UNIVERSITY OF VESZPRÉM

DEPARTMENT OF MATHEMATICS AND COMPUTING

COMPARISON OF INTERPOLATION METHODS FOR PREDICTING THE VAPOUR PRESSURE OF AQUEOUS GLYCEROL SOLUTIONS

by

István SZALKAI[1,*], Attila SEBESTYÉN[2]
Ferenc BÓDI[3] and László KÓTAI[3]

PREPRINT No. 036

(1995)

V E S Z P R É M
EGYETEM U. 10.
P. O. Box: 158.
H-8201

0 Introduction

Recently evaporation of aqueous solutions with small glycerol concentrations obtained from the combined processes of ion-exchange and chemisorption on diluted solutions is ever increasingly preferred instead of expensive purification of concentrated glycerol solutions required by the cosmetic industry. Considering that glycerol loses water very easily at higher concentrations, resuling in acrolein or polycondensed derivatives, respectively, the control of the appropriate temperature limits is very important in the industrial technologcal processes.

In the present work mathematical relationships between data-triads of *concentration - vapour pressure - boiling point temperature* of the clear glycerol are studied. The possibilities of data interpolation with the application of several two variable spline-interpolation methods are studied with the aim of calculation of evaporating conditions at low pressure and upper temperature limits of the glycerol solutions with given concentrations.

In the present paper we compare four basically different methods with two variations to the third and fourth ones, in five sections. These methods were used to approximate the prediction of the vapour pressure, depending on the temperature and concentration as a two variable function. That is, $t=f(c,p)$ for some function $f:\mathbb{R}^2\to\mathbb{R}$ where t stands for temperature (Celsius), c for concentration (%) and p for pressure (MPa).

In the present paper we compare these methods both theoretically and practically. The difficulties for programming as well computing results, especially the errors are discussed in details, too. In the last section we also discuss computing experiments.

All the computations are based on the data in Table 1 taken from [K, Table 44]. These measured values are drawn on Fig. 1 in order to compare them with the approximated values.

Though most of these methods are *general methods* for approximating *any* two variable function, not the glycerin one, but this was a good example to compare these methods in the practice.

concentration (%)	pressure (MPa) 0,0053	0,0133	0,0202	0,0266	0,0333	0,0399	0,0466	0,0533	0,0599	0,0666	0,0733	0,0799	0,0866	0,0933	0,1013
	temperature (°C)														
0	34,0	51,6	60,1	66,4	71,6	75,9	79,6	82,9	85,9	88,7	91,2	93,5	95,7	97,7	100,0
10	34,4	52,1	60,7	67,0	72,3	76,6	80,3	83,7	86,7	89,5	92,0	94,3	96,6	98,6	100,9
20	34,9	52,7	61,3	67,7	73,0	77,3	81,1	84,4	87,5	90,3	92,9	95,2	97,4	99,5	101,8
30	35,5	53,4	62,1	68,6	73,8	78,2	82,0	85,3	88,4	91,3	93,8	96,2	98,4	100,4	102,8
40	36,5	54,4	63,1	69,5	74,9	79,3	83,1	86,4	89,5	92,4	95,0	97,3	99,6	101,7	104,0
50	37,5	55,7	64,5	71,1	76,4	80,9	84,8	88,2	91,3	94,2	96,8	99,2	101,5	103,6	106,0
60	39,5	58,0	66,9	73,5	79,0	83,5	87,4	90,9	94,1	97,0	99,7	102,1	104,4	106,0	109,0
70	43,0	61,7	70,8	77,6	83,1	87,7	91,7	95,2	98,4	101,4	104,1	106,6	109,0	111,1	113,6
80	49,1	68,2	77,4	84,3	90,0	94,6	98,7	102,3	105,6	108,7	111,4	113,9	116,3	118,5	121,0
90	59,5	80,2	90,3	97,7	103,9	109,1	113,5	117,4	121,0	124,4	127,4	130,2	132,8	135,2	138,8

Table 1 The measured boiling temperature of aquaeous glycerol solutions in Celsius grades, as in [K, Table 44]

1 A Heuristic Formula

In [SVPK] we find the formula

$$ln(p)=A(c)\cdot t^{\alpha(c)} \qquad \text{that is} \qquad t = \sqrt[\alpha(c)]{\frac{\ln(p)}{A(c)}}$$

where

$$A(c) = 0.5379-0.1626c+0.4271c^2-0.7497c^3$$

and

$$\alpha(c) = 0.5307+0.2310c-0.6996c^2+0.7539c^3$$

where p denotes the pressure in MPa, c is the relative concentration in % and t stands for temperature in °C. The authors of [SVPK] state that the error is less than 7% in the interval $10\%\le c\le 90\%$, $0.0053\text{MPa}\le p\le 0.1013\text{MPa}$ and $35^\circ\text{C}\le t\le 105^\circ\text{C}$. The authors of [SVPK] state that they found this formula using mass correlation.

The graph of the approximating function, given by the above formula is shown on Fig.2 while the absolute error of this method can be seen in Fig.3 and (for separate concentrarions) in Fig.4 .

The other methods, discussed in the following sections, are independent of temperature, concentration or vapour pressure. They are only general pure mathematical methods for approximating two variable functions.

2 Iterated One Dimensional Interpolation

In [DRM] and [PFTV] one finds the well known method of two variable spline interpolation which repeatedly uses one dimensional spline interpolations. Though this method is well known and one can find both in [DRM] and [PFTV] we shortly describe it to make this paper self-contained.

The present method requires that the domain of the function we are to approximate be a rectangle $[a,b]\times[c,d]$ and that the measured points lie

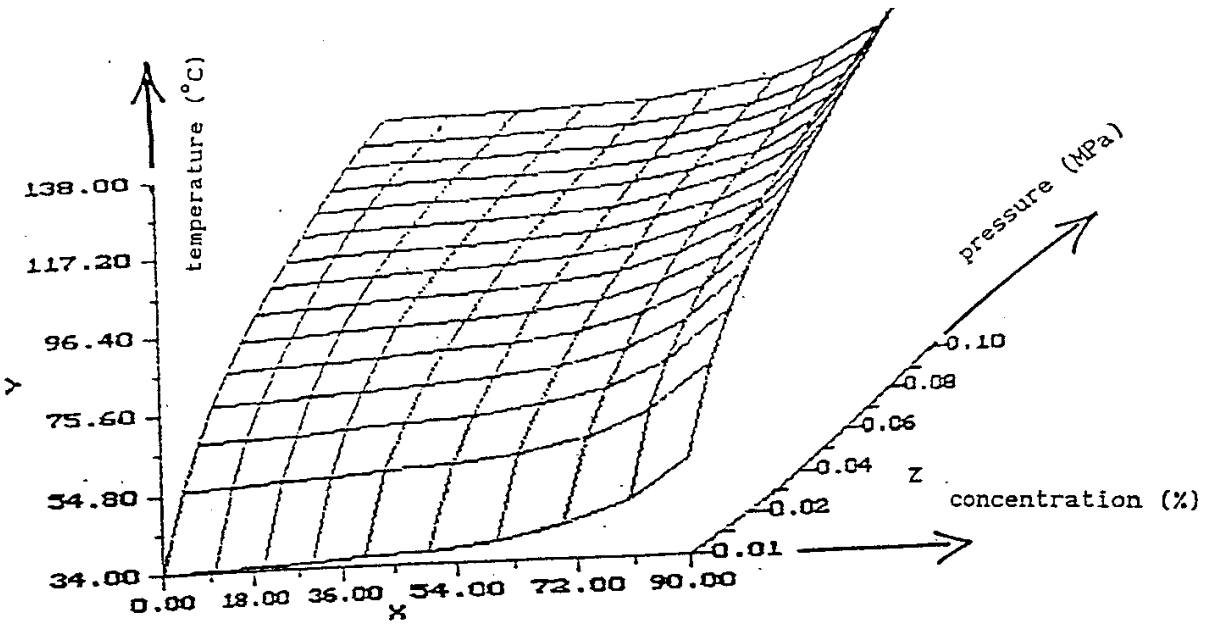


Figure 1 : The measured datas from Table 1 (ie. [K, Table 44])

