RATIONAL FUNCTION METHOD OF INTERPOLATION

by

Gerald I. Kerley

ABSTRACT

A new method for interpolating functions of one and two variables from tables is presented. The technique uses a ratio of polynomials to represent the function on an interpolation interval. A quadratic formula is used to estimate derivatives at the tabular points. The method is particularly useful for functions which have rapid or even discontinuous changes in the first derivative at certain points.

I. INTRODUCTION

In this report, we present new algorithms for interpolating functions of one and two independent variables from tabulations of discrete points.

This study began with the development of a computer-based library of equation of state tables, called Sesame. Although numerous interpolation methods already exist, we found that it was necessary to develop a new technique. Thermodynamic quantities present special interpolation problems because they are not smooth functions when phase transitions are present. Familiar methods, such as spline[1] and Lagrangian[2] interpolation, produce unphysical oscillations in the neighborhood of the phase transitions. The interpolated surface can have regions of thermodynamic instability, resulting in imaginary sound speeds and negative heat capacities.

We have found a simple, but effective, way to deal with discontinuities in the derivative of a function. Our technique is also computationally efficient, and it is competitive with other methods when speed is an important factor.

In Sec. II, we discuss the application of our method to functions of one independent variable. The results are summarized in Eqs. (11), (12), and (13), and a FORTRAN program is given in Appendix A. In Sec. III, we present three examples and compare with other techniques. In Sec. IV, we discuss a simple method for functions of two variables. An alternative, but more complicated, algorithm for bivariant functions is described in Appendix B.

No interpolation scheme is foolproof. As with other schemes, our technique must be used with some care. In particular, attention should be given to choosing the mesh on which to tabulate the functions. However, we believe that our method does not have some of the pitfalls which are common to many other techniques, and that it can be applied to a wide class of problems.

II. FUNCTIONS OF ONE VARIABLE

Consider a function f(x), tabulated at N points, $x_1 < x_2 < \cdots < x_N$. We assume that $x_i \neq x_{i+1}$ and that f(x) is continuous but not necessarily smooth. We want to estimate the function f(x) from the tabular data.

Define the quantities

$$\left.\begin{aligned} \Delta_i &= x_{i+1} - x_i \quad , \\ S_i &= \left(f_{i+1} - f_i\right) \Delta_i^{-1} \quad , \\ t_i &= (df/dx)_i \quad . \end{aligned}\right\} \tag{1}$$

In some applications, the derivatives $\{t_i\}$ may be specified along with the function, but in most cases they must be estimated. For example, a simple rule for estimating t_i is to fit a quadratic polynomial through the points f_{i-1}, f_i, and f_{i+1}. The result is

$$t_i = \left(\Delta_{i-1}\, S_i + \Delta_i\, S_{i-1}\right)\Big/\left(\Delta_{i-1} + \Delta_i\right) \quad . \tag{2}$$

To estimate the function f(x) over the interval $x_i \leq x \leq x_{i+1}$, many methods use the cubic polynomial defined by the four quantities f_i, f_{i+1}, t_i, t_{i+1}. The result is

$$\left.\begin{aligned}
p(x) &= f_i + t_i\left(x - x_i\right) + b_i\left(x - x_i\right)^2 + c_i\left(x - x_i\right)^3 \quad , \\
b_i &= \left(3S_i - t_{i+1} - 2t_i\right) \Delta_i^{-2} \quad , \\
c_i &= \left(t_{i+1} - 2S_i + t_i\right) \Delta_i^{-2} \quad .
\end{aligned}\right\} \tag{3}$$

An advantage of this procedure is that it gives an estimate of the function which has continuous derivatives everywhere. However, it can also give spurious oscillations. An example is shown in Fig. 1, where we display the function $f(x) = .2x + .8x^6$. On the interval $0 \leqslant x \leqslant 1$, the cubic polynomial gives $p(x) = .2x - 2.4x^2 + 3.2x^3$. This approximation is not monotonic, like the true function, and its second derivative has the wrong sign at $x = 0$.

The above example is a special case of the condition $t_i < S_i < t_{i+1}$. In the absence of additional information, it is reasonable to assume that the first derivative of the function is monotonic. Hence f(x) should be bounded as follows.

$$\left.\begin{aligned}
f(x) &\geqslant f_\ell(x) = f_i + t_i\left(x - x_i\right) \quad , \\
f(x) &\geqslant f_u(x) = f_{i+1} + t_{i+1}\left(x - x_{i+1}\right) \quad .
\end{aligned}\right\} \tag{4}$$

Equation (3) does not necessarily satisfy these conditions, because a cubic polynomial cannot "turn a corner" abruptly.

Suppose we try to estimate the function with a ratio of polynomials.

$$\left.\begin{aligned}
r(x) &= \frac{f_i + A_i\left(x - x_i\right) + B_i\left(x - x_i\right)^2}{1 + C_i\left(x - x_i\right)} \quad , \\
C_i &= \left[\left(S_i - t_i\right)/\left(t_{i+1} - S_i\right) - 1\right] \Delta_i^{-1} \quad , \\
B_i &= S_i C_i + \left(S_i - t_i\right) \Delta_i^{-1} \quad , \\
A_i &= t_i + f_i\, C_i \quad .
\end{aligned}\right\} \tag{5}$$

When Eq. (5) is applied to the example discussed above, we obtain the result shown in Fig. 2. The new approximation is much more satisfactory for this case. In general, it is found that the rational function satisfies the conditions

$$r(x) = f_\ell(x) + \frac{\left(S_i - t_i\right)^2\left(x - x_i\right)^2}{\left(t_{i+1} - S_i\right)\left(x_{i+1} - x\right) + \left(S_i - t_i\right)\left(x - x_i\right)} > f_\ell(x) \quad ,$$

$$r(x) = f_u(x) + \frac{\left(t_{i+1} - S_i\right)^2\left(x_{i+1} - x\right)^2}{\left(t_{i+1} - S_i\right)\left(x_{i+1} - x\right) + \left(S_i - t_i\right)\left(x - x_i\right)} > f_u(x) \quad ,$$

as required.

