

# Interpolation of One- and Two-Dimensional Images with Pixelwise Photon Number Conservation

Takashi Sakurai<sup>1,2</sup> and Junho Shin<sup>1</sup>

<sup>1</sup>*National Astronomical Observatory, Mitaka, Tokyo 181-8588  
sakurai@solar.mtk.nao.ac.jp, jshin@solar.mtk.nao.ac.jp*

<sup>2</sup>*National Solar Observatory, Tucson, AZ 85719-6732, U.S.A.*

(Received 2000 December 20; accepted 2001 February 15)

## Abstract

The interpolation of one- and two-dimensional images is considered. The usual linear or cubic spline interpolations try to connect given data points. On the other hand, image data give area integrals of the radiation intensity (i.e., photon counts) over pixels. Based on variational principles, we have developed schemes of quadratic and fourth-order polynomial interpolation which reproduce the given pixelwise photon counts exactly. These methods were tested on some simple examples and showed satisfactory results.

**Key words:** interpolation — methods: analytical — methods: numerical — splines — techniques: image processing

## 1. Introduction

When data are given on discrete points and data values are needed between the sampling points, one has to interpolate the data (for a recent review of this topic, see, e.g., Thévenaz et al. 2000). The most common ways are piecewise-linear interpolation and cubic spline interpolation. In both methods, the interpolated lines, or curves, go through the sampling points. If the data to be interpolated are like temperature, pressure, or velocity measured at the sampling points, these interpolation methods are quite adequate. In astronomy or image processing, one often encounters a situation in which photon counts or radiation energy integrated over pixels are given. In such a case, one may assign the measured values to the centers of the pixels and apply ordinary interpolation formulae. However, when one integrates the interpolated values over the pixels, one does not exactly reproduce the measured (pixelwise) photon counts.

Spline interpolations are used in a wide variety of fields in astronomy (Adorf 1995; Akerlof et al. 1994; Platais 1991), but the usual cubic splines do not reproduce pixelwise photon counts exactly. Interpolation schemes that reproduce the photon counts exactly are less popular but in the literature (e.g., de Boor 1978; Späth 1995a, b), and the routines of de Boor (1978) are included in the mathematical library of IRAF (Tody 1986). However, compared to the usual cubic splines, interpolations conserving photon numbers are not widely used in astronomy. One reason for this could be that the interpolation schemes whose accuracy is equivalent to, or better than, the cubic splines are rather complicated and not easy to use in practical applications. In the present paper, we revisit this problem and propose a scheme of formulation of this problem based on the variational principle. We consider one- and two-dimensional data; for both cases the interpolating solutions can be obtained easily with a recurrence formula. Tests have been performed on several simple examples, and interpolation errors are evaluated.

The topic studied in this paper is interpolation, not image improvement, although these two are closely related. Image improvement or restoration procedures generally seek a solution which is as smooth as possible and reproduces the observed photon counts within their statistical uncertainties (Narayan, Nityananda 1986; Piña, Puetter 1993). Our procedures are constrained to exactly reproduce the observed photon counts, and therefore may not give an optimum solution as far as the smoothness is concerned. Which approach is more appropriate may depend on the actual cases under study. We believe that there are cases where conservation of the photon numbers is more important than an overall smoothness in the solution; the procedures developed in this paper are aimed at such applications.

## 2. Variational Formulation of Interpolation: 1-D Cases

First, we consider one-dimensional cases. Let the data points be  $x_i$  ( $i = 0, 1, \dots, n$ ). The corresponding data values are  $\phi_i$ . In order to find the interpolating function,  $\phi(x)$  ( $x_0 \leq x \leq x_n$ ), we introduce the following minimization principle:

$$\frac{1}{2} \int_{x_0}^{x_n} \phi''(x)^2 dx = \min. \quad (1)$$

Here the prime ( $'$ ) denotes  $d/dx$ . The function to be minimized can be regarded as the energy associated with some elastic material that reacts against bending, because  $\phi''(x)$  is the curvature of  $\phi(x)$ . The Euler–Lagrange equation for this variational problem is

$$\phi'''(x) = 0, \tag{2}$$

namely the interpolating function,  $\phi(x)$ , is a cubic function. This gives a variational interpretation for the cubic spline interpolation (e.g., Hoschek, Lasser 1993, p.77). The cubic splines are the curves produced by an elastic material, like a steel ruler, when it is constrained to go through the specified node points,  $\phi_i$ .

On the other hand, the variational problem,

$$\frac{1}{2} \int_{x_0}^{x_n} \phi'(x)^2 dx = \min, \tag{3}$$

leads to the Euler–Lagrange equation,

$$\phi''(x) = 0. \tag{4}$$

Therefore, we obtain a (piecewise) linear function. If the quantity to be minimized is  $\sqrt{1 + \phi'^2}$  instead of  $(1/2)\phi'^2$ , the integral gives the length of the curve connecting the points  $\phi_i$ . Here, the piecewise linear segments are obviously the minimizing solution. As long as  $\phi'$  is small,  $\sqrt{1 + \phi'^2}$  can be approximated as  $1 + (1/2)\phi'^2$ , and we may apply the same interpretation to the minimizing solution. Even for an arbitrary magnitude of  $\phi'$ , it is a useful approximation to regard the minimizing solution as is realized by a rubber band, which tries to shrink along its length with its tension. [The same factor,  $\sqrt{1 + \phi'^2}$ , should appear in equation (1) if it exactly represents the elastic energy. We drop this factor because, otherwise, the resulting equation becomes nonlinear.]

2.1. Quadratic Interpolation with a Fixed Area

We now proceed to a problem in which not the node values  $\phi_i$ , but the integrated values of  $\phi(x)$  over the intervals  $[x_i, x_{i+1}]$  ( $i = 0, 1, 2, \dots, n - 1$ ), namely,

$$N_i = \int_{x_i}^{x_{i+1}} \phi(x) dx \tag{5}$$

are specified. If  $\phi$  is the radiation intensity,  $N_i$  corresponds to the radiation energy, or photon counts, detected by the  $i$ -th pixel, which extends from  $x_i$  to  $x_{i+1}$ . In the geometrical or mechanical analog,  $N_i$  is the area under the curve  $\phi(x)$  in the interval  $[x_i, x_{i+1}]$ . A constraint like equation (5) can be taken into account by introducing a Lagrange multiplier.

First we consider the form of energy given by equation (3). Since the areas under the curve are constrained, we can anticipate that piecewise linear functions are no longer the solution, and the solution has to be curved. One may imagine a plastic foil pressed against an incompressible fluid.

The quantity to be minimized is, with the inclusion of Lagrange multipliers,  $\lambda_i$  ( $i = 0, 1, \dots, n - 1$ ),

$$\frac{1}{2} \int_{x_0}^{x_n} \phi'(x)^2 dx + \sum_{i=0}^{n-1} \lambda_i \int_{x_i}^{x_{i+1}} \phi(x) dx = \min. \tag{6}$$

The variation of this equation is

$$\sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} [\lambda_i - \phi''(x)] \delta\phi(x) dx + \sum_{i=1}^{n-1} [\phi'(x_i + 0) - \phi'(x_i - 0)] \delta\phi(x_i) + \phi'(x_0) \delta\phi(x_0) - \phi'(x_n) \delta\phi(x_n) = 0. \tag{7}$$

Since  $\delta\phi$  is arbitrary, we obtain the following Euler–Lagrange equation:

$$\phi''(x) - \lambda_i = 0 \quad (x_i \leq x \leq x_{i+1}). \tag{8}$$

Namely, the minimizing solution is a quadratic function. Because the node values,  $\phi(x_i)$ , are not constrained,  $\delta\phi(x_i)$  is arbitrary. Therefore, at the nodes the following conditions have to be satisfied:

$$\phi'(x_0) = 0, \tag{9}$$

$$\phi'(x_n) = 0, \tag{10}$$

$$\phi'(x) = \text{continuous at } x_i \quad (i = 1, 2, \dots, n - 1). \tag{11}$$

A quadratic function is characterized by three coefficients. For practical computations, it is useful to adopt the following form for the minimizing solution:

$$\phi(x) = \phi_i P(\xi) + \phi_{i+1} Q(\xi) + N_i R(\xi)/\Delta_i \quad (x_i \leq x \leq x_{i+1}, \quad i = 0, 1, 2, \dots, n - 1). \tag{12}$$

Here, the  $\phi_i$ 's are the node values to be determined later,  $\Delta_i = x_{i+1} - x_i$ , and  $\xi = (x - x_i)/\Delta_i$ . In each interval,  $[x_i, x_{i+1}]$ , the normalized variable,  $\xi$ , runs from 0 to 1. Three functions [ $P(\xi)$ ,  $Q(\xi)$ , and  $R(\xi)$ ] are quadratic functions of  $\xi$  and should satisfy the following conditions:

$$P(0) = 1, \quad P(1) = 0, \quad \int_0^1 P(\xi) d\xi = 0, \tag{13}$$

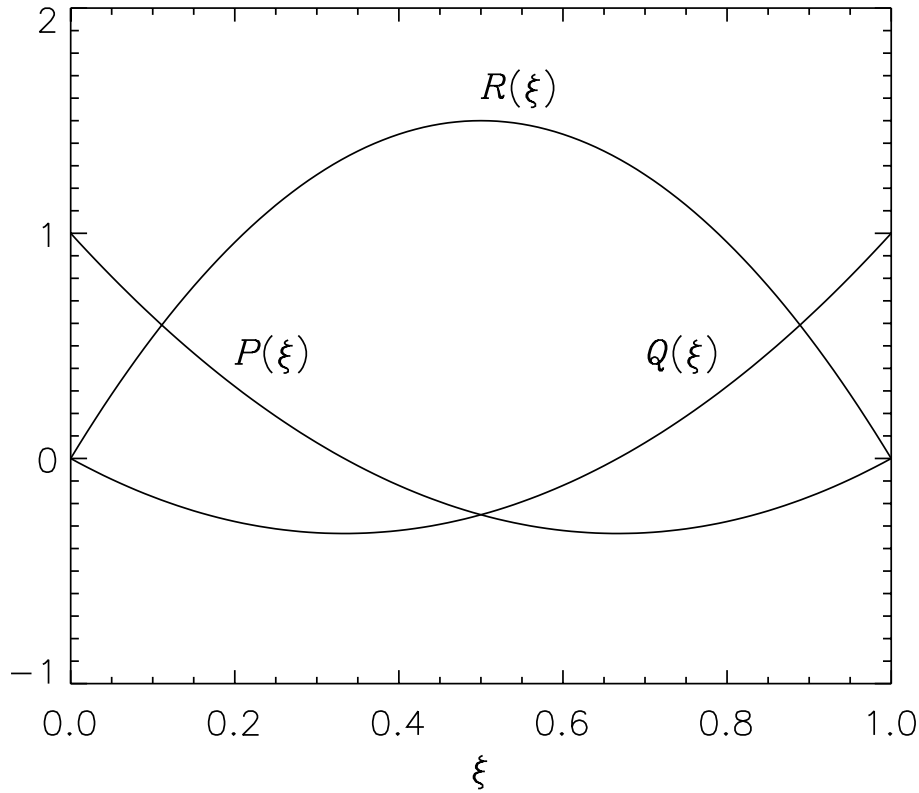


Fig. 1. Blending functions in the case of quadratic interpolation.

$$Q(0) = 0, \quad Q(1) = 1, \quad \int_0^1 Q(\xi) d\xi = 0, \tag{14}$$

$$R(0) = 0, \quad R(1) = 0, \quad \int_0^1 R(\xi) d\xi = 1. \tag{15}$$

The explicit forms for these three functions are easily obtained as

$$P(\xi) = (1 - \xi)(1 - 3\xi), \tag{16}$$

$$Q(\xi) = \xi(3\xi - 2) \equiv P(1 - \xi), \tag{17}$$

$$R(\xi) = 6\xi(1 - \xi), \tag{18}$$

and are shown graphically in figure 1. As in the case of cubic spline functions, we may call these functions blending functions, because they blend the contributions from three independent constraints ( $\phi_i$ ,  $\phi_{i+1}$ , and  $N_i$ ).