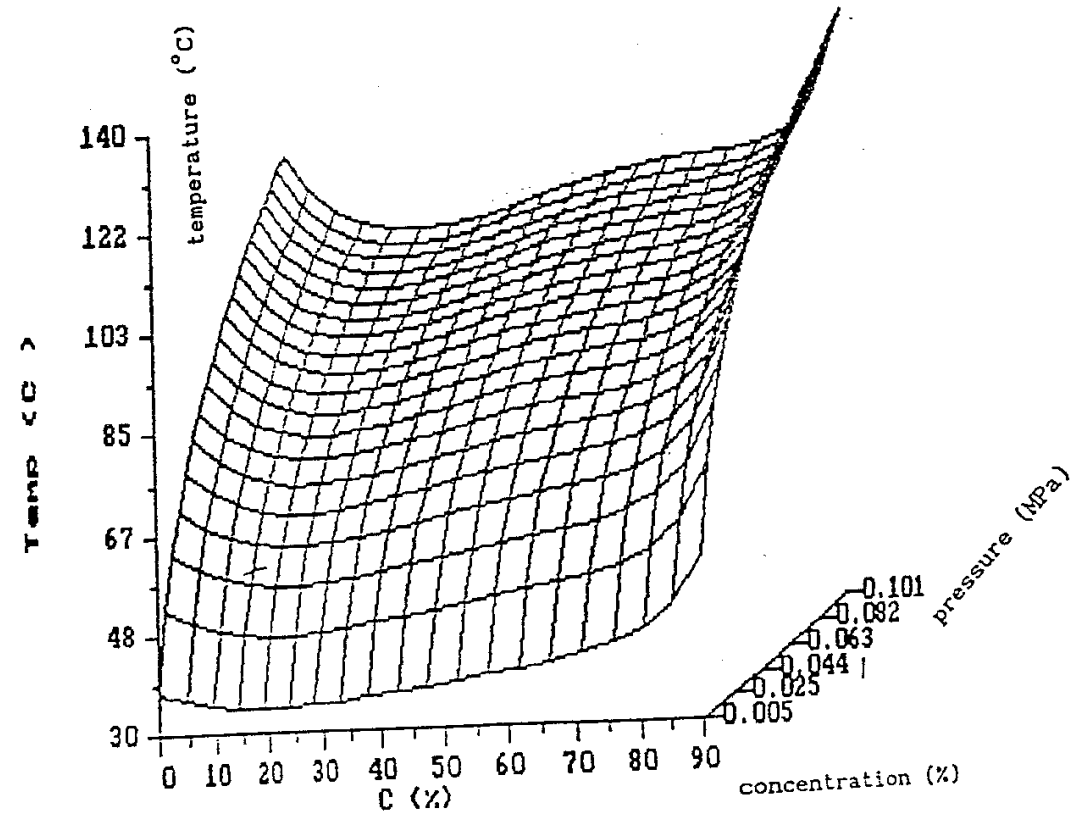
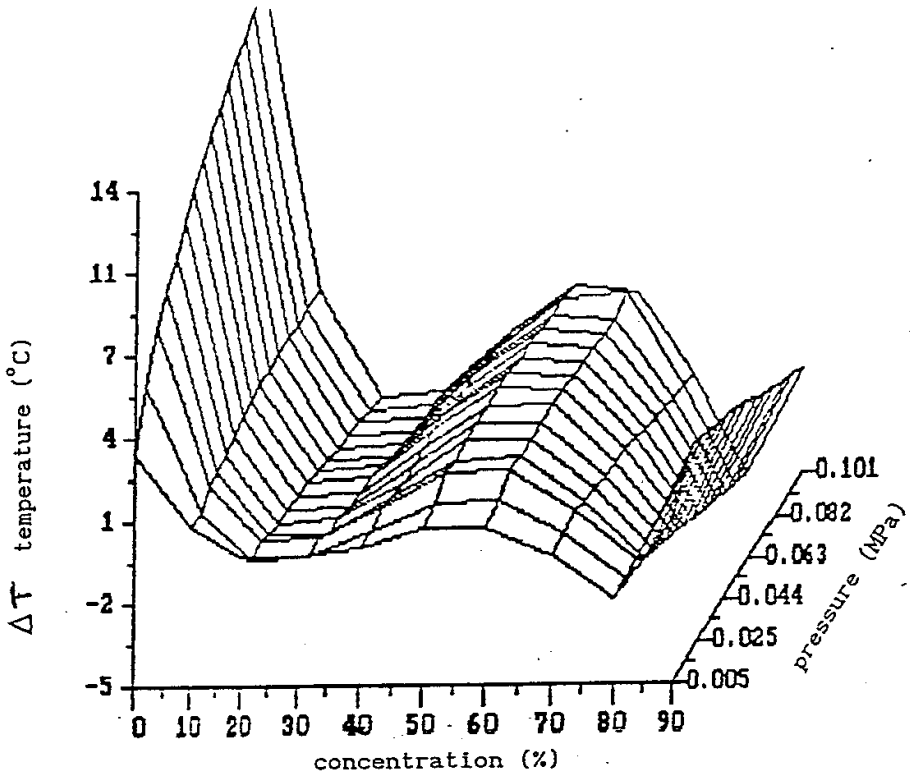
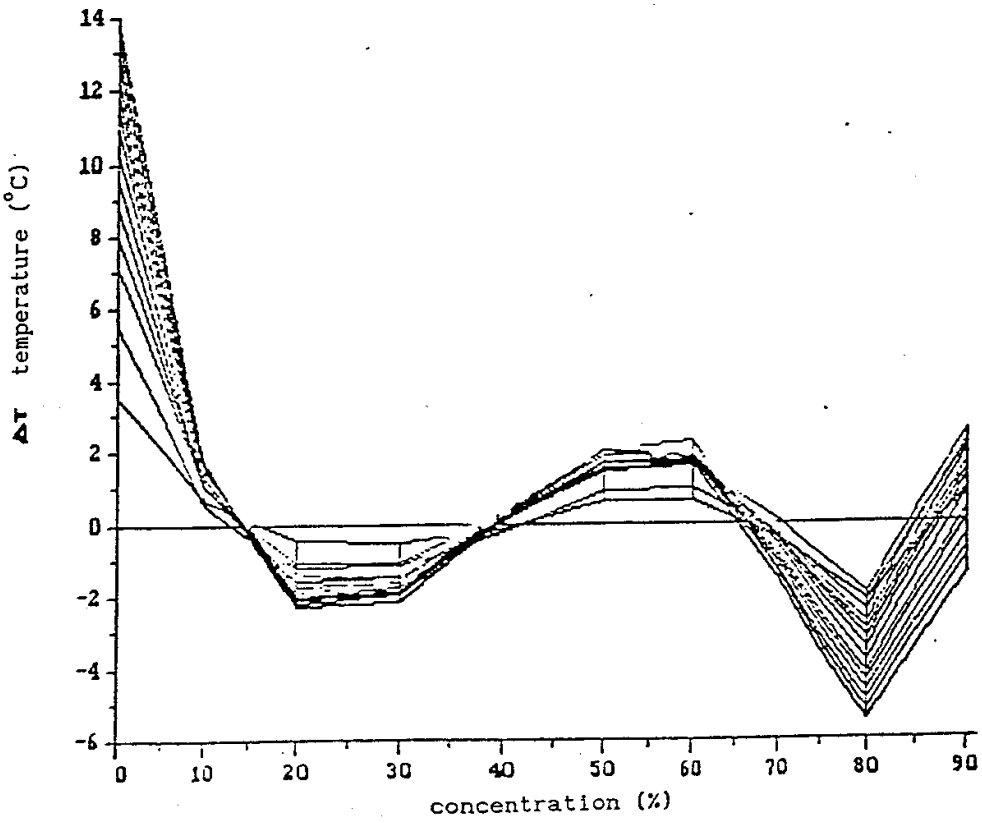


Figure 2 : Saburov & al's method



Figures 3 and 4 : The error of Saburovs' method



on lines parallel to the ordinate x axis (see below). So we are given the measured values $z_{i,j}$ at the measured points $P_{i,j}=(x_{i,j},y_i)$ such that $1\le i\le M$, $1\le j\le N$, $a\le x_{i,j}\le b$, $c\le y_i\le d$. We want to construct a spline function **S(x,y)** such that $S(x_{i,j},y_i)=z_{i,j}$ for $1\le i\le M$, $1\le j\le N$.