Equation (5) also gives satisfactory results for the case $t_i > S_i > t_{i+1}$, where the derivative is monotonic. For the other possibilities, however, we find that $C_i < -\Delta_i^{-1}$, and the denominator of Eq. (5) vanishes at a point within the interval. These deficiencies can be corrected by modifying the rational function as follows.

$$\left.\begin{aligned}
r(x) &= \frac{f_i + A_i\left(x - x_i\right) + B_i\left(x - x_i\right)^2 + D_i\left(x - x_i\right)^3}{1 + C_i\left(x - x_i\right)} \quad , \\
C_i &= \left[\left|\left(S_i - t_i\right)\Big/\left(t_{i+1} - S_i\right)\right| - 1\right] \Delta_i^{-1} \quad , \\
D_i &= \left[t_i - S_i + \left(t_{i+1} - S_i\right)\left(1 + C_i\Delta_i\right)\right] \Delta_i^{-2} \quad , \\
B_i &= S_iC_i + \left(S_i - t_i\right) \Delta_i^{-1} - D_i\Delta_i \quad , \\
A_i &= t_i + f_iC_i \quad .
\end{aligned}\right\} \qquad (6)$$

Equation (6) reduces to Eq. (5) for $t_i < S_i < t_{i+1}$ and $t_i > S_i > t_{i+1}$. It is interesting to examine two limiting cases. We find that

$$r(x) = f_i + S_i\left(x - x_i\right), \text{ when } t_i = S_i \text{ or } t_{i+1} = S_i \quad . \qquad (7)$$

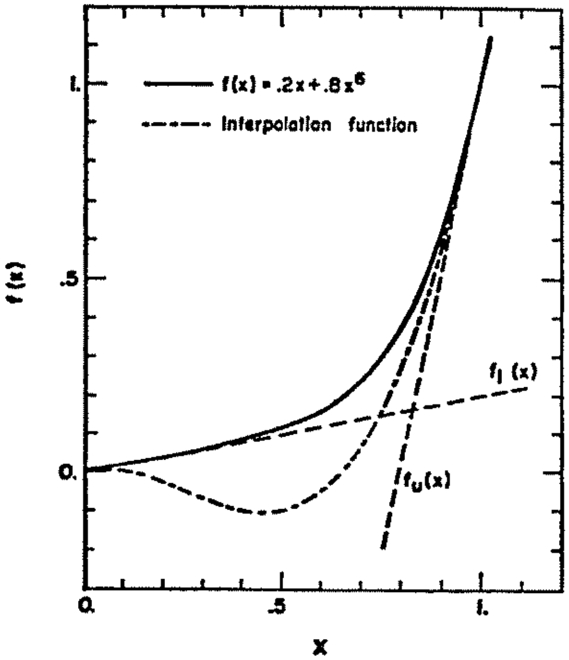

Fig. 1.
Comparison of function $f(x) = .2x + .8x^6$ with an estimate using a cubic polynomial, Eq. (3).

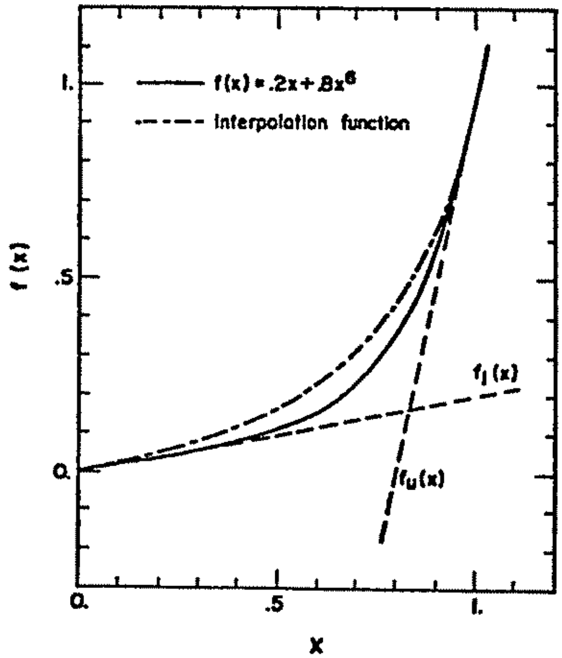

Fig. 2.
Comparison of function $f(x) = .2x + .8x^6$ with an estimate using a ratio of polynomials, Eq. (5).

In these cases, the derivative of the function is found to be discontinuous at either x_{i+1} or x_i, respectively. Under all other conditions, our interpolation function is smooth to first order.

We have tested Eq. (6) on many different functions, tabulated on a number of different grids. We compared the results obtained with several algorithms for evaluating the derivatives, t_i, including the rule given by Akima.[3] In our tests, Eq. (2) gave the best results in most cases. This rule is also attractive for its simplicity and computational speed. Although our study was not exhaustive, we believe that Eq. (2) should be adequate for most purposes.