From the construction of these functions, the solution,  $\phi(x)$  [equation (12)], goes through the node points,  $\phi_i$ , and has specified areas,  $N_i$ , under the curve. The unknown node values,  $\phi_i$ , can be determined by applying conditions (9)–(11). The results are cast into a tri-diagonal matrix equation:

$$b_0\phi_0 + c_0\phi_1 = d_0, \tag{19}$$

$$a_i\phi_{i-1} + b_i\phi_i + c_i\phi_{i+1} = d_i \quad (i = 1, 2, \dots, n - 1), \tag{20}$$

$$a_n\phi_{n-1} + b_n\phi_n = d_n, \tag{21}$$

where

$$a_i = 1 \quad (i = 1, 2, \dots, n - 1), \tag{22}$$

$$a_n = 1, \tag{23}$$

$$b_0 = 2, \tag{24}$$

$$b_i = 2 \left( 1 + \frac{\Delta_{i-1}}{\Delta_i} \right) \quad (i = 1, 2, \dots, n - 1), \tag{25}$$

$$b_n = 2, \tag{26}$$

$$c_0 = 1, \tag{27}$$

$$c_i = \frac{\Delta_{i-1}}{\Delta_i} \quad (i = 1, 2, \dots, n - 1), \tag{28}$$

$$d_0 = 3 \frac{N_0}{\Delta_0}, \tag{29}$$

$$d_i = 3 \left( \frac{N_{i-1}}{\Delta_{i-1}} + 3 \frac{N_i}{\Delta_i} \frac{\Delta_{i-1}}{\Delta_i} \right) \quad (i = 1, 2, \dots, n - 1), \tag{30}$$

$$d_n = 3 \frac{N_{n-1}}{\Delta_{n-1}}. \tag{31}$$

Then, the following recurrence formulae solve the equation:

$$c_0 \leftarrow \frac{c_0}{b_0}, \quad d_0 \leftarrow \frac{d_0}{b_0}, \tag{32}$$

$$c_i \leftarrow \frac{c_i}{b_i - a_i c_{i-1}}, \quad d_i \leftarrow \frac{d_i - a_i d_{i-1}}{b_i - a_i c_{i-1}} \quad (i = 1, 2, \dots, n - 1), \tag{33}$$

$$\phi_n \leftarrow \frac{d_n - a_n d_{n-1}}{b_n - a_n c_{n-1}}, \tag{34}$$

$$\phi_i \leftarrow d_i - c_i d_{i+1} \quad (i = n - 1, \dots, 1, 0). \tag{35}$$

The solution has the specified area under the curve within each interval,  $[x_i, x_{i+1}]$ , and its value and the derivative are continuous at the nodes. The same solution was obtained by Späth (1995a) with a different approach. His method is more straightforward than ours in derivation, while our method shows more clearly why the interpolating function is a quadratic polynomial.

### 2.2. Fourth-Order Interpolation with a Fixed Area

In the previous subsection we considered a variational problem in which the energy density is given by  $(1/2)\phi'^2$ . In this subsection we discuss a case in which the energy density is  $(1/2)\phi''^2$ . When no constraints are imposed, the minimizing solution is a cubic function. In parallel to the treatment of the previous subsection, we impose the condition of fixed areas under the curve, and introduce the following integral to be minimized:

$$\sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} \left[ \frac{1}{2} \sigma_i \phi''(x)^2 + \lambda_i \phi(x) \right] dx = \min. \tag{36}$$

Here,  $\sigma_i$  represents the stiffness of a hypothetical elastic material, which may have different values in different intervals. The variation of this equation is

$$\begin{aligned} & \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} [\lambda_i + \sigma_i \phi''''(x)] \delta\phi(x) dx \\ & + \sum_{i=1}^{n-1} [\sigma_{i-1} \phi''(x_i + 0) - \sigma_i \phi''(x_i - 0)] \delta\phi'(x_i) + \sigma_0 \phi''(x_0) \delta\phi'(x_0) - \sigma_{n-1} \phi''(x_n) \delta\phi'(x_n) \\ & - \sum_{i=1}^{n-1} [\sigma_{i-1} \phi''''(x_i + 0) - \sigma_i \phi''''(x_i - 0)] \delta\phi(x_i) - \sigma_0 \phi''''(x_0) \delta\phi(x_0) + \sigma_{n-1} \phi''''(x_n) \delta\phi(x_n) = 0. \end{aligned} \tag{37}$$

Since  $\delta\phi$  is arbitrary, we obtain the following Euler–Lagrange equation:

$$\sigma_i \phi''''(x) + \lambda_i = 0 \quad (x_i \leq x \leq x_{i+1}). \tag{38}$$

Namely, the minimizing solution is a fourth-order (quartic) polynomial. At the nodes the following conditions have to be satisfied:

$$\phi''(x_0) = 0, \tag{39}$$

$$\phi'''(x_0) = 0, \tag{40}$$

$$\phi''(x_n) = 0, \tag{41}$$

$$\phi'''(x_n) = 0, \tag{42}$$

$$\sigma_{i-1} \phi''(x_i - 0) = \sigma_i \phi''(x_i + 0) \quad (i = 1, 2, \dots, n - 1), \tag{43}$$

$$\sigma_{i-1} \phi''''(x_i - 0) = \sigma_i \phi''''(x_i + 0) \quad (i = 1, 2, \dots, n - 1). \tag{44}$$

A fourth-order polynomial is characterized by five coefficients. For practical computations, it is useful to adopt the following form for the minimizing solution:

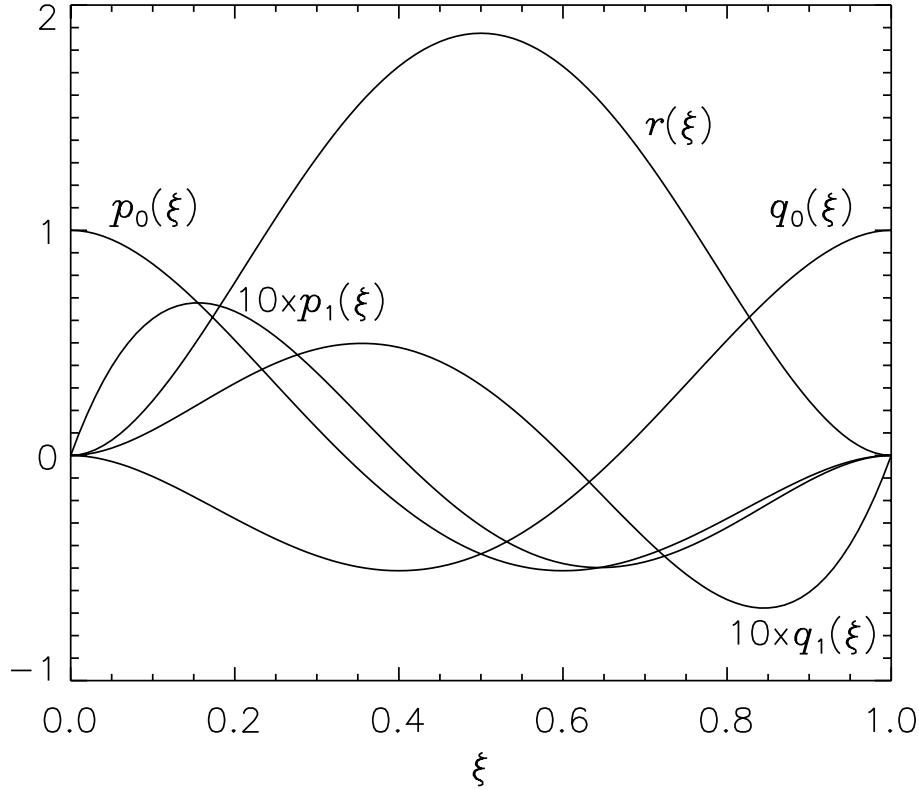


Fig. 2. Blending functions in the case of fourth-order polynomial interpolation.

$$\phi(x) = \phi_i p_0(\xi) + \phi_{i+1} q_0(\xi) + \Delta_i [\phi'_i p_1(\xi) + \phi'_{i+1} q_1(\xi)] + N_i r(\xi) / \Delta_i \quad (x_i \leq x \leq x_{i+1}, \quad i = 0, 1, 2, \dots, n - 1). \quad (45)$$

The definitions for  $\Delta_i$ ,  $\xi$ , and  $N_i$  are the same as before. Five functions [ $p_0(\xi)$ ,  $q_0(\xi)$ ,  $p_1(\xi)$ ,  $q_1(\xi)$ , and  $r(\xi)$ ] are fourth-order polynomials of  $\xi$ , and should satisfy the following conditions:

$$p_0(0) = 1, \quad p_0(1) = 0, \quad p'_0(0) = 0, \quad p'_0(1) = 0, \quad \int_0^1 p_0(\xi) d\xi = 0, \quad (46)$$

$$q_0(0) = 0, \quad q_0(1) = 1, \quad q'_0(0) = 0, \quad q'_0(1) = 0, \quad \int_0^1 q_0(\xi) d\xi = 0, \quad (47)$$

$$p_1(0) = 0, \quad p_1(1) = 0, \quad p'_1(0) = 1, \quad p'_1(1) = 0, \quad \int_0^1 p_1(\xi) d\xi = 0, \quad (48)$$

$$q_1(0) = 0, \quad q_1(1) = 0, \quad q'_1(0) = 0, \quad q'_1(1) = 1, \quad \int_0^1 q_1(\xi) d\xi = 0, \quad (49)$$

$$r(0) = 0, \quad r(1) = 0, \quad r'(0) = 0, \quad r'(1) = 0, \quad \int_0^1 r(\xi) d\xi = 1. \quad (50)$$

The explicit forms for these five blending functions are obtained as

$$p_0(\xi) = (1 - \xi)^2(1 + 5\xi)(1 - 3\xi), \quad (51)$$

$$q_0(\xi) = \xi^2(3\xi - 2)(6 - 5\xi) \equiv p_0(1 - \xi), \quad (52)$$

$$p_1(\xi) = \xi(1 - \xi)^2(1 - \frac{5}{2}\xi), \quad (53)$$

$$q_1(\xi) = -\frac{1}{2}\xi^2(1 - \xi)(5\xi - 3) \equiv -p_1(1 - \xi), \quad (54)$$

$$r(\xi) = 30\xi^2(1 - \xi)^2, \quad (55)$$

and are shown graphically in figure 2.

The unknown node values,  $\phi_i$  and  $\phi'_i$ , can be determined by applying conditions (39)–(44). The results are cast into a block tri-diagonal matrix equation with  $2 \times 2$  matrices ( $A_i$ ,  $B_i$ ,  $C_i$ ) as coefficients:

$$B_0\Phi_0 + C_0\Phi_1 = D_0, \tag{56}$$

$$A_i\Phi_{i-1} + B_i\Phi_i + C_i\Phi_{i+1} = D_i \quad (i = 1, 2, \dots, n-1), \tag{57}$$

$$A_n\Phi_{n-1} + B_n\Phi_n = D_n, \tag{58}$$

where

$$\Phi_i = \begin{pmatrix} \phi_i \\ \Delta_i\phi'_i \end{pmatrix}. \tag{59}$$

For  $i = n$ , we defined  $\Delta_n = \Delta_{n-1}$ . Further, we introduce the following quantities:

$$f_i = \frac{\Delta_i}{\Delta_{i+1}} \quad (i = 0, 1, \dots, n-1), \quad g_i = \frac{\sigma_{i+1}}{\sigma_i} \quad (i = 0, 1, \dots, n-2). \tag{60}$$

Then we obtain:

$$A_i = \begin{pmatrix} -8 & -1 \\ 14 & 2 \end{pmatrix} \quad (i = 1, 2, \dots, n-1), \tag{61}$$

$$A_n = \begin{pmatrix} -8 & -1 \\ 14 & 2 \end{pmatrix}, \tag{62}$$

$$B_0 = \begin{pmatrix} 12 & 3 \\ 16 & 3 \end{pmatrix}, \tag{63}$$

$$B_i = \begin{pmatrix} 12(g_{i-1}f_{i-1}^2 - 1) & 3f_{i-1} + 3g_{i-1}f_{i-1}^2 \\ 16(1 + g_{i-1}f_{i-1}^3) & -3f_{i-1} + 3g_{i-1}f_{i-1}^3 \end{pmatrix} \quad (i = 1, 2, \dots, n-1), \tag{64}$$

$$B_n = \begin{pmatrix} -12 & 3 \\ 16 & -3 \end{pmatrix}, \tag{65}$$

$$C_0 = \begin{pmatrix} 8 & -f_0 \\ 14 & -2f_0 \end{pmatrix}, \tag{66}$$

$$C_i = \begin{pmatrix} 8g_{i-1}f_{i-1}^2 & -g_{i-1}f_i f_{i-1}^2 \\ 14g_{i-1}f_{i-1}^3 & -2g_{i-1}f_i f_{i-1}^3 \end{pmatrix} \quad (i = 1, 2, \dots, n-1), \tag{67}$$

$$D_0 = \begin{pmatrix} 20N_0/\Delta_0 \\ 30N_0/\Delta_0 \end{pmatrix}, \tag{68}$$

$$D_i = \begin{pmatrix} -20N_{i-1}/\Delta_{i-1} + 20g_{i-1}f_{i-1}^2 N_i/\Delta_i \\ 30N_{i-1}/\Delta_{i-1} + 30g_{i-1}f_{i-1}^3 N_i/\Delta_i \end{pmatrix} \quad (i = 1, 2, \dots, n-1), \tag{69}$$

$$D_n = \begin{pmatrix} -20N_{n-1}/\Delta_{n-1} \\ 30N_{n-1}/\Delta_{n-1} \end{pmatrix}. \tag{70}$$

The following recurrence formulae give the solution:

$$C_0 \leftarrow B_0^{-1}C_0, \quad D_0 \leftarrow B_0^{-1}D_0, \tag{71}$$

$$C_i \leftarrow (B_i - A_i C_{i-1})^{-1}C_i \quad (i = 1, 2, \dots, n-1), \tag{72}$$

$$D_i \leftarrow (B_i - A_i C_{i-1})^{-1}(D_i - A_i D_{i-1}) \quad (i = 1, 2, \dots, n-1), \tag{73}$$

$$\Phi_n \leftarrow (B_n - A_n C_{n-1})^{-1}(D_n - A_n D_{n-1}), \tag{74}$$

$$\Phi_i \leftarrow D_i - C_i D_{i+1} \quad (i = n-1, \dots, 1, 0). \tag{75}$$

The solution has the specified area under the curve within each interval,  $[x_i, x_{i+1}]$ , and its value and the first derivative are continuous at the nodes. If the stiffness,  $\sigma_i$ , is uniform over all of the intervals, the solution is also continuous up to the third derivative at the nodes. If  $\sigma_i$  varies from one interval to another, the second and third derivatives show jumps at the nodes.

