FREE
A PROGRAM FOR CONSTRAINED APPROXIMATION BY SPLINES WITH FREE KNOTS

TORSTEN SCHÜTZE

Abstract. In this paper a program for the computation and visualization of univariate splines with free knots is described. For given noisy data, the algorithm delivers a polynomial spline with an optimal knot sequence subject to constraints on derivatives.

Contents

Disclaimer: The software is provided "as is". The program is distributed in the hope that it will be useful, but *without any warranty.*

The primary purpose of this software is to provide a convenient framework for the numerical tests of the methods developed in [SS95b] and [SS95a], not to provide production software for the computation of splines with free knots.

1. Overview

The program **FREE** is intended for the computation and visualization of approximating univariate splines with free knots. It has been developed by the author during the last years and has to be considered as an experimental research code accompanying the theoretical investigations. However, we tried to achieve a reasonable amount of efficiency for the program by exploiting the structure of the problems as far as possible. We made some efforts to provide the source code with comments on the details of the implementation. The algorithms are based on the theory

Research was supported by Deutsche Forschungsgemeinschaft under grant Schm 968/2-1,2-2.

The author wishes to thank V. Kunert for the source files of [SK93] on which the algorithms for fixed knots rely.

developed in [SS95b] and [SS95a]. Therefore, in this paper we only give a short description of the main features of the program. The notations used in this manual are those of [SS95b] and [SS95a].

For given data (x_i, y_i) $(i = 1, \dots, m)$ **FREE** computes an univariate polynomial spline s of order k to the knot sequence

$$t_1 = \cdots = t_k = a < t_{k+1} \le \cdots \le t_n < b = t_{n+1} = \cdots = t_{n+k}$$

where $m \ge n$. The spline s is given in its B-spline representation $s(x) = \sum_{j=1}^{n} B_{j,k,\mathbf{t}}(x)\alpha_j$. The parameters of the spline are chosen in such a way that the following approximation or smoothing functional becomes minimal, respectively:
Approximation problem

$$\frac{1}{2}\sum_{i=1}^{m}[y_i - s(x_i)]^2 = \frac{1}{2}\left\|\mathbf{y} - \mathbf{B}(\mathbf{t})\boldsymbol{\alpha}\right\|^2 \to \min \tag{1}$$

Smoothing problem

$$\frac{1}{2}\sum_{i=1}^{m}[y_i - s(x_i)]^2 + \mu\frac{1}{2}\int_a^b \left[s^{(r)}(x)\right]^2 dx = \frac{1}{2}\left\|\begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix} - \begin{bmatrix}\mathbf{B}(\mathbf{t})\\ \sqrt{\mu}\mathbf{S}_r(\mathbf{t})\end{bmatrix}\boldsymbol{\alpha}\right\|^2 \to \min \tag{2}$$

Given a fixed number of knots, the minimization can be performed either with both the coefficients and the knots as variables or only with the coefficients as variables. If the location of knots is variable, we include a subset

$$\tilde{\mathbf{t}} = \left(t_{p(1)}, \dots, t_{p(l)}\right)^T \in \mathbb{R}^l$$

of the knot sequence $\mathbf{t}$, the so-called *free knots*, into the optimization process. The number of free knots is denoted by l. To avoid the coalescing of knots, the relative distance of two consecutive knots is bounded from below according to

$$t_{p(j)-1} + \epsilon\left(t_{p(j)+1} - t_{p(j)-1}\right) \le t_{p(j)} \le t_{p(j)+1} - \epsilon\left(t_{p(j)+1} - t_{p(j)-1}\right), \tag{3}$$

with the relative knot separation parameter $\epsilon > 0$. Analogously, also the absolute distance can be bounded

$$t_{p(j)} - t_{p(j)-1} \ge \epsilon \qquad t_{p(j)+1} - t_{p(j)} \ge \epsilon \tag{4}$$

using an absolute knot separation parameter $\epsilon > 0$. The numerical tests have shown that the relative knot separation parameter should be preferred. Both of the knot constraints (3) or (4) lead to a matrix formulation

$$\mathbf{C}\tilde{\mathbf{t}} - \mathbf{h} \ge \mathbf{0} \quad \text{with} \quad \mathbf{C} \in \mathbb{R}^{ncstr,l}, \mathbf{h} \in \mathbb{R}^{ncstr}. \tag{5}$$

In the program **FREE** either no shape constraints or constraints

$$l_i^{(p)} \le s^{(p)}(x) \le u_i^{(p)} \quad \forall x \in [t_i, t_{i+1}),\ i = k, \dots, n \tag{6}$$

on the derivative of the spline are considered. Here p is a fixed order of derivative, and $l_i^{(p)}$, $u_i^{(p)}$ $(i = k, \dots, n)$ are given bounds. The constraints (6) lead to the sufficient condition

$$\mathbf{L} \le \mathbf{D}_p(\mathbf{t})\boldsymbol{\alpha} \le \mathbf{U}. \tag{7}$$

1.1. **Approximation by splines with fixed knots.** If the knot sequence $\mathbf{t}$ is fixed we obtain linear least squares problems. Specifically the following problems can be solved with the program

- Approximation: compute an approximating least squares spline with prescribed, fixed knots and without shape constraints

$$\frac{1}{2}\left\|\mathbf{y} - \mathbf{B}(\tilde{\mathbf{t}})\boldsymbol{\alpha}\right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n}$$

- Smoothing: compute a smoothing spline with prescribed, fixed knots and without shape constraints

$$\frac{1}{2}\left\| \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} - \begin{bmatrix} \mathbf{B}(\tilde{\mathbf{t}}) \\ \sqrt{\mu}\mathbf{S}_r(\tilde{\mathbf{t}}) \end{bmatrix} \boldsymbol{\alpha} \right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n}$$