To complete the specifications of our interpolation algorithm, it is necessary to propose a method for estimating the derivatives at the end points, x_1 and x_N. To compute t_1, the simplest procedure is to use the quadratic polynomial fit through the points f_1, f_2, and f_3. The result is found to be

$$t_1 = 2S_1 - t_2 \quad , \tag{8}$$

where t_2 is given by Eq. (2). Unfortunately, this rule is not always satisfactory.

Once again, we consider the case $f(x) = .2x + .8x^6$. A quadratic polynomial, fit at $x = 0$, $x = .5$, and $x = 1$, is $p(x) = -.55x + 1.55x^2$. The result is shown in Fig. 3. As might be expected, the polynomial is a poor fit to the function. However, the real problem is that it gives the wrong <u>sign</u> for the derivative at $x = 0$. If this algorithm were applied to a table of internal energy vs temperature, this effect would predict a negative heat capacity, which is unphysical.

To eliminate this problem, we require t_1 to have the same sign as S_1 (hence $t_1 \cdot S_1 \geq 0$). This constraint can be achieved by modifying the calculation of the slope t_2. We proceed as follows.

$$\left.\begin{aligned} & t_2 = \left(\Delta_1 S_2 + \Delta_2 S_1\right)\Big/\left(\Delta_1 + \Delta_2\right) \quad . \quad \text{If} \\ & 2S_1^{\,2} - S_1 \cdot t_2 < 0,\ t_2 = 2S_1 \quad . \\ & t_1 = 2S_1 - t_2 \quad . \end{aligned}\right\} \tag{9}$$

Our estimate of the derivative at x_N is

$$t_N = 2\ S_{N-1} - t_{N-1} \quad . \tag{10}$$

To prohibit unphysical behavior, the derivative t_{N-1} can be modified by the same procedure discussed above, for t_2. However, our present applications do not require such a refinement.

Finally, we summarize the equations developed in this section and write them in a form suitable for efficient computation. For $i > 1$, $i < N - 1$,

$$\left.\begin{aligned} & C_1 = \left(S_i - S_{i-1}\right)\Big/\left(\Delta_i + \Delta_{i-1}\right) \quad , \text{ if} \\ & \quad i = 2 \text{ and } S_{i-1}\left(S_{i-1} - \Delta_{i-1}\ C_1\right) \leq 0,\ C_1 = \left(S_i - 2S_{i-1}\right)\Big/\Delta_i \quad , \\ & C_2 = \left(S_{i+1} - S_i\right)\Big/\left(\Delta_{i+1} + \Delta_i\right) \quad , \end{aligned}\right\} \tag{11}$$

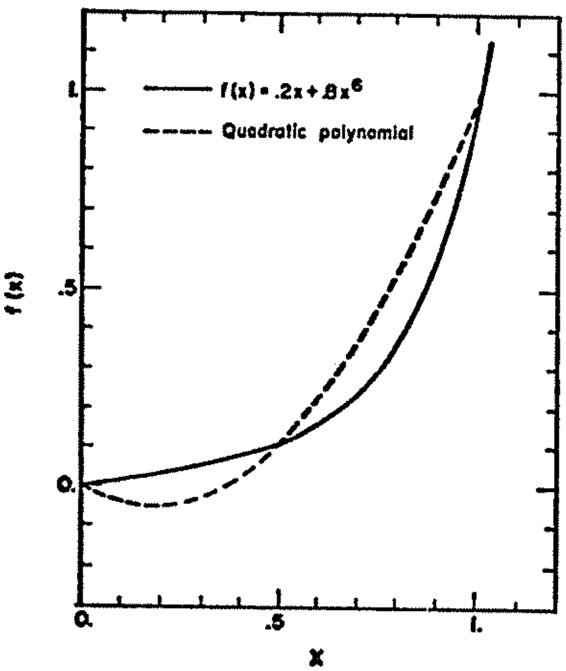

Fig. 3.
Comparison of function $f(x) = .2x + .8x^6$ with a quadratic polynomial used to estimate the derivative at $x = 0$.

$$\left.\begin{aligned}\mu_1 &= \left|C_2\left(x_{i+1} - x\right)\right| \;, \\ \mu_2 &= \left|C_1\left(x - x_i\right)\right| \;, \\ r(x) &= f_i + \left(x - x_i\right)\left[S_i - \frac{C_1\mu_1 + C_2\mu_2}{\mu_1 + \mu_2}\left(x_{i+1} - x\right)\right] \;.\end{aligned}\right\} \qquad (11)$$

For $i = 1$,

$$\left.\begin{aligned}& C_2 = \left(S_2 - S_1\right)\Big/\left(\Delta_2 + \Delta_1\right) \;, \text{ if} \\ & S_1\left(S_1 - \Delta_1\, C_2\right) \leq 0,\; C_2 = S_1/\Delta_1 \;, \\ & r(x) = f_1 + \left(x - x_1\right)\left[S_1 - C_2\left(x_2 - x\right)\right] \;.\end{aligned}\right\} \qquad (12)$$

For $i = N - 1$,

$$\left.\begin{aligned}& C_1 = \left(S_{N-1} - S_{N-2}\right)\Big/\left(\Delta_{N-1} + \Delta_{N-2}\right) \\ & r(x) = f_{N-1} + \left(x - x_{N-1}\right)\left[S_{N-1} - C_1\left(x_N - x\right)\right] \;.\end{aligned}\right\} \qquad (13)$$

These equations have been coded into a simple FORTRAN function subprogram, called FRF1. A listing is given in Appendix A.