For calculating $S(X,Y)$ at each point $(X,Y)\in[a,b]\times[c,d]$ we use a couple of one dim. spline interpolations in each horizontal line and a single further one along a vertical line. That is:

First for each *fixed* $i\le M$ use one 1- dimensional spline approximation along the points $x_{i,j}$ with the measured values $z_{i,j}$ $(1\le j\le N)$ to construct spline functions $s_i(x)$ such that $s_i(x_{i,j})=z_{i,j}$ for $1\le j\le N$. Call this procedure *horizontal steps*. (Observe that $Dom(s_i)=[a,b]$ and for horizontal steps we do not need the point (X,Y).)

After this use a single one dimensional spline interpolation based on the points y_i and values $s_i(X)$ for $1\le i\le M$ to get the spline function $S_X(y)$ with $Dom(S_X)=[c,d]$ such that $S_X(y_i)=s_i(X)$. Finally we take the desired approximation function S as $S(X,Y)=S_X(Y)$. The method is illustrated in Fig. 5.

In practice, of course, we can compute the splines s_i in advance, and after this for each point $(X,Y)\in[a,b]\times[c,d]$ we can determine $S(X,Y)$ by using *one* more 1-dimensional spline interpolation to get S_X .

In our computations we used cubic spline interpolations (described in more details in [DRM] and [PFTV], and here below) in each case. The approximating function for our glycerol data $S(x,y)$ is drawn in Fig. 6.

Advantages of this method: the one-dimensional approximation method is widely known with easily understable theory and an easy implementation in programming. No equidistance tabulated (measuring) points y_i or $x_{i,j}$ are needed. We need only that these points $P_{i,j}$ lie on parallel lines (eg. parallel to the x-axis[1]). The method also gives approximation at the margins of the closed rectangle, ie. on the whole Dom(f). Further, the method can be easily generalized for higher dimensions, eg. using iterated less-dimensional iterations, by induction on the dimension of the space $\mathbb{R}^n$.

Disadvantages of this method: the repeated use of the final spline

[1] That is $P_{i,j}=(x_{i,j},y_i)$ for $1<i<M$, $1<j<N$, which means that the second coordinate depends upon i only.

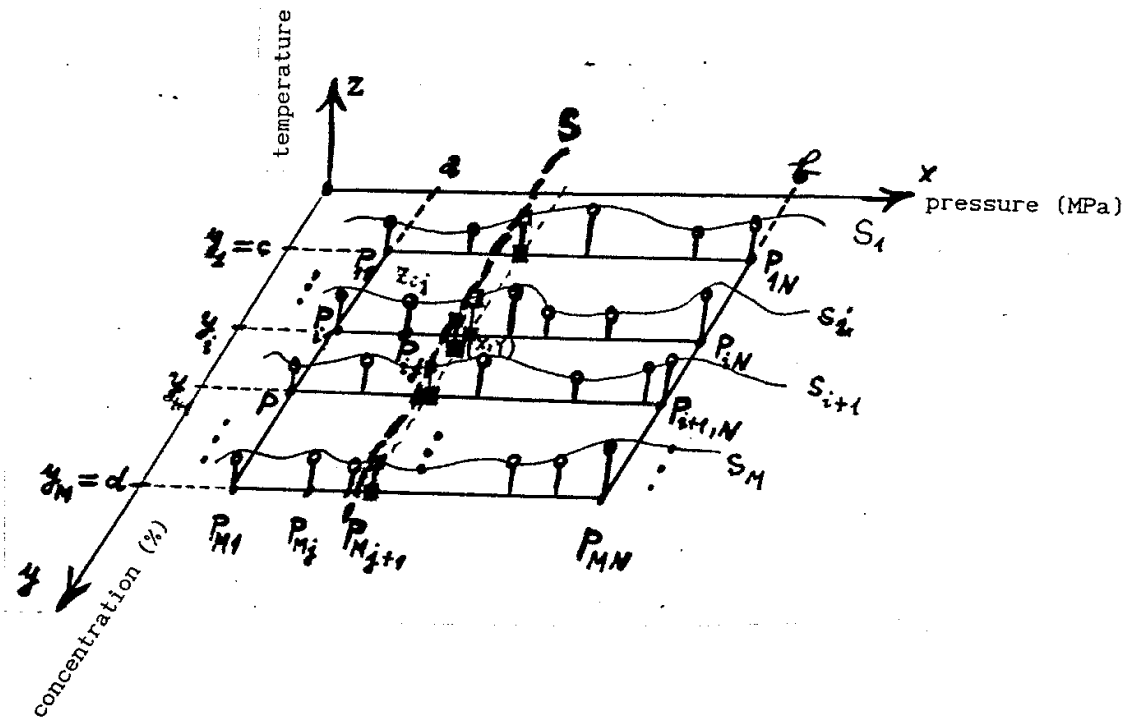


Figure 5 : The principle of iterated one dimensional spline function for two dimensional interpolation

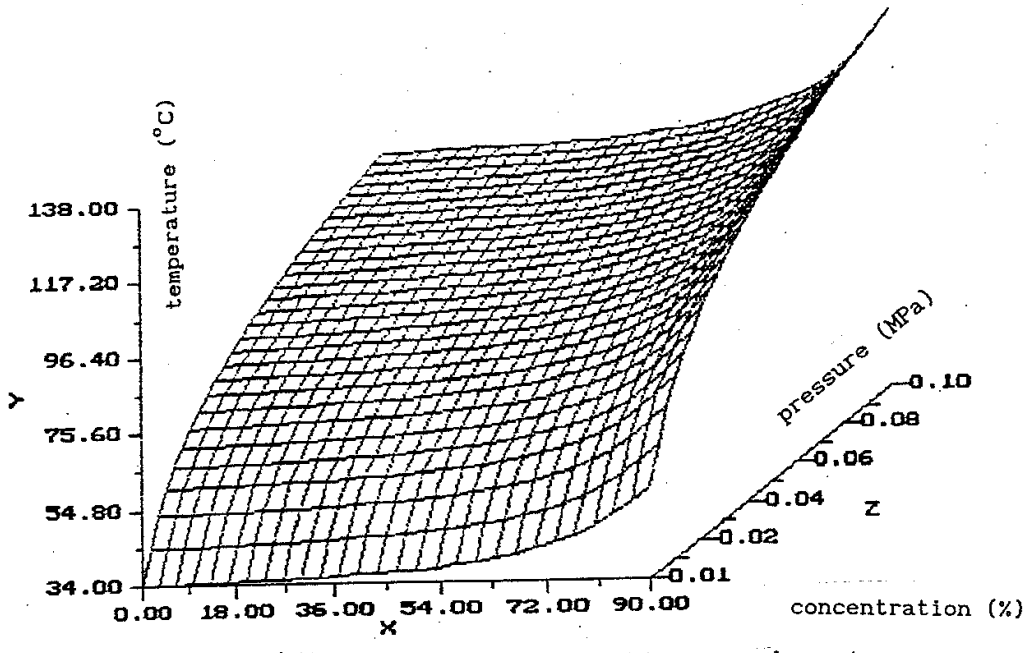


Figure 6 : Iterated one dimensional splines

approximation ("vertical step") for getting S(X,Y) at each point (X,Y) makes the computation (slightly) slower: we repeatedly have to build up and solve a tridiagonal $N\times N$ system of linear equations.

