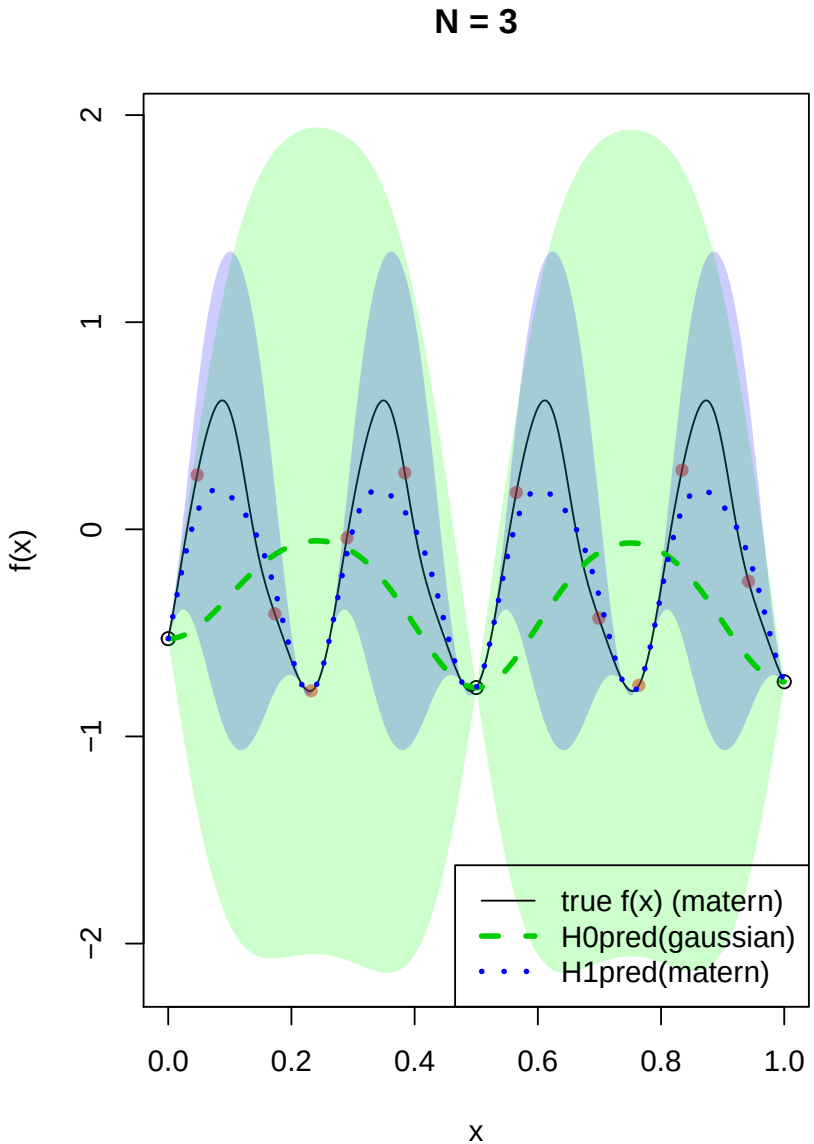


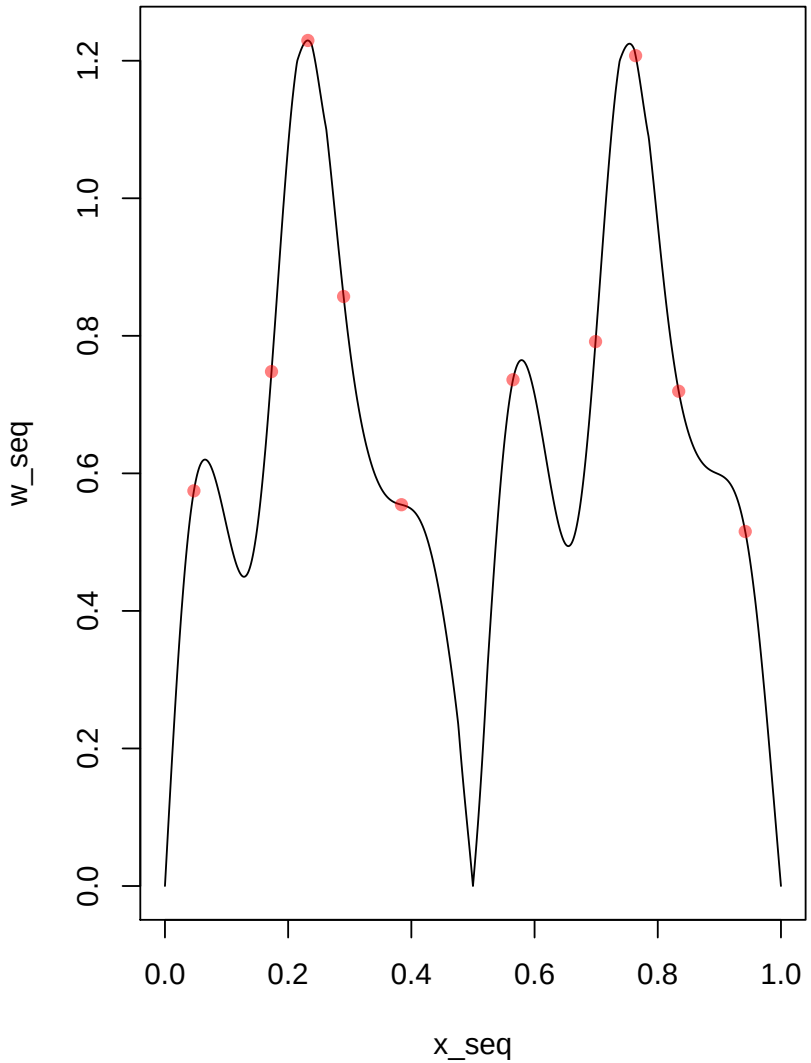
MMED


Frequency
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.2
0.4
0.6
0.8
1.0
design points


y from training set
−0.55
−0.60
−0.65
−0.70
−0.75
0.0
0.2
0.4
0.6
0.8
1.0
design points