III. EXAMPLES

In Fig. 4, we demonstrate what happens when several standard interpolation schemes are applied to a function with a discontinuous derivative. The function depicted is the intersection of two straight lines, tabulated at five points. This problem is a trivial one, and our algorithm gives the exact solution. However, we find that the spline[1] and Lagrangian[2] methods introduce spurious oscillations in the neighborhood of the intersection. The spline method gives continuous derivatives. The Lagrangian method does not. The method of Akima[3] is much better in this problem. However, it makes a significant error in the derivative near the intersection and underestimates the function itself.

As a second example, we consider the test function

$$\left.\begin{array}{ll} f(x) = 2.9154519\, x^3 & x \leqslant .5 \\ f(x) = .36443149 & .5 \leqslant x \leqslant .8 \\ f(x) = 2.9154519\, (x - .3)^3 & .8 \leqslant x \quad . \end{array}\right\} \qquad (14)$$

In Fig. 5, the function is tabulated at 8 points on the interval $0 \leqslant x \leqslant 1$, with 3 points on the interval $.5 \leqslant x \leqslant .8$. Using the rational function method, the solid curve is obtained. Next consider what happens when the 3 points on $.5 \leqslant x \leqslant .8$ are replaced by only 2 points (as shown by "x"). In this case, the interpolation algorithm does not have enough information to discern the discontinuity in slope. Hence it creates the oscillations shown by the dashed line. The example illustrates the importance of using a good mesh.

Our third example is chosen from a physical application. For many materials, the pressure on the cold curve can be represented by the expression[4]

$$\left.\begin{array}{l} P_C(\eta) = A\eta^{2/3} \left(\eta\, e^{B_r \nu} - e^{B_a \nu}\right) \; , \\ \nu = 1 - \eta^{-1/3} \quad , \end{array}\right\} \qquad (15)$$

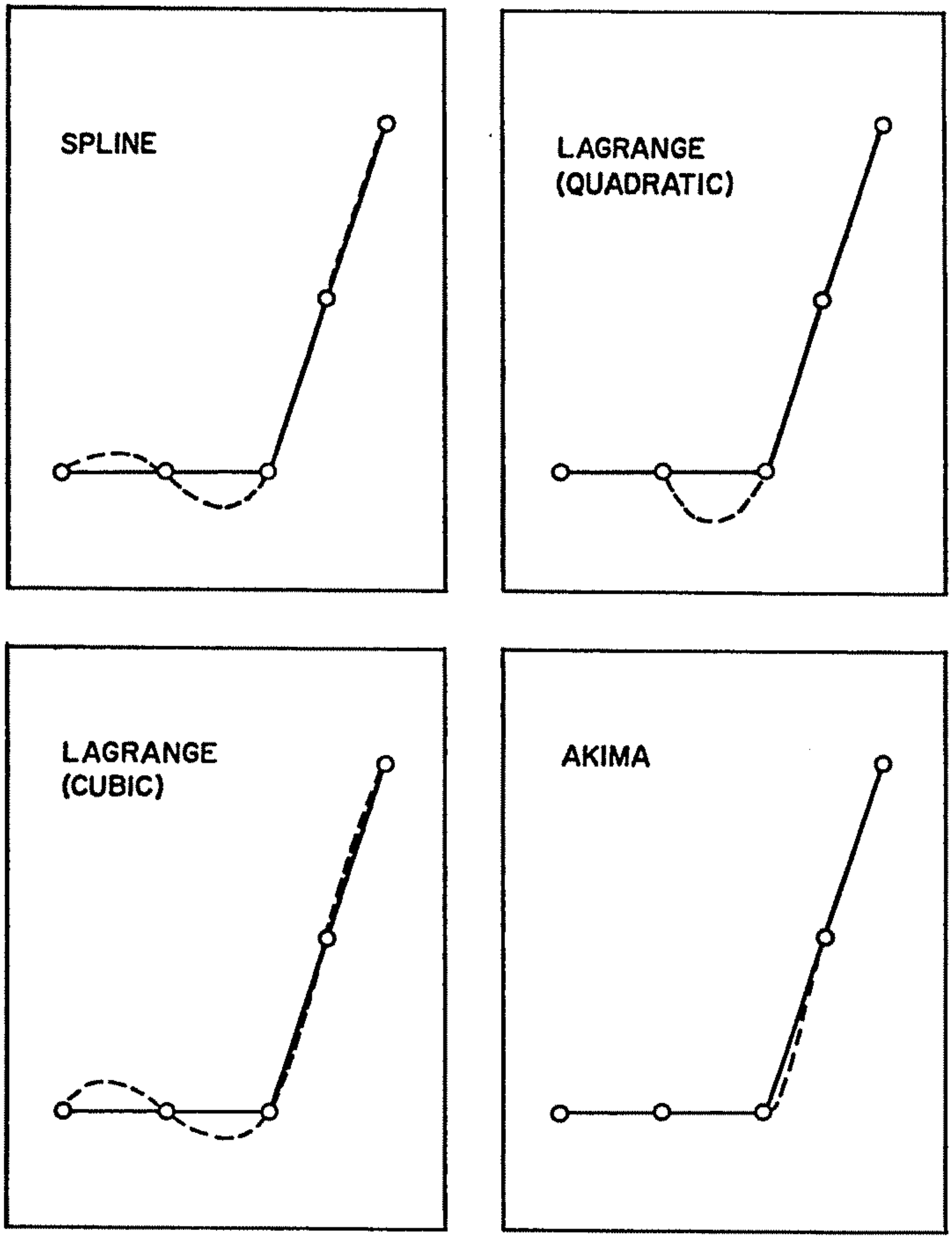

Fig. 4.
Interpolation schemes applied to the intersection of two straight lines. The tabulated points are circles.