- Constrained Approximation: compute an approximating least squares spline with prescribed, fixed knots subject to shape constraints

$$\frac{1}{2}\left\|\mathbf{y} - \mathbf{B}(\tilde{\mathbf{t}})\boldsymbol{\alpha}\right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \text{ subject to } \mathbf{L} \le \mathbf{D}_p(\tilde{\mathbf{t}})\boldsymbol{\alpha} \le \mathbf{U}$$

- Constrained Smoothing: compute a smoothing spline with prescribed, fixed knots subject to shape constraints

$$\frac{1}{2}\left\| \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} - \begin{bmatrix} \mathbf{B}(\tilde{\mathbf{t}}) \\ \sqrt{\mu}\mathbf{S}_r(\tilde{\mathbf{t}}) \end{bmatrix} \boldsymbol{\alpha} \right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \text{ subject to } \mathbf{L} \le \mathbf{D}_p(\tilde{\mathbf{t}})\boldsymbol{\alpha} \le \mathbf{U}$$

The constrained approximation and smoothing problems are linear least squares problems with linear constraints where the occurring matrices have band structure. An efficient method for the solution of these problems that has been implemented in **FREE** is described in [SK93].

1.2. **Approximation by splines with free knots.** If we consider the location of the knots as variable, a nonlinear least squares problems with a special structure is obtained.

In the case of splines with free knots but without shape constraints, we are lead to separable least squares problems. The theory and the numerical method that apply in this case have been presented in [SS95b].

In the case of splines with free knots and with shape constraints a constrained semi-linear least squares problem is obtained. This problem has been investigated in [SS95a].

Summarizing, the following problems are considered

- Approximation: compute an approximating least squares spline with free knots and without shape constraints

  FAP – full approximation problem

$$\frac{1}{2}\left\|\mathbf{y} - \mathbf{B}(\tilde{\mathbf{t}})\boldsymbol{\alpha}\right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n,\tilde{\mathbf{t}}\in\mathbb{R}^l} \text{ subject to } \mathbf{C}\tilde{\mathbf{t}} - \mathbf{h} \ge \mathbf{0}$$

- Smoothing: compute a smoothing spline with free knots and without shape constraints

  FSP – full smoothing problem

$$\frac{1}{2}\left\| \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} - \begin{bmatrix} \mathbf{B}(\tilde{\mathbf{t}}) \\ \sqrt{\mu}\mathbf{S}_r(\tilde{\mathbf{t}}) \end{bmatrix} \boldsymbol{\alpha} \right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n,\tilde{\mathbf{t}}\in\mathbb{R}^l} \text{ subject to } \mathbf{C}\tilde{\mathbf{t}} - \mathbf{h} \ge \mathbf{0}$$

- Constrained Approximation: compute an approximating least squares spline with free knots subject to shape constraints

  FCAP – full constrained approximation problem

$$\frac{1}{2}\left\|\mathbf{y} - \mathbf{B}(\tilde{\mathbf{t}})\boldsymbol{\alpha}\right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n,\tilde{\mathbf{t}}\in\mathbb{R}^l} \text{ subject to } \mathbf{L} \le \mathbf{D}_p(\tilde{\mathbf{t}})\boldsymbol{\alpha} \le \mathbf{U} \text{ and } \mathbf{C}\tilde{\mathbf{t}} - \mathbf{h} \ge \mathbf{0}$$

- Constrained Smoothing: compute a smoothing spline with free knots subject to shape constraints

  FCSP – full constrained smoothing problem

$$\frac{1}{2}\left\| \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} - \begin{bmatrix} \mathbf{B}(\tilde{\mathbf{t}}) \\ \sqrt{\mu}\mathbf{S}_r(\tilde{\mathbf{t}}) \end{bmatrix} \boldsymbol{\alpha} \right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n,\tilde{\mathbf{t}}\in\mathbb{R}^l} \text{ subject to } \mathbf{L} \le \mathbf{D}_p(\tilde{\mathbf{t}})\boldsymbol{\alpha} \le \mathbf{U} \text{ and } \mathbf{C}\tilde{\mathbf{t}} - \mathbf{h} \ge \mathbf{0}$$

If we fix the knot sequence in the above four nonlinear problems we obtain linear least squares problems with respect to the coefficients $\boldsymbol{\alpha}$, namely the problems of Section 1.1 which can efficiently be solved. The general idea for solving the original ("full") nonlinear problems consists of a reduction to problems where only the free knots occur as variables. In the definition of these reduced problems the linear problems of Section 1.1 appear as subproblems. The following reduced problems are considered

- Approximation:

  RAP – reduced approximation problem

$$\frac{1}{2}\left\|\mathbf{y}-\mathbf{B}(\tilde{\mathbf{t}})\boldsymbol{\alpha}(\tilde{\mathbf{t}})\right\|^2 \to \min_{\tilde{\mathbf{t}}\in\mathbb{R}^l} \text{ subject to } \mathbf{C}\tilde{\mathbf{t}}-\mathbf{h}\geq\mathbf{0}$$

  where $\boldsymbol{\alpha}(\tilde{\mathbf{t}})$ solves

$$\frac{1}{2}\left\|\mathbf{y}-\mathbf{B}(\tilde{\mathbf{t}})\boldsymbol{\alpha}\right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n}$$

- Smoothing:

  RSP – reduced smoothing problem

$$\frac{1}{2}\left\|\begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix}-\begin{bmatrix}\mathbf{B}(\tilde{\mathbf{t}})\\ \sqrt{\mu}\mathbf{S}_r(\tilde{\mathbf{t}})\end{bmatrix}\boldsymbol{\alpha}(\tilde{\mathbf{t}})\right\|^2 \to \min_{\tilde{\mathbf{t}}\in\mathbb{R}^l} \text{ subject to } \mathbf{C}\tilde{\mathbf{t}}-\mathbf{h}\geq\mathbf{0}$$

  where $\boldsymbol{\alpha}(\tilde{\mathbf{t}})$ solves

$$\frac{1}{2}\left\|\begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix}-\begin{bmatrix}\mathbf{B}(\tilde{\mathbf{t}})\\ \sqrt{\mu}\mathbf{S}_r(\tilde{\mathbf{t}})\end{bmatrix}\boldsymbol{\alpha}\right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n}$$

- Constrained Approximation:

  RCAP – reduced constrained approximation problem

$$\frac{1}{2}\left\|\mathbf{y}-\mathbf{B}(\tilde{\mathbf{t}})\boldsymbol{\alpha}(\tilde{\mathbf{t}})\right\|^2 \to \min_{\tilde{\mathbf{t}}\in\mathbb{R}^l} \text{ subject to } \mathbf{C}\tilde{\mathbf{t}}-\mathbf{h}\geq\mathbf{0}$$

  where $\boldsymbol{\alpha}(\tilde{\mathbf{t}})$ solves

$$\frac{1}{2}\left\|\mathbf{y}-\mathbf{B}(\tilde{\mathbf{t}})\boldsymbol{\alpha}\right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \text{ subject to } \mathbf{L}\leq\mathbf{D}_p(\tilde{\mathbf{t}})\boldsymbol{\alpha}\leq\mathbf{U}$$

- Constrained Smoothing:

  RCSP – reduced constrained smoothing problem

$$\frac{1}{2}\left\|\begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix}-\begin{bmatrix}\mathbf{B}(\tilde{\mathbf{t}})\\ \sqrt{\mu}\mathbf{S}_r(\tilde{\mathbf{t}})\end{bmatrix}\boldsymbol{\alpha}(\tilde{\mathbf{t}})\right\|^2 \to \min_{\tilde{\mathbf{t}}\in\mathbb{R}^l} \text{ subject to } \mathbf{C}\tilde{\mathbf{t}}-\mathbf{h}\geq\mathbf{0}$$

  where $\boldsymbol{\alpha}(\tilde{\mathbf{t}})$ solves

$$\frac{1}{2}\left\|\begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix}-\begin{bmatrix}\mathbf{B}(\tilde{\mathbf{t}})\\ \sqrt{\mu}\mathbf{S}_r(\tilde{\mathbf{t}})\end{bmatrix}\boldsymbol{\alpha}\right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \text{ subject to } \mathbf{L}\leq\mathbf{D}_p(\tilde{\mathbf{t}})\boldsymbol{\alpha}\leq\mathbf{U}$$

In all of the above four cases (approximation or smoothing, with or without shape constraints) the reduced problem is a nonlinear least squares problem with linear inequality constraints, i.e.,

$$\frac{1}{2}\left\|\mathbf{F}(\tilde{\mathbf{t}})\right\|^2 \to \min_{\tilde{\mathbf{t}}\in\mathbb{R}^l} \quad \text{subject to} \quad \mathbf{C}\tilde{\mathbf{t}}-\mathbf{h}\geq\mathbf{0}$$

with appropriate residual function $\mathbf{F}$. The reduced problem will be solved by a generalized Gauss-Newton method. This Gauss-Newton method is described in [SS95b], [SS95a], and in more detail in [Sch93].

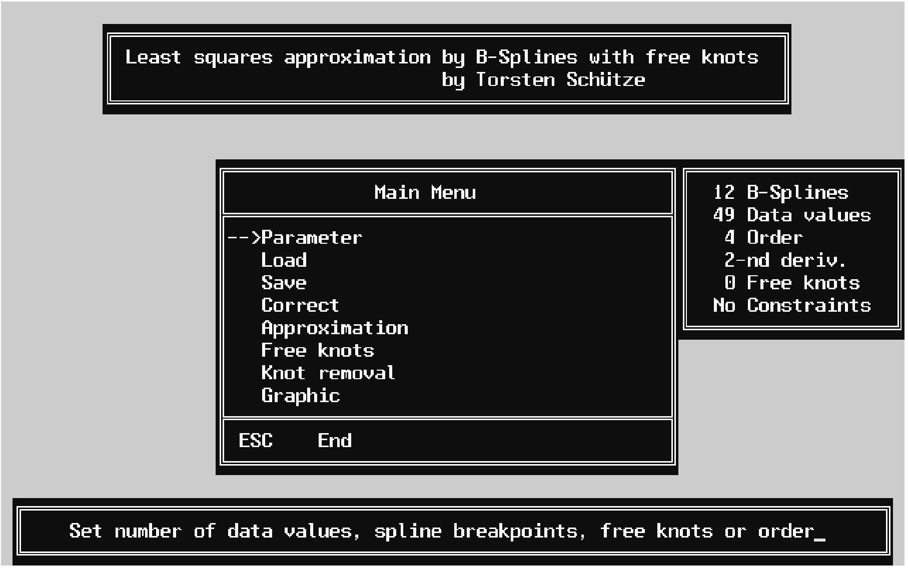

Figure 1. Main menu of program **FREE**

2. Hard- and software requirements

The program **FREE** was compiled under MS-DOS with Borland Pascal 7.0 using IEEE double precision arithmetic. It requires at least a 386 PC with arithmetic coprocessor and standard graphics card to run the program. To recompile the program **FREE**, Borland Pascal 6.0 or higher and the accompanying source files are required. No further source files or libraries are necessary.