Wasserstein distance

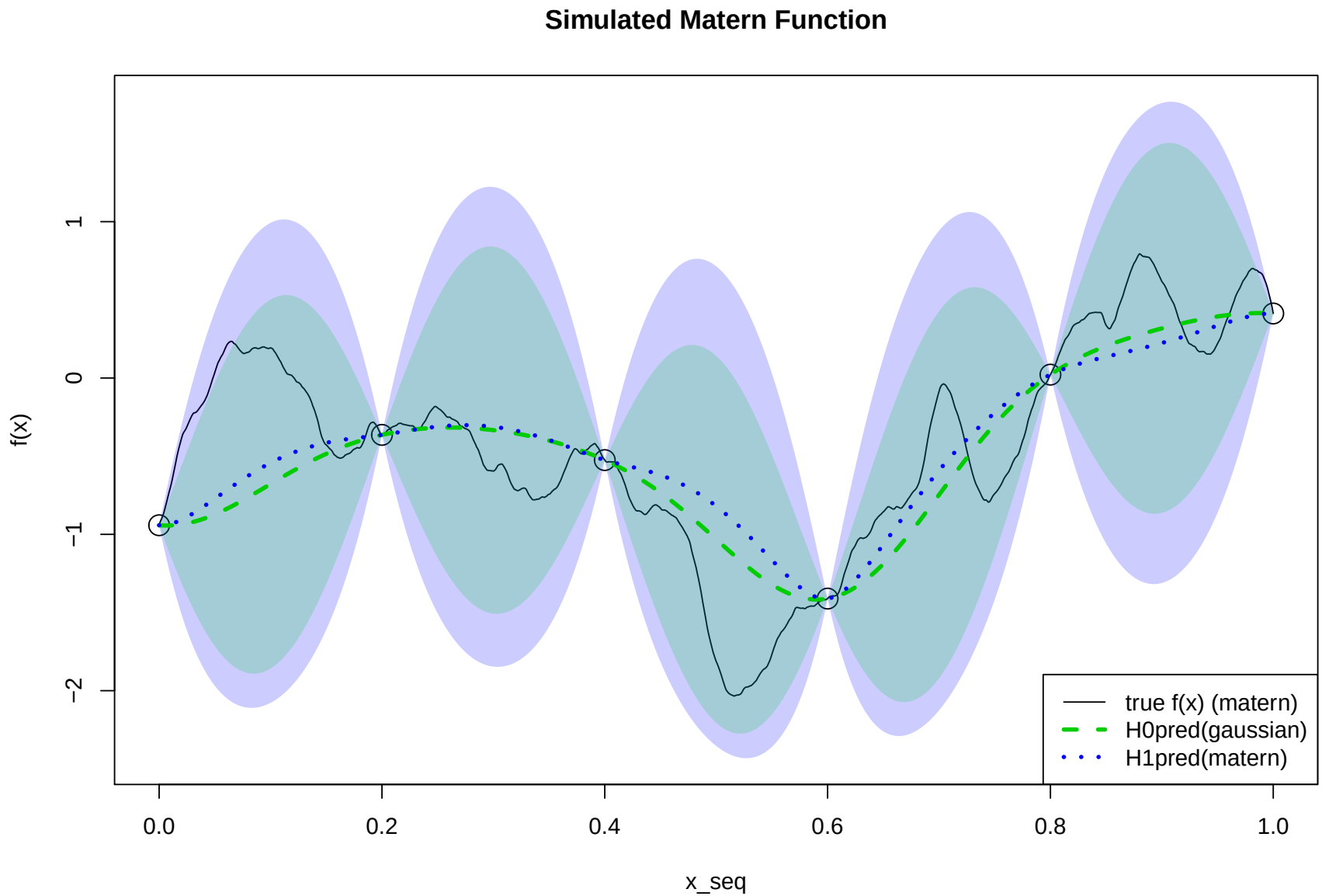




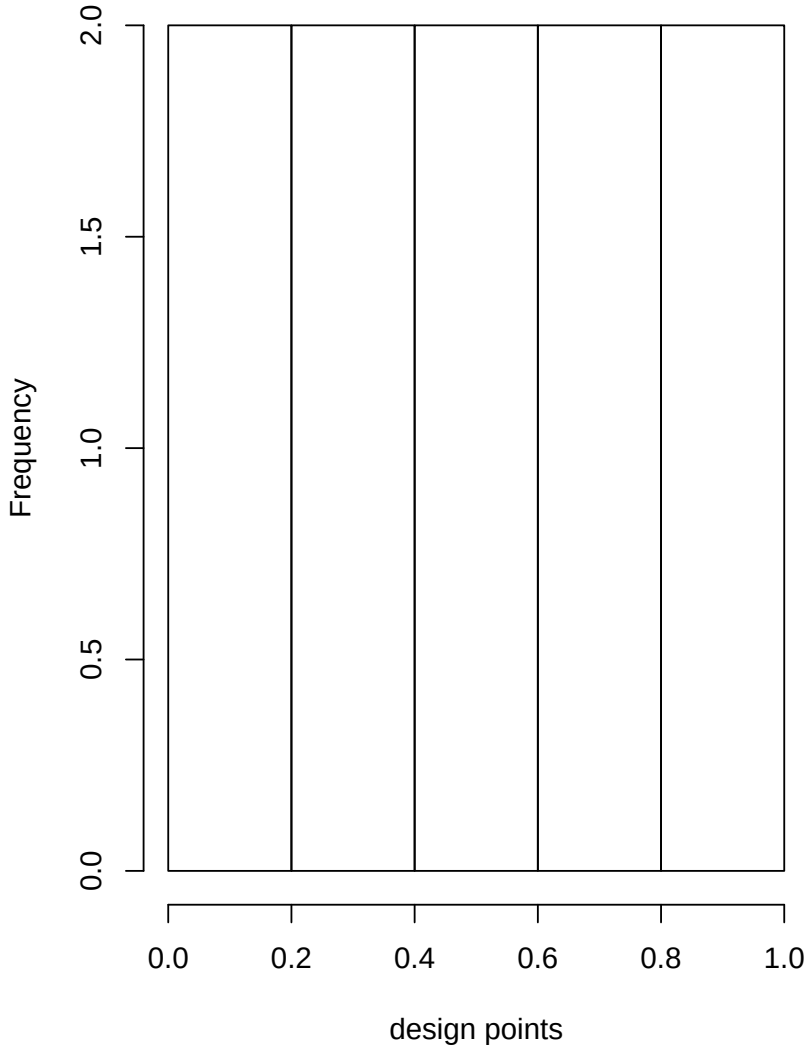
GP: Space-filling (N = 6)

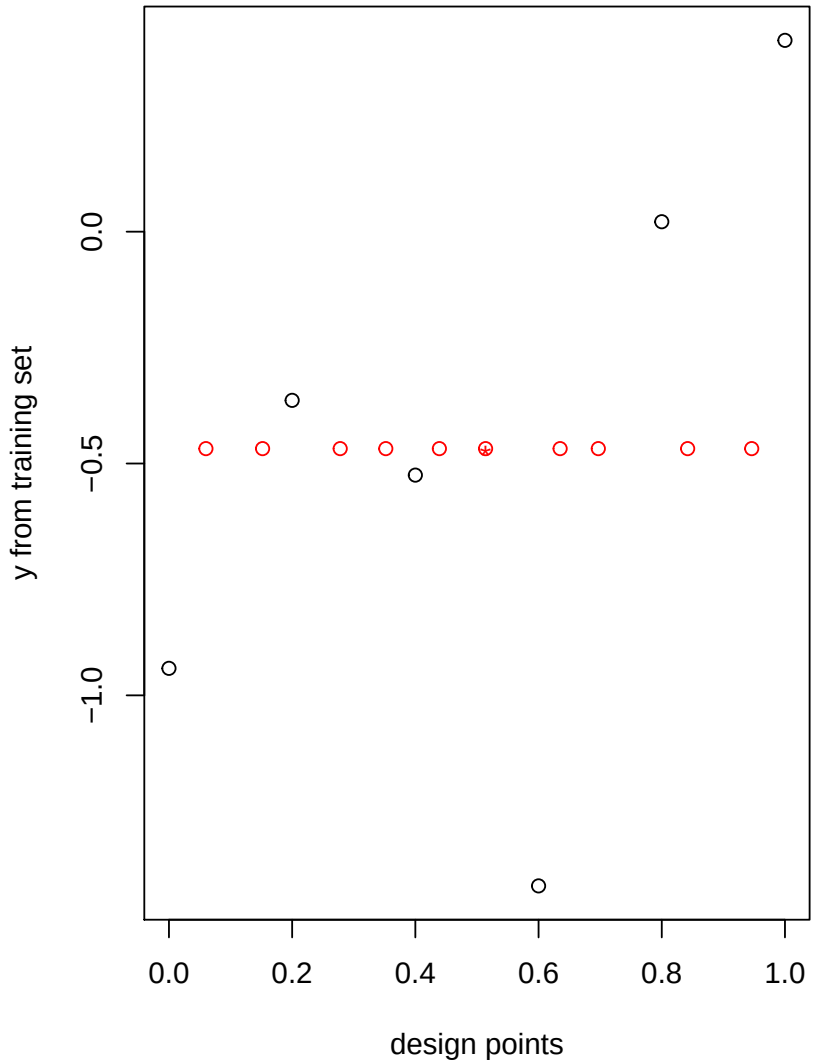
Function: Matern (N = 6)

For this example, I consider a function generated from the matern. H_0 is Gaussian, and H_1 is Matern, both with $\ell = 0.1$