Though the *one dimensional cubic spline approximation* method is well known (see eg. [DRM], [PFTV]), we shortly describe it for the readers' convenience.

Let $a=x_1<x_2< \dots <x_N=b$ and $y_1,y_2, \dots ,y_N\in\mathbb{R}$ be given. We want to construct a one variable function $S(x)$ of which *the first derivative is continuous* on the interval $[a,b]$ and such that $S(x_j)=y_j$ for $1\le j\le N$. We set (omitting theoretical computations)

$$S(x) = S_j(x) := A(x)\cdot y_j + B(x)\cdot y_{j+1} + C(x)\cdot y''_j + D(x)\cdot y''_{j+1}$$

for $x\in[x_j,x_{j+1}]$, $1\le j\le N$ where the functions A(x) through D(x) and the real numbers $y''_j, y''_{j+1}\in\mathbb{R}$ are to be determined as follows. The method is a modification of Lagrange's formula. So

$$A(x) = \frac{x_{j+1} - x}{x_{j+1} - x_j} \qquad , \qquad B(x) = 1 - A(x)$$

$$C(x) = \frac{1}{6}(A^3(x)-A(x))\cdot(x_{j+1}-x_j)^2 \quad \text{and} \quad D(x) = \frac{1}{6}(B^3(x)-B(x))\cdot(x_{j+1}-x_j)^2$$

We require the continuity of $S'(x)$ (the derivative of S(x)) on the whole interval $[a,b]$, that is $S'_j(x_{j+1})=S'_{j+1}(x_{j+1})$ for $1\le j<N$. Further we can think of y''_j and y''_{j+1} as the second derivatives $S''_j(x_j)$ and $S''_{j+1}(x_{j+1})$. This gives us a system of N-2 linear equations in N unknown real numbers y''_j $(1\le j<N)$ as

$$\frac{x_j-x_{j-1}}{6}\cdot y''_{j-1} + \frac{x_{j+1}-x_{j-1}}{3}\cdot y''_j + \frac{x_{j+1}-x_j}{6}\cdot y''_{j+1} = \frac{y_{j+1}-y_j}{x_{j+1}-x_j} - \frac{y_j-y_{j+1}}{x_j-x_{j+1}} \qquad (1<j<N)$$

and for the unique solution we may take $y''_1=y''_N=0$. Solving this system of linear equations we get S(x).

Summarizing the above: to find a cubic spline function we need to solve an N-2 dim. (tridiagonal) system of linear equations, at each vertical step.

3 A Direct Two-Dimensional Interpolation

In [L1] one can find a general direct method, invented by M.Lénárd, for constructing directly a two dimensional spline interpolation of minimal degree. Since this general method requires that the measuring points form an equidistant lattice, some precalculations are needed to apply this method for our problem. In this section we briefly sketch this method; the mathematical background can be found in [L1].

We are given the measuring points $(x_j, y_k) \in [a,b]\times[c,d]$ and the values $u_{j,k} \in \mathbb{R}$ for $0 \le j \le N+1$, $0 \le k \le M+1$ such that $x_{j+1}-x_j=h$ and $y_{j+1}-y_j=\ell$ for some positive real numbers h, ℓ (ie. the measuring points form an equidistant lattice), and further $x_o=a$, $x_{N+1}=b$, $y_o=c$, $y_{M+1}=d$. We have to find a two-variable spline function $S_\Delta(x,y)$ such that $S_\Delta(x_j, y_k)=u_{jk}$ for $0 \le j \le N+1$, $0 \le k \le M+1$ .Of course we want certain continuity and minimality properties for $S_\Delta(x,y)$. More precisely, we require the following properties:

(a) S_Δ is twice continuously partially differerentiable in both variables on its whole domain $[a,b]\times[c,d]$,

(b) S_Δ is a polynomial *in both variables* of minimal degree on each lattice subrectangle $[x_j, x_{j+1}]\times[y_k, y_{k+1}]$ $(0 \le j \le N, 0 \le k \le M)$,

(c) on each lattice rectangle, its degree, as regarded as a *two-variable* polynomial, is minimal.

In [L1] M.Lénárd constructed S_Δ and proved that it satisfied (a) through (c). We describe only the construction itself.

For $(x,y) \in [x_j, x_{j+1}]\times[y_k - y_{k+1}]$ $(0 \le j \le N, 0 \le k \le M)$ let

$$S_\Delta(x,y) = S_{j,k}(x,y) = \sum_{\substack{0 \le \mu, \nu \le 5 \\ \mu+\nu \le 6}} A_{j,k}^{(\mu,\nu)} \cdot (x-x_j)^\mu \cdot (y-y_k)^\nu$$

where the constants $A_{j,k}^{(\mu,\nu)} \in \mathbb{R}$ are determined in [L1], as the solution of a system of linear equations arising from (a). The solution is listed below:

$$A_{j,k}^{(0,0)} = u_{j,k},$$

$$A_{j,k}^{(1,0)} = \frac{1}{2h}[\Delta^{1,0}u_{j,k} + \Delta^{1,0}u_{j-1,k}],$$

$$A_{j,k}^{(0,1)} = \frac{1}{2\ell}[\Delta^{0,1}u_{j,k} + \Delta^{0,1}u_{j,k-1}],$$

$$A_{j,k}^{(2,0)} = \frac{1}{2h^2}\Delta^{2,0}u_{j-1,k},$$

$$A_{j,k}^{(1,1)} = \frac{1}{2h\ell}[\Delta^{1,1}u_{j,k} + \Delta^{1,1}u_{j-1,k-1}],$$

$$A_{j,k}^{(0,2)} = \frac{1}{2\ell^2}\Delta^{0,2}u_{j,k-1},$$

$$A_{j,k}^{(3,0)} = -\frac{3}{2h^3}\Delta^{2,0}u_{j-1,k}$$

$$A_{j,k}^{(2,1)} = \frac{1}{2h^2\ell}[\Delta^{2,1}u_{j-1,k-1} - \Delta^{2,2}u_{j-1,k-1}]$$

$$A_{j,k}^{(1,2)} = \frac{1}{2h\ell^2}[\Delta^{1,2}u_{j-1,k-1} - \Delta^{2,2}u_{j-1,k-1}]$$

$$A_{j,k}^{(0,3)} = -\frac{3}{2\ell^3}\Delta^{0,3}u_{j,k-1}$$

$$A_{j,k}^{(4,0)} = \frac{5}{2h^4}\Delta^{3,0}u_{j-1,k}$$

$$A_{j,k}^{(3,1)} = \frac{1}{2h^3\ell}[\Delta^{2,2}u_{j-1,k-1} - 3\Delta^{3,1}u_{j-1,k}]$$

$$A_{j,k}^{(2,2)} = \frac{1}{2h^2\ell^2}[\Delta^{2,2}u_{j,k} + 4\Delta^{2,2}u_{j-1,k-1}]$$

$$A_{j,k}^{(1,3)} = \frac{1}{2h\ell^3}[\Delta^{2,2}u_{j-1,k-1} - 3\Delta^{1,3}u_{j,k-1}]$$

$$A_{j,k}^{(0,4)} = \frac{5}{2\ell^4}\Delta^{0,3}u_{j,k-1}$$

$$A_{j,k}^{(5,0)} = -\frac{1}{h^5}\Delta^{3,0}u_{j-1,k}$$

$$A_{j,k}^{(4,1)} = \frac{5}{2h^4l}\Delta^{3,1}u_{j-1,k}$$

$$A_{j,k}^{(3,2)} = -\frac{1}{2h^3l^2}[\Delta^{2,2}u_{j,k}+2\Delta^{2,2}u_{j-1,k-1}]$$

$$A_{j,k}^{(2,3)} = -\frac{1}{2h^2l^3}[\Delta^{2,2}u_{j,k}+2\Delta^{2,2}u_{j-1,k-1}]$$