3. The structure of the program

When the program **FREE** is started, the main menu appears, see Figure 1. In the center of the screen the menu items are shown. The bottom line contains a short, descriptive help text on the current menu point. The box on the right hand side shows important parameters of the current approximation problem. These parameters are in detail

n	number of B-splines
m	number of data values
k	order of B-splines
r	order of derivative in smoothing term
l	number of free knots
	type of shape constraints

The user can select specific menu items by moving the arrow either using the cursor keys $\uparrow$ (up) and $\downarrow$ (down), or using the keys CTRL E (up) and CTRL X (down) and finally confirming with ENTER. Menus can be escaped from with ESC. In the rest of this section we will shortly describe each menu item of Figure 1.

3.1. Parameter.

Under the menu point Parameter you can enter the following parameters which describe the approximation problem to be solved

- Number of data points m
- Dimension of spline space n
- Order of B-splines k (default: $k = 4$)
- Number of free knots l (default: $l = 0$)
- Order of derivative in smoothing term r (default: $r = 2$)
- Smoothing parameter $\mu \geq 0$ (default: $\mu = 0$)

Note that for some variables as m, n, and k there are upper bounds on their values. These upper bounds are predefined constants in the source code of **FREE**. You cannot exceed these values at run-time, e.g., if you want to approximate a data set with m larger than the actual constant $MMAX$ you have to recompile the program (definition of constants in `def.pas`). The author of the program is aware of this hard restriction. A portable C version of this code with full dynamic memory allocation is in preparation but has not been finished yet. It should be mentioned that you cannot increase the constants in `def.pas` arbitrary due to MS-DOS memory limits (64K barrier).

3.2. **Load.** After specifying the dimension of the problem you can Load the actual data values, spline parameters and constraints. There are three ways to input the data values (x_i, y_i) $(i = 1, \dots, m)$ and spline parameters t_j $(j = 1, \dots, n+k)$, α_j $(j = 1, \dots, n)$ under Read data values and Read spline parameters, respectively.

- Disk: Read file from disk
- Keyboard: Input data values or spline parameters via keyboard
- Test: Generate data or knot sequence using a test function

If you have input the data values or spline parameters once via keyboard you can later process these values by loading them from disk. The correct file structure will be guaranteed by saving the data or spline parameters to disk using **FREE** (see Save) or by generating data values externally. The source code of the accompanying program `gen-dat.pas` may be used as a template for this purpose (see also the procedures `Data_File_Read` and `Spline_File_Read` in `inout.pas`).

Under the menu item Test function of Read data values you can choose between two common test data sets

- Titanium Heat Data Set: $m = 49$, see [dB78]
- Fritsch-Carlson RPN 14 Radiochemical Data: $m = 9$, see [FC80]

For every data set the user can select among three "standard" knot sequences under Read spline parameter/Test knot sequence

- equidistant with respect to abscissae:
$$\begin{array}{ll} t_j = x_1 & j = 1, \dots, k \\ t_j = x_1 + (j-k) * (x_m - x_1)/(n-k+1) & j = k+1, \dots, n \\ t_j = x_m & j = n+1, \dots, n+k \end{array}$$
- equidistant with respect to chord length: Choose t_j so that the scaled chord length d is equidistantly subdivided, see [CHJ87].
$$d := \sum_{i=2}^{m} \sqrt{\left(\frac{x_i - x_{i-1}}{x_m}\right)^2 + \left(\frac{y_i - y_{i-1}}{\max_{1 \le j \le m} y_j}\right)^2}$$
The scaling of the chord length causes an approximately equal influence of changes in x and y.
- per data point:
$$\begin{array}{ll} t_j = x_1 & j = 1, \dots, k \\ t_j = y_{j-k+1} & j = k+1, \dots, n \\ t_j = x_m & j = n+1, \dots, n+k \end{array}$$

The remaining menu items of Load are concerned with the specification of the free knots and the shape constraints. With Free knots you can either set all inner knots free or set the entries of the index vector $\mathbf{p} = (p(1), \dots, p(l))^T$ one at a time. Moreover, you can specify the value of the knot separation parameter ϵ. Under Set constraints you can impose constraints on derivatives of the spline. These constraints are in detail

- Positivity ("Po"): $s(x) \ge 0$, $\forall x \in [a, b]$
- Monotonicity ("Mo"): $s'(x) \ge 0$, $\forall x \in [a, b]$
- Convexity ("Co"): $s''(x) \ge 0$, $\forall x \in [a, b]$
- Interval-wise bounds for spline ("0"): $l_i^{(0)} \le s(x) \le u_i^{(0)}$, $\forall x \in [t_i, t_{i+1})$, $i = k, \dots, n$

- Interval-wise bounds for first derivative ("1"): $l_i^{(1)} \leq s'(x) \leq u_i^{(1)}$, $\forall x \in [t_i, t_{i+1})$, $i = k, \ldots, n$
- Interval-wise bounds for second derivative ("2"): $l_i^{(2)} \leq s''(x) \leq u_i^{(2)}$, $\forall x \in [t_i, t_{i+1})$, $i = k, \ldots, n$

The characters "Po", "Mo" etc. denote the type of constraints as they appear in the parameter box on the right hand side of the screen. With Check consistency of bounds you can verify the strict consistency relation $\mathbf{L} < \mathbf{U}$ necessary for constrained spline approximation with free knots. Clear bounds for derivatives restores the default, i.e., no constraints on derivatives (No).