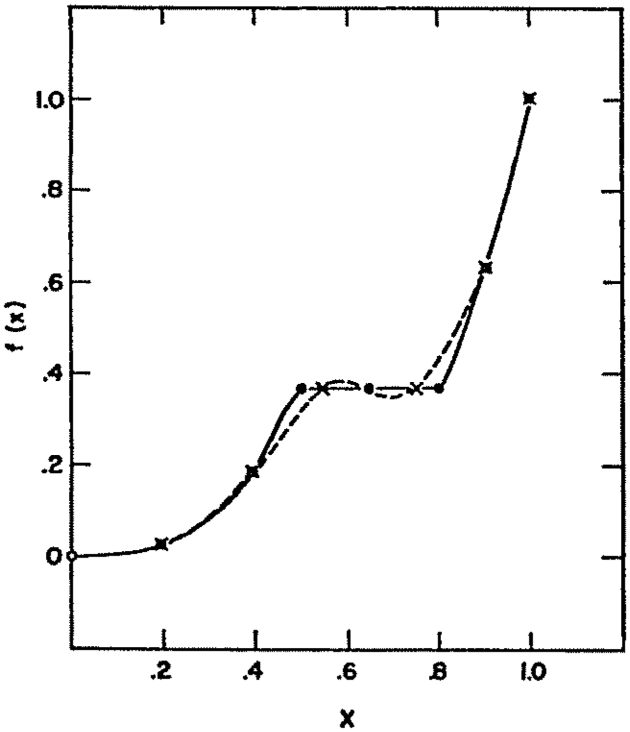

Fig. 5.
Test of rational function interpolation. The dashed curve is obtained when there are too few points on the tabulated curve.

where η is the compression (density divided by a reference density). For aluminum, we take $A = .292$ (Mbar), $B_r = 4.304$, $B_a = -.054$.[4] In many applications, functions similar to Eq. (15) must be tabulated over a large range of the independent and dependent variables. Hence, it is necessary to choose the mesh carefully.

In our studies, we have found that excellent results can be obtained by choosing an exponential mesh in the dependent variable. For example, select a mesh $\eta_1 < \eta_2 < \cdots < \eta_N$ such that

$$P_C(\eta_{i+1}) = w\, P_C(\eta_i) \quad , \tag{16}$$

where w is a constant. In Table I we give such a mesh for our sample function. Only 12 points have been used to tabulate the pressure over 6 decades.

In Table II, we compare the results when several algorithms are used to interpolate on the data in Table I. Because we have chosen the mesh carefully, all five methods do fairly well. The spline method is best in this test, which involves a smooth function. Our rational function method is comparable to the cubic

TABLE I

MESH USED IN TABULATING EQUATION (15)

η	P_C(Mbar)
1.	0.
1.01372	.01
1.04189	.0316228
1.12060	.1
1.31065	.316228
1.69626	1.
2.38651	3.16228
3.55549	10.
5.51261	31.6228
8.81618	100.
14.4799	316.228
24.3631	1000.

Lagrange method in accuracy, giving errors on the order of .1%. The quadratic Lagrange and Akima methods give larger errors, on the order of 1.0%.

In summary, these examples demonstrate that our method is competitive with other schemes for smooth functions, and that it is significantly better for functions which have discontinuities or rapid changes in the derivative.

IV. FUNCTIONS OF TWO VARIABLES

Next we consider a bivariant function, $F(x,y)$, tabulated on a grid of $N \times M$ points, $x_1 < x_2 < \cdots < x_N$, $y_1 < y_2 < \cdots < y_M$. An interpolation scheme can be constructed by making two successive applications of our rational function method. In general, however, the surface so defined will depend upon the order of the interpolations (the form will not be symmetric in x and y), and it will not be smooth.

TABLE II

COMPARISON OF INTERPOLATION ALGORITHMS

η	P(Mbar)	Rational Function	Spline	Lagrange (Quadratic)	Lagrange (Cubic)	Akima
1.007	.0050581	.0050586	.0050587	.0050586	.0050589	.0050472
1.03	.022313	.022315	.022316	.022313	.022316	.022364
1.08	.063243	.063226	.063245	.063181	.063247	.063482
1.22	.20380	.20361	.20380	.20314	.20382	.20524
1.50	.60892	.60825	.60894	.60503	.60919	.61377
1.87	1.4253	1.4269	1.4254	1.4131	1.4265	1.4187
2.04	1.9169	1.9154	1.9171	1.9028	1.9187	1.9330
2.2	2.4505	2.4463	2.4506	2.4407	2.4519	2.4816
3.00	6.2221	6.2189	6.2229	6.1840	6.2293	6.2777
4.00	13.750	13.769	13.751	13.678	13.764	13.661
4.25	16.151	16.165	16.154	16.065	16.171	16.149
4.50	18.768	18.770	18.771	18.681	18.791	18.888
4.75	21.603	21.591	21.605	21.526	21.625	21.833
5.00	24.658	24.637	24.660	24.601	24.676	24.940
7.20	61.380	61.391	61.385	61.232	61.441	61.795
12.0	205.90	205.90	205.93	205.80	206.04	193.60
20.0	650.07	650.10	649.97	650.10	649.46	655.70

We have developed two methods for generalizing our rational function algorithm to a function of two variables. In this section, we describe the simpler technique, which we prefer for most applications. The other approach is discussed in Appendix B.