$$A_{j,k}^{(1,4)} = \frac{5}{2hl^4}\Delta^{1,3}u_{j,k-1}$$

$$A_{j,k}^{(0,5)} = -\frac{1}{l^5}\Delta^{0,3}u_{j,k-1}$$

$$A_{j,k}^{(5,1)} = -\frac{1}{h^5l}\Delta^{3,2}u_{j-1,k}$$

$$A_{j,k}^{(3,3)} = \frac{1}{2h^3l^3}[\Delta^{2,2}u_{j,k}+\Delta^{2,2}u_{j-1,k-1}]$$

$$A_{j,k}^{(1,5)} = -\frac{1}{hl^5}\Delta^{1,3}u_{j,k-1}.$$

$$A_{j,k}^{(4,2)} = A_{j,k}^{(2,4)} = 0$$

$$(j = 1, \ldots, N+1,\ k = 1, \ldots, M+1).$$

where $\Delta^{\mu,\nu}u_{j,k}$ denotes the finite partial difference of the μ -th order in the first variable and of the ν -th order in the second variable computed from the set of measured values $\{u_{j,k}: 0\le j\le N+1, 0\le k\le M+1\}$ for $0\le j\le N+1$, $0\le k\le M+1$.

Theoretical error estimates for this method (if the measured points form an equidistant lattice) can be found in [L1].

Advantages of this method: Although the precomputation of the higher dimensional arrays $\Delta^{p,q}_{j,k}$ and $A^{p,q}_{j,k}$ takes some time, S(X,Y) can then be computed as a polynomial at any point (X,Y), making the computation fast. Further, the method can be easily generalized for higher dimensions as described in [L3] (4) .

Disadvantages of this method: We ultimately need equidistant tabulated (measuring) points $P_{i,j}=(y_i,x_j)$ where $y_{i+1}-y_i=h$ and $x_{j+1}-x_j=\ell$ for some fixed h,ℓ. In some cases the set of measuring points can be transformed to be equidistant. This problem is discussed in the next section. Further, the method gives no approximation at certain margins of the rectangle, ie. of Dom(f), since we can not compute the finite partial differences $\Delta^{\mu,\nu}u_{j,k}$ of higher order, for $j>N+1-\mu$ or $k\ge M+1-\nu$ at the margin of Dom(f), and the maximum value of μ and ν is 6.

4 Modifying the Set of the Measuring Points

As we saw in the previous section, Lénárd's method requires that the measuring points form an equidistant lattice, which was not the case in our original chemical problem for glycerol solutions. In this section we discuss some precalculation methods which can be used to modify the physical set of data in order to use Lénárd's method.

---

(4) To be more precise: [L3] generalizes the methods from *[L2]* to higher dimensions, where [L2] describes *improved* variants of the method we presented here, after [L1]. In [L2] M.Lénárd eliminates the restriction of the equidistant measuring points, preserves the shape of the dataset (monotonicity, convexness, etc.), avoids the "bumps" shown in Figures 7 and 8. We intend to investigate the method from [L2] in practice in a forthcoming paper.

4.a) Using preliminary approximation

Our first method can be applied for problems where the tabulated (measuring) points $P_{i,j}$ lie on lines parallel to the x axis; that is $P_{i,j}=(x_{i,j},y_i)$ for $1\le i\le M$, $1\le j\le N$. Further, the sequence y_i is an equidistant one; that is $y_{i+1}-y_i=h$ for some positive constant $h\in R$. (In our chemical example this was the case.) Before we use Lénárd's method, we first use one-dimensional interpolations $s_i(x)$ along each line $L_i=\{P_{i,j}: j\le N\}$ for each fixed $1\le i\le M$ to approximate the measuring values at the equidistant tabulated points $R_{i,j}$, $j\le N$, ($1\le i\le M$, i is fixed). After these approximations the tabulated points $R_{i,j}$ form an equidistant set for $1\le i\le M$, $1\le j\le N$. Using the values given by the functions $s_i(x)$ for $1\le i\le M$, we can apply Lénárd's method to obtain a final two-dimensional interpolation for our two variable problem on the whole domain of f.

However this method is slow and the error is not small enough, because of the pre-approximation. Further, the computed approximating values are not equal to respective measured data, at the measuring points ! (See Fig. 7)

4.b) Transforming the measuring points

Our next transformation is applicable only for problems where the tabulated (measuring) points $P_{i,j}=(y_i,x_j)$ form a *rectangular lattice*, not neccessarily equidistant for $1\le i\le M$, $1\le j\le N$. That is, $y_{i+1}-y_i=h$ or $x_{i+1}-x_i=\ell$ for some $h,\ell\in R$ is NOT required. (In our chemical example this is also in the case.)

First we use piecewise linear transformations in both of the variables to transform this set of measuring points into equidistant ones; second we perform Lénárd's method; and third, we use the inverse of the piecewise linear transformations to get the approximation required. In more detail:

We are given the measuring points $(x_j,y_k)\in[a,b]\times[c,d]$ and the values $u_{j,k}\in R$ for $0\le j\le N+1$, $0\le k\le M+1$ such that $x_o=a$, $x_{N+1}=b$, $y_o=c$ and $y_{M+1}=d$. Let

$$t : [a,b]\times[c,d] \longrightarrow [a,b]\times[c,d]$$

be a two variable function such that the points $t(x_i,y_j)$ for $0\le j\le N+1$, $0\le k\le M+1$ form an equidistant lattice in $[a,b]\times[c,d]$. It is easy to see that

$$\mathbf{t(x,y) = (t_1(x),t_2(y))}$$

and further that $t_1(x_{j+1})-t_1(x_j)=h$ and $t_2(y_{k+1})-t_2(y_k)=\ell$ for some positive real numbers h,ℓ for $0\le j\le N$, $0\le k\le M$.
More exactly, for points $(x,y)\in Dom(t)$, $x_j\le x<x_{j+1}$ and $y_k\le y<y_{k+1}$ we have

$$t_1(x) = \frac{(x-x_j)\cdot h}{x_{j+1}-x_j} + t_1(x_j) \quad \text{and} \quad t_2(y) = \frac{(y-y_k)\cdot \ell}{y_{k+1}-y_k} + t_2(y_k)$$

Both t_1 and t_2 are continuous, piecewise linear functions. Then applying Lénárd's method to the equidistant "measuring" points $R_{i,j}=(t_1(x_j),t_2(y_j))$ and the measured values u_{ij} assigned to $R_{i,j}$ for $0\le j\le N+1$, $0\le k\le M+1$ we get an interpolating spline $S_o(x,y)$. Finally the function composition

$$S_t(x,y) := (S_o\circ t)(x,y) = S_o(t(x),t(y))$$

will interpolate our original problem.

Comparing this transformation method to the previous one we can observe that the present one is not slow and the error is small enough. The error is exactly Lénárd's method's error since t transforms the points (x,y) with their error together. The presented method gives exact values at the measured points. The function **t** can be calculated fast and easily. However this piecewise transformation is not differentiable at tabulated (measuring) points, so S_t need not be differentiable anymore but it is continous anyway, in spite that Lénárd's S_o is twice continously partially differentiable! However the obtained two variable function is not smooth enough since the grade of the transformation is different in each cells. The wawes in Fig.8 are caused by these piecewise linear transformations.