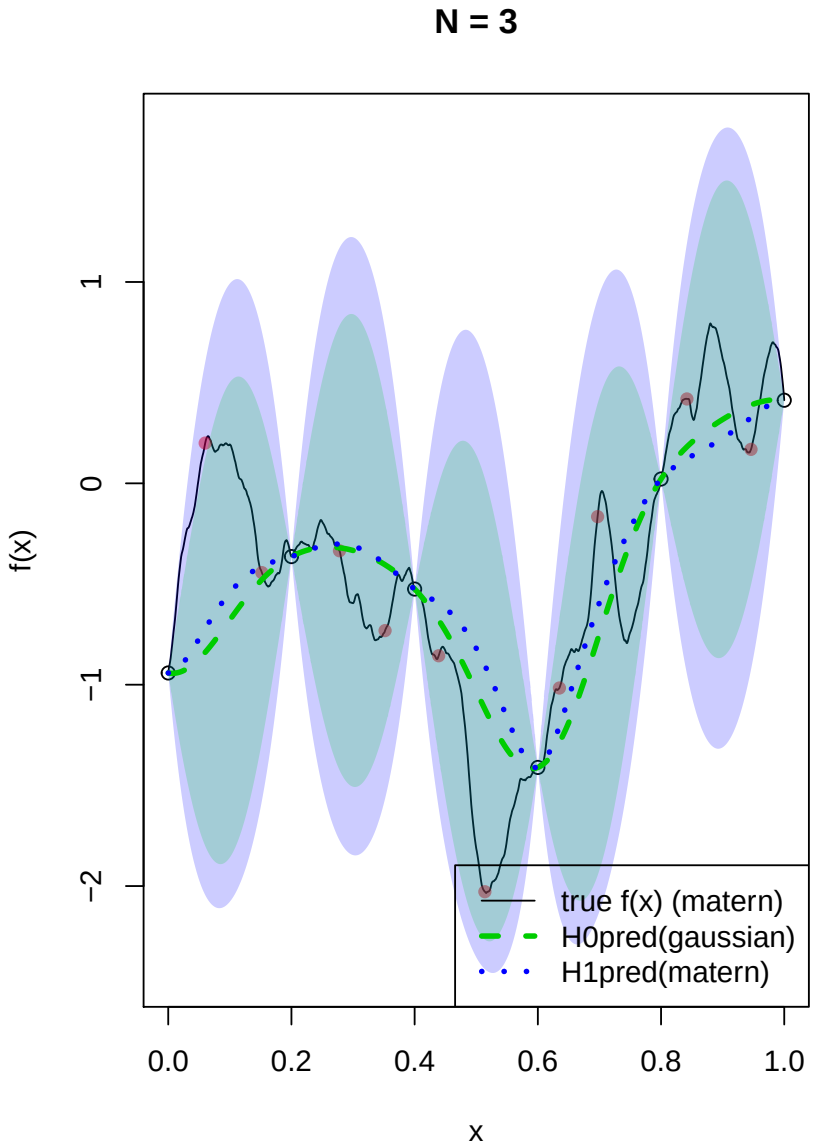


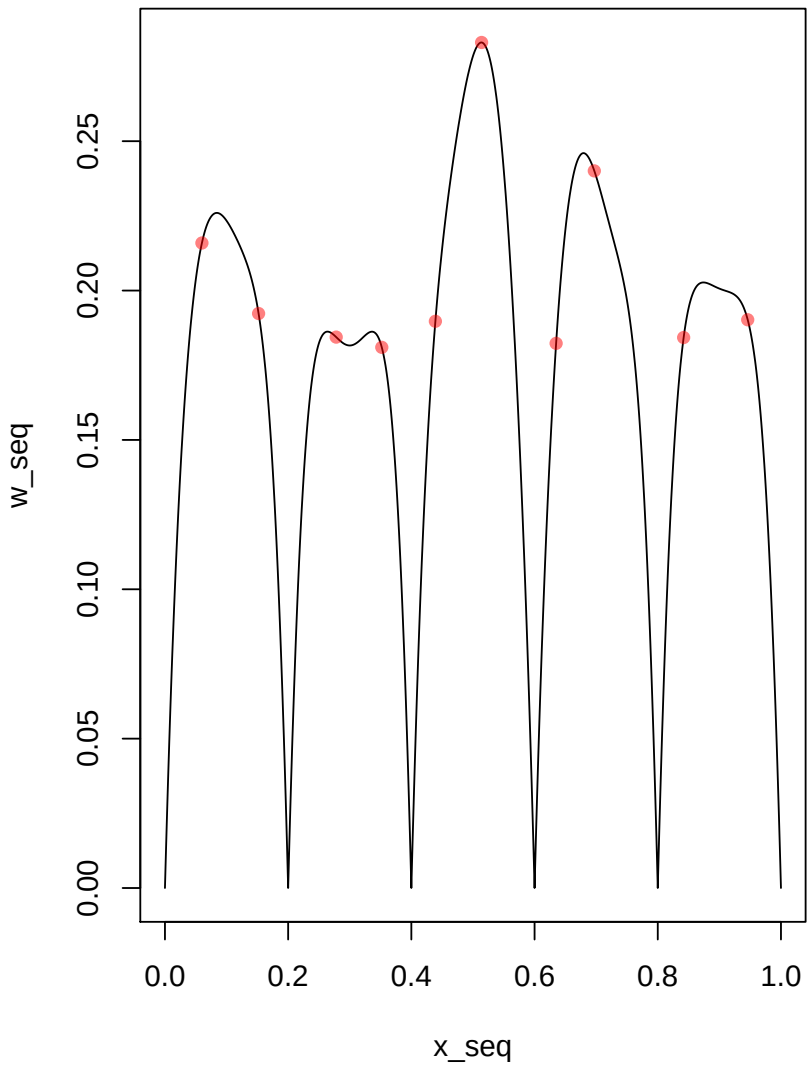
MMED


Frequency
0.0
0.5
1.0
1.5
2.0
0.0
0.2
0.4
0.6
0.8
1.0
design points


y from training set
−1.0
−0.5
0.0
0.0
0.2
0.4
0.6
0.8
1.0
design points

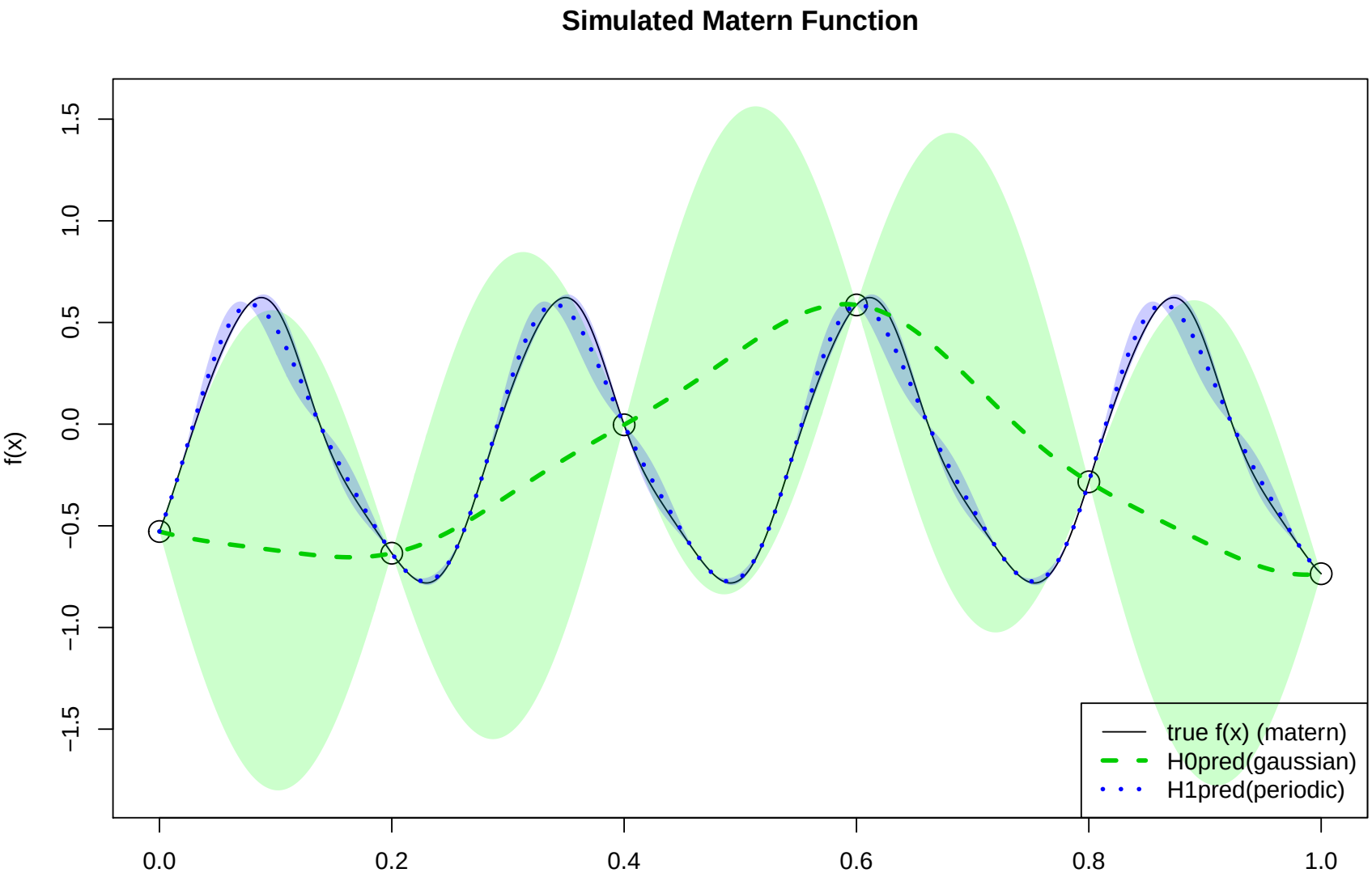