Let R(x,y) be our estimate of the function F(x,y). First we apply our one-dimensional scheme along the grid lines. We obtain

$$\left.\begin{aligned} r_j(x) &= R\left(x,y_j\right) \quad , \\ r_i(x) &= R\left(x_i,y\right) \quad . \end{aligned}\right\} \tag{17}$$

We want to obtain $R(x,y)$ on the interval $x_i \leq x \leq x_{i+1}$, $y_j \leq y \leq y_{j+1}$. Our procedure is to weight the four functions, $r_j(x)$, $r_{j+1}(x)$, $r_i(y)$, $r_{i+1}(y)$, as follows.

$$\left.\begin{aligned} R(x,y) = {} & r_j(x)\left(1 - q_y\right) + r_{j+1}(x)\, q_y + r_i(y)\left(1 - q_x\right) + r_{i+1}(y)\, q_x \\ & -F(i,j)\left(1 - q_x\right)\left(1 - q_y\right) - F(i,j+1)\left(1 - q_x\right) q_y \\ & - F(i+1,j)\, q_x\left(1 - q_y\right) - F(i+1, j+1)\, q_x\, q_y \quad , \end{aligned}\right\} \quad (18)$$

where

$$q_x = \left(x - x_i\right)/\left(x_{i+1} - x_i\right) \quad ,$$

$$q_y = \left(y - y_j\right)/\left(y_{j+1} - y_j\right) \quad .$$

Note that Eq. (18) is symmetric in the variables x and y.

A drawback of Eq. (18) is that it does not have continuous derivatives everywhere. It can be shown that

$$\lim_{\varepsilon \to 0} \left(\frac{\partial R}{\partial y}\right)_{y = y_j + \varepsilon} \neq \lim_{\varepsilon \to 0} \left(\frac{\partial R}{\partial y}\right)_{y = y_j - \varepsilon} \quad ,$$

and that

$$\lim_{\varepsilon \to 0} \left(\frac{\partial R}{\partial x}\right)_{x = x_i + \varepsilon} \neq \lim_{\varepsilon \to 0} \left(\frac{\partial R}{\partial x}\right)_{x = x_i - \varepsilon} \quad .$$

This deficiency is corrected in our other scheme, which we describe in Appendix B. However, we have found the above method to be adequate for many applications, and we prefer it for its computational speed.

A more serious problem with the method is illustrated in Fig. 6, where we show a portion of the pressure table for a Sesame material, in the vicinity of

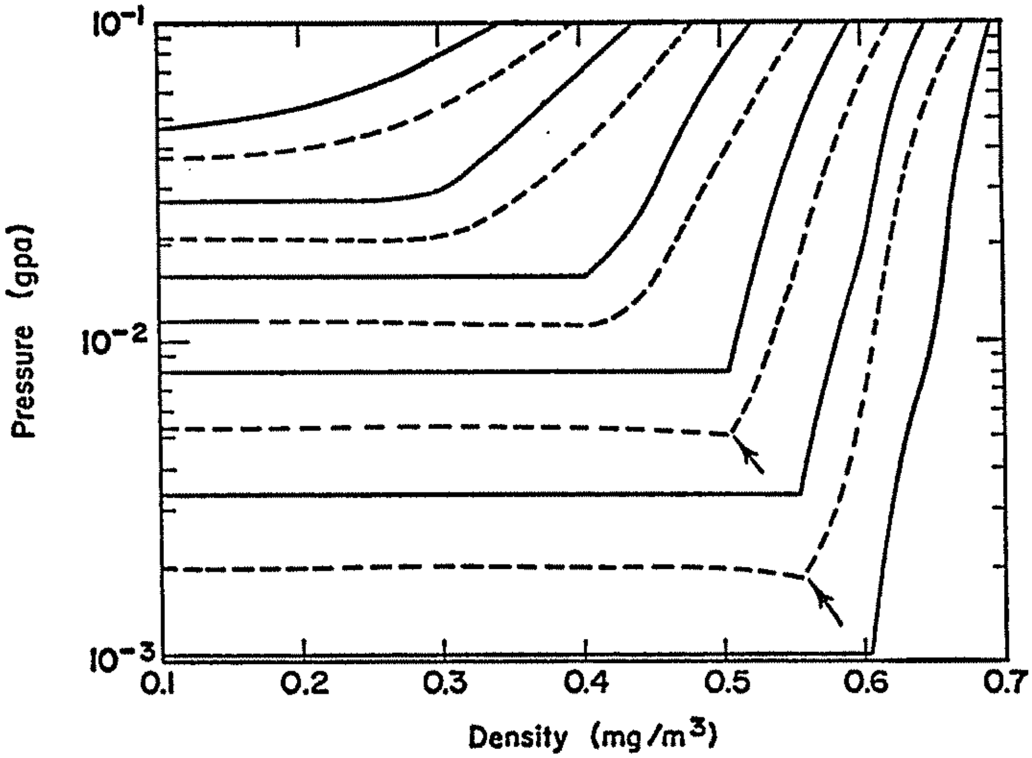

Fig. 6.
Interpolated pressure-density-temperature surface.

the vaporization boundary. The solid curves are pressure vs density isotherms (constant temperature loci), interpolated at tabulated temperature values. The dashed curves are interpolated isotherms at intermediate temperatures. Thermodynamic stability requires that the pressure be a monotonically increasing function of density. While the tabulated isotherms have the correct behavior, the intermediate ones do not, in a small region indicated by arrows. Thus far, we have been unable to develop a satisfactory scheme for bivariant functions which does not have this difficulty. Until additional work is done on the problem, it is necessary to tailor the mesh to minimize this effect.