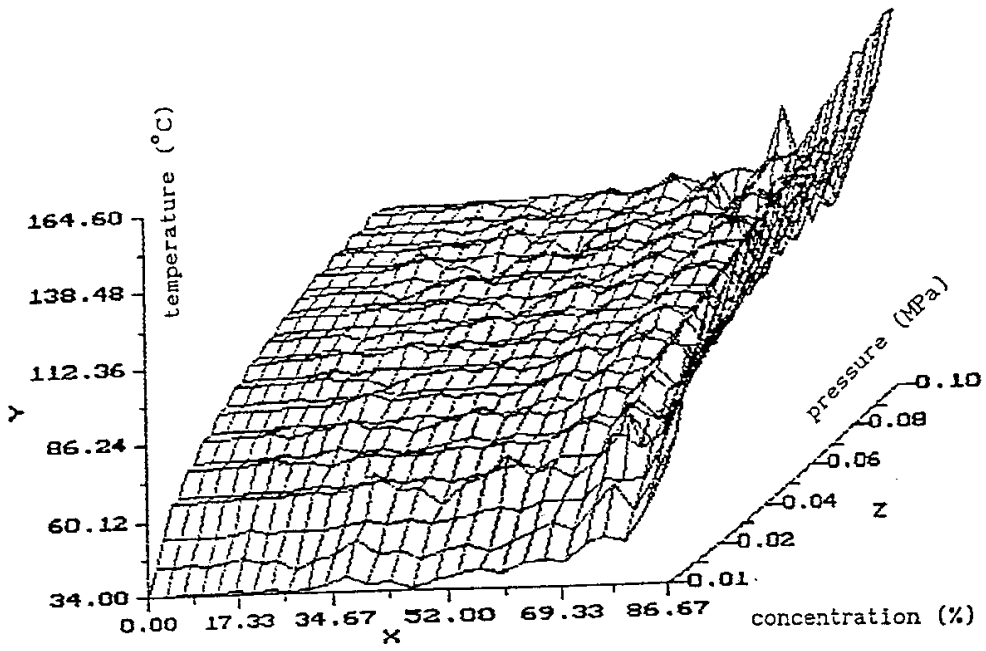


Figure 7 : Lénárd's method with pre-approximated values

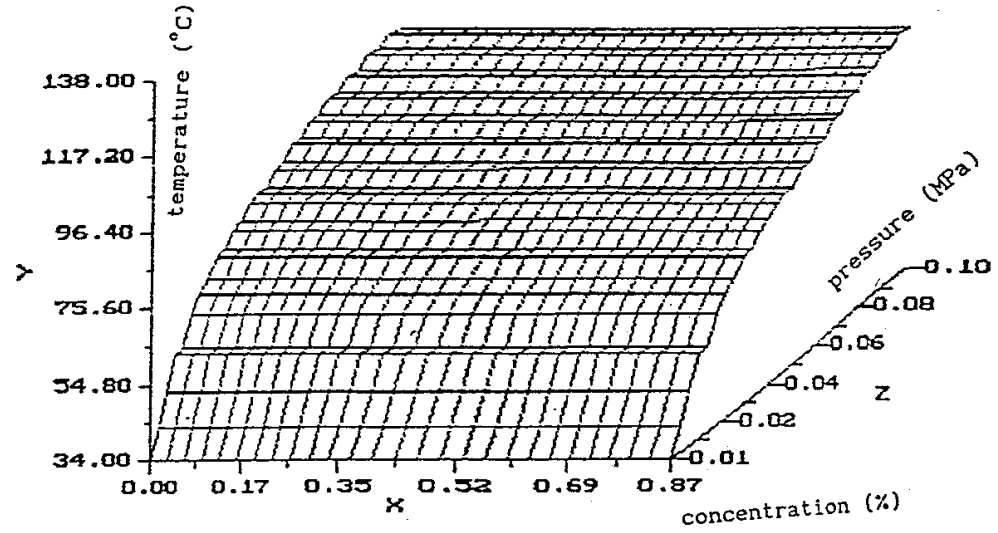


Figure 8 : Lénárd's method with piecewise linear transformations of the function's domain

5 Shepard's method

In [S] D.Shepard presented a general method for continuous approximation of more variable functions $f:\mathbb{R}^n\to\mathbb{R}$ (see also [GW]). This method does *not* require any special assumption on the positions of the measuring points: it works for *arbitrary* distribution of the dataset. Now we present this general method briefly.

So, we are given *arbitrary* measuring points $P_1,P_2,\ldots,P_N \in \mathbb{R}^n$ with the corresponding values $F_1,F_2,\ldots,F_N \in \mathbb{R}$. Now then the below function

$$U(P) := \frac{\sum_{i=1}^{N}\left(F_i \cdot \prod_{j\neq i}\rho(P,P_j)\right)}{\sum_{i=1}^{N}\left(\prod_{j\neq i}\rho(P,P_j)\right)}$$

defined for all $P\in\mathbb{R}^n$ is *continuous on the whole* $\mathbb{R}^n$, and $U(P_i)=F_i$ holds for each $i\leq N$ where $\rho(P,Q)$ denotes the Eucledian distance of the points $P,Q\in\mathbb{R}^n$. This fact can be seen easily, and moreover is also true for any metric $\rho:TxT\to\mathbb{R}^+$ in any metric space T .

This easy fact gives a general method for approximating *any* function with arbitrary set of measuring points. In [GW] we find nice figures of U for *small* N (below 10 (!)). However, the *above* formula for larger N requires a lot of computation, even with tremendous large numbers. In our example in Table 1, N=150, so the products would be about 10^{+400} ! Fortunately U(P) has an other form which is, in fact, equivalent to the above formula:

$$U(P) := \frac{\sum_{i=1}^{N}\frac{F_i}{\rho(P,P_i)}}{\sum_{i=1}^{N}\frac{1}{\rho(P,P_i)}}$$

In other words, U(P) is a weighted arithmetic mean of the measured values $F_1,F_2,\ldots,F_N$ with weights: the *reciprocials* of the distances of the point P from the given points P_i .

The latter formula can be easily used in practice: neither complicated nor slow, continuous on the whole space $\mathbb{R}^n$, and is convergent as P runs (in any way) to ∞ . The smooth surface shown in Fig.9.a) is generated by this method. However, investigating Shepard's function via more grid points, we can observe "bumps" (hills and valleys) *inside* the convex hull of the given points $F_1, F_2, \ldots, F_N$ (see Fig.9.b): this is the *real* face of U(P) ! These bumps show, that U(P) treates to take his limit values $\bar{U} = \lim_{P \to \infty} U(P)$ almost everywhere in $\mathbb{R}^n$. *This* disadvantage of Shepard's method is not highlighted in the literature. Elimination methods for this problem (choosing another weight functions) will be discussed in detail in [Sz].

Another trick to avoid these "bumps" is to delimite ourselves for using for small subset of the given dataset for each approximation. In our glicerous example (see Table 1) we used 4×4 data points surrounding the requested approximated point.

We have run two variants of Shepard's method on computer. The original version is shown on Fig.9., while using only 4×4 neighbouring measured points for each approximation point is presented in Fig.10. This modification of the present method of course makes the method inaccurate, and must not be continuous since, turning from one cell to another, the 4×4 points, and so the value of U(P) are changed.

6. Computing experiments

Our data set was taken from [K] where in the table there are 10 rows (concentration) and 15 columns (pressure). The concentrations are equidistant but the pressures not. We used an IBM 386 compatible PC on 33 MHz and Turbo Pascal 6.0 language. The computation for each method took only some minutes. We approximated the temperature with the following six variants of the methods described in this paper.

method 1 : Saburov & als' [SVPK] heuristic formula, see our Section 1,

method 2 : iterated one dimensional splines, see our Section 2,

method 3 : Lénárd's two dimensional interpolation with preapproximating in order to make the pressure data set to be equidistant, see Sections 3 and 4.a,