The reason why we did not consider the form  $(1/2)\sigma_i\phi'^2$  as a quantity to be minimized in the previous subsection is that the resulting minimizing solution does not have a continuous derivative at the nodes if  $\sigma_i$  is non-uniform. For the purpose of interpolation, we are only interested in solutions with continuous first derivative.

### 2.3. Combination of Tension and Elastic Energies

In the previous two subsections, we studied the energy densities of the forms  $(1/2)\phi'^2$  and  $(1/2)\phi''^2$ . The former corresponds to a rubber band with tension, and the latter corresponds to an elastic material like a steel ruler. A system which has both characteristics can be represented by an energy principle,

$$\frac{1}{2} \int_{x_0}^{x_n} [k^2\phi'(x)^2 + \phi''(x)^2] dx + \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} \lambda_i \phi(x) dx = \min, \tag{76}$$

where  $k$  is a constant. The Euler–Lagrange equation for this variational problem is

$$\phi''''(x) - k^2 \phi''(x) + \lambda_i = 0. \quad (77)$$

Therefore, the minimizing solution is the sum of a quadratic function and a combination of exponential functions,  $\exp(\pm k\xi)$ , in each sampling interval. The details of the solution and a procedure for practical solution are given in appendix 1. For a sufficiently large  $k$  ( $k > 10$ ), the solution is nearly the same as the quadratic function model. For a small value of  $k$  ( $k < 1$ ), the solution is basically the same as the fourth-order polynomial model. A change-over of the characteristics takes place at around  $k = 3$ –5. When the constraint by the Lagrange multiplier is not applied, the solution has been known as the exponential splines (Hoschek, Lasser 1993, p.98).

### 3. Examples of 1-D Interpolation

We next discuss our test of the procedure on several simple examples. First, we assumed a model function,  $\phi_{\text{model}}(x)$ , and numerically integrated it over the intervals and generated a discrete data set,  $N_i$  ( $i = 0, 1, 2, \dots, n - 1$ ). Our interpolation method was then applied, and the difference between the interpolated results and the original function evaluated.

We considered the following three kinds of model functions:

$$\phi_{\text{model 1}}(x) = 1 / \left[ 1 + \left( \frac{x - x_c}{a} \right)^2 \right]^b, \quad (78)$$

$$\phi_{\text{model 2}}(x) = \frac{1}{2} \left[ 1 + \tanh \left( \frac{x - x_c}{a} \right) \right], \quad (79)$$

$$\phi_{\text{model 3}}(x) = \frac{1}{2} \left[ 1 + \sin \left( \frac{x - x_c}{a} \right) \right]. \quad (80)$$

The first one is a Moffat function (Moffat 1969), which has often been used to approximate the point-spread function of various (optical or X-ray) telescopes (King 1971; Walker et al. 1994; Martens et al. 1995; Sakurai, Shin 2000). Here,  $x_c$  is the location of the peak,  $a$  is a parameter representing the width of the core, and  $b$  controls the decay of the function in the wing. In the following examples the value of  $b$  is fixed at 1.5. The second (hyperbolic tangent) function models a transition from 0 to 1 at  $x = x_c$  with a characteristic length scale  $a$ . The last example is a sinusoidal function of wavelength  $2\pi a$ . Since the interpolating function is assumed to have vanishing derivatives at the boundaries ( $x = x_0, x_n$ ), the sinusoid is not an adequate model to evaluate the performance of our interpolation schemes, however.

All of the three model functions vary between 0 and 1. The maximum values of their second derivatives are  $2b/a^2$ ,  $2/(3\sqrt{3}a^2)$ , and  $1/(2a^2)$  for models 1, 2, and 3, respectively. Therefore, for the same value of  $a$ , the curvature is largest in the Moffat function.

In the following examples, we set  $n = 21$ , and  $x_0 = -10.5, x_1 = -9.5, \dots, x_{21} = 10.5$ . For  $x_c$  we take the values 0, 0.25, and 0.5. In the case of Moffat function, the sampled data  $N_i$  are symmetric about the peak if  $x_c = 0$  or 0.5, while for other values of  $x_c$  the sampled  $N_i$  is not symmetric.

#### 3.1. Uniform $\sigma_i$

First we consider (1) the usual cubic spline interpolation, (2) our quadratic interpolation, and (3) our fourth-order interpolation with  $\sigma_i = \text{constant}$ . For each model and interpolation scheme, we computed the errors,

$$\delta\phi = \phi_{\text{interpolation}} - \phi_{\text{model}}, \quad (81)$$

and evaluated their root-mean-square and maximum values. For the cubic spline interpolation, we also evaluated the errors in  $N_i$ , namely

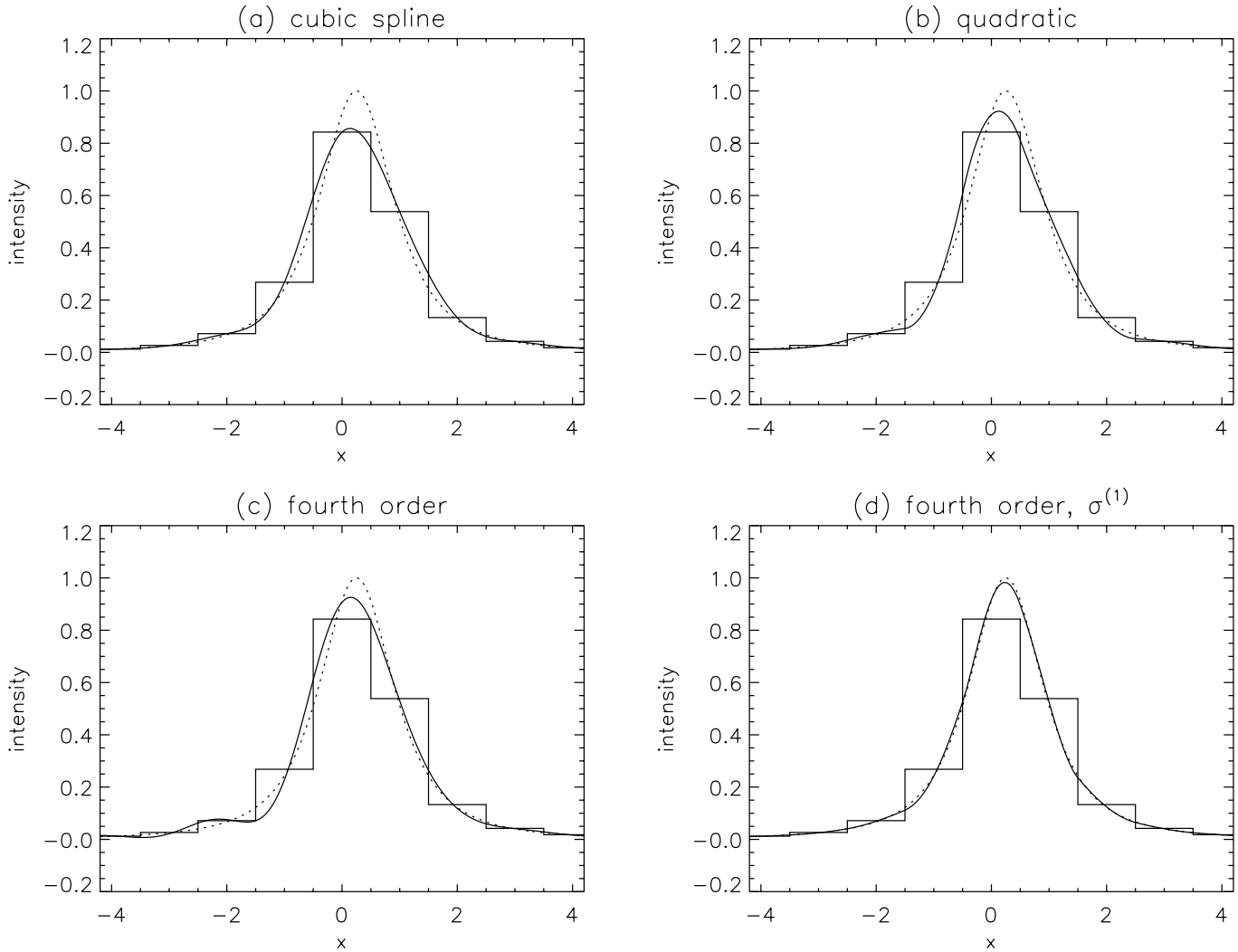
$$\delta N_i = \int_{x_i}^{x_{i+1}} \phi_{\text{interpolation}}(x) dx - N_i, \quad (82)$$

and evaluated their root-mean-square and maximum values. The quadratic and fourth-order interpolations developed in this paper have no errors in  $N_i$ .

Figures 3–5 show the results for models 1–3, respectively. In each figure, panels (a)–(c) represent (a) cubic spline interpolation, (b) quadratic interpolation, and (c) fourth-order interpolation. (Other panels will be explained in the next subsection.) The values of  $a$  [1.0 for model 1, 0.5 for model 2,  $2/\pi$  for model 3] were selected to yield maximum errors of roughly 10–20%. Since these values of  $a$  are close to the size of the pixel, they are not adequate as far as the interpolation is concerned, but were adopted for demonstration purposes. The values of  $x_c$  are 0.25 for figure 3, and 0 for figures 4 and 5. Only the central part ( $-4 \leq x \leq 4$ ) is shown in figures 3 and 4. In figure 5, only the range of positive  $x$  is shown.

The results are summarized in the upper half of table 1 for two values of  $a$  for each model: one that used in figures 3–5, and the other, which is twice of that. The tabulated errors are the largest values of errors among  $x_c = 0, 0.25, 0.5$ .

From table 1 we can derive the following properties:



**Fig. 3.** Example of interpolation applied to the Moffat function with  $a = 1$ ,  $b = 1.5$ ,  $x_c = 0.25$ . The dotted curve is the original Moffat function, and the sampled data are represented by step functions. The solid curve shows the interpolated results: (a) cubic spline interpolation, (b) quadratic interpolation, (c) fourth-order interpolation, and (d) fourth-order interpolation with stiffness parameter  $\sigma_i = \sigma_i^{(1)}$ .

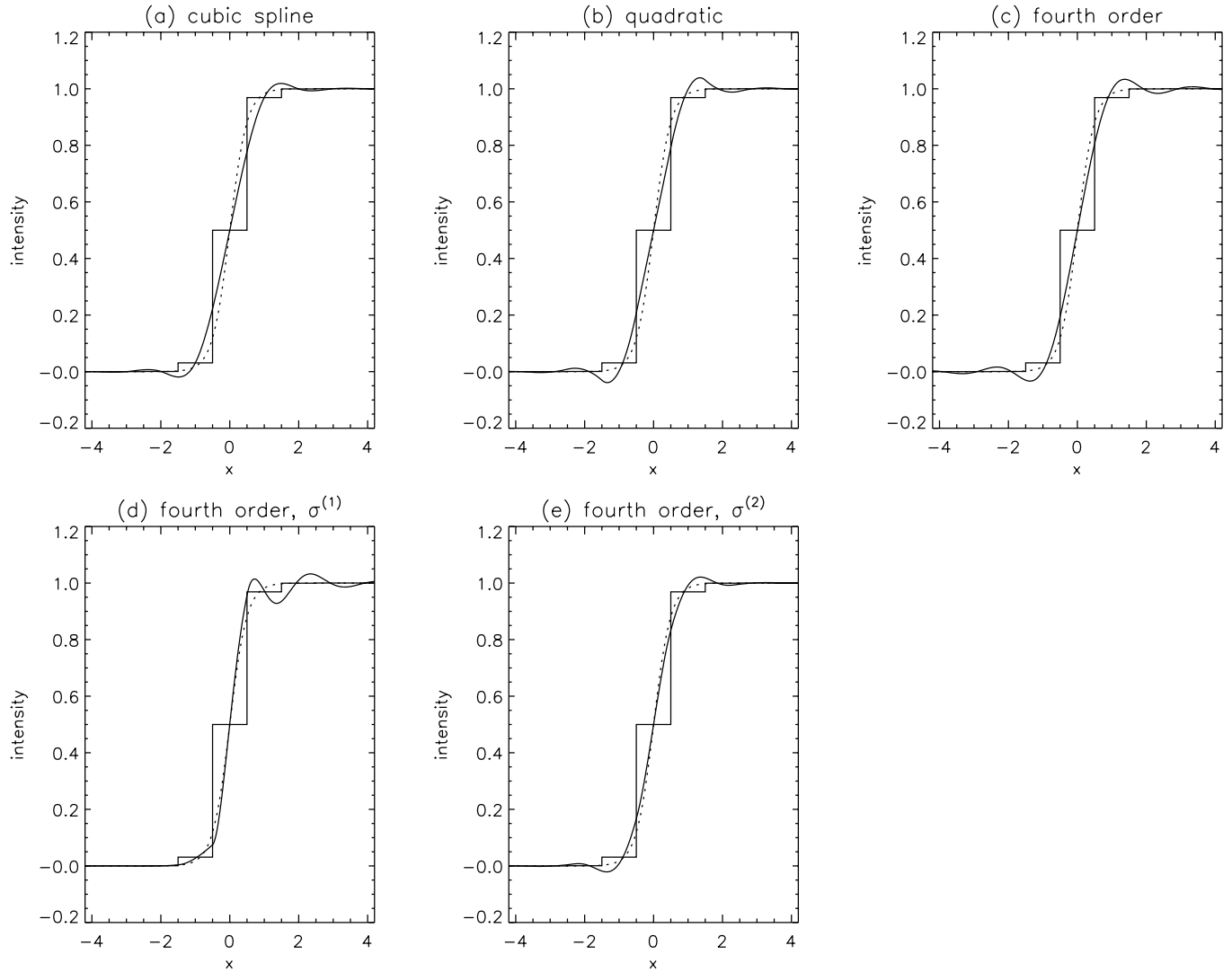
- (1) The accuracy is lowest for the cubic interpolation, and highest for the fourth-order interpolation for the Moffat function and hyperbolic tangent function models.
- (2) For a sinusoidal function, the errors are larger because of inadequate boundary conditions. The quadratic interpolation gave the worst results because the interpolation assumes a vanishing derivative at the boundaries. (The cubic spline interpolation assumes a vanishing second derivative at the boundary. The fourth-order interpolation assumes second and third-order derivatives to vanish at the boundary.)
- (3) The errors in  $N_i$  in the cubic interpolation in these examples amounted to 7%.
- (4) As  $a$  becomes small, the interpolated curves tend to ‘undershoot’ (to go below zero). This tendency is largest for the fourth-order interpolation.