Wasserstein distance



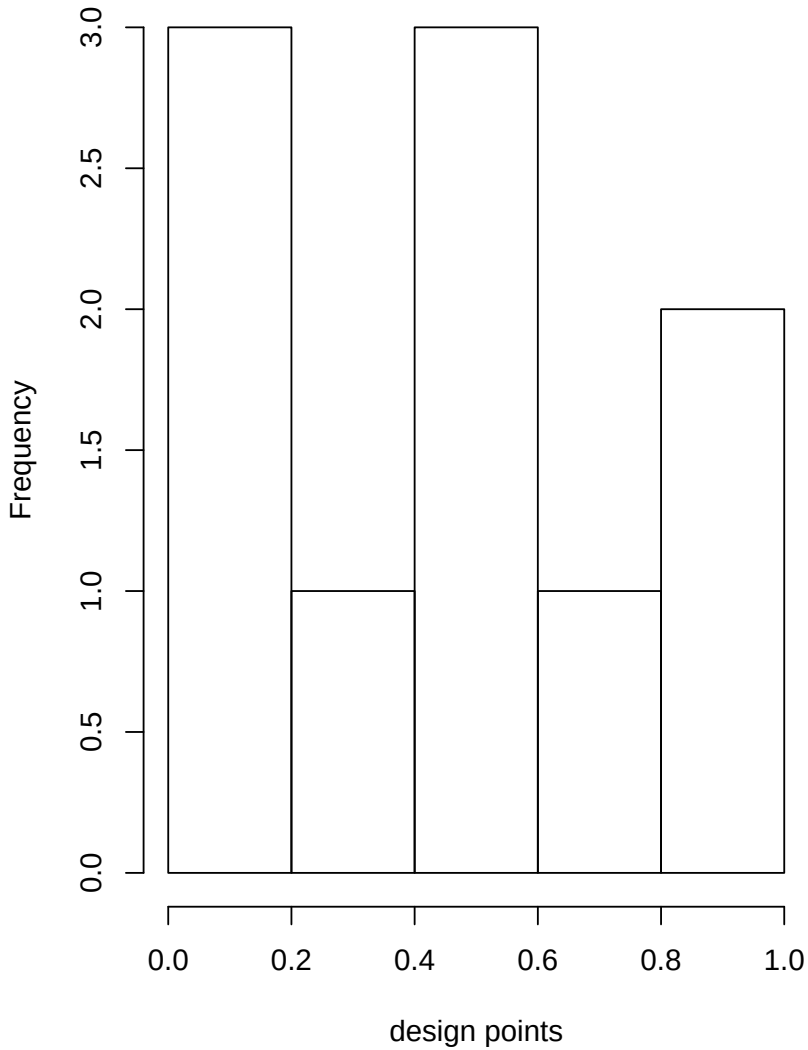


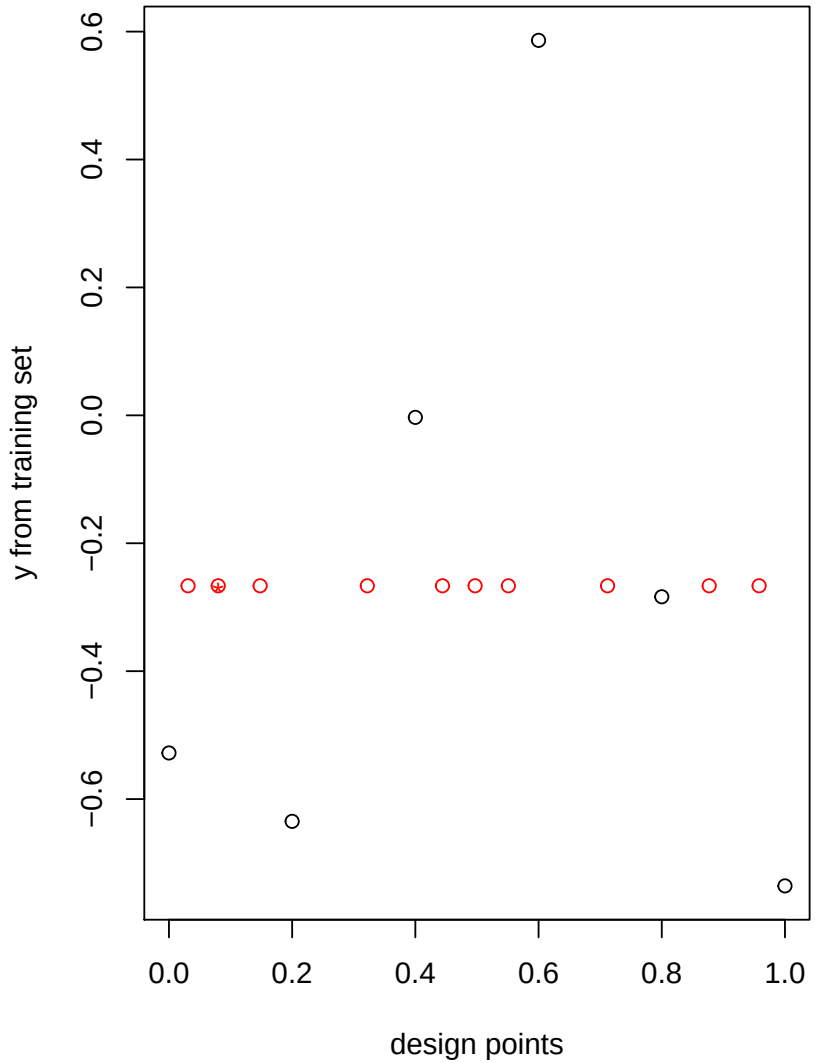
Function: Periodic

For this example, I consider a function generated from the periodic kernel. H_0 is Gaussian, and H_1 is Periodic, both with $\ell_G = 0.1$ and $\ell_P = 2$ respectively.

