第7卷 第2期　　自动化学报　　Vol. 7, No. 2
1981年4月　　ACTA AUTOMATICA SINICA　　April, 1981

字符方程初探*

蒋卡林
（中国科学院沈阳自动化研究所）

摘　　要

本文提出 α 指数技巧以建立种种字符状图样的单一方程，说明一般常用字符的近似的单一方程式都可列出来. 最后，还指出了数据与方程的符合验算，并进行若干讨论.

一、引　言

本文提出用单一的方程式描述一简单图案、一个常用字符或一个简单的物景轮廓图，以利于自动绘制、识别或分析. 一般常用印刷体字符，经接收和预处理细化过程后，变成为线条图. 我们只要对各字符形状作出最合理的规定，那么，这些线条图是可以用单一的方程式表示的. 这样一来，各种字符或图案的大量信息储存就可以用方程式的程序语句代替了. 至于如何用单一的方程式表示，这是一项数学技巧，将在后面讲解.

用方程式表示字符的另一价值在于能使字符精确化、标准化. 目前各国所出产的各种印字装置的字符形状都尚未统一，而且不少字符是未精确化的. 制造商或有关人员进行有关字符方面的种种实际工作时，必须首先向国家标准局购取巨册的字符图纸拷贝. 现在，我们有了字符方程以后，这种拷贝就不再需要了.

目前，无论在国内或国外都尚未做到字符的统一. 这一件事也许是好事，因为我们还有机会采用我们认为是更为合理的例如最便于编入计算机程序的种种字符图样. 当然，我们也不故意标新立异. 在设计字符时尽量作到传统性与科学性兼顾.

二、字符区和字符部位

安排或设计字符区和字符部位时，应考虑当前的现实和今后的发展. 当前各国所用的各种打字机、高速印字机、现金记录器、加法器和信贷卡片印字机等的字体有大有小，间距也不统一. 这是允许的. 但我们应随着识别技术的发展，逐步地改用较接近于正规书刊的那种字体，即应使字体稍许细高一些，间距也稍许减一些.

本文对字型的标准尺寸不作轻意的规定，而只对字高和字宽的比值作出规定，以供参考. 这比值的规定，在字符方程式的推导上是必需的.

作者经慎重考虑、研究后，建议国际标准组织（ISO）考虑采用图1所示的矩形字符

* 本文修改稿于1980年9月29日收到.

区. 每一字符区包括10行4列. 坐标横轴位于4,5行之间,纵轴则位于2,3列之间. 这图还表示各种字符的部位. 小写字母占行最多,共占10行. 各小写字母,都像 b d p q 一样,可划分成:

1) 上部;即芽部,占3行;
2) 中部;即豆部,占5行;
3) 下部;即根部,占2行.

