STUDIA UNIV. "BABEŞ–BOLYAI", MATHEMATICA, Volume **XLVIII**, Number 4, December 2003

INTERPOLATING ON SOME NODES OF A GIVEN TRIANGLE

TEODORA CĂTINAŞ

Abstract. We consider the interpolation problem for some data on several nodes of a given triangle. We show that an interpolant may be found by dividing the initial problem into two subproblems, each one with some fewer nodes. The main result is given in Theorem 5.

1. Introduction

We are interested in interpolation on certain nodes on a given triangle, using the generalized Newton algorithm [1], [2]. This algorithm enables us to divide the interpolation problem into two smaller subproblems.

We shall recall first some known results. Denote by $\Pi_k(\mathbb{R}^s)$ the space of polynomials in s variables and of degree at most k and by $\#(A)$ the cardinal of a set A.

Theorem 1. *(Gasca-Maeztu) see* [2]. *Let N be a set of $\frac{1}{2}(k+1)(k+2)$ nodes in $\mathbb{R}^s$, where $s \geq 2$. Suppose that there exist the hyperplanes H_0, $H_1, ...,$ H_k in $\mathbb{R}^s$ such that*

a) $N \subset H_0 \cup H \cup ... \cup H_k$;

b) $\#(N \cap H_i) = i + 1, \quad 0 \leq i \leq k$.

Then arbitrary data on N can be interpolated by elements of $\Pi_k(\mathbb{R}^s)$.

The previous result generalizes the following theorem of Micchelli:

Theorem 2. *see* [1]. *Interpolation of arbitrary data by an element of $\Pi_m(\mathbb{R}^2)$ is uniquely possible on a set N of $\frac{1}{2}(m+1)(m+2)$ nodes if there exist $m+1$ lines $L_0, L_1, ..., L_m$ whose union contains N and that have the property that each L_i contains exactly $i+1$ nodes, $i = 0, ..., m$.*

Next we present the Newton algorithm and its generalization (see [1] and [2]).

Algorithm 3. *(The Newton algorithm for univariate polynomial interpolation). Let g be a polynomial that interpolates a function f at the distinct nodes $x_1, ..., x_n$ and let $h = \prod_{i=1}^{n}(x - x_i)$. Then for suitable c, $g + ch$ will interpolate f at $x_1, ..., x_n, x_{n+1}$, where x_{n+1} is a new node. The algorithm is applied repeatedly, starting with $n = 1$. The polynomial g can be of degree $n-1$, but this is not necessarily.*

Algorithm 4. *(The generalized Newton algorithm). Let X be an arbitrary linear space. Let g be a function (not necessarily a polynomial) that interpolates the given function $f : X \to \mathbb{R}$ at the distinct nodes $x_1, ..., x_n$. Let x_{n+1} be a new node. We require a function h that takes the value 0 at $x_1, ..., x_n$, but has a nonzero value at x_{n+1}. For an appropriate value of c, $g + ch$ will interpolate f at $x_1, ..., x_n, x_{n+1}$.*

Received by the editors: 11.02.2003.

At the next level of generalization, we replace $\{x_1, ..., x_n\}$ by any set of nodes N, which needs not be finite. We assume that g interpolates f on N (in symbols, $g|N = f|N$). Let y be a new node, $y \notin N$. We require a functional h such that $h|N = 0$ and $h(y) \neq 0$. We may assume $h(y) = 1$. Then $g + f(y)h$ interpolates f on $N \cup \{y\}$.

In a further level of generalization we use $g+rh$ as an interpolant, but permit r to be a function more general than simply a constant. We use the notation

$$Z = \{x \in X : h(x) = 0\}.$$

Consider a set N of nodes, let g interpolate f on $N \cap Z(h)$ and r interpolate $(f-g)/h$ on $N \backslash Z(h)$. Then $g + rh$ interpolates f on N.

As pointed out in [1] and [2], this last generalization of the Newton algorithm is successfully applied in Theorem 2.

An immediate conclusion is that this abstract version of the Newton algorithm enables the dividing of an interpolation problem into two smaller subproblems, where smallness refers to the number of interpolation conditions.