Errors due to the boundary conditions mentioned in (2) can be improved by adopting the so-called ‘not-a-knot’ boundary condition (de Boor 1978, p.55). The modifications to the procedure are given in appendix 2. In this paper, however, we keep the basic variational formulation as close as possible, and in this sense we only use the boundary conditions derived from the variational principle.

The negative values found in the interpolated curves mentioned in (4) need to be suppressed if  $\phi$  represents the radiation intensity. This point is considered in the following.

### 3.2. Suppression of Negative Values by Manipulating $\sigma_i$

In a mechanical analogy,  $\sigma_i$  represents the stiffness of a material. Therefore, it is expected that a better fitting may be obtained if  $\sigma_i$  is locally reduced near to the sharp peaks or transitions in  $\phi$ . By properly selecting  $\sigma_i$ , the accuracy in fitting will improve,



**Fig. 4.** Example of interpolation applied to the hyperbolic tangent function with  $a = 0.5$ ,  $x_c = 0$ . The format is the same as figure 3 for panels (a)–(d). Panel (e) shows the result for  $\sigma_i = \sigma_i^{(2)}$ .

which also implies better suppression of negative values in the interpolation.

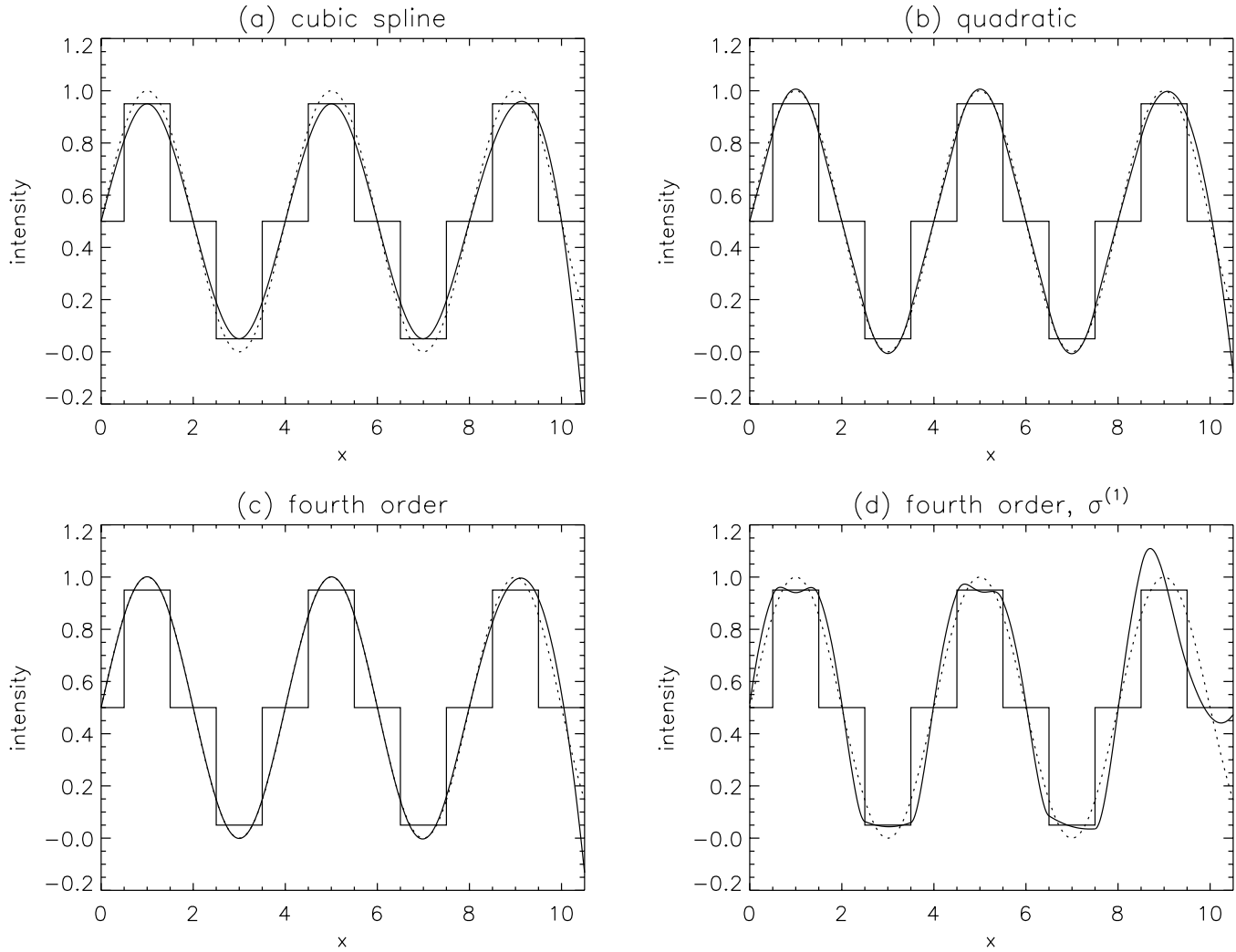
We tried two forms of  $\sigma_i$ :

$$\sigma_i^{(1)} = [f / (f + N_i / N_{\max})]^{\alpha_1}, \quad (83)$$

$$\sigma_i^{(2)} = 1 / [1 + (N_i'')^2 / \langle (N_i'')^2 \rangle]^{\alpha_2}. \quad (84)$$

Here,  $N_{\max}$  is the maximum value of data,  $N_i$ ;  $N_i''$  stands for the second derivative of  $N_i$  calculated numerically ( $N_i'' = N_{i+1} + N_{i-1} - 2N_i$ ); and  $\langle (N_i'')^2 \rangle$  denotes the mean square value of  $N_i''$ . Unless  $N_i$  is a linear function of  $i$ ,  $\langle (N_i'')^2 \rangle$  never vanishes, but generally some care has to be taken to handle such cases. A small number,  $f$ , is introduced to avoid a singularity when  $N_i = 0$ . We tried  $f = 0.1$  and  $f = 0.01$ , but the results were not sensitive to these values. In the following examples we used  $f = 0.01$ . The power indices  $\alpha_1$  and  $\alpha_2$  were varied between 1 and 2, and we found slightly better results for  $\alpha_1, \alpha_2 = 2$ , which were adopted in the following examples. The first form of  $\sigma_i$  makes the stiffness reduced where  $N_i$  is large. This works well when the data have sharp peaks, but fails if the data have sharp valleys. The second form of  $\sigma_i$  makes the stiffness reduced where the curvature of  $N_i$  is large. The latter looks to be applicable to more general cases, but the former one shows better suppression of negative interpolated values.

In the present examples, we have 21 values of  $\sigma_i$  that are adjustable, so that as an extreme case we would apply a nonlinear least squares minimization procedure and manipulate all the  $\sigma_i$ 's to minimize the square sum of the residuals. Thus, we could find the optimum distribution of  $\sigma_i$  (denoted as  $\sigma_i^{(\text{opt})}$ ) in the sense of least-squares errors, but this is only possible when we have a model and a generated set of data  $N_i$ . (In a real situation we are only given the data  $N_i$ , hence the optimum  $\sigma_i$  cannot be defined.)



**Fig. 5.** Example of interpolation applied to the sinusoidal function with  $a = 2/\pi$ ,  $x_c = 0$ . The format is the same as figure 3. Only a portion of positive  $x$  is shown because of symmetry.

In panels (d) of figures 3–5, the results for  $\sigma_i = \sigma_i^{(1)}$  are shown. For model 2, the result for  $\sigma_i = \sigma_i^{(2)}$  is shown in panel (e). The lower half of table 1 summarizes the results.

Compared to the fourth-order interpolation with  $\sigma_i = \text{constant}$ , we can notice that the errors are reduced for models 1 and 2. For model 3 (sinusoidal function), no improvements are seen. (In sinusoidal functions whose wavelengths are only a few pixels, one cannot identify particular intervals to reduce the stiffness  $\sigma_i$ .) In model 2, the application of  $\sigma_i^{(1)}$  suppresses the undershoot, but amplifies the overshoot. The application of  $\sigma_i^{(2)}$  suppresses both the under- and overshoot errors equally, and gives better results in terms of the root-mean-square errors.

If we strictly follow the variational formulation, the negative values in the interpolated results can be excluded, as is shown in appendix 3. However, the scheme is complicated and does not seem to be practical. For practical applications we prefer to use an appropriate form of the  $\sigma_i$  parameter to control the interpolation.

#### 4. Interpolation in 2-D

In the case of an ordinary spline fitting, in the extension from one- to two-dimensional models (from curve fitting to surface fitting), it is hard to retain the basic variational formalism because it leads to a rather complicated form of equations. Instead, a more practical approach toward computational efficiency is taken (e.g., Hoschek, Lasser 1993, p.251). Namely, a one-dimensional cubic spline fitting is applied both in the  $x$ - and  $y$ -directions along the mesh points, and the spline curves connecting the mesh data are generated. The construction of two-dimensional surfaces defined by these spline curves at their edges is done by the so-called ‘tensor product’ form of one-dimensional spline functions (see below). Here, we only consider a uniformly-spaced two-dimensional grid  $(x_i, y_j)$  ( $i = 0, 1, 2, \dots, n_x, j = 0, 1, 2, \dots, n_y$ ). The grid spacings  $\Delta x = x_{i+1} - x_i$  and  $\Delta y = y_{j+1} - y_j$  are

**Table 1.** Summary of errors in various one-dimensional interpolation schemes.

	Moffat		tanh		sin	
	$a = 2$	$a = 1$	$a = 1$	$a = 0.5$	$a = 4/\pi$	$a = 2/\pi$
Cubic spline						
$\delta N_i$ (rms)	0.007	0.017	0.005	0.011	0.009	0.036
$\delta N_i$ (max)	0.028	0.066	0.014	0.034	0.014	0.049
$\delta\phi$ (rms)	0.009	0.037	0.006	0.024	0.011	0.044
$\delta\phi$ (max)	0.043	0.195	0.024	0.106	0.013	0.050
Quadratic						
$\delta\phi$ (rms)	0.005	0.034	0.004	0.022	0.008	0.033
$\delta\phi$ (max)	0.022	0.163	0.018	0.099	0.084	0.276
Fourth order, $\sigma_i = \text{constant}$						
$\delta\phi$ (rms)	0.003	0.029	0.003	0.019	0.007	0.024
$\delta\phi$ (max)	0.013	0.137	0.011	0.082	0.056	0.206
Fourth order, $\sigma_i = \sigma_i^{(\text{opt})}$						
$\delta\phi$ (rms)	0.001	0.003	0.001	0.003	0.002	0.011
$\delta\phi$ (max)	0.006	0.018	0.004	0.014	0.013	0.131
Fourth order, $\sigma_i = \sigma_i^{(1)}$						
$\delta\phi$ (rms)	0.002	0.007	0.002	0.023	0.040	0.105
$\delta\phi$ (max)	0.008	0.041	0.012	0.104	0.141	0.613
Fourth order, $\sigma_i = \sigma_i^{(2)}$						
$\delta\phi$ (rms)	0.004	0.022	0.001	0.012	0.009	0.028
$\delta\phi$ (max)	0.020	0.114	0.003	0.055	0.064	0.198

assumed to be constants. The area subtended by four corner points  $(x_i, y_j)$ ,  $(x_{i+1}, y_j)$ ,  $(x_i, y_{j+1})$ , and  $(x_{i+1}, y_{j+1})$  will be called the  $(i, j)$ -th cell. The given data are

$$N_{i,j} = \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} \phi(x, y) dx dy, \quad (85)$$

which represents the volume under the surface  $\phi(x, y)$  assigned to the  $(i, j)$ -th cell. For astronomical applications, the cells are detector pixels and  $N_{i,j}$ 's represent photon counts.