3.3. **Save.** Under this menu item you have the following possibilities to inspect the numerical results and to save your work:

- Save data values: (x_i, y_i) $(i = 1, \ldots, m)$
- Save spline parameter: knots t_j $(j = 1, \ldots, n+k)$
  coefficients α_j $(j = 1, \ldots, n)$
- Save statistic: residual vector $y_i - s(x_i)$ $(i = 1, \ldots, m)$
  norm of residual $\|\mathbf{r}\| = \|\mathbf{y} - \mathbf{B}\boldsymbol{\alpha}\|$
  index of free knots $p(1), \ldots, p(l)$
  knot separation parameter ϵ
- Save smoothing statistic: smoothing parameter μ
  norm of smoothing residual $\left\| \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} - \begin{bmatrix} \mathbf{B} \\ \sqrt{\mu}\mathbf{S} \end{bmatrix} \boldsymbol{\alpha} \right\|$

While you can view the output of the last two items only on screen, there is a variety of options to save the data values and spline parameters. These are

- Output to screen
- Output to printer
- Output to disk
- Output in Matlab format

The last item can be used to obtain an ASCII-File with correct Matlab syntax for later usage (the pictures of [SS95b] and [SS95a] were produced with Matlab).

3.4. **Correct.** For minor corrections of data values and spline parameters you can use the menu item Correct. Under Correct data values you can change (x_i, y_i) for an individual index i. Analogously the spline knots t_j and the coefficients α_j can be changed under Correct spline parameter.

The remaining entries Condense spline knots and Remove spline knots under Correct are intended for the manipulation of the knot sequence $\mathbf{t}$. Under Condense spline knots you can automatically reduce the number of knots by removing knots where the knot constraints $t_i - t_{i-1} \geq 0$ are almost active in the sense of $t_i - t_{i-1} \leq 2\epsilon$ or the relative error of the knots is to small, i.e., $|t_i - t_{i-1}| < 0.002\,|t_i|$. Under Remove spline knots you can manually either

- remove a simple knot,
- remove a multiple knot, or
- do nothing.

3.5. **Approximation.** The next two menu items Approximation and Free knots contain the essential parts of the program **FREE**. While you can compute approximating and smoothing splines for a *fixed* knot sequence under Approximation, under Free knots parts of the knot sequence are considered *variable* and included into the optimization process.

Although the approximation problem ($\mu = 0$) can be regarded as special case of the smoothing problem, special algorithms have been implemented in **FREE** to exploit the structure as far as possible. Specifically the following algorithms are available

- Approximation: compute an approximating least squares spline with prescribed, fixed knots and without shape constraints

$$\frac{1}{2}\left\|\mathbf{y} - \mathbf{B}(\tilde{\mathbf{t}})\boldsymbol{\alpha}\right\|^2 \to \min_{\boldsymbol{\alpha} \in \mathbb{R}^n}$$

TABLE 1. Return codes and termination criteria

1	$\left\|\mathbf{F}^{\nu+1}\right\| \leq \varepsilon_0^t$
2	$\left\|\mathbf{J}^{\nu T}\mathbf{F}^{\nu}\right\| \leq \varepsilon_1^t$
3	$\left\|\mathbf{F}^{\nu T}\mathbf{J}^{\nu}\mathbf{s}^{\nu}\right\| \leq \varepsilon_2^t$
4	$\left\|\mathbf{x}^{\nu+1}-\mathbf{x}^{\nu}\right\| \leq \varepsilon_3^t\,(\|\mathbf{x}^{\nu}\|+\varepsilon_4^t)$
5	$\left\|\|\mathbf{F}^{\nu+1}\|-\|\mathbf{F}^{\nu}\|\right\| \leq \varepsilon_5^t\,\|\mathbf{F}^{\nu}\|$
6	$\nu > \nu max$
7	failed otherwise

- Smoothing: compute a smoothing spline with prescribed, fixed knots and without shape constraints

$$\frac{1}{2}\left\|\begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix}-\begin{bmatrix}\mathbf{B}(\tilde{\mathbf{t}})\\ \sqrt{\mu}\mathbf{S}_r(\tilde{\mathbf{t}})\end{bmatrix}\boldsymbol{\alpha}\right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n}$$

- Constrained Approximation: compute an approximating least squares spline with prescribed, fixed knots subject to shape constraints

$$\frac{1}{2}\left\|\mathbf{y}-\mathbf{B}(\tilde{\mathbf{t}})\boldsymbol{\alpha}\right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \text{ subject to } \mathbf{L}\leq\mathbf{D}_p(\tilde{\mathbf{t}})\boldsymbol{\alpha}\leq\mathbf{U}$$

- Constrained Smoothing: compute a smoothing spline with prescribed, fixed knots subject to shape constraints

$$\frac{1}{2}\left\|\begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix}-\begin{bmatrix}\mathbf{B}(\tilde{\mathbf{t}})\\ \sqrt{\mu}\mathbf{S}_r(\tilde{\mathbf{t}})\end{bmatrix}\boldsymbol{\alpha}\right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \text{ subject to } \mathbf{L}\leq\mathbf{D}_p(\tilde{\mathbf{t}})\boldsymbol{\alpha}\leq\mathbf{U}$$

3.6. **Free knots.** All algorithms that are concerned with *variable* knots can be found under Free knots. The original problem, where both the variables $\boldsymbol{\alpha}$ and $\tilde{\mathbf{t}}$ appear, is reduced to a problem that has only the knots of the spline as variables. The reduced problem is a nonlinear least squares problem with linear inequality constraints, i.e.,

$$\frac{1}{2}\left\|\mathbf{F}(\tilde{\mathbf{t}})\right\|^2 \to \min_{\tilde{\mathbf{t}}\in\mathbb{R}^l} \quad \text{subject to} \quad \mathbf{C}\tilde{\mathbf{t}}-\mathbf{h}\geq\mathbf{0}$$

with appropriate residual function $\mathbf{F}$. The reduced problem will be solved with a generalized Gauss-Newton method. The method terminates if at least one of the conditions in Table 1 is satisfied.