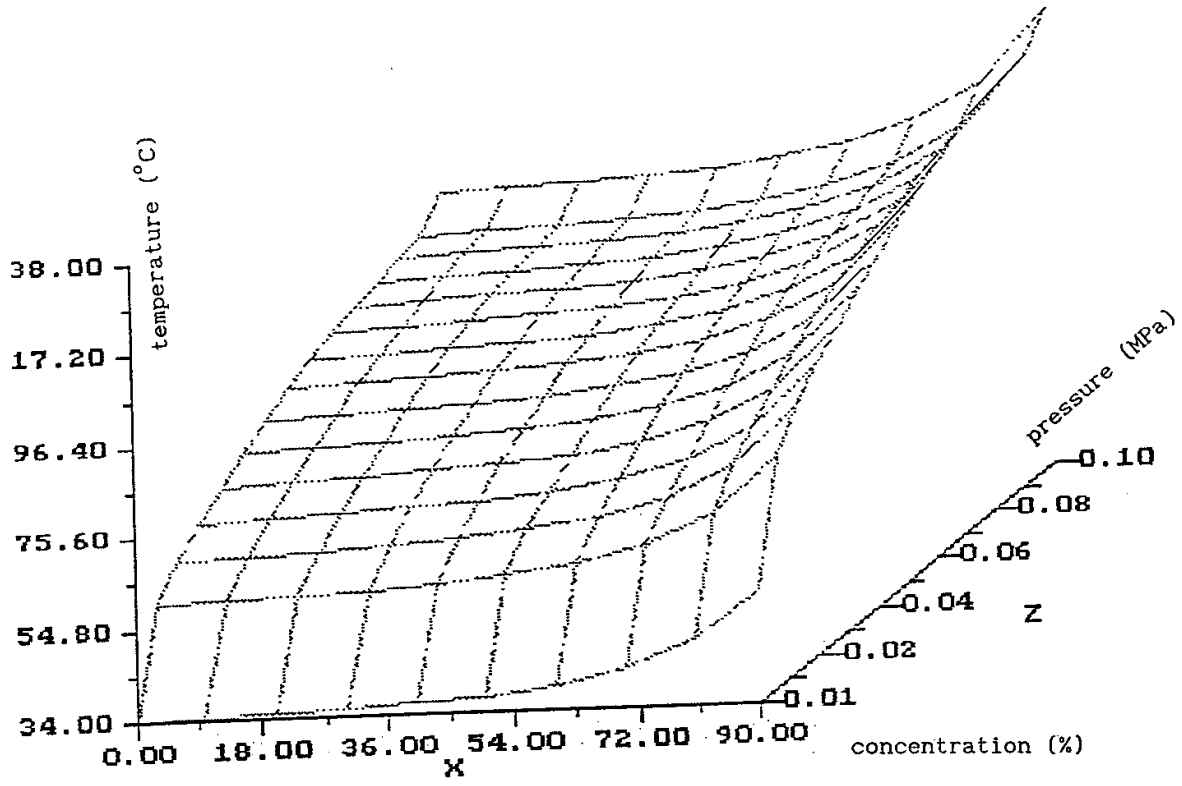

temperature (°C)
38.00
17.20
96.40
75.60
54.80
34.00
0.00
18.00
36.00
54.00
72.00
90.00
X
pressure (MPa)
0.10
0.08
0.06
0.04
0.02
0.01
Z
concentration (%)

Fig. 9.a.

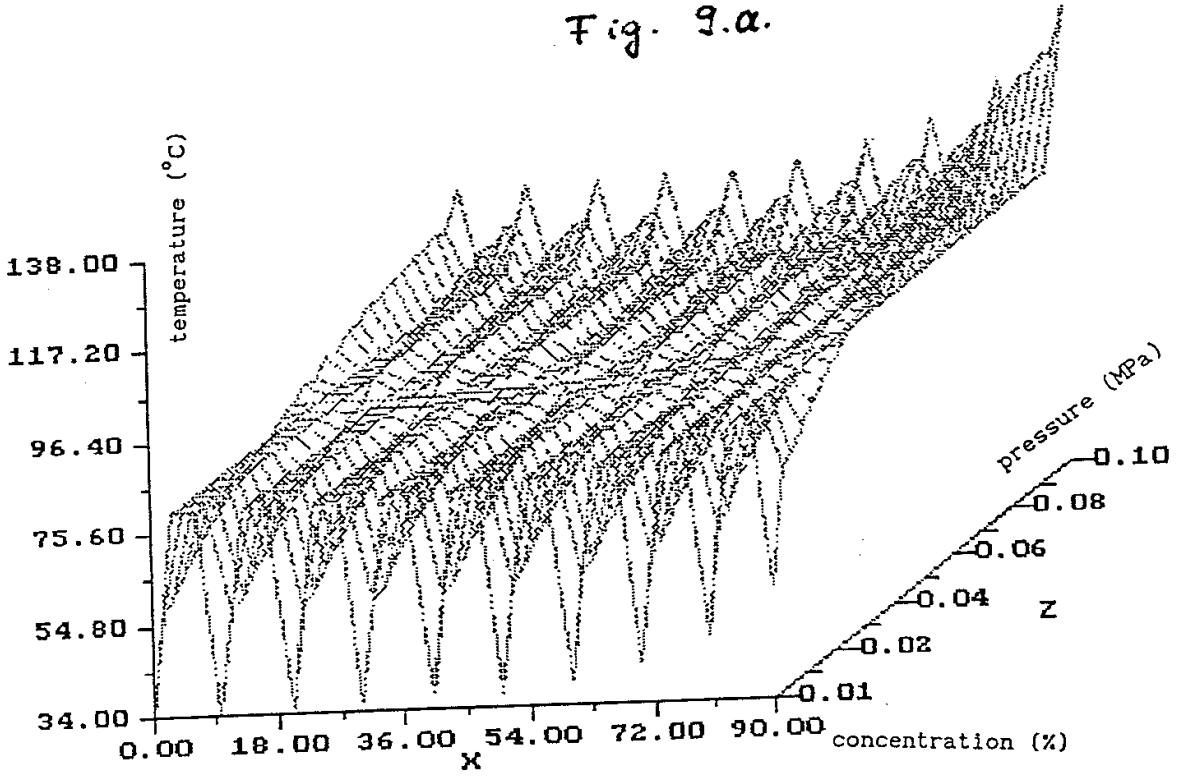

temperature (°C)
138.00
117.20
96.40
75.60
54.80
34.00
0.00
18.00
36.00
54.00
72.00
90.00
X
pressure (MPa)
0.10
0.08
0.06
0.04
0.02
0.01
Z
concentration (%)

Fig. 9.b)

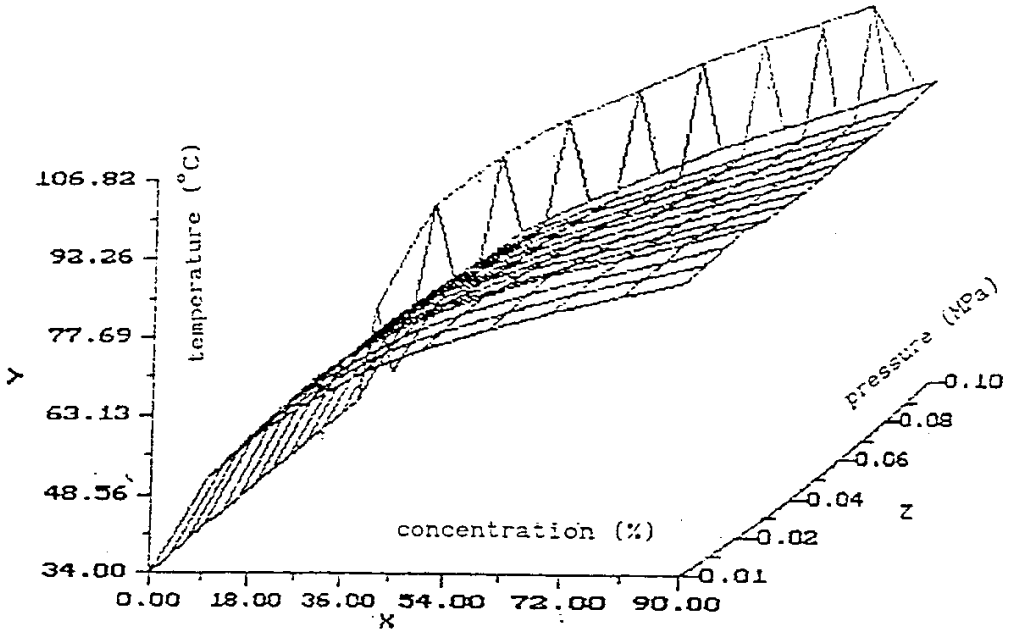


Figure 10 : Shepard's method, using 4x4 neighbour points for each approximation

method 4 : Lénárd's two dimensional interpolation with transformation of the measuring points, see Sections 3 and 4.b,

method 5 : Shepard's original method, see Section 5,

method 6 : Shepard's method using 4×4 tabulated (measuring) points for each approximation, see Section 5.