2. **Interpolating on some nodes of a given triangle**

Let $f : T_h \to \mathbb{R}$ be a function defined on the triangle

$$T_h = \left\{(x, y) \in \mathbb{R}^2 : 0 \le x \le h,\, 0 \le y \le h,\, x + y \le h\right\}, \quad h \in \mathbb{N}^*. \tag{1}$$

Let $N = \{X_1, X_2, X_3, X_4, X_5, X_6\}$ be a set of six nodes situated on the edges of the triangle T_h, where $X_1(0, \frac{h}{3})$, $X_2(0, \frac{2h}{3})$, $X_3(\frac{h}{4}, 0)$, $X_4(\frac{h}{2}, 0)$, $X_5(\frac{3h}{4}, 0)$, $X_6(\frac{h}{2}, \frac{h}{2})$. We consider the Lagrange interpolation functionals

$$\Lambda_L = \{\lambda_i(f) \,|\, \lambda_i(f) = f(x_i),\ 1 \le i \le 6\}.$$

The problem we deal with here is to

$$\text{find } r \in \Pi_2(\mathbb{R}^2) \text{ such that } r \text{ interpolates } f \text{ with regard to } \Lambda_L. \tag{2}$$

We find an answer in the following way. Let L_0, L_1, L_2 denote the lines of the triangle, such that $\{X_6\} \subset L_0$, $\{X_1, X_2\} \subset L_1$, $\{X_3, X_4, X_5\} \subset L_2$. The problem here satisfies the hypothesis of the Micchelli's theorem. Therefore, this result assures that there exists an interpolant in $\Pi_2(\mathbb{R}^2)$ for f with regard to Λ_L and this interpolant is unique.

Next our purpose is to find this interpolant. For doing this we use the generalized Newton algorithm.

Let l_2 denote an element of $\Pi_1(\mathbb{R}^2)$ whose zero set is the line L_2,

$$Z(l_2) = \{(x, y) : l_2(x, y) = 0\} = L_2.$$

We have $N \cap L_2 = \{X_3, X_4, X_5\}$, $l_2(x, y) = y$.

Let $p_2 \in \Pi_2(\mathbb{R}^2)$ interpolate f on $N \cap L_2 = \{X_3, X_4, X_5\}$. Therefore, p_2 has the form

$$p_2(x, y) = a_0 x^2 + b_0 x + c_0,$$

where a_0, b_0 and c_0 can be determined from the interpolation conditions:

$$\begin{cases} p_2(\frac{h}{4}, 0) = f(\frac{h}{4}, 0) \\ p_2(\frac{h}{2}, 0) = f(\frac{h}{2}, 0) \\ p_2(\frac{3h}{4}, 0) = f(\frac{3h}{4}, 0). \end{cases} \tag{3}$$

Its expression is

$$p_2(x,y) = (\tfrac{8}{h^2}x^2 - \tfrac{6}{h}x + 1)f(\tfrac{3h}{4},0) + (\tfrac{8}{h^2}x^2 - \tfrac{10}{h}x + 3)f(\tfrac{h}{4},0) + (-\tfrac{16}{h^2}x^2 + \tfrac{16}{h}x - 3)f(\tfrac{h}{2},0).$$

Let $q_1 \in \Pi_1(\mathbb{R}^2)$ interpolate $(f - p_2)/l_2$ on $N \backslash Z(l_2) = \{X_1, X_2, X_6\}$. Therefore, q_1 has the form

$$q_1(x,y) = a_1 x + b_1 y + c_1,$$

where a_1, b_1 and c_1 can be determined from the interpolation conditions:

$$\begin{cases} q_1(0,\frac{h}{3}) = \frac{f(0,\frac{h}{3})-p_2(0,\frac{h}{3})}{\frac{h}{3}} \\ q_1(0,\frac{2h}{3}) = \frac{f(0,\frac{2h}{3})-p_2(0,\frac{2h}{3})}{\frac{2h}{3}} \\ q_1(\frac{h}{2},\frac{h}{2}) = \frac{f(\frac{h}{2},\frac{h}{2})-p_2(\frac{h}{2},\frac{h}{2})}{\frac{h}{2}}. \end{cases} \tag{4}$$

According to the generalized Newton algorithm we have that $r = p_2 + l_2 q_1$ interpolates f on N. So the interpolation problem is divided into two smaller subproblems, with fewer interpolation conditions. The subproblems involve the determination of p_2 and q_1, each regarding three interpolation conditions.