Some of the termination parameters for the generalized Gauss-Newton method can be changed under the menu item Set parameter

Bound for defect of $\left\|\mathbf{F}^{\nu+1}\right\|$:	ε_0^t (default: $1.0E-10$)
Bound for relative change of $\|\mathbf{x}^{\nu}\|$:	ε_3^t (default: $1.0E-06$)
Bound for relative change of $\|\mathbf{F}^{\nu}\|$:	ε_5^t (default: $1.0E-10$)
Bound for norm of gradient $\left\|\mathbf{J}^{\nu T}\mathbf{F}^{\nu}\right\|$:	ε_1^t (default: $1.0E-10$)
Bound for $\left\|\mathbf{J}^{\nu T}\mathbf{F}^{\nu}\mathbf{s}^{\nu}\right\|$:	ε_2^t (default: $1.0E-10$)
Maximal number of Gauss-Newton steps:	νmax (default: 100)

It should be mentioned that for testing purposes the default values of the constants in Table 1 are rather stringent. For real applications the tests should be weakened, e.g., $\varepsilon_3^t = 10^{-2}\ldots10^{-3}$.

The menu for the selection of the computational method for splines with free knots has the same structure as in Section 3.5 for splines with fixed knots.

- Approximation: compute an approximating least squares spline spline with free knots and without shape constraints

  RAP – reduced approximation problem

$$\frac{1}{2}\left\|\mathbf{y}-\mathbf{B}(\tilde{\mathbf{t}})\boldsymbol{\alpha}(\tilde{\mathbf{t}})\right\|^2 \to \min_{\tilde{\mathbf{t}}\in\mathbb{R}^l} \text{ subject to } \mathbf{C}\tilde{\mathbf{t}}-\mathbf{h}\geq\mathbf{0}$$

  where $\boldsymbol{\alpha}(\tilde{\mathbf{t}})$ solves

$$\frac{1}{2}\left\|\mathbf{y}-\mathbf{B}(\tilde{\mathbf{t}})\boldsymbol{\alpha}\right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n}$$

- Smoothing: compute a smoothing spline with free knots and without shape constraints

  RSP – reduced smoothing problem

$$\frac{1}{2}\left\|\begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix}-\begin{bmatrix}\mathbf{B}(\tilde{\mathbf{t}})\\ \sqrt{\mu}\mathbf{S}_r(\tilde{\mathbf{t}})\end{bmatrix}\boldsymbol{\alpha}(\tilde{\mathbf{t}})\right\|^2 \to \min_{\tilde{\mathbf{t}}\in\mathbb{R}^l} \text{ subject to } \mathbf{C}\tilde{\mathbf{t}}-\mathbf{h}\geq\mathbf{0}$$

  where $\boldsymbol{\alpha}(\tilde{\mathbf{t}})$ solves

$$\frac{1}{2}\left\|\begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix}-\begin{bmatrix}\mathbf{B}(\tilde{\mathbf{t}})\\ \sqrt{\mu}\mathbf{S}_r(\tilde{\mathbf{t}})\end{bmatrix}\boldsymbol{\alpha}\right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n}$$

- Constrained Approximation: compute an approximating least squares spline with free knots subject to shape constraints

  RCAP – reduced constrained approximation problem

$$\frac{1}{2}\left\|\mathbf{y}-\mathbf{B}(\tilde{\mathbf{t}})\boldsymbol{\alpha}(\tilde{\mathbf{t}})\right\|^2 \to \min_{\tilde{\mathbf{t}}\in\mathbb{R}^l} \text{ subject to } \mathbf{C}\tilde{\mathbf{t}}-\mathbf{h}\geq\mathbf{0}$$

  where $\boldsymbol{\alpha}(\tilde{\mathbf{t}})$ solves

$$\frac{1}{2}\left\|\mathbf{y}-\mathbf{B}(\tilde{\mathbf{t}})\boldsymbol{\alpha}\right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \text{ subject to } \mathbf{L}\leq\mathbf{D}_p(\tilde{\mathbf{t}})\boldsymbol{\alpha}\leq\mathbf{U}$$

- Constrained Smoothing: compute a smoothing spline with free knots subject to shape constraints

  RCSP – reduced constrained smoothing problem

$$\frac{1}{2}\left\|\begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix}-\begin{bmatrix}\mathbf{B}(\tilde{\mathbf{t}})\\ \sqrt{\mu}\mathbf{S}_r(\tilde{\mathbf{t}})\end{bmatrix}\boldsymbol{\alpha}(\tilde{\mathbf{t}})\right\|^2 \to \min_{\tilde{\mathbf{t}}\in\mathbb{R}^l} \text{ subject to } \mathbf{C}\tilde{\mathbf{t}}-\mathbf{h}\geq\mathbf{0}$$

  where $\boldsymbol{\alpha}(\tilde{\mathbf{t}})$ solves

$$\frac{1}{2}\left\|\begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix}-\begin{bmatrix}\mathbf{B}(\tilde{\mathbf{t}})\\ \sqrt{\mu}\mathbf{S}_r(\tilde{\mathbf{t}})\end{bmatrix}\boldsymbol{\alpha}\right\|^2 \to \min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \text{ subject to } \mathbf{L}\leq\mathbf{D}_p(\tilde{\mathbf{t}})\boldsymbol{\alpha}\leq\mathbf{U}$$

For each of the above four problems either the relative knot condition (3) or the absolute knot condition (4) can be chosen.