V. CONCLUSIONS

We have presented new methods for interpolating functions of one and two variables. Our procedures offer significant advantages over standard methods, particularly when the function has discontinuities in the derivatives. Our bivariant scheme is not completely satisfactory, in that spurious oscillations are introduced in particular difficult cases. However, our techniques should be useful in a wide range of applications.

ACKNOWLEDGMENTS

The spline interpolation calculations were carried out with LASL computer routines written by Tom Jordan. Gerald T. Rood provided the Lagrangian and Akima interpolation programs. The author is grateful to Bard I. Bennett, who carried out many tests of our rational function algorithm.

REFERENCES

1. T. N. E. Greville, "Numerical Procedures for Interpolation by Spline Functions," J. SIAM, Numer. Anal. Ser. B., 1, 53-68 (1964).

2. A. Ralston, A First Course in Numerical Analysis (McGraw-Hill, New York, 1965) pp. 42-46.

3. H. Akima, "A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures," Jour. of the A.C.M. 17, 589-602 (1970).

4. J. F. Barnes, "Statistical Atom Theory and the Equation of State of Solids," Phys. Rev. 153, 269-275 (1967).

APPENDIX A

FORTRAN LISTING FOR RATIONAL FUNCTION ALGORITHM

```
      FUNCTION FRF1(X,XT,FT,N,DF)
C-----------------------------------------------------------------------
C
C    PURPOSE:         INTERPOLATE FOR A FUNCTION F(X) FROM TABLES OF
C                     F AND X.  USES RATIONAL FUNCTION METHOD WITH
C                     QUADRATIC ESTIMATE OF DERIVATIVES AT END POINTS.
C
C    ARGUMENTS:       X  (INP) - INDEPENDENT VARIABLE
C                     XT (INP) - TABLE OF INDEPENDENT VARIABLE
C                     FT (INP) - TABLE OF DEPENDENT VARIABLE
C                     N  (INP) - LENGTH OF ARRAYS XT AND FT
C                     DF (OUT) - DERIVATIVE OF FUNCTION
C                     FRF1 IS THE VALUE OF THE FUNCTION AT X
C
C    PROGRAMMER:      G. I. KERLEY, T-4.
```

```
C
C     DATE:           5 JULY 1977
C
C-----------------------------------------------------------------------
         DIMENSION XT(1),FT(1)
C     SEARCH FOR INDEX
         I = 1
         J = N
    1    IF(J-I.EQ.1) GO TO 3
         JP = .5*(J+I)
         IF(X.LT.XT(JP))GO TO 2
         I = JP
         GO TO 1
    2    J = JP
         GO TO 1
C     COMPUTE INTERPOLATION FUNCTION
    3    Q = X-XT(I)
         D = XT(I+1)-XT(I)
         R = D-Q
         S = (FT(I+1)-FT(I))/D
         IF(I.GT.1) GO TO 4
         SP = (FT(I+2)-FT(I+1))/(XT(I+2)-XT(I+1))
         C2 = (SP-S)/(XT(I+2)-XT(I))
         IF(S*(S-D*C2).LE.0.) C2=S/D
         FRF1 = FT(I)+Q*(S-R*C2)
         DF = S+(Q-R)*C2
         RETURN
    4    DM = XT(I)-XT(I-1)
         SM = (FT(I)-FT(I-1))/DM
         C1 = (S-SM)/(D+DM)
         IF(I.LT.N-1) GO TO 5
         FRF1 = FT(I)+Q*(S-R*C1)
         DF = S+(Q-R)*C1
         RETURN
    5    IF(I.GT.2) GO TO 6
         IF(SM*(SM-DM*C1).LE.0.) C1=(S-SM-SM)/D
    6    SP = (FT(I+2)-FT(I+1))/(XT(I+2)-XT(I+1))
         C2 = (SP-S)/(XT(I+2)-XT(I))
         C3 = ABS(C2*R)
         C4 = C3+ABS(C1*Q)
         IF(C4.GT.0.) C3=C3/C4
         C4 = C2+C3*(C1-C2)
         FRF1 = FT(I)+Q*(S-R*C4)
         DF = S+(Q-R)*C4+D*(C4-C2)*(1.-C3)
         RETURN
         END
```

APPENDIX B

ALTERNATE METHOD FOR A FUNCTION OF TWO INDEPENDENT VARIABLES

Let $F(i,j)$, $F_x(i,j)$, $F_y(i,j)$, and $F_{xy}(i,j)$ be the function and estimates of its x-derivative, y-derivative, and cross derivative $(\partial^2F/\partial x\partial y)$ at the point (x_i, y_j). Let $R(x,y)$ be our estimate of the function $F(x,y)$ on the interval $x_i \leqslant x \leqslant x_{i+1}$, $y_j \leqslant y \leqslant y_{j+1}$.