Since r obeys the interpolation conditions we obtain that r solves the interpolation problem on N.

The problem becomes easier to solve if we apply twice the generalized Newton algorithm. We have to find an interpolant for q_1 on the set $M := \{X_1,\ X_2,\ X_6\}$. Let l_1 denote an element of $\Pi_1(\mathbb{R}^2)$ whose zero set is the line L_1,

$$Z(l_1) = \{(x,y) : l_1(x,y) = 0\} = L_1.$$

We have $M \cap L_1 = \{X_1,\ X_2\}$, $l_1(x,y) = x$.

Let $p_1 \in \Pi_1(\mathbb{R}^2)$ interpolate q_1 on $M \cap L_1 = \{X_1,\ X_2\}$. Therefore, p_1 has the form

$$p_1(x,y) = a_2 y + b_2,$$

where a_2 and b_2 can be determined from the interpolation conditions:

$$\begin{cases} p_1(0,\frac{h}{3}) = q_1(0,\frac{h}{3}) \\ p_1(0,\frac{2h}{3}) = q_1(0,\frac{2h}{3}). \end{cases} \tag{5}$$

By (4), (5) becomes

$$\begin{cases} p_1(0,\frac{h}{3}) = \frac{f(0,\frac{h}{3})-p_2(0,\frac{h}{3})}{\frac{h}{3}} \\ p_1(0,\frac{2h}{3}) = \frac{f(0,\frac{2h}{3})-p_2(0,\frac{2h}{3})}{\frac{2h}{3}}. \end{cases} \tag{6}$$

Its expression is

$$p_1(x,y) = \tfrac{3}{2h}(\tfrac{3}{h}y - 1)f(0,\tfrac{2h}{3}) + \tfrac{3}{h}(-\tfrac{3}{h}y + 2)f(0,\tfrac{h}{3}) + \tfrac{3}{2h}(-\tfrac{3}{h}y + 1)p_2(0,\tfrac{2h}{3}) + \tfrac{3}{h}(\tfrac{3}{h}y - 2)p_2(0,\tfrac{h}{3}).$$

Let $q_0 \in \Pi_0(\mathbb{R}^2)$ interpolate $(q_1 - p_1)/l_1$ on $M \backslash Z(l_1) = \{X_6\}$. Therefore, q_0 is constant:

$$q_0 = \frac{q_1(\frac{h}{2},\frac{h}{2}) - p_1(\frac{h}{2},\frac{h}{2})}{\frac{h}{2}}.$$

According to the generalized Newton algorithm we have that $p_1 + l_1 q_0$ interpolates q_1 on $M = \{X_1,\ X_2,\ X_6\}$. So the interpolation problem here involves the determination of p_1, regarding two interpolation conditions, the initial interpolation problem becoming much easier to solve. The polynomial $p_1 + l_1 q_0$ verifies the interpolation conditions so it interpolates $q_1$on M. We conclude with the following result.

Theorem 5. *The initial interpolation problem (2) on N is solved by*

$$r = p_2 + l_2 q_1 = p_2 + l_2(p_1 + l_1 q_0).$$

We shall illustrate the above theory with two practical examples. Consider $h = 10$ in (1), $f_1 : T_{10} \to \mathbb{R}$, $f_1(x, y) = x^2 + y^2$ and $f_2 : T_{10} \to \mathbb{R}$, $f_2(x, y) = \sqrt{x^2 + y^2}$. Consider r_i the interpolant of f_i, $i = 1, 2$. Figures 1 and 2 display the error $|f_i(x, y) - r_i(x, y)|$, plotted in Matlab.

3. The generalized Newton algorithm for linear functionals

As pointed out in [2], the generalized Newton algorithm can be applied not only for point-evaluation functionals, but for arbitrary linear functionals. Consider a linear space E and Φ_1, $\Phi_2, \ldots$ some linear functionals defined on E. Let f be an element of E to be interpolated. We assume that an element g is available in E such that $\Phi_i(g) = \Phi_i(f)$ for $1 \leq i \leq n$. Next, select h in E so that $\Phi_i(h) = 0$ for $1 \leq i \leq n$ and $\Phi_{n+1}(h) = 1$. The new interpolant will be of the form $g + ch$, where $c = \Phi_{n+1}(f)$.