As explained in [SS95b] and [SS95a], the structure of the Jacobian $\mathbf{J}=\mathbf{F}'$ of the residual function $\mathbf{F}$ is rather complicated. Therefore, besides a finite difference approximation of the exact Jacobian (*Golub/Pereyra model*, **GP**) an approximate Jacobian (*Kaufman model*, **Ka**) has been implemented. For both the Golub/Pereyra model and the Kaufman model, derivatives of the B-spline matrix $\mathbf{B}$ and the smoothing matrix $\mathbf{S}_r$ with respect to the free knots are needed. Depending on the type of computation we introduce the following notations

- outer discretization (**OD**): compute $\mathbf{J}\mathbf{e}^\tau \approx (\mathbf{F}(\tilde{\mathbf{t}}+h_\tau\mathbf{e}^\tau)-\mathbf{F}(\tilde{\mathbf{t}}))/h_\tau$
- inner discretization (**ID**): compute $\mathbf{J}\mathbf{e}^\tau$ using $\mathbf{B}'\mathbf{e}^\tau \approx (B_{j,k}(x,\tilde{\mathbf{t}}+h_\tau\mathbf{e}^\tau)-B_{j,k}(x,\tilde{\mathbf{t}}))/h_\tau$ (analogously for $\mathbf{S}_r'\mathbf{e}^\tau$)
- exact derivatives (**ED**): compute $\mathbf{J}\mathbf{e}^\tau$ using exact derivatives

TABLE 2. Diagnostic output during the Gauss-Newton process

Item		Description
Iteration	ν	number of iteration
residual norm	$\lVert \mathbf{F}^{\nu+1} \rVert$	residual sum after ν-th step of iteration
condition number	$\mathrm{cond}(\mathbf{J}^\nu)$	estimated condition number of Jacobian
projected gradient	$\lvert \mathbf{F}^{\nu T} \mathbf{J}^\nu \mathbf{s}^\nu \rvert$	projected gradient at $\tilde{\mathbf{t}}^\nu$
rel. error	$\lVert \tilde{\mathbf{t}}^{\nu+1} - \tilde{\mathbf{t}}^\nu \rVert \ / \ \lVert \tilde{\mathbf{t}}^\nu \rVert$	approximates relative error of $\tilde{\mathbf{t}}^\nu$
rel. error in $\mathbf{F}$	$\lVert \mathbf{F}^{\nu+1} - \mathbf{F}^\nu \rVert \ / \ \lVert \mathbf{F}^\nu \rVert$	approximates relative error of $\lVert \mathbf{F}^\nu \rVert$
backtracking		
steps		number of backtracking steps in line search
step length	γ	step length in line search
mode		regularized Gauss-Newton step (y/n)

By choosing the type of the approximation of the Jacobian the user selects simultaneously the discretization used. In the current version of the program the Golub/Pereyra model can only be computed using outer discretization, the Kaufman model can be computed using inner discretization or exact derivatives. [1]

If the algorithm stops, the values of the quantities in the termination tests at the last iterate are displayed on the screen. Additionally the computing time is shown. For diagnostic purposes a file `diag.txt` with detailed information about the convergence behaviour is generated during the Gauss-Newton process, see [SS95b] for a description of the generalized Gauss-Newton method used. Specifically, the quantities described in Table 2 are listed.

3.7. **Knot removal.** The procedures for the computation of splines with free knots can be combined with an automatic knot removal strategy, see [SS95b]. We want to approximate the data with as few B-splines as possible so that the least squares error $\lVert \mathbf{F}(\tilde{\mathbf{t}}) \rVert$ of the corresponding spline to knot sequence $\tilde{\mathbf{t}}$ is not greater than a prescribed bound Δ, i.e., $\lVert \mathbf{F}(\tilde{\mathbf{t}}) \rVert \leq \Delta$. We call such a knot sequence acceptable.

We start the process with an acceptable knot sequence consisting of a large number of knots. In one step of the iterative process, at first the current knot sequence is optimized using our algorithm. Then the interior knots are ranked according to their importance for the approximation which is measured by the jump in the first discontinuous derivative. If removing the less important knot leads to a knot sequence which is not acceptable, then the iteration is terminated, and the old (last acceptable) knot sequence is taken as final knot sequence. Otherwise, the iteration is repeated with the new, shortened knot sequence where the less important knot is dropped.

We have implemented the following algorithms for automatic knot removal

- Phase 1: In the first stage, we start with a fixed, acceptable knot sequence. Then we remove knots as described above but without optimizing the new knot sequence. This cheap process is repeated as long as the knot sequence remains acceptable. When the acceptability gets lost, we start the second stage.

  The user is asked for an upper bound Δ and whether he wants to use approximating or smoothing splines.
- Phase 2: In the second stage, we try to remove further knots by relocating the knots in every step. Either the Kaufman model or the Golub/Pereyra model can be chosen for the

[1] You cannot switch between **ID** and **ED** during run-time, i.e., to use **ID** instead of the default **ED** you have to change the source code.

nonlinear problems. The user can switch between an absolute or a relative knot separation parameter, c.f. (3) and (4).

- Remove one knot: Under the third menu item of Knot removal, one can remove the knot with the smallest jump in the $(k-1)$-th derivative.

Note that the automatic knot removal procedures are intended for splines without shape constraints.