In Table 2 we present a part of our computational results, simultaneously for all the six method. Figures 1, 2 and 6 through 10 show the (global) original measured datas and the approximating functions given by these six methods. Fig. 11 is a *common graph* of sections of the above six methods' surfaces to the fixed concentration c=30% .

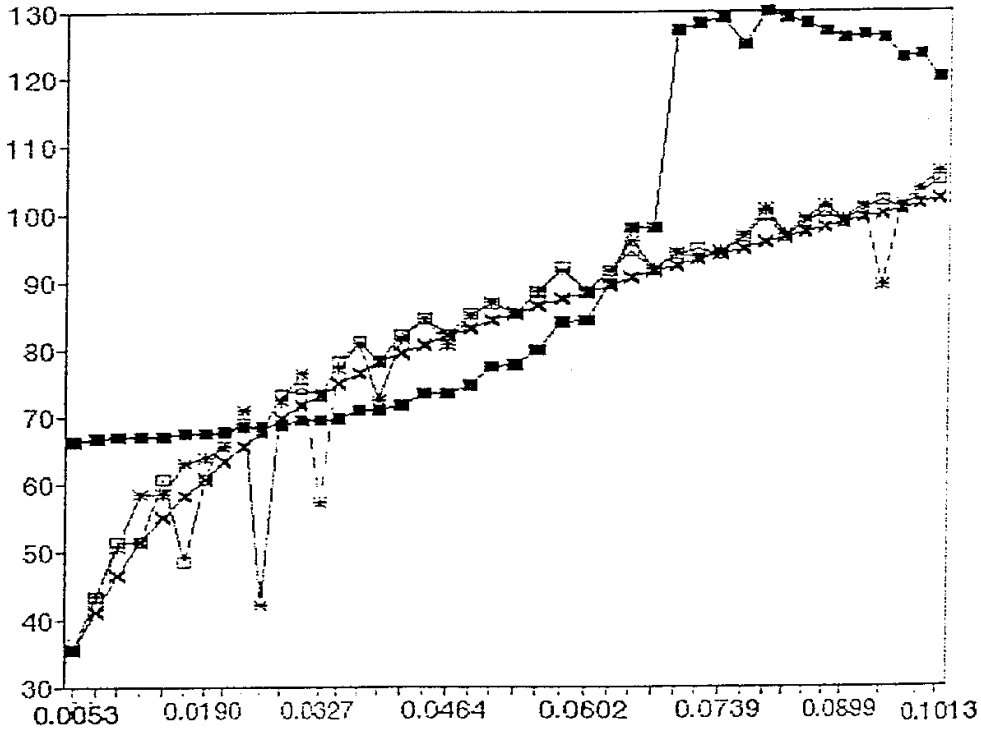


Figure 11 : The common intersections of the previuos methods at concentration 30%

x - Iterated one dimensional splines
□ - Lénárd's method with pre-approximation
* - Lénárd's method with linear transformations
✱ - Shepard's original formula

Table 2 : The comparison table of the methods at cc=30%

Pressure MPa)	Iterated 1-dim. splines	Lénárd's method transform	Lénárd's method preapprox	Shepard's method 4x4 points
0.0053	35.500	35.500	35.500	66.400
0.0075	41.170	43.171	43.412	66.684
0.0098	46.544	50.842	51.325	67.128
0.0121	51.325	58.514	51.325	67.057
0.0144	55.228	58.439	60.848	67.088
0.0167	58.287	63.110	58.645	67.643
0.0190	60.848	64.132	60.848	67.758
0.0213	63.263	65.667	67.924	67.796
0.0235	65.659	71.207	69.181	68.469
0.0258	67.924	72.040	67.924	68.623
0.0281	69.949	72.321	73.398	68.803
0.0304	71.745	76.336	74.332	69.525
0.0327	73.398	67.338	73.398	69.526
0.0350	74.989	77.282	78.004	69.885
0.0373	76.531	80.892	81.033	71.040
0.0395	78.004	72.492	78.004	71.110
0.0418	79.389	81.580	81.918	71.779
0.0441	80.690	84.670	84.502	73.421
0.0464	81.918	80.700	81.918	73.503
0.0487	83.083	85.149	85.300	74.702
0.0510	84.203	86.962	86.679	77.369
0.0533	85.300	85.300	85.300	77.600
0.0555	86.386	88.516	88.517	79.769
0.0578	87.461	91.517	91.998	84.059
0.0601	88.517	88.744	88.517	84.330
0.0624	89.545	91.671	91.464	89.586
0.0647	90.532	95.973	94.660	98.145
0.0670	91.464	91.791	91.464	98.030
0.0693	92.337	94.304	94.009	127.24
0.0715	93.176	93.318	94.812	206.38
0.0738	94.009	94.434	94.009	233.99
0.0761	94.849	96.884	96.488	135.71
0.0784	95.682	100.68	99.696	118.89
0.0807	96.488	97.039	96.488	103.48
0.0830	97.257	99.212	98.701	115.56
0.0853	97.994	101.17	100.29	103.49
0.0875	98.701	99.276	98.701	101.57
0.0898	99.387	101.29	100.73	106.52
0.0921	100.06	99.410	101.80	101.31
0.0944	100.73	101.37	100.73	101.09
0.0967	101.42	103.55	102.80	104.97
0.0990	102.11	106.03	104.88	102.74
0.1013	102.11	102.80	102.80	100.03

Pressure (MPa)	measured values
0.0053	35.5
0.0133	53.4
0.0202	62.1
0.0266	68.6
0.0333	73.8
0.0399	78.2
0.0466	82.0
0.0533	85.3
0.0599	88.4
0.0666	91.3
0.0733	93.8
0.0799	96.2
0.0866	98.4
0.0933	100.4
0.1013	102.8

REFERENCES

[DRM] Djuric, M., Ranogajec, J., Marinkov-Neducin, R.: *Predicting of Floor Tiles Behaviour by Using Two-Dimensional Spline Interpolation,* Hung. J. Ind. Chem., 20 (1992), pp. 39-44

[GW] Gordon, W. J., Wixon, J. A.: *Shepard's method of "metric interpolation" to bivariate and multivariate interpolation,* Math. of Computation 32 (1978) 253-264

[K] Kiss, B. : *Növényolajipari és háztartási vegyipari táblázatok,* Mezőgazdasági Kiadó, Budapest, 1988 (in Hungarian)

[L1] Lénárd, M.: *Spline Interpolation in Two Variables,* Studia Sci. Math. Hung., 20 (1985), pp. 145-154

[L2] ———— : *On the two dimensional spline interpolation of Hermite-type,* Colloquia M. Soc. J. Bolyai, vol. 49, 1985, pp. 531-541

[L3] ———— : *On an n - dimensional Quadratic Spline Approximation,* J. Approx. Theory 68 (1992), pp. 113-135

[PFTV] Press, W. H., Flannery, B. P., Teukolsky, S. A., Vetterling, W. T.: *Numerical Recipes - The Art of Scientific Computing,* Cambridge Univ. Press, 1985

[S] Shepard, D.: *A two dimensional interpolation function for irregularly spaced data,* Proc. 1968. ACM Nat. Conf, 517-524

[SVPK] Saburov, A. G., Vasilienec, I. M., Postolov, Ju. M., Klibov, N. P.: *Numerical Formulas for Thermodynamic Properties of Aqueous Glycerol Solutions,* Maslo-Chim. Promusl., 1987, pp. 21-22 (in Russian)

[Sz] Szalkai, I.: *Improving Shepard's Method,* in preparation