We illustrate this by solving a problem proposed in [2].

Problem 6. *Find $p \in \Pi_3(\mathbb{R})$ such that $p(0) = 3$, $p'(1) = 4$, $\int_0^1 p(x)dx = 5$ and $\int_0^1 x^2 p(x)dx = 6$.*

Proof. We use the generalized Newton algorithm for linear functionals. We consider the linear functionals defined by: $\Phi_1(f) = f(0)$, $\Phi_2(f) = \int_0^1 f(x)dx$, $\Phi_3(f) = \int_0^1 x^2 f(x)dx$, $\Phi_4(f) = f'(1)$, for some given f.

We assume that there exists g such that

$$\begin{aligned} &\Phi_i(g) = \Phi_i(p), \qquad i = 1, 2, 3 \\ &\Phi_4(g) = 0. \end{aligned} \tag{7}$$

Select now h such that

$$\begin{aligned} &\Phi_i(h) = 0, \quad i = 1, 2, 3, \\ &\Phi_4(h) = 1. \end{aligned} \tag{8}$$

The interpolant of p is $g + ch$, where $c = \Phi_4(f) = 4$. Therefore we have to find the interpolant of p, $r := g + 4h$ from $\Pi_3(\mathbb{R})$. We do this taking into account (7) and (8). We have $g \in \Pi_3(\mathbb{R})$ and $h \in \Pi_3(\mathbb{R})$ so g and h have the following expressions

$$\begin{aligned} g(x) &= a_1 x^3 + b_1 x^2 + c_1 x + d_1, \\ h(x) &= a_2 x^3 + b_2 x^2 + c_2 x + d_2, \end{aligned}$$

where a_1, b_1, c_1, d_1, a_2, b_2, c_2, $d_2 \in \mathbb{R}$, and moreover $a_1^2 + a_2^2 \neq 0$.

Solving the systems (7) and (8) we obtain the polynomials

$$\begin{aligned} g(x) &= -\tfrac{984}{5} x^3 + 366 x^2 - \tfrac{708}{5} x + 3, \\ h(x) &= \frac{6}{5} x^3 - \frac{3}{2} x^2 + \frac{2}{5} x. \end{aligned}$$

Therefore, the interpolant of p is

$$r(x) = -192x^3 + 360x^2 - 140x + 3 \in \Pi_3(\mathbb{R}).$$

$\square$

References

[1] E. W. Cheney, *Multivariate Approximation Theory*, Selected Topics, CBMS51, SIAM, Philadelphia, Pennsylvania, 1986.

[2] W. Cheney, W. Light, *A Course in Approximation Theory*, Brooks/Cole Publishing Company, Pacific Grove, 2000.

[3] Gh. Coman, *Numerical Analysis*, Ed. Libris, Cluj-Napoca, 1995 (in Romanian).

[4] Gh. Coman, *Numerical multivariate approximation operator*, "Tiberiu Popoviciu" Itinerant Seminar of Functional Equations, Approximation and Convexity, Cluj-Napoca, May 22-26, 2001.

[5] Gh. Coman, K. Böhmer, *Blending interpolation schemes on triangle with error bounds*, Lecture Notes in Mathematics, no. **571**, Springer-Verlag, Berlin, 1977, 14-37.

[6] M. Gasca and T. Sauer, *Multivariate polynomial interpolation*, Adv. Comput. Math., **12** (2000), 377-410.

[7] M. Gasca and T. Sauer, *On the history of multivariate polynomial interpolation*, J. Comput. Appl. Math., **122** (2000), 23-35.

[8] R. A. Lorentz, *Multivariate Hermite interpolation by algebric polynomials: A survey*, J. Comput. Appl. Math. **122** (2000), 167-201.

[9] T. Sauer, *Polynomial interpolation of minimal degree*, Numer. Math., **78** (1997), 59-85.

[10] T. Sauer and Y. Xu, *On multivariate Lagrange interpolation,* Math. Comp., **64** (1995), 1147-1170.

"Babeş-Bolyai" University, Faculty of Mathematics and Computer Science, Str. Kogălniceanu 1, 3400 Cluj-Napoca, Romania
E-mail address: `tgulea@math.ubbcluj.ro`