#### 4.1. Quadratic Interpolation

First, we will apply the quadratic formula obtained in subsection 2.1 to two-dimensional cases. For the  $(i, j)$ -th cell, we assume the following 'tensor product' (bi-quadratic) form for  $\phi(x, y)$ :

$$\begin{aligned} \phi(x, y) = & \phi_{i,j} P(\xi)P(\eta) + \phi_{i+1,j} Q(\xi)P(\eta) + \phi_{i,j+1} P(\xi)Q(\eta) + \phi_{i+1,j+1} Q(\xi)Q(\eta) \\ & + [X_{i,j} R(\xi)P(\eta) + X_{i,j+1} R(\xi)Q(\eta)] / \Delta x + [Y_{i,j} P(\xi)R(\eta) + Y_{i+1,j} Q(\xi)R(\eta)] / \Delta y \\ & + N_{i,j} R(\xi)R(\eta) / (\Delta x \Delta y). \end{aligned} \quad (86)$$

Here, the normalized variables within a cell are defined as

$$\xi = (x - x_i) / \Delta x, \quad (87)$$

$$\eta = (y - y_j) / \Delta y, \quad (88)$$

and functions  $P$ ,  $Q$ , and  $R$  are taken from equations (16)–(18). This construction guarantees that  $\phi(x, y)$  is continuous across the cell boundaries, and the constraint (85) is automatically satisfied.

When we cut the surface  $\phi(x, y)$  with  $y = \text{constant}$  and  $x = \text{constant}$  planes, the areas under the curves in the  $(i, j)$ -th cell are defined as

$$X_i(y) = \int_{x_i}^{x_{i+1}} \phi(x, y) dx, \quad (89)$$

$$Y_j(x) = \int_{y_j}^{y_{j+1}} \phi(x, y) dy. \quad (90)$$

$X_{i,j}$  and  $Y_{i,j}$  in equation (86) are the values of these functions at the cell boundaries.

From the given data  $N_{i,j}$  we can obtain  $\phi_{i,j}$ ,  $X_{i,j}$ , and  $Y_{i,j}$  as follows (bi-quadratic scheme).

- (i) For each  $j$  ( $j = 0, 1, \dots, n_y - 1$ ) apply one-dimensional formulae (32)–(35) in the  $x$ -direction on  $N_{i,j}$  ( $i = 0, 1, \dots, n_x - 1$ ) and obtain  $Y_{i,j}$  ( $i = 0, 1, \dots, n_x$ ).
- (ii) For each  $i$  ( $i = 0, 1, \dots, n_x - 1$ ) apply one-dimensional formulae in the  $y$ -direction on  $N_{i,j}$  ( $j = 0, 1, \dots, n_y - 1$ ) and obtain  $X_{i,j}$  ( $j = 0, 1, \dots, n_y$ ).
- (iii) For each  $j$  ( $j = 0, 1, \dots, n_y$ ) apply one-dimensional formulae in the  $x$ -direction on  $X_{i,j}$  ( $i = 0, 1, \dots, n_x - 1$ ) and obtain  $\phi_{i,j}$  ( $i = 0, 1, \dots, n_x$ ).
- (iv) For each  $i$  ( $i = 0, 1, \dots, n_x$ ) if we apply one-dimensional formulae in the  $y$ -direction on  $Y_{i,j}$  ( $j = 0, 1, \dots, n_y - 1$ ), we also obtain  $\phi_{i,j}$  ( $j = 0, 1, \dots, n_y$ ). The results are the same as those of (iii).

If the coefficients  $\phi_{i,j}$ ,  $X_{i,j}$ , and  $Y_{i,j}$  thus obtained are inserted in equation (86), the interpolating function has continuous first derivative across the cell boundaries.

#### 4.2. Fourth-Order Interpolation

Next, we will apply the fourth-order (quartic) formula obtained in subsection 2.2 to two-dimensional cases. For the  $(i, j)$ -th cell, we assume the following bi-quartic form for  $\phi(x, y)$ :

$$\begin{aligned} \phi(x, y) = & \phi_{i,j} p_0(\xi) p_0(\eta) + \phi_{i+1,j} q_0(\xi) p_0(\eta) + \phi_{i,j+1} p_0(\xi) q_0(\eta) + \phi_{i+1,j+1} q_0(\xi) q_0(\eta) \\ & + \phi_{i,j}^{(x)} p_1(\xi) p_0(\eta) + \phi_{i+1,j}^{(x)} q_1(\xi) p_0(\eta) + \phi_{i,j+1}^{(x)} p_1(\xi) q_0(\eta) + \phi_{i+1,j+1}^{(x)} q_1(\xi) q_0(\eta) \\ & + \phi_{i,j}^{(y)} p_0(\xi) p_1(\eta) + \phi_{i+1,j}^{(y)} q_0(\xi) p_1(\eta) + \phi_{i,j+1}^{(y)} p_0(\xi) q_1(\eta) + \phi_{i+1,j+1}^{(y)} q_0(\xi) q_1(\eta) \\ & + \phi_{i,j}^{(xy)} p_1(\xi) p_1(\eta) + \phi_{i+1,j}^{(xy)} q_1(\xi) p_1(\eta) + \phi_{i,j+1}^{(xy)} p_1(\xi) q_1(\eta) + \phi_{i+1,j+1}^{(xy)} q_1(\xi) q_1(\eta) \\ & + \left[ X_{i,j} r(\xi) p_0(\eta) + X_{i,j+1} r(\xi) q_0(\eta) + X_{i,j}^{(y)} r(\xi) p_1(\eta) + X_{i,j+1}^{(y)} r(\xi) q_1(\eta) \right] / \Delta x \\ & + \left[ Y_{i,j} p_0(\xi) r(\eta) + Y_{i+1,j} q_0(\xi) r(\eta) + Y_{i,j}^{(x)} p_1(\xi) r(\eta) + Y_{i+1,j}^{(x)} q_1(\xi) r(\eta) \right] / \Delta y \\ & + N_{i,j} r(\xi) r(\eta) / (\Delta x \Delta y). \end{aligned} \tag{91}$$

Functions  $p_0, q_0, p_1, q_1$ , and  $r$  are taken from equations (51)–(55).  $X_{i,j}^{(y)}$  and  $Y_{i,j}^{(x)}$  are the derivative of  $X_i(y)$  and  $Y_j(x)$  [equations (89)–(90)] at the cell boundaries. This construction of  $\phi(x, y)$  guarantees that  $\phi(x, y)$ ,  $\phi^{(x)}(x, y)$ ,  $\phi^{(y)}(x, y)$ , and  $\phi^{(xy)}(x, y)$  (superscripts denote differentiations with respect to the indicated variables) are continuous across the cell boundaries.

From the given data  $N_{i,j}$  we can obtain  $\phi_{i,j}, \phi_{i,j}^{(x)}, \phi_{i,j}^{(y)}, \phi_{i,j}^{(xy)}, X_{i,j}, X_{i,j}^{(y)}, Y_{i,j}$ , and  $Y_{i,j}^{(x)}$  as follows (bi-quartic scheme):

- (i) For each  $j$  ( $j = 0, 1, \dots, n_y - 1$ ) apply one-dimensional formula (71)–(75) in the  $x$ -direction on  $N_{i,j}$  ( $i = 0, 1, \dots, n_x - 1$ ) and obtain  $Y_{i,j}$  and  $Y_{i,j}^{(x)}$  ( $i = 0, 1, \dots, n_x$ ).
- (ii) For each  $i$  ( $i = 0, 1, \dots, n_x - 1$ ) apply one-dimensional formula in the  $y$ -direction on  $N_{i,j}$  ( $j = 0, 1, \dots, n_y - 1$ ) and obtain  $X_{i,j}$  and  $X_{i,j}^{(y)}$  ( $j = 0, 1, \dots, n_y$ ).
- (iii) For each  $j$  ( $j = 0, 1, \dots, n_y$ ) apply one-dimensional formula in the  $x$ -direction on  $X_{i,j}$  ( $i = 0, 1, \dots, n_x - 1$ ) and obtain  $\phi_{i,j}$  and  $\phi_{i,j}^{(x)}$  ( $i = 0, 1, \dots, n_x$ ).
- (iv) For each  $j$  ( $j = 0, 1, \dots, n_y$ ) apply one-dimensional formula in the  $x$ -direction on  $X_{i,j}^{(y)}$  ( $i = 0, 1, \dots, n_x - 1$ ) and obtain  $\phi_{i,j}^{(y)}$  and  $\phi_{i,j}^{(xy)}$  ( $i = 0, 1, \dots, n_x$ ).
- (v) For each  $i$  ( $i = 0, 1, \dots, n_x$ ) if we apply one-dimensional formula in the  $y$ -direction on  $Y_{i,j}$  ( $j = 0, 1, \dots, n_y - 1$ ), we also obtain  $\phi_{i,j}$  and  $\phi_{i,j}^{(y)}$  ( $j = 0, 1, \dots, n_y$ ). The results are the same as those of (iii) and (iv).
- (vi) Likewise, for each  $j$  ( $j = 0, 1, \dots, n_y$ ) if we apply one-dimensional formula in the  $x$ -direction on  $Y_{i,j}^{(x)}$  ( $i = 0, 1, \dots, n_x - 1$ ), we obtain  $\phi_{i,j}^{(x)}$  and  $\phi_{i,j}^{(xy)}$  ( $i = 0, 1, \dots, n_x$ ). The results are the same as those of (iii) and (iv).

If the coefficients  $\phi_{i,j}, \phi_{i,j}^{(x)}, \phi_{i,j}^{(y)}, \phi_{i,j}^{(xy)}, X_{i,j}, X_{i,j}^{(y)}, Y_{i,j}$ , and  $Y_{i,j}^{(x)}$  thus obtained are inserted in equation (91), the interpolating function has continuous third derivatives across the cell boundaries if  $\sigma_i$  in equation (38) is uniform. However, if  $\sigma_i$  is not uniform, the results obtained in steps (iii)–(iv) and (v)–(vi) become inconsistent. In such a case, we will assign mean values of those obtained in steps (iii)–(iv) and (v)–(vi). Whether this scheme gives satisfactory results is examined below.

#### 4.3. Variational Approach

The solutions described above are not the minimizing solution for an energy principle. To see this point, we may consider the following two forms of energy:

$$W_1 = \frac{1}{2} \int [\phi^{(x)}(x, y)^2 + \phi^{(y)}(x, y)^2] dx dy + \lambda \int \phi(x, y) dx dy \tag{92}$$

and

$$W_2 = \frac{1}{2} \int [\phi^{(xx)}(x, y)^2 + 2\phi^{(xy)}(x, y)^2 + \phi^{(yy)}(x, y)^2] dx dy + \lambda \int \phi(x, y) dx dy. \quad (93)$$

Here, the superscripts denote differentiations with respect to the indicated variables. The Euler–Lagrange equation for  $W_1 = \min$  and  $W_2 = \min$  are, respectively,

$$\Delta\phi - \lambda = 0 \quad (94)$$

and

$$\Delta^2\phi + \lambda = 0. \quad (95)$$

Unlike the one-dimensional cases, solutions for  $\phi$  may not be polynomials of finite order, but will involve infinite-series expansions. Therefore, a truncated polynomial form, like equation (91), is an approximation to the solution, or may be regarded as a trial function to the variational principle. In the following we attempt to determine the coefficients by minimizing the energy.

The trial function (91) contains 24 unknown coefficients per each cell, but they are shared by adjacent cells. Each node point is assigned four unknowns ( $\phi_{i,j}$ ,  $\phi_{i,j}^{(x)}$ ,  $\phi_{i,j}^{(y)}$ , and  $\phi_{i,j}^{(xy)}$ ), and there are  $(n_x + 1)(n_y + 1)$  node points. At each vertical ( $x = \text{constant}$ ) boundary between the cells,  $Y_{i,j}$  and  $Y_{i,j}^{(x)}$  are the unknowns; there are  $(n_x + 1)n_y$  such boundaries. A similar consideration applies to the horizontal ( $y = \text{constant}$ ) boundaries. Therefore, the total number of unknowns is  $4(n_x + 1)(n_y + 1) + 2(n_x + 1)n_y + 2n_x(n_y + 1)$  or, roughly speaking,  $O(8n_x n_y)$ .

We consider the following quantity to be minimized, which is a combination of  $W_1$  and  $W_2$  defined by equations (92) and (93):

$$W = \frac{1}{2} \sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} \sigma_{i,j} \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} \left\{ \tau [\phi^{(x)}(x, y)^2 + \phi^{(y)}(x, y)^2] + (1 - \tau) [\phi^{(xx)}(x, y)^2 + \phi^{(yy)}(x, y)^2 + 2\phi^{(xy)}(x, y)^2] \right\} dx dy = \min. \quad (96)$$

The parameter  $\sigma_{i,j}$  represents the weight assigned to the  $(i, j)$ -th cell, and can be regarded as the stiffness of a hypothetical material. The parameter  $\tau$  expresses the relative contributions to the energy from tension and elasticity;  $\tau = 0$  means an elasticity-dominant, steel-like material, and  $\tau = 1$  means a tension-dominant, membrane-like material. Since the trial functions satisfy the constraint for volumes  $N_{i,j}$ , there is no need to add Lagrange multipliers to the above expression.