We define the quantities

$$\left.\begin{aligned}
\Delta_x &= x_{i+1} - x_i \quad , \\
q_x &= \left(x - x_i\right) \Delta_x^{-1} \quad , \\
\Delta_y &= y_{j+1} - y_j \quad , \\
q_y &= \left(y - y_j\right) \Delta_y^{-1} \quad , \\
S(j) &= [F(i + 1,j) - F(i,j)]\, \Delta_x^{-1} \quad , \\
S(i) &= [F(i,j + 1) - F(i,j)]\, \Delta_y^{-1} \quad , \\
S_y(j) &= \left[F_y(i + 1,j) - F_y(i,j)\right] \Delta_x^{-1} \quad , \\
S_x(i) &= \left[F_x(i,j + 1) - F_x(i,j)\right] \Delta_y^{-1} \quad .
\end{aligned}\right\} \qquad \text{(B-1)}$$

First, let us apply our one-variable scheme to the function along the grid lines. We will omit the steps, but the result can be written in the following form.

$$\left.\begin{aligned} r_j(x) &\equiv R\left(x,y_j\right) \\ &= F(i,j)\ e_j(x) + F(i+1,j)\ g_j(x) \\ &+ \Delta_x \left[F_x(i,j)\ h_j(x) + F_x(i+1,j)\ k_j(x)\right] \quad , \end{aligned}\right\} \tag{B-2}$$

where

$$h_j(x) = q_x\left(1 - q_x\right)^2\Big/\left(1 + c_j q_x\right) \quad ,$$

$$k_j(x) = h_j(x) - q_x\left(1 - q_x\right) \quad ,$$

$$e_j(x) = \left(1 - q_x\right)^2 + 2h_j(x) \quad ,$$

$$g_j(x) = 1 - e_j(x) \quad ,$$

$$c_j = -1 + \left|F_x(i,j) - S(j)\right|\Big/\left|F_x(i+1,j) - S(j)\right| \quad .$$

In a similar manner, we obtain

$$\left.\begin{aligned} r_i(y) &\equiv R\left(x_i,y\right) \\ &= F(i,j)\ e_i(y) + F(i,\ j+1)\ g_i(y) \\ &+ \Delta_y \left[F_y(i,j)\ h_i(y) + F_y(i+1,y)\ k_i(y)\right] \quad , \end{aligned}\right\} \tag{B-3}$$

where

$$h_i(y) = {q_y}^2\left(1 - q_y\right)\Big/\left(1 + c_i q_y\right) \quad ,$$

$$c_i = -1 + \left|F_y(i,j) - S(i)\right|\Big/\left|F_y(i,j+1) - S(i)\right| \quad \text{, etc.}$$

The same algorithm can be applied to derivatives of the function. We find

$$\left.\begin{aligned} d_j(x) &\equiv (\partial R/\partial y)_{y\,=\,y_j} \\ &= F_y(i,j)\ E_j(x) + F_y(i+1,j)\ G_j(x) \\ &+ \Delta_x \left[F_{xy}(i,j)\ H_j(x) + F_{xy}(i+1,j)\ K_j(x)\right] \ , \end{aligned}\right\} \tag{B-4}$$

where

$$H_j(x) = q_x\left(1 - q_x\right)^2\Big/\left(1 + \gamma_i q_x\right) \ ,$$

$$K_j(x) = H_j(x) - q_x\left(1 - q_x\right) \ ,$$

$$E_j(x) = \left(1 - q_x\right)^2 + 2H_j(x) \ ,$$

$$G_j(x) = 1 - E_j(x) \ ,$$

$$\gamma_j = -1 + \left|F_{xy}(i,j) - S_y(j)\right|\Big/\left|F_{xy}(i+1,j) - S_y(j)\right| \ .$$

Finally, we obtain a similar expression for $d_i(y) \equiv (\partial R/\partial x)_{x\,=\,x_i}$.

To estimate the function on the interval $x_i \leqslant x \leqslant x_{i+1}$, $y_j \leqslant y \leqslant y_{j+1}$, the following form can be used.

$$\left.\begin{aligned} R(x,y) &= F(i,j)\ e_j(x)\ e_i(y) + F(i+1,j)\ q_j(x)\ e_{i+1}(y) \\ &+ F(i,j+1)\ e_{j+1}(x)\ q_i(y) + F(i+1,j+1)\ g_{j+1}(x)\ g_{i+1}(y) \\ &+ \Delta_x \Big[F_x(i,j)\ h_j(x)\ E_i(y) + F_x(i+1,j)\ k_j(x)\ E_{i+1}(y) \\ &+ F_x(i,j+1)\ h_{j+1}(x)\ G_i(y) + F_x(i+1,j+1)\ k_{j+1}(x)\ G_{i+1}(y)\Big] \end{aligned}\right\} \tag{B-5}$$

$$+ \Delta y \Big[F_y(i,j)\ E_j(x)\ h_i(y) + F_y(i,j+1)\ E_{j+1}(x)\ k_i(y)$$

$$+ F_y(i+1,j)\ G_j(x)\ h_{i+1}(y) + F_y(i+1,j+1)\ G_{j+1}(x)\ K_{i+1}(y)\Big]$$

$$+ \Delta_x\Delta_y \Big[F_{xy}(i,j)\ H_j(x)\ H_i(y) + F_{xy}(i+1,j)\ K_j(x)\ H_{i+1}(y)$$

$$+ F_{xy}(i,j+1)\ H_{j+1}(x)\ K_i(y) + F_{xy}(i+1,j+1)\ K_{j+1}(x)\ K_{i+1}(y)\Big] \ .$$

It is easily seen that Eq. (B-5) reduces to Eq. (B-2) for $y = y_j$ and to Eq. (B-3) for $x = x_i$. Unlike Eq. (18), it has continuous derivatives everywhere. However, it does exhibit the same kind of behavior as is illustrated in Fig. 6.