3.8. **Graphic.** Under the menu item Graphic procedures for the graphical presentation of splines and their derivatives can be found. The following topics are available

- 2D-Plot: two-dimensional presentation of the r-th order derivative of the spline in the interval $x_{\min}, x_{\max}$ $(0 \leq r \leq k)$
- 2D-Plot with TeX Output: same as under the above menu item, but with additional output to files `1.tex`, `2.tex`, ... with TEX commands for drawing the picture
- Hardcopy: activates DOS-interrupt for a hardcopy of the screen
- 3D-Plot: three-dimensional plot of the residual of the spline

  Given two indices i_1, i_2, this procedure plots the residual of the spline as function of the two knots t_{i_1} and t_{i_2}. The residual will be computed on a $N \times N$ grid of the set $\{(t_{i_1}, t_{i_2}) \in [t_{i_1-1}, t_{i_1+1}] \times [t_{i_2-1}, t_{i_2+1}]\}$ and visualized. The values $\|\mathbf{F}(t_{i_1}, t_{i_2})\|$ can be saved to an ASCII-file for later processing, e.g., by GNUPLOT.

4. The structure of the source files

The source code of **FREE** consists of the main program `free.pas` and several Pascal units. For your convenience we have included `gen-dat.pas`, a template for a program to generate test data that can be read by **FREE**. The source code is well-documented with a concise description of the in- and output parameters of a routine. The whole program consists of approximately 13,000 lines of source code.

Part of the distribution are the binary files containing the data for the examples in [Sch93], [SS95b], and [SS95a], see Table 4.

References

[Bjö90] Å. Björck, *Least squares methods*, Handbook of Numerical Analysis, volume I, Solution of equations in R^n (P. G. Ciarlet and J. L. Lions, eds.), North Holland, 1990, pp. 589–652.

[CHJ87] M. G. Cox, P. M. Harris, and H. M. Jones, *Strategies for knot placement in least squares data fitting by splines*, NPL Report DITC 101/87, National Physical Laboratory, Teddington, September 1987.

[dB78] C. de Boor, *A practical guide to splines*, Springer-Verlag, New York, Heidelberg, Berlin, 1978.

[Die87] P. Dierckx, *FITPACK user guide, part 1: Curve fitting routines*, Tech. Report TW Report 89, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, 1987.

[Die93] P. Dierckx, *Curve and surface fitting with splines*, Oxford University Press, 1993.

[FC80] F. N. Fritsch and R. E. Carlson, *Monotone piecewise cubic interpolation*, SIAM J. Numer. Anal. **17** (1980), no. 2, 238–246.

[Hu93] Y. Hu, *An algorithm for data reduction using splines with free knots*, IMA J. Numer. Anal. **13** (1993), no. 3, 365–381.

[LH95] C. L. Lawson and R. J. Hanson, *Solving least squares problems*, SIAM Publications, Philadelphia, 1995, Reprint of edition by Prentice Hall, 1974.

[Sch93] T. Schütze, *Quadratmittelapproximation durch B-Splines mit freien Knoten*, Diplomarbeit, Martin-Luther-Universität Halle/Wittenberg, Fachbereich Mathematik und Informatik, 1993.

[SK93] H. Schwetlick and V. Kunert, *Spline smoothing under constraints on derivatives*, BIT **33** (1993), 512–528.

[SS95a] T. Schütze and H. Schwetlick, *Constrained approximation by splines with free knots*, Preprint MATH-NM-18-1995, Technical University of Dresden, 1995.

[SS95b] H. Schwetlick and T. Schütze, *Least squares approximation by splines with free knots*, BIT **35** (1995), no. 3, 361–384.

(T. Schütze) Department of Mathematics, Technical University of Dresden, D-01062 Dresden, Germany, http://www.math.tu-dresden.de/~schuetze

E-mail address: `schuetze@math.tu-dresden.de`

TABLE 3. Structure of the source files

	MAIN PROGRAM
`free.pas`	Least Squares Approximation by B-splines with free knots
	UNITS
`blas.pas`	Basic Linear Algebra Subroutines
`blsunit.pas`	procedure for the solution of linear least squares problems with simple bounds (BLS), algorithm see [Bjö90]
`bspline.pas`	procedures for the computation of polynomial B-Splines and l_2-approximations
`csspline.pas`	procedures for the computation of constrained smoothing B-Splines
`d3.pas`	routines for three-dimensional plots, by B. Goldschmidt, University of Halle
`def.pas`	definitions of constants, types and global variables
`error.pas`	implements a simple error handler
`gauss.pas`	generalized Gauss-Newton methods for nonlinear least squares problems
`givens.pas`	low level procedures for GIVENS plane rotations
`graphic.pas`	graphical presentation of splines and their derivatives
`inout.pas`	procedures for in- and output
`jacobi.pas`	provide the Jacobian for Gauss Newton methods
`knotrem.pas`	automatic knot removal procedures
`lawson.pas`	procedures for the solution of constrained linear least squares problems, algorithms see [LH95]
`linpack.pas`	procedures from LINPACK for the QR-factorization of a matrix and the solution of linear least squares problems
`math.pas`	low level mathematical subroutines
`smooth.pas`	procedures for the computation of smoothing polynomial B-Splines
`various.pas`	various procedures for menu handling
	AUXILIARY PROGRAM
`gen-dat.pas`	template for a program to generate test data that can be read by **FREE**

TABLE 4. Data files

File	Name	$m =$	Reference
`arctan41.dta`	Arctan Data	41	[SS95a]
`hu90.dta`	Hu Data	90	[SS95b], [Hu93]
`spec90.dta`	Spectral Data	90	[Sch93]
`titan49.dta`	Titanium Heat Data	41	[SS95b], [SS95a]
`vol16.dta`	Volumetric Moisture Content Data	16	[SS95a], [Die93], [Die87]