When we substitute the form for the trial function [equation (91)] into the above expression,  $W$  becomes a quadratic function of the unknown coefficients. By minimizing with respect to these coefficients, we obtain a system of linear equations. The explicit form of the equation is not expressible in a compact form, and can only be worked out numerically. The size of the matrix is roughly  $8n_x n_y \times 8n_x n_y$ , and the usual algorithm of solution by the sweep-out method is practically limited to  $n_x, n_y \lesssim 15$ . However, the matrix is sparse, and iterative methods (e.g., the conjugate gradient method, Press et al. 1992) are more suitable to solve the equation. Because the band width of the matrix is found to be at most 60, the effective number of matrix elements is less than  $480n_x n_y$ .

## 5. Examples of 2-D Interpolation

We applied our method to the following three examples:

$$\phi_{\text{model 1}}(x, y) = 1 \left/ \left[ 1 + \left( \frac{x - x_c}{a} \right)^2 + \left( \frac{y - y_c}{a} \right)^2 \right]^b \right., \quad (97)$$

$$\begin{aligned} \phi_{\text{model 2}}(x, y) &= \frac{1}{8} \left[ 1 + \tanh \left( \frac{x + w - x_c}{a} \right) \right] \left[ 1 - \tanh \left( \frac{x - w - x_c}{a} \right) \right] \\ &\quad \times \left[ 1 + \tanh \left( \frac{y + w - y_c}{a} \right) \right] \left[ 1 - \tanh \left( \frac{y - w - y_c}{a} \right) \right], \end{aligned} \quad (98)$$

$$\phi_{\text{model 3}}(x, y) = \frac{1}{2} \left[ 1 - \tanh \left( \frac{\sqrt{(x - x_c)^2 + (y - y_c)^2} - w}{a} \right) \right]. \quad (99)$$

The first example is a two-dimensional Moffat function. The second and the third examples are flat-top functions of half width,  $w$ ; models 1 and 2 are here called square-table and round-table functions, respectively. We set  $n_x = n_y = 21$ , and  $x_0 = -10.5, x_1 = -9.5, \dots, x_{21} = 10.5, y_0 = -10.5, y_1 = -9.5, \dots, y_{21} = 10.5$ . The value of  $w$  was set to 5. For the values of  $(x_c, y_c)$ , we took  $(0, 0), (0.25, 0), (0.5, 0), (0.25, 0.25), (0.5, 0.25),$  and  $(0.5, 0.5)$ , and the largest errors were recorded.

In parallel to the one-dimensional cases, we applied to these models (1) the usual bi-cubic spline interpolation, (2) our bi-quadratic interpolation, and (3) our bi-quartic (fourth-order) interpolation with  $\sigma_i = \text{constant}$ . For the bi-cubic interpolation, the errors in  $N_{i,j}$  were also evaluated. For bi-quartic interpolation, we also introduced the stiffness parameter as  $\sigma_{i,j} = \sigma_{i,j}^{(1)}$  and  $\sigma_{i,j} = \sigma_{i,j}^{(2)}$ . The definition for  $\sigma_{i,j}^{(1)}$  is the same as equation (83), while for  $\sigma_{i,j}^{(2)}$   $N_i''$  in equation (84) should be replaced by the

**Table 2.** Summary of errors in various two-dimensional interpolation schemes.

	Moffat		Square table		Round table	
	$a = 2$	$a = 1$	$a = 1$	$a = 0.5$	$a = 1$	$a = 0.5$
Cubic spline						
$\delta N_{i,j}(\text{rms})$	0.003	0.006	0.006	0.015	0.006	0.012
$\delta N_{i,j}(\text{max})$	0.052	0.110	0.024	0.063	0.018	0.044
$\delta\phi(\text{rms})$	0.004	0.011	0.009	0.034	0.007	0.023
$\delta\phi(\text{max})$	0.083	0.333	0.038	0.171	0.028	0.111
Quadratic						
$\delta\phi(\text{rms})$	0.002	0.009	0.005	0.029	0.003	0.018
$\delta\phi(\text{max})$	0.044	0.280	0.025	0.154	0.018	0.100
Fourth order, $\sigma_i = \text{constant}$						
$\delta\phi(\text{rms})$	0.001	0.008	0.004	0.026	0.002	0.016
$\delta\phi(\text{max})$	0.025	0.239	0.016	0.130	0.011	0.086
Fourth order, $\sigma_i = \sigma_i^{(1)}$						
$\delta\phi(\text{rms})$	0.001	0.002	0.003	0.030	0.002	0.025
$\delta\phi(\text{max})$	0.017	0.092	0.015	0.273	0.012	0.187
Fourth order, $\sigma_i = \sigma_i^{(2)}$						
$\delta\phi(\text{rms})$	0.005	0.005	0.001	0.017	0.001	0.011
$\delta\phi(\text{max})$	0.067	0.168	0.005	0.096	0.005	0.062
Minimization, $\tau = 0, \sigma_i = \sigma_i^{(2)}$						
$\delta\phi(\text{rms})$	0.001	0.002	0.003	0.031	0.003	0.027
$\delta\phi(\text{max})$	0.016	0.093	0.015	0.246	0.014	0.142
Minimization, $\tau = 1, \sigma_i = \sigma_i^{(2)}$						
$\delta\phi(\text{rms})$	0.007	0.006	0.009	0.011	0.006	0.010
$\delta\phi(\text{max})$	0.073	0.208	0.050	0.132	0.050	0.084

Laplacian:  $\Delta N_{i,j} = N_{i+1,j} + N_{i-1,j} + N_{i,j+1} + N_{i,j-1} - 4N_{i,j}$ . Finally, the minimization solutions described in subsection 4.2 were obtained for the same setting of  $\sigma_{i,j}$  as in the fourth-order interpolation calculation. For the parameter  $\tau$ , we tested  $\tau = 0$  and  $\tau = 1$ . The results are summarized in table 2.

Figures 6–8 show the results of bi-quartic interpolation, with weighting by  $\sigma_{i,j}^{(2)}$ . Figure 6 shows the case of the Moffat function with  $a = 1$ ,  $(x_c, y_c) = (0.25, 0.5)$ . Figures 7 and 8 show, respectively, the square-table function and round-table function. For these we took  $a = 0.5$ ,  $(x_c, y_c) = (0, 0)$ .

Our simulations show the following properties:

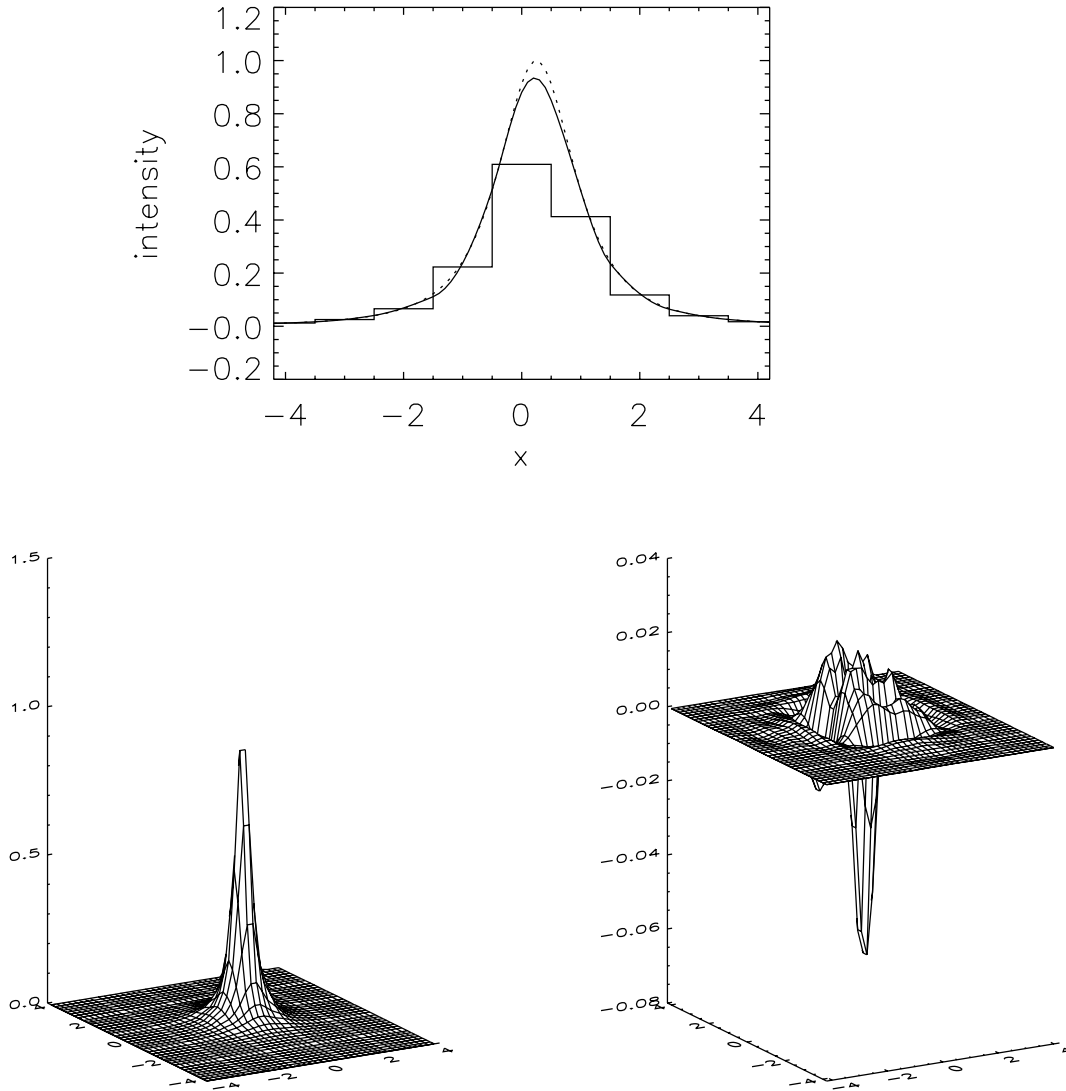
- (1) The accuracy is lowest for the bi-cubic interpolation, and highest for the fourth-order interpolation, as in the one-dimensional cases.
- (2) In the minimizing solutions,  $\tau = 0$  gives results better than  $\tau = 1$ . Probably the form of the trial function (91), which is a fourth order polynomial, better conforms with the energy defined by equation (93).
- (3) The bi-quartic algorithm gives almost the same results as those of minimizing solutions based on the variational formulation, regardless of whether  $\sigma_{i,j}$  is uniform or not.
- (4) Undershooting of interpolated curves can be suppressed if we use a stiffness parameter,  $\sigma_{i,j}^{(1)}$ .

As the application of minimizing solutions is limited to relatively small sizes of image and the computation is time-consuming, the bi-quartic scheme described in subsection 4.2 is a practical and most accurate choice.

## 6. Summary

In this paper, we have discussed the interpolation of astronomical image data in which photon counts or energy integrated over pixels are specified. In one-dimensional cases, we considered quadratic and fourth-order interpolations. The quadratic interpolation is defined in equation (12), and its solution scheme is given in equations (32)–(35). The fourth-order interpolation is defined in equation (45), and its solution scheme is given in equations (71)–(75). There is no big difference in computational complexity between the two methods, and from the accuracy of the interpolation, the fourth-order scheme is recommended. In the fourth-order scheme it is also possible to suppress negative values in the interpolated curves by manipulating the parameter  $\sigma_i$ .

In two-dimensional cases, we considered an extension of one-dimensional schemes (both quadratic and fourth-order schemes) and also a more basic minimization scheme based on the variational principle. In the former approach, the one-dimensional



**Fig. 6.** Two-dimensional interpolation applied to the Moffat function with  $a = 1$ ,  $b = 1.5$ ,  $x_c = 0.25$ ,  $y_c = 0.5$ . The top panel shows the slice at  $y = y_c$ . The meanings of the solid, dotted, and step curves are the same as in figures 3–5. On the bottom left is a two-dimensional plot of the original model function, and the bottom-right panel shows the fitting errors,  $\phi_{\text{interpolation}} - \phi_{\text{model}}$ .

scheme is applied in the  $x$ - and  $y$ -directions separately. In the latter approach, a matrix equation has to be solved, which is time-consuming. The fourth-order scheme gives better results than the quadratic scheme, and comparable to the full minimization solution. Therefore, the fourth-order scheme given in subsection 4.2 is recommended. The parameter  $\sigma_{i,j}$  can be used to suppress negative values in the interpolated data.

T.S. is grateful for the hospitality provided by the National Solar Observatory, Tucson, where part of this work was carried out. He thanks Dr. J.W. Harvey for comments on the draft of this paper.