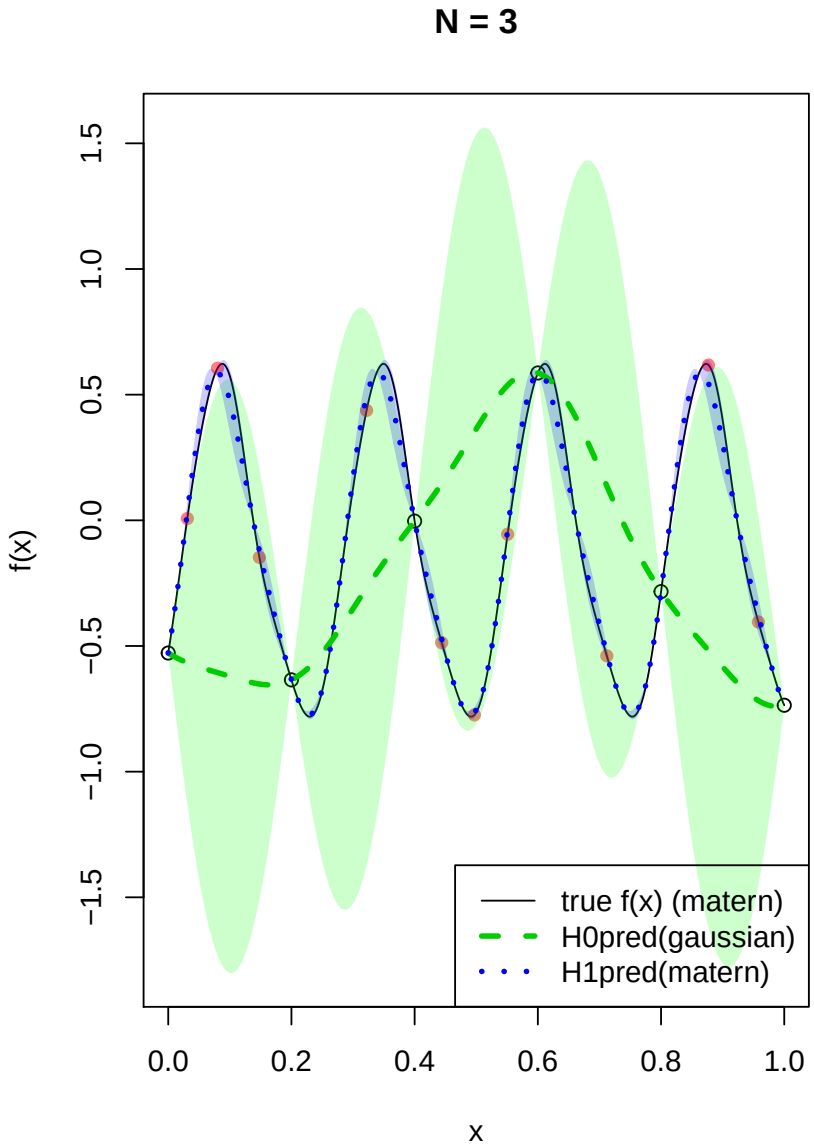


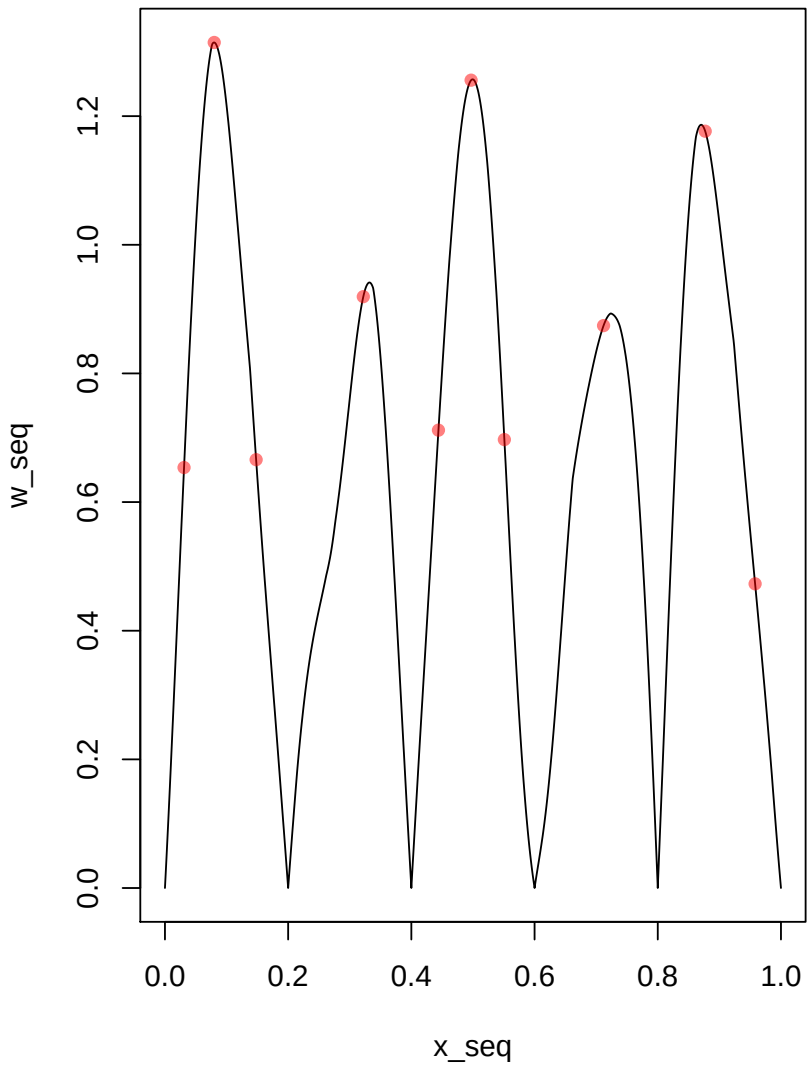
MMED


Frequency
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.2
0.4
0.6
0.8
1.0
design points


y from training set
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.0
0.2
0.4
0.6
0.8
1.0
design points

Wasserstein distance

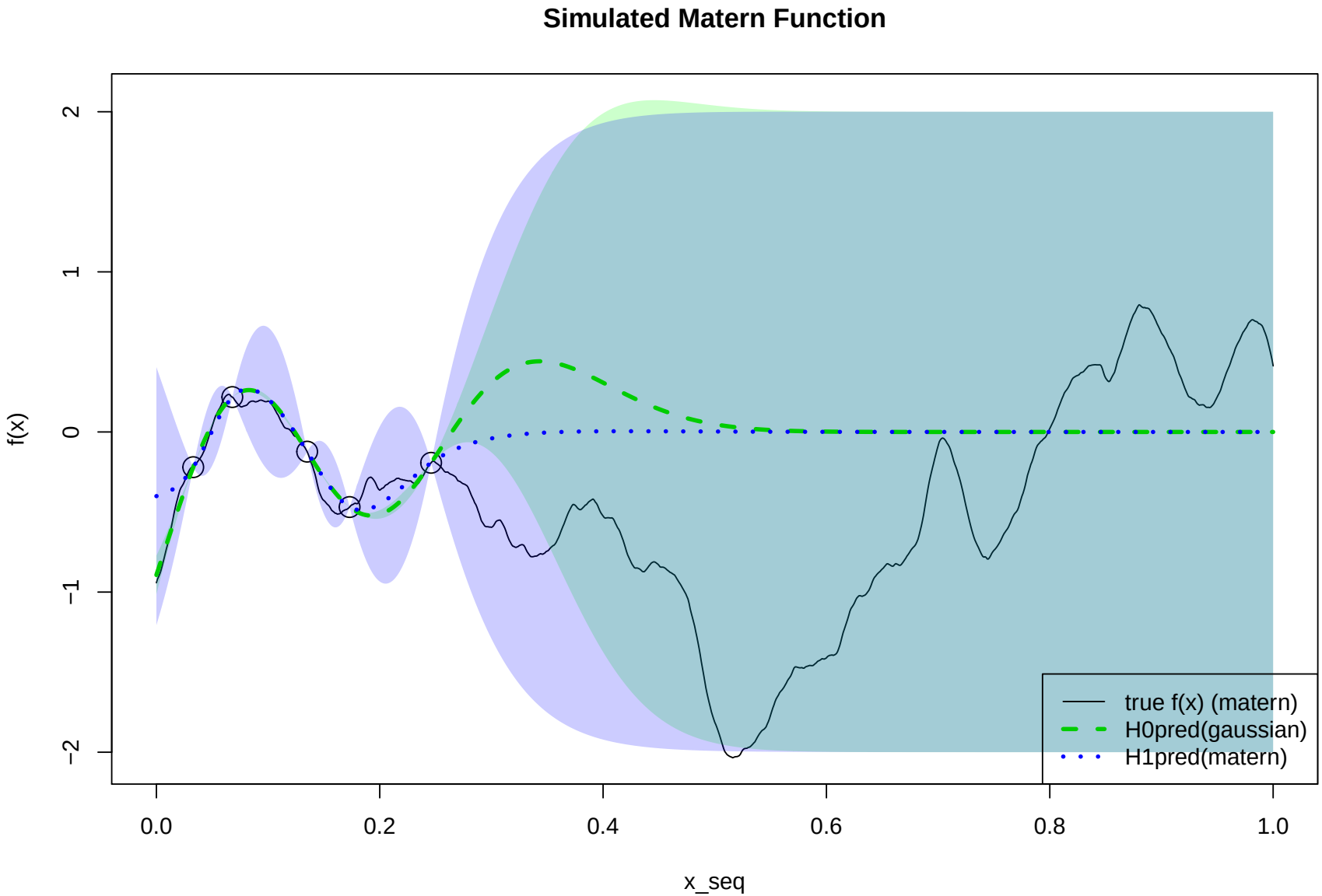




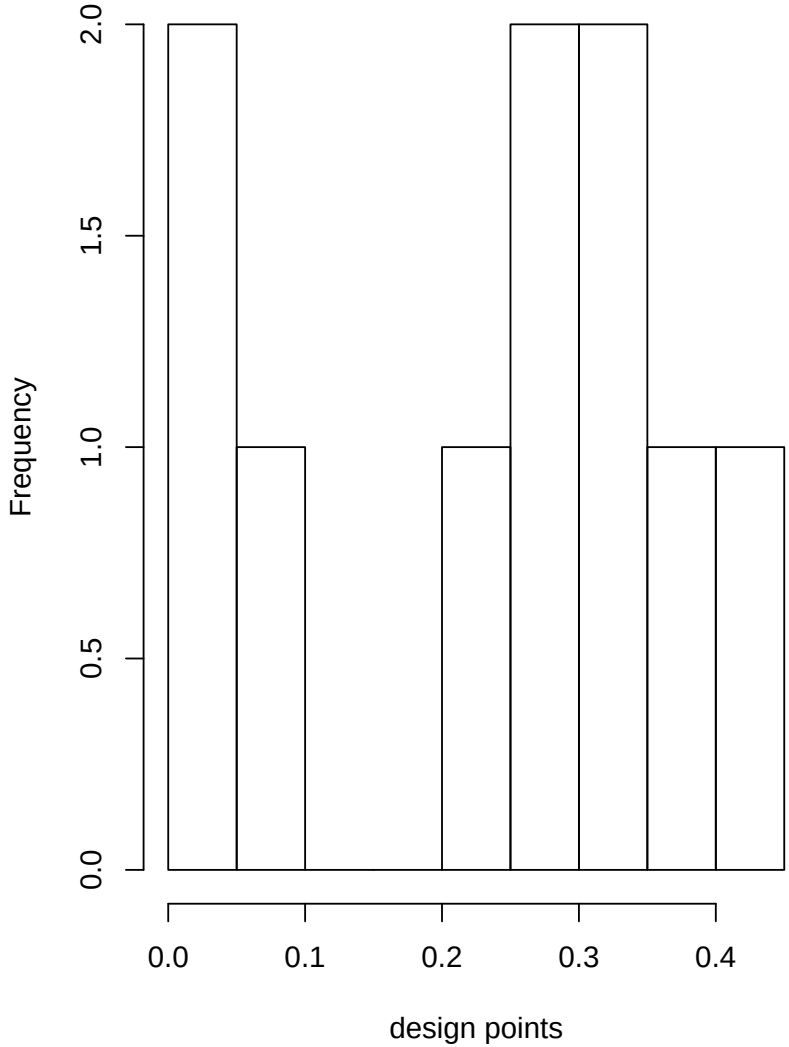
GP: MMED with Training Points in one region of support

