

---

# Introduction to SPLINTER

---

A tutorial for downloading, installing and using  
SPLINTER, with example problems.

March 8, 2016

## Contents

<b>1</b>	<b>Introduction <b>SPLINTER</b></b>	<b>2</b>
1.1	Intent and reader prerequisites . . . . .	2
1.2	Introduction to multivariate function approximation . . . . .	2
1.2.1	Sampling . . . . .	2
1.3	Mathematical background . . . . .	3
1.4	Prerequisites . . . . .	3
1.4.1	Eigen . . . . .	3

# 1 Introduction SPLINTER

The intention of this document is to inform and guide the user of SPLINTER, a library of method for multivariate function approximation with splines.

SPLINTER is an open-source library implemented in C++. It is publicly available at [github.com/bgrimstad/splinter](https://github.com/bgrimstad/splinter), and is distributed under the MPL 2.0 license [1].

## 1.1 Intent and reader prerequisites

This manual intends to give the reader a quick introduction to the SPLINTER framework, so she will be able set up and solve optimization problems. It will focus on the practical aspects and concepts rather than the mathematical theory of splines. The reader should, however, be acquainted with the following topics:

- Linear algebra: solving linear systems, the concept of sparsity, inner product, and vector norms.
- Regression... OLS, regularization.
- Polynomials, smoothness, support...

## 1.2 Introduction to multivariate function approximation

SPLINTER is a library of method for multivariate function approximation with splines. This introduction gives a quick exposition of the mathematics behind SPLINTER.

### 1.2.1 Sampling

Describe sampling here.

$$X = \{(x_i, y_i)\}_{i=1}^N \tag{1}$$

where  $y_i = f(x_i) + \epsilon_i$  for some unknown function  $f$ . The noise  $\epsilon_i$  is a stochastic term assumed to be normally distributed, i.e.  $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$

- Sampling on a regular grid versus scattered samples. Show figure.
- When sampling is computationally expensive (hence, time consuming) or limited (only a few samples are available)

The unknown function  $f$  is approximated by

$$\hat{f}(x) = \sum_{i=1}^M c_i \phi_i(x) \tag{2}$$

The functions  $\phi_i(x)$  are referred to as the *basis functions* of  $\hat{f}$ . An important property of  $\hat{f}$  is that it is linear in the coefficients  $c_i$ . All approximation models in SPLINTER are on the functional form in (REF). The models do however differ in what type of basis functions that are used.

The design matrix  $\Phi \in \mathbb{R}^{M \times N}$  is given as  $\Phi_{ij} = \phi_i(x_j)$ . For a vector of coefficients  $c$ ,  $\Phi c = \hat{y}$

### 1.3 Mathematical background

This section discusses a few of the mathematical concepts needed to better understand how SPLINTER works. The *epigraph form* is a transformation of an optimization problem to a form with a linear objective, which we need in SPLINTER. *Convexity* and *convex relaxations* are important topics in optimization theory in general, and in Branch-and-Bound algorithms such as SPLINTER, good convex relaxations are essential for a global solution to be found. The *B-spline* is a powerful approximation and interpolation technique that can be used in SPLINTER to approximate nonlinear functions with piecewise polynomials. Also, we can easily find good convex relaxations of B-splines, which is a very useful property. We also introduce global optimization and the Branch-and-Bound algorithm.

### 1.4 Prerequisites

SPLINTER relies on a few open-source libraries to function. The most important of these are listed below.

#### 1.4.1 Eigen

Eigen is a C++ template library for linear algebra. It includes functionality for Matlab-like matrix and vector operations, solvers and other related algorithms such as matrix decompositions. For more information on Eigen and installation instructions, the reader is referred to the [Eigen web site](#) [2].

## References

- [1] Bjarne Grimstad et al. SPLINTER: A library for multivariate function approximation with splines. <https://github.com/bgrimstad/splinter>, 2012. Accessed: 2015-05-16.
- [2] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. <http://eigen.tuxfamily.org>, 2010.