**Appendix 1. Solution for a Mixed Form of Energies**

For the energy density of the form of equation (76), the solution can also be represented as equation (45). The explicit forms for the functions  $p_0$ ,  $q_0$ ,  $p_1$ ,  $q_1$ , and  $r$  are

$$\begin{aligned}
 p_0(\xi) = & \left(\frac{h_1}{2} + \frac{h_2}{2}\right) (1 - e^{-k\xi}) + \left(\frac{h_1}{2} - \frac{h_2}{2}\right) [e^{-k(1-\xi)} - e^{-k}] \\
 & + 1 - \frac{kh_1}{2} (1 + e^{-k}) \xi - \frac{kh_2}{2} \xi(1 - \xi)(1 - e^{-k}),
 \end{aligned}
 \tag{A1}$$

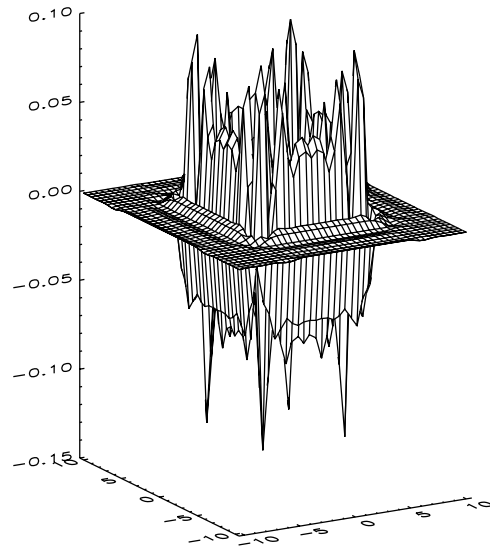
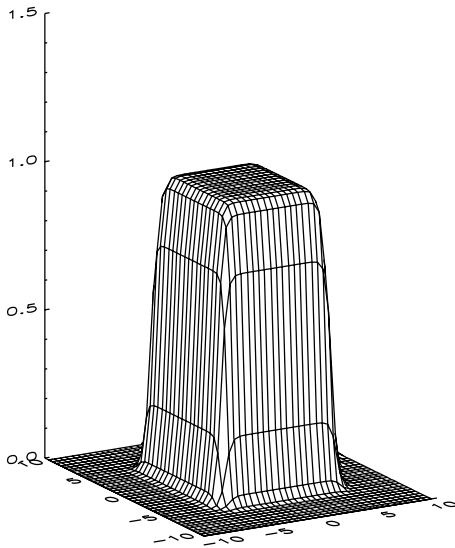
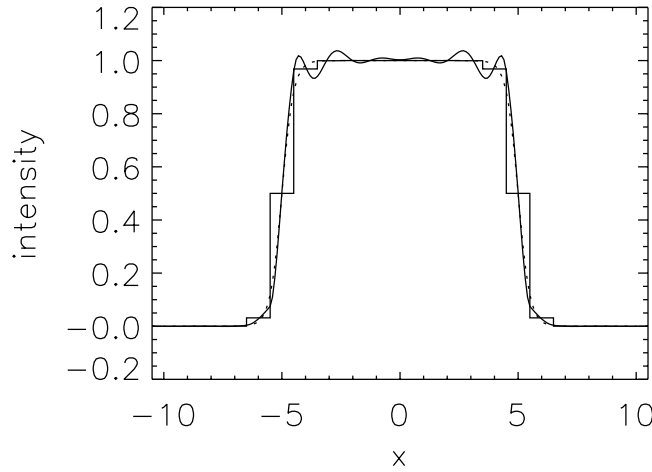


Fig. 7. Two-dimensional interpolation applied to a square-shaped, flat-top function with  $a = 0.5$ ,  $x_c = y_c = 0$ . The format is the same as figure 6.

$$q_0(\xi) = p_0(1 - \xi), \tag{A2}$$

$$p_1(\xi) = \left(\frac{h_1}{4} + \frac{h_2}{12}\right)(1 - e^{-k\xi}) + \left(\frac{h_1}{4} - \frac{h_2}{12}\right)[e^{-k(1-\xi)} - e^{-k}] - \frac{\xi^2}{2} + \left[1 - \frac{kh_1}{4}(1 + e^{-k})\right]\xi - \frac{kh_2}{12}(1 - e^{-k})\xi(1 - \xi), \tag{A3}$$

$$q_1(\xi) = -p_1(1 - \xi), \tag{A4}$$

$$r(\xi) = h_2[k(1 - e^{-k})\xi(1 - \xi) + e^{-k\xi} - 1 + e^{-k(1-\xi)} - e^{-k}], \tag{A5}$$

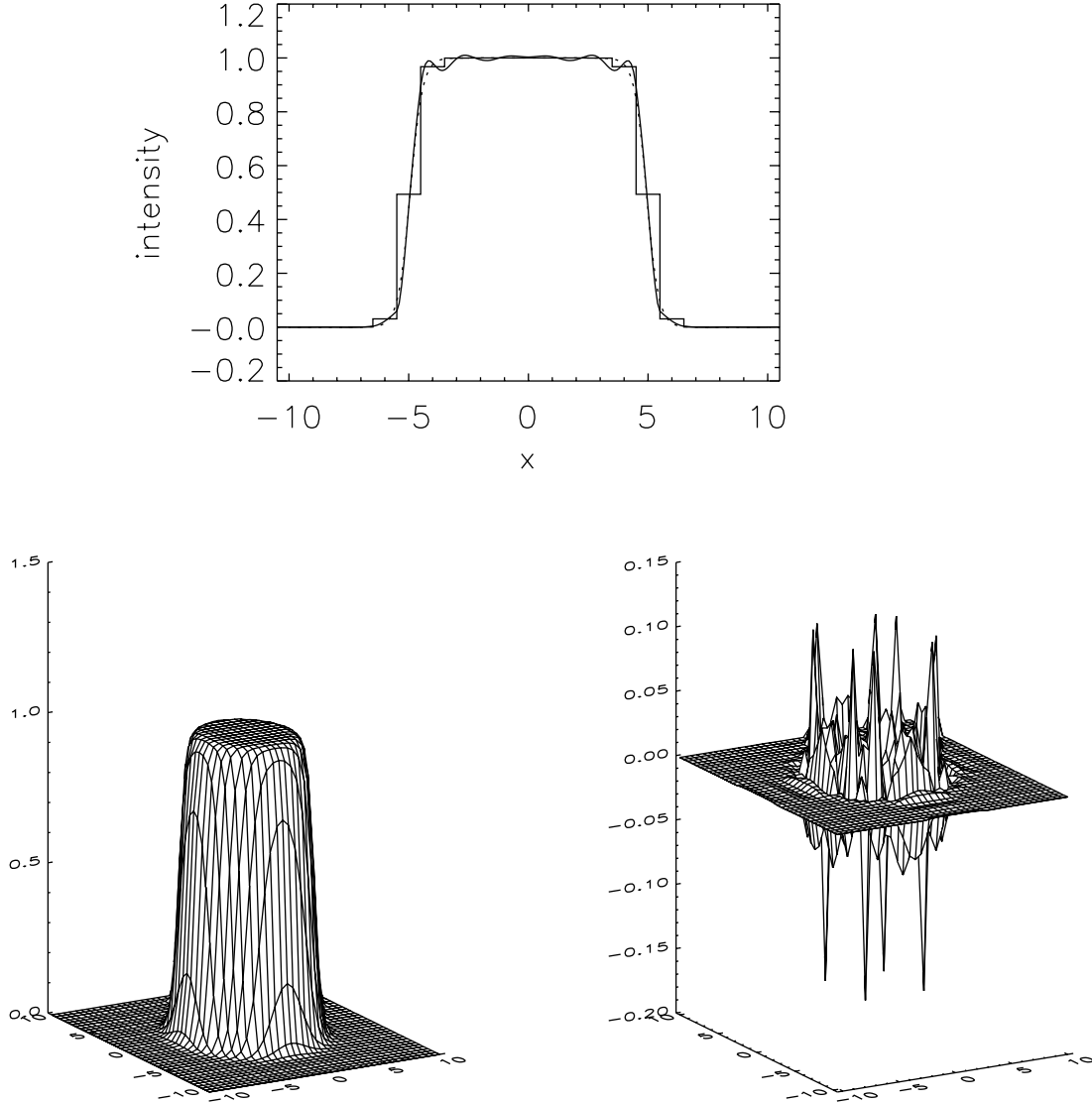
where

$$h_1^{-1} = \frac{k}{2} - 1 + \left(\frac{k}{2} + 1\right)e^{-k}, \tag{A6}$$

$$h_2^{-1} = \frac{2}{k} + \frac{k}{6} - 1 - \left(\frac{2}{k} + \frac{k}{6} + 1\right)e^{-k}. \tag{A7}$$

These functions satisfy conditions (46)–(50). Figure 9 shows these functions when  $k = 5$ .

The equations for  $\phi_i$  and  $\phi'_i$  take the form of equations (56)–(58), with the same definition for  $\Phi_i$  [equation (59)]. We then found



**Fig. 8.** Two-dimensional interpolation applied to a round, flat-top function with  $a = 0.5$ ,  $x_c = y_c = 0$ . The format is the same as figure 6.

$$A_i = \begin{pmatrix} p_0''(1) & p_1''(1) \\ p_0'''(1) & p_1'''(1) \end{pmatrix} \quad (i = 1, 2, \dots, n), \quad (\text{A8})$$

$$B_0 = \begin{pmatrix} p_0''(0) & p_1''(0) \\ p_0'''(0) & p_1'''(0) \end{pmatrix}, \quad (\text{A9})$$

$$B_i = \begin{pmatrix} q_0''(1) - f_{i-1}^2 p_0''(0) & f_{i-1} q_1''(1) - f_{i-1}^2 p_1''(0) \\ q_0'''(1) - f_{i-1}^2 p_0'''(0) & f_{i-1} q_1'''(1) - f_{i-1}^3 p_1'''(0) \end{pmatrix} \quad (i = 1, 2, \dots, n-1), \quad (\text{A10})$$

$$B_n = \begin{pmatrix} q_0''(1) f_{n-1} q_1''(1) \\ q_0'''(1) f_{n-1} q_1'''(1) \end{pmatrix}, \quad (\text{A11})$$

$$C_0 = \begin{pmatrix} q_0''(0) f_0 q_1''(0) \\ q_0'''(0) f_0 q_1'''(0) \end{pmatrix}, \quad (\text{A12})$$

$$C_i = \begin{pmatrix} -f_{i-1}^2 q_0''(0) - f_{i-1}^2 f_i q_1''(0) \\ -f_{i-1}^3 q_0'''(0) - f_{i-1}^3 f_i q_1'''(0) \end{pmatrix} \quad (i = 1, 2, \dots, n-1), \quad (\text{A13})$$

$$D_0 = \begin{pmatrix} -r''(0) N_0 / \Delta_0 \\ -r'''(0) N_1 / \Delta_1 \end{pmatrix}, \quad (\text{A14})$$

$$D_i = \begin{pmatrix} f_{i-1}^2 r''(0) N_i / \Delta_i - r''(1) N_{i-1} / \Delta_{i-1} \\ f_{i-1}^3 r'''(0) N_i / \Delta_i - r'''(1) N_{i-1} / \Delta_{i-1} \end{pmatrix} \quad (i = 1, 2, \dots, n-1), \quad (\text{A15})$$

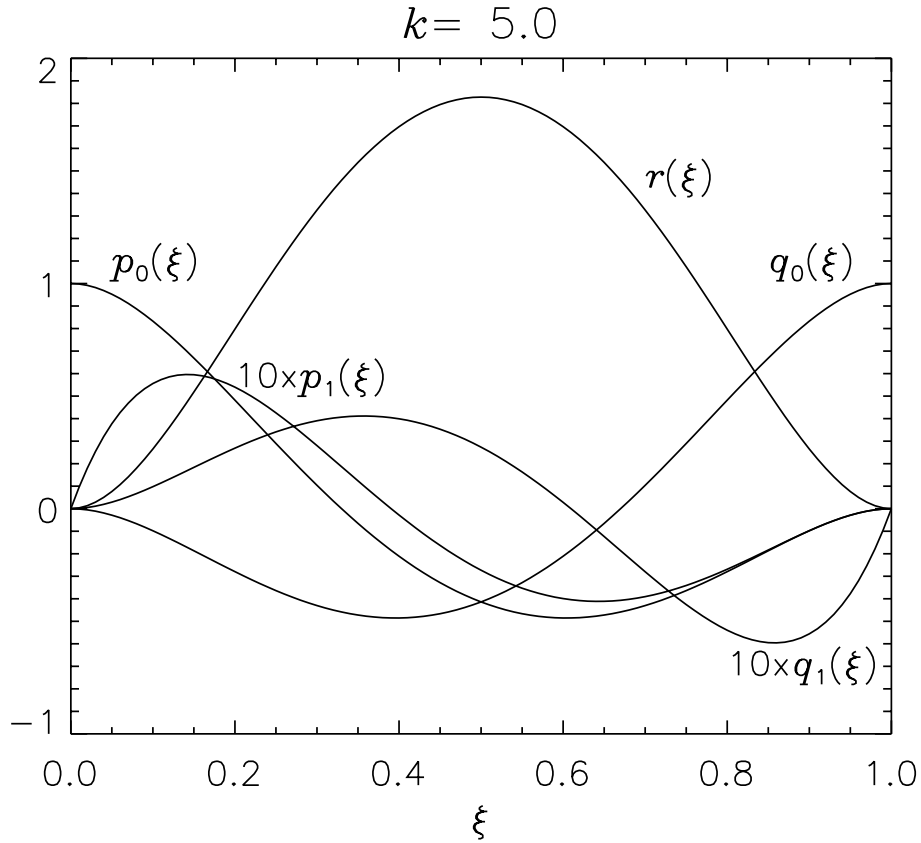


Fig. 9. Blending functions in the case of hybrid energy (tension plus elasticity), with  $k = 5$ .

$$D_n = \begin{pmatrix} -r''(1)N_{n-1}/\Delta_{n-1} \\ -r'''(1)N_{n-1}/\Delta_{n-1} \end{pmatrix}. \tag{A16}$$

The solution procedure is the same as equations (71)–(75).