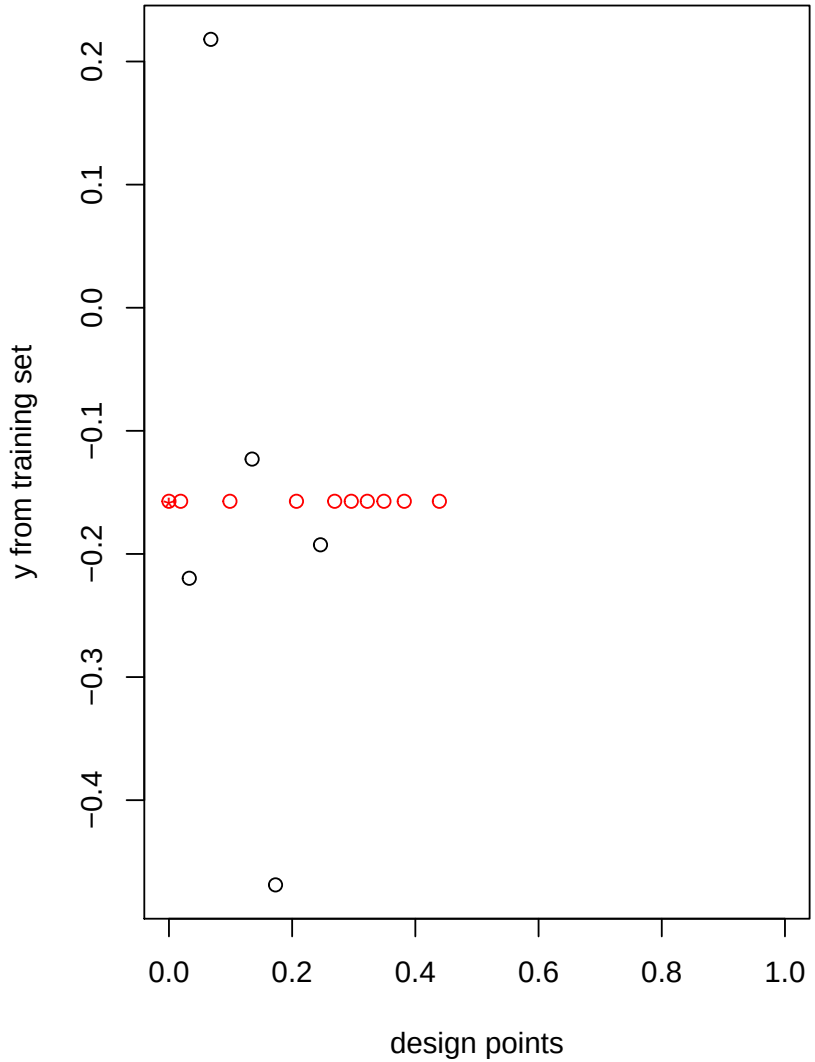
Function: Matern

I consider the Matern vs. Gaussian example again.

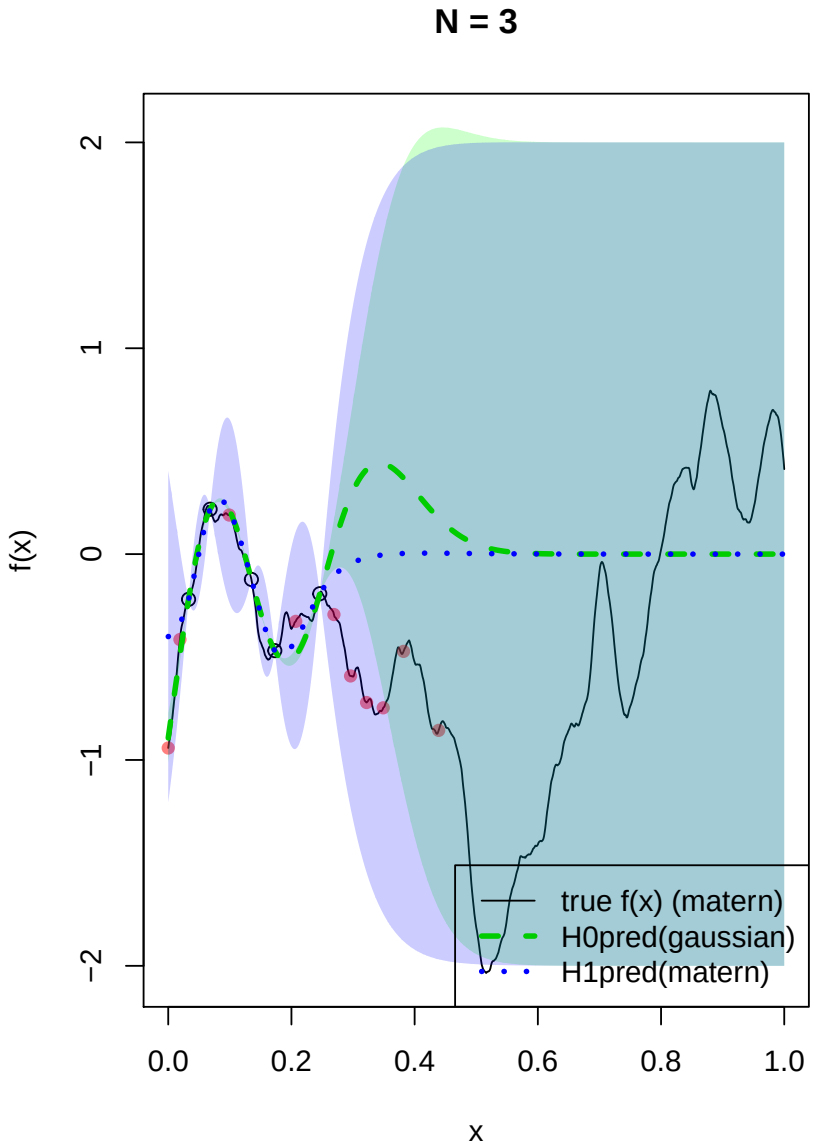


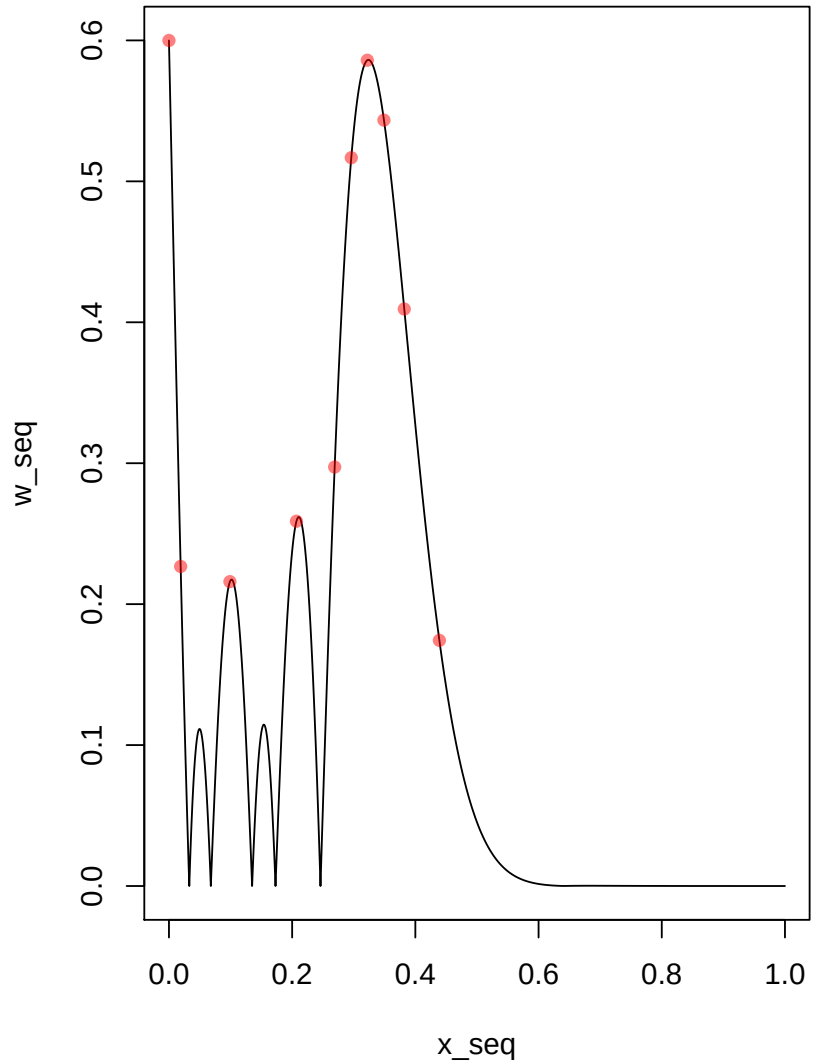
MMED


Frequency
0.0
0.5
1.0
1.5
2.0
0.0
0.1
0.2
0.3
0.4
design points


y from training set
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
0.0
0.2
0.4
0.6
0.8
1.0
design points