**Appendix 2. Not-a-Knot Boundary Conditions**

In the case of quadratic interpolation, boundary conditions (9) and (10) are replaced by

$$\phi'' = \text{continuous at } x_1 \text{ and } x_{n-1}. \tag{A17}$$

Correspondingly, equations (23), (24), (26), (27), (29), and (31) are replaced by the following equations:

$$a_n = \left(1 + \frac{\Delta_{n-2}}{\Delta_{n-1}}\right)^2, \tag{A18}$$

$$b_0 = 1 + \frac{\Delta_0}{\Delta_1}, \tag{A19}$$

$$b_n = \frac{\Delta_{n-2}}{\Delta_{n-1}} \left(1 + \frac{\Delta_{n-2}}{\Delta_{n-1}}\right), \tag{A20}$$

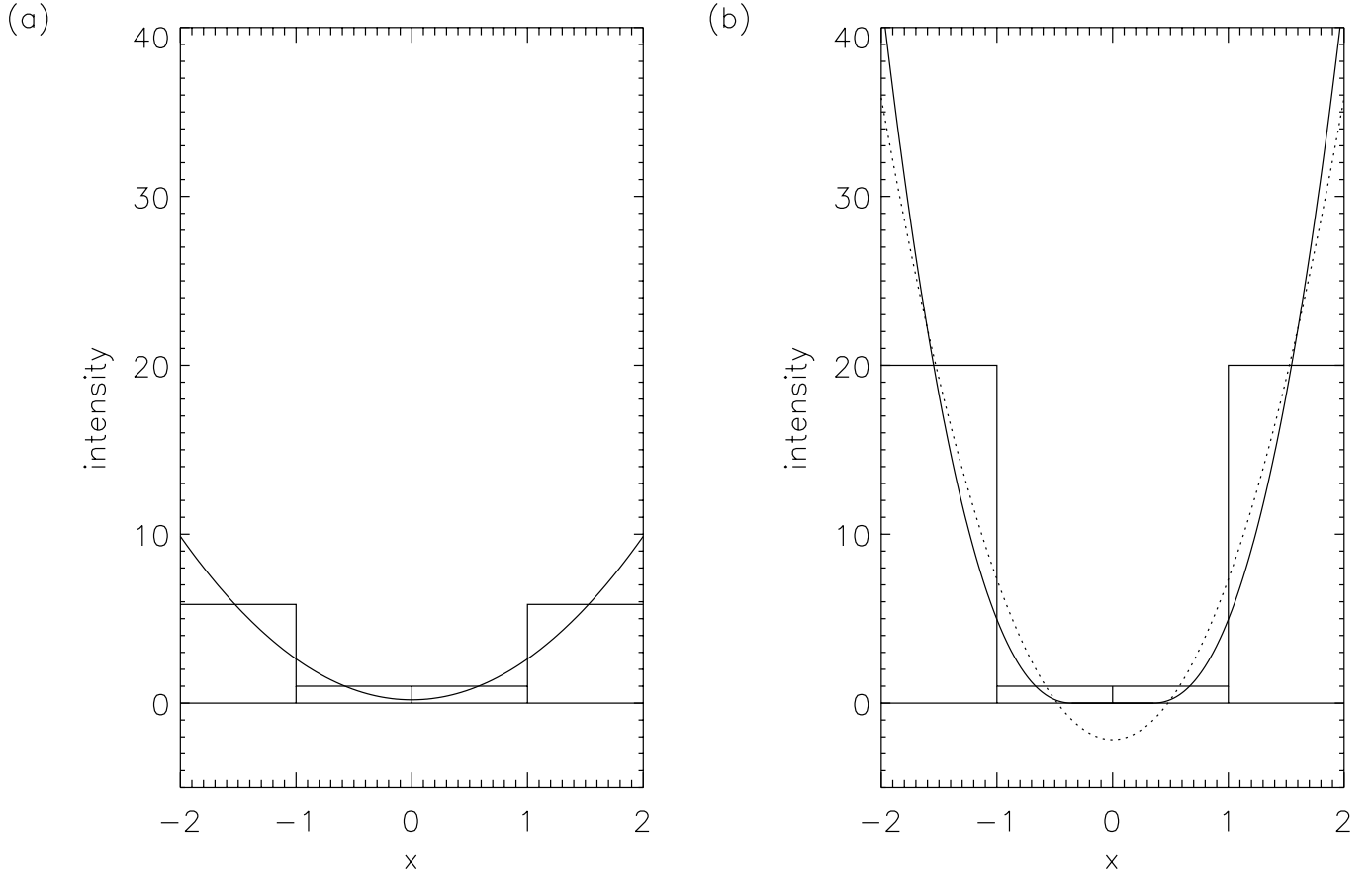
$$c_0 = \left(1 + \frac{\Delta_1}{\Delta_2}\right)^2, \tag{A21}$$

$$d_0 = \frac{N_0}{\Delta_0} \left(2 + 3\frac{\Delta_0}{\Delta_1}\right) + \frac{N_1}{\Delta_1} \left(\frac{\Delta_0}{\Delta_1}\right)^2, \tag{A22}$$

$$d_n = \frac{N_{n-2}}{\Delta_{n-2}} + \frac{N_{n-1}}{\Delta_{n-1}} \frac{\Delta_{n-2}}{\Delta_{n-1}} \left(3 + 2\frac{\Delta_{n-2}}{\Delta_{n-1}}\right). \tag{A23}$$

In the case of fourth-order interpolation, boundary conditions (39) and (41) are replaced by

$$\sigma_0\phi''''(x_1 - 0) = \sigma_1\phi''''(x_1 + 0), \tag{A24}$$



**Fig. 10.** Simple one-dimensional example for considering non-negativity constraint. (a) The interpolated curve just touches the  $x$ -axis. (b) A sub-interval of  $\phi \equiv 0$  develops near the center. The dotted curve is the result when a non-negativity constraint is not imposed.

$$\sigma_{n-2}\phi''''(x_{n-1} - 0) = \sigma_{n-1}\phi''''(x_{n-1} + 0). \quad (\text{A25})$$

Correspondingly, equations (62), (63), (65), (66), (68), and (70) are replaced by the following:

$$A_n = \begin{pmatrix} -2 + g_{n-2}f_{n-2}^2 \left( \frac{12 + 16f_{n-2} + 6f_{n-2}^2}{14} \right) & f_{n-2} \left( 1 + \frac{3g_{n-2}f_{n-2} + 3g_{n-2}f_{n-2}^2 + g_{n-2}f_{n-2}^3}{2} \right) \end{pmatrix}, \quad (\text{A26})$$

$$B_0 = \begin{pmatrix} 6 + 14f_0 + 8f_0^2 & 1 + 2f_0 + f_0^2 \\ 16 & 3 \end{pmatrix}, \quad (\text{A27})$$

$$B_n = \begin{pmatrix} g_{n-2}f_{n-2}^2 \left( \frac{8 + 14f_{n-2} + 6f_{n-2}^2}{16} \right) & -g_{n-2}f_{n-2}^2 \left( \frac{1 + 2f_{n-2} + f_{n-2}^2}{-3} \right) \end{pmatrix}, \quad (\text{A28})$$

$$C_0 = \begin{pmatrix} 6 + 16f_0 + 12f_0^2 - 2g_0f_0^4 & -f_0 \left( 1 + 3f_0 + 3f_0^2 + 3g_0f_0^3 \right) \\ 14 & -2 \end{pmatrix}, \quad (\text{A29})$$

$$D_0 = \begin{pmatrix} \left( \frac{12 + 30f_0 + 20f_0^2}{\Delta_0} \frac{N_0}{\Delta_0} - 2g_0f_0^4 \frac{N_1}{\Delta_1} \right) \\ 30 \frac{N_0}{\Delta_0} \end{pmatrix}, \quad (\text{A30})$$

$$D_n = \begin{pmatrix} -2 \frac{N_{n-2}}{\Delta_{n-2}} + g_{n-2}f_{n-2}^2 \frac{N_{n-1}}{\Delta_{n-1}} \left( 20 + 30f_{n-2} + 12f_{n-2}^2 \right) \\ 30 \frac{N_{n-1}}{\Delta_{n-1}} \end{pmatrix}. \quad (\text{A31})$$

### Appendix 3. Non-Negativity Constraint

If  $\phi$  represents the radiation intensity, it should be non-negative. The methods presented in the main text, however, do not themselves exclude negative values, and even if  $N_i$ 's or  $N_{i,j}$ 's are all non-negative, the interpolated function  $\phi$  can take negative values.

The introduction of non-negativity constraint is possible, at least in one-dimensional cases. In a geometrical analogy in which  $\phi$  represents the shape of some material covering the surface of liquid in a vessel, the non-negativity means that  $\phi$  should not go below the bottom of the vessel.

Let us consider a simple one-dimensional model made of five node points:  $x_0 = -2, x_1 = -1, x_2 = 0, x_3 = 1, x_4 = 2$ , and specify  $N_i$  as  $N_1 = N_2 = 1.0, N_3 = N_0$ . If we apply the fourth-order interpolation described in subsection 2.2, we find that the fitted curve is non-negative if  $N_0 < 5.84$ . Figure 10a shows the marginal case of  $N_0 = 5.84$ , in which the curve is tangent to the  $x$ -axis. For  $N_0 > 5.84$ , the interpolated curve goes negative around  $x = 0$ . If non-negativity is imposed, we expect that there will appear an interval around  $x = 0$  where  $\phi \equiv 0$ . Let such an interval be  $[-x_s, x_s]$ , considering the symmetry of the problem.

Because the energy principle does not hold in  $[-x_s, x_s]$ , this region is disconnected into two intervals  $[-2, -x_s]$  and  $[x_s, 2]$ , and can be treated separately. Further, the boundary conditions at  $\pm x_s$  must be re-considered. For example, at  $x_s$ , instead of requiring  $\phi''(x_s) = \phi'''(x_s) = 0$  [equations (39)–(44)], we have to impose  $\phi(x_s) = \phi'(x_s) = 0$  in order for the solution to be connected smoothly to the region  $\phi \equiv 0$ . The location of  $x_s$  is determined by requiring  $\phi''(x_s) = 0$ . Generally,  $\phi'''(x_s + 0) \neq 0$ , which means that  $\phi'''$  has a jump at  $x_s$ , and  $\phi''''$  behaves as a delta function. This delta function represents the force from the bottom of the vessel. Figure 10b shows the solution for  $N_0 = 20$ . The value of  $x_s$  is determined to be 0.354.

The situation becomes more complicated if the interpolated curve goes negative at the central part of an interval. The correct solution will then have a central sub-interval where  $\phi \equiv 0$ , and two sub-intervals at the ends with  $\phi > 0$ . However, there is no a priori way to partition the  $N_i$  values to these two sub-intervals. From the standpoint of variational formulation, the partitioning depends on the history of the system, namely how the partition was made when the curve first touched the bottom. As a matter of fact, any partitioning of  $N_i$  gives a local minimum of energy. In order to uniquely fix the solution, one has to evaluate the energy and look for the global energy minimum.

In considering these complexities, we prefer to use the practical method described in the text, by manipulating the stiffness parameter ( $\sigma_i$ ) in the definition of the energy to be minimized.

### References

- Adorf, H. M. 1995, in ASP Conf. Ser. 77, *Astronomical Data Analysis Software and Systems IV* (San Francisco: ASP), 460
- Akerlof, C., Alcock, C., Allsman, R., Axelrod, T., Bennett, D. P., Cook, K. H., Freeman, K., Griest, K., et al. 1994, *ApJ*, 436, 787
- de Boor, C. 1978, *A Practical Guide to Splines* (New York: Springer Verlag)
- Hoschek, J., & Lasser, D. 1993, *Fundamentals of Computer Aided Geometric Design* (Wellesley: A.K. Peters)
- King, I. R. 1971, *PASP*, 83, 199
- Martens, P. C., Acton, L. W., & Lemen, J. R. 1995, *Sol. Phys.*, 157, 141
- Moffat, A. F. J. 1969, *A&A*, 3, 455
- Narayan, R., & Nityananda, R. 1986, *ARA&A*, 24, 127
- Piña, R. K., & Puetter, R. C. 1993, *PASP*, 105, 630
- Platais, I. 1991, *A&A*, 241, 656
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, *Numerical Recipes in Fortran* (2nd ed., Cambridge: Cambridge University Press), 77
- from Sakurai, T., & Shin, J. 2000, *Adv. Sp. Res.* 25, 1761
- Späth, H. 1995a, *One-Dimensional Spline Interpolation Algorithms* (Wellesley: A.K. Peters)
- Späth, H. 1995b, *Two-Dimensional Spline Interpolation Algorithms* (Wellesley: A.K. Peters)
- Thévenaz, P., Blu, T., & Unser, M. 2000, *IEEE Trans. Medical Imaging*, 19, 739
- Tody, D. 1986, in *Proc. SPIE*, 627, *Instrumentation in Astronomy VI*, 733
- Walker, G. A. H., Walker, A. R., Racine, R., Fletcher, J. M., & McClure, R. D. 1994, *PASP*, 106, 356