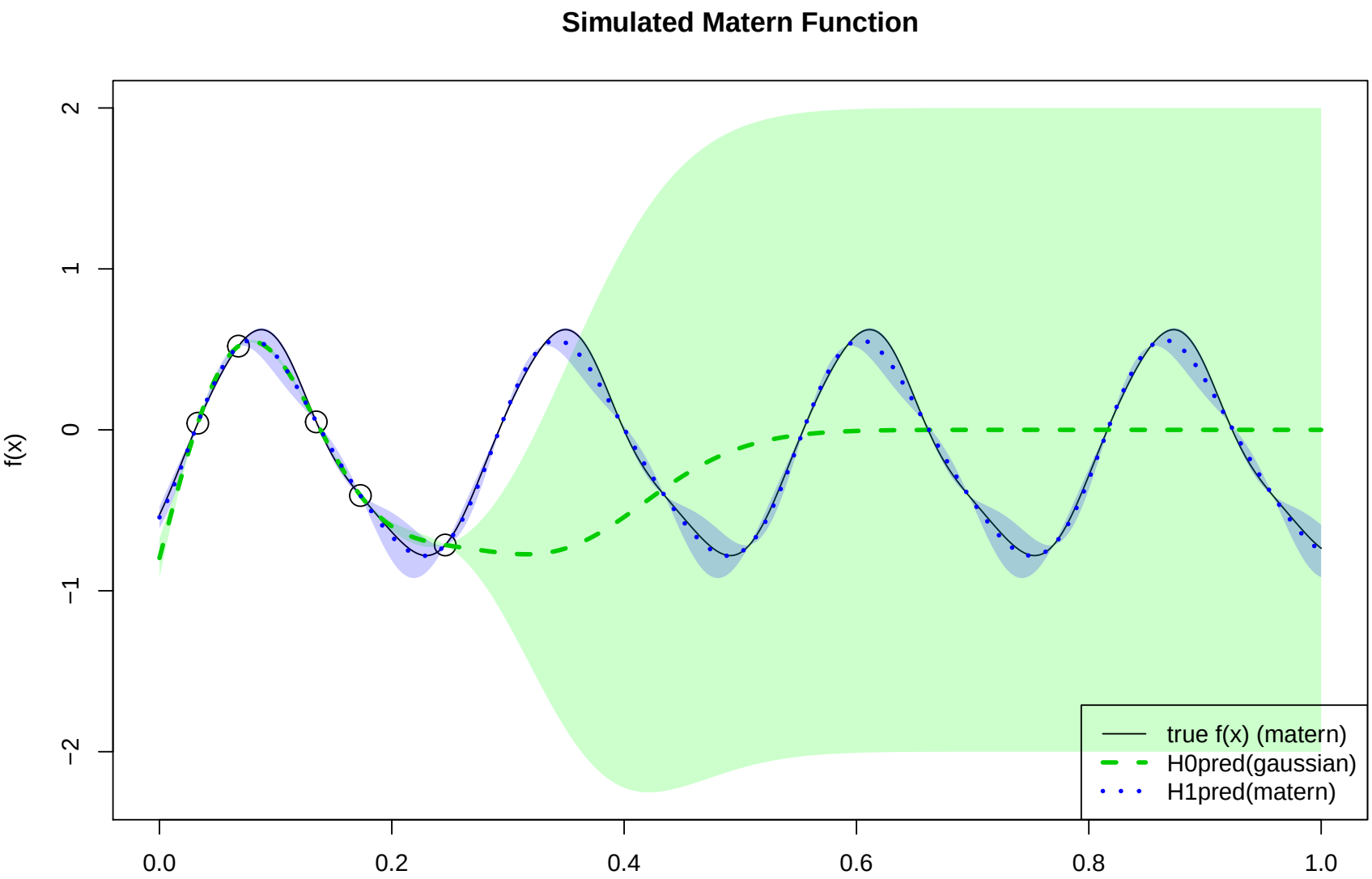
Wasserstein distance





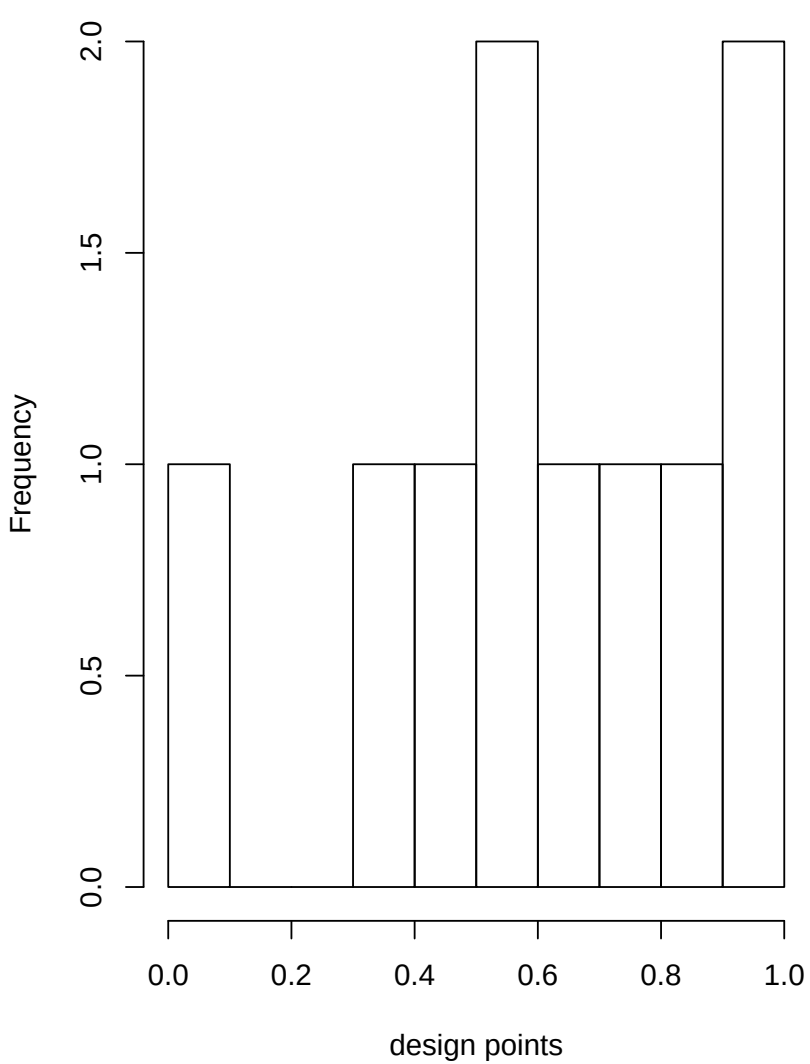
Function: Periodic

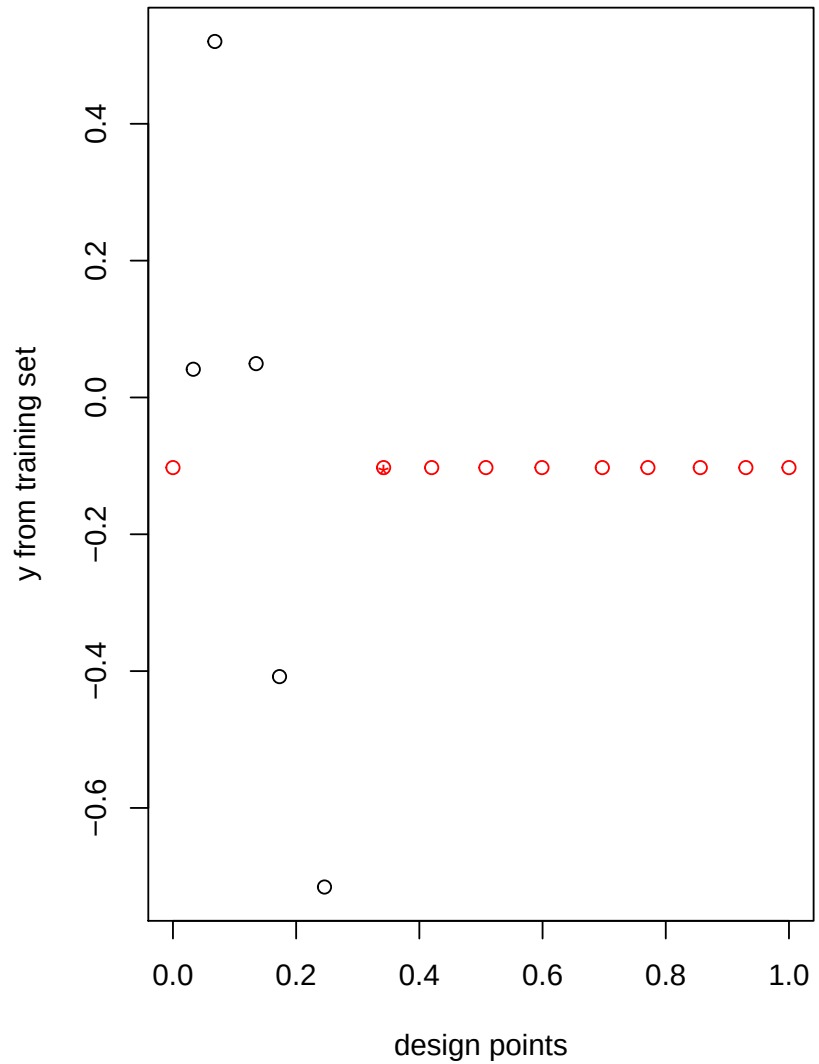
For this example, I consider a function generated from the periodic kernel. H_0 is Gaussian, and H_1 is Periodic, both with $\ell_G = 0.1$ and $\ell_P = 2$ respectively.



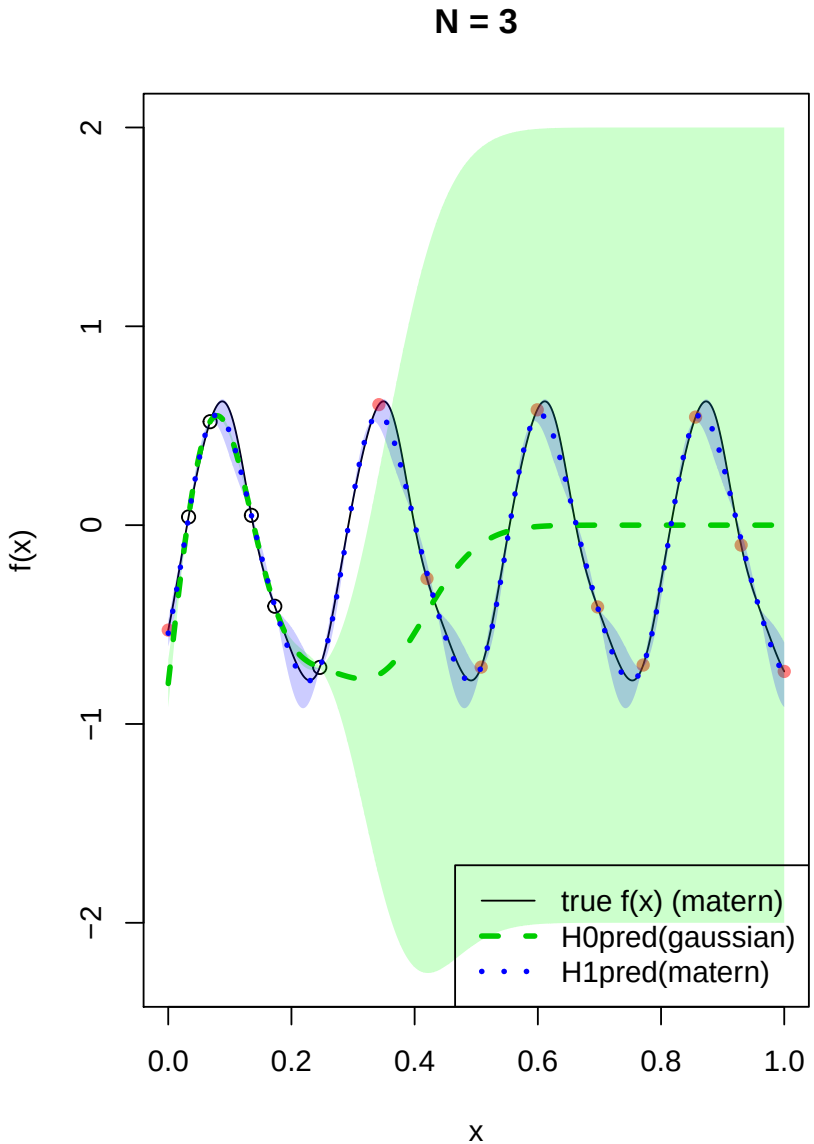
MMED localized training data

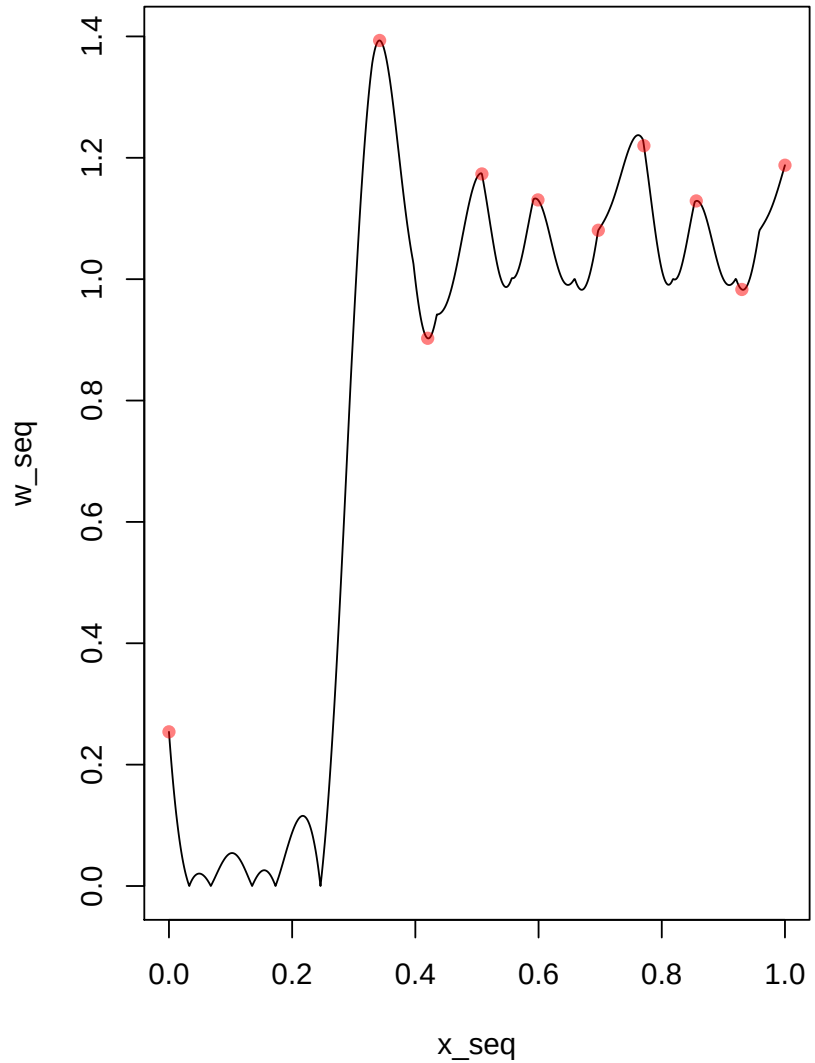
With the training data restricted to one area, the MMED is then applied over the entire support.





Wasserstein distance





Space-filling by necessity? (again)

- It seems like when the goal is to distinguish the two hypotheses, the MMED will tend towards a space-filling design.
- If this is done iteratively, the result would likely be a space-filling design that may not necessarily distinguish the two hypotheses because the Wasserstein distance might be largely based on where the uncertainty is greatest, which corresponds to the areas where the predicted functions differ the most, I think.
  - At least, this seems to be the case with comparing two stationary processes.
  - The result might be different when a stationary GP is compared to a non-stationary one, or when there is some other kind of structure happening in the data, such as with the periodic kernel.
- However, maybe the situation is more nuanced than it appears. Whatever the case, I think it would be wise to prioritize the situation where there are few points in general, in my simulations.