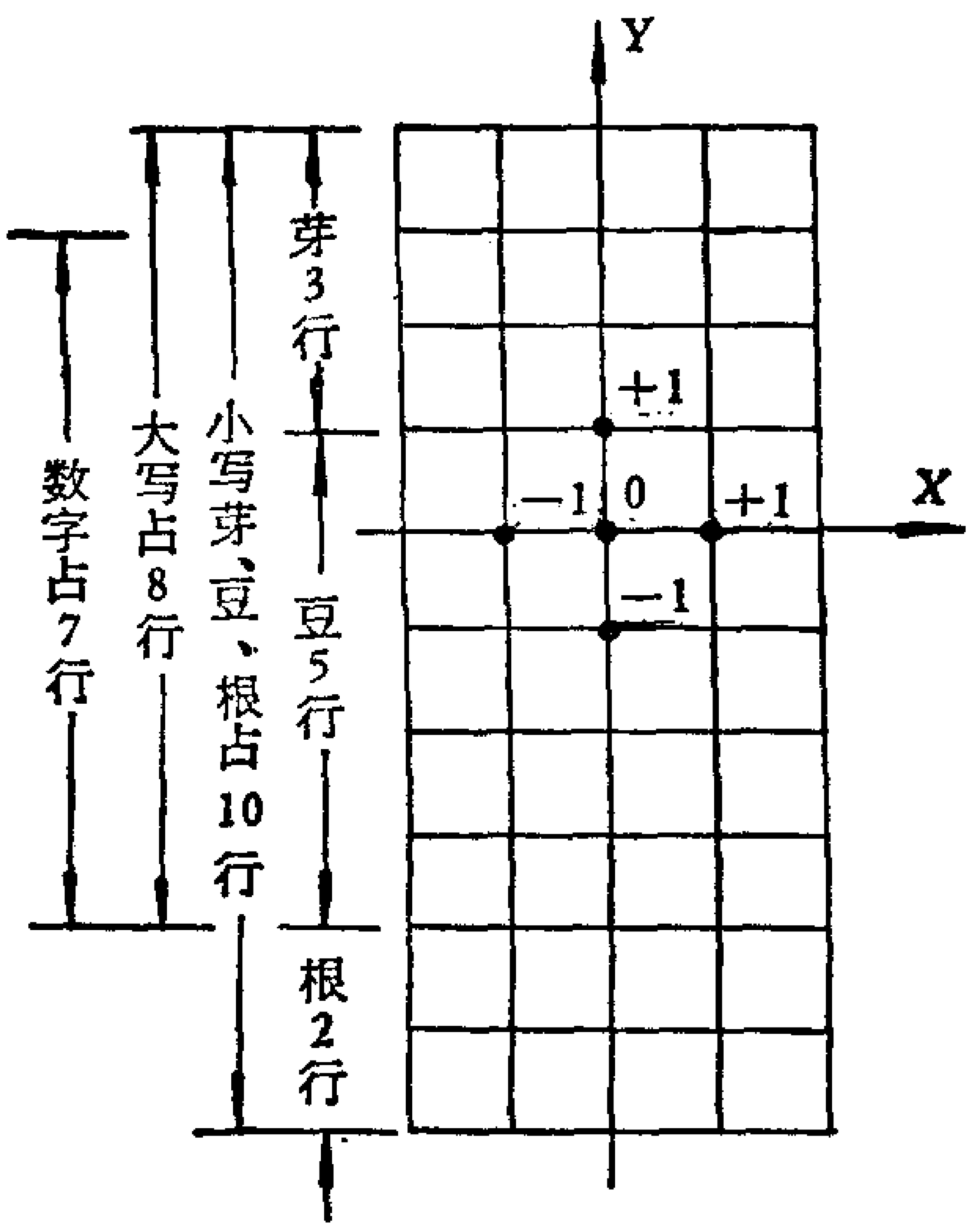

图1 字符区与坐标

至于根部取消问题,那是今后文字改革的事,暂且不谈.

为了避免字符边缘被抹去,各字符区的上下左右都应留出0.2的空隙.

此外还应避免字符的互相包含. 例如,大写字母O占8行,数字0占7行,这就可避免它们互相包含以致引起误识. 总之,为了避免某一字符(例如C)的数据碰巧能满足另一字符(例如O)方程,设计字符时,要绝对避免一字符刚好能嵌进另一字符内的可能性.

三、O类字符方程

现在我们首先列出最简单的O类字符的方程,以供进一步研究,讨论. 这类字符的图样,都是采用一个圈或者两个圈. 采用一个圈的三字符是最容易混淆的. 例如,欧洲国家的旧标准中的大写O和数字零就非常相像. 下列三式和其相应表示的图样是绝不会混淆的.

大写字母O:
$$(x/1.8)^2 + (y/3.8)^2 - 1 = 0 \tag{1}$$

数字0:
$$\left(\frac{x}{1.8}\right)^{2.3} + \left(\frac{y+0.4}{3.4}\right)^{2.3} - 1 = 0 \tag{2}$$

小写字母o:
$$\left(\frac{x}{1.8}\right)^{2.6} + \left(\frac{y+1.5}{2.3}\right)^{2.6} - 1 = 0 \tag{3}$$

字符□:
$$(x/1.8)^8 + (y/3.8)^8 - 1 = 0 \tag{4}$$

字符8:
$$[x^2 + (y-1.5)^2 - 1.5^2]\left[\left(\frac{x}{1.8}\right)^{2.3} + \left(\frac{y+1.9}{1.9}\right)^{2.3} - 1\right] = 0 \tag{5}$$

四、关于字符方程的实际应用

字符方程的第一个价值是能将字符或图样直接编入计算机程序内,从而避免巨量的图象信息储存.

字符方程的第二个价值是不依靠大量的图纸拷贝而能做到字符图形标准化、精确化.

在字符识别中如何应用字符方程的问题,应根据具体情况确定. 当我们识别某几个

具有非常突出特征的印刷体字符例如 1 和 L 时，也许不必依靠字符方程．用它去识别它们，可能是杀鸡用牛刀了．但在容易误识的情况下，字符方程是大有可为的．

字符方程不仅可应用于易混淆文字的识别，还可应用于不清洁文字的识别以及附有花样背景文字的识别．这类文字由人去识别是非常容易的，而由机器去识别目前还不可能．有了字符方程以后，就出现了由机器识别这种字符的可能性．可由机器判断在被扫描的图片上，是否有许多密接的点恰巧或相当近似地满足某一字符方程，而其余的点则形成别的陌生图样或零乱地分布着，无两个以上的点相连贯的现象．

基于事实上的需要，我们应教会计算机：

1）字符方程图象内可能存在的极个别的孤立点不要求去满足；

2）实际图象内可能存在的多余的几个孤立点不必去计较．

字符方程的目的当然不只限于识别．所以字符不论繁简，统统都可以方程式化，以准备作字符标准化、字符识别以及其它各种用途之所需．

五、α 指数技巧[1)]

这个数学技巧很简单，但很有用．它可叫作“无穷大偶指数法”或“无穷大偶指数术”．现在说明一下它是怎样引出来的．作者在逐步地研究前面的方程（1）、（2）、（3）、（4）后，发现方次愈高就愈接近于矩形．如果把（4）式的指数 8 提高到真正的无穷大，即写成为

$$(x/1.8)^{\alpha}+(y/3.8)^{\alpha}-1=0 \tag{6}$$

那么，就得到真正的矩形．作者经过一番分析后，发现某些多角形也可用类似的方程式表示．因此，为了描述各种各样的断断折折的图形，我们不妨普遍地采用无穷大偶数 α 作为各种函数的指数．我们之所以采用无穷大偶数而不采用无穷大奇数，这是因为无论正数或负数的偶数次方都等于正数，处理起来比较方便．当然，无穷大并不是一个单一的数，而是各种无限地增长的数列的最终极限．从工程实用上说，我们也不需要真正的无穷大，而是相当大的数就行了．例如（4）式的指数是 8，并不等于真正无穷大，但这式的图象已近似是矩形了．从理论上说，我们应该以真正的无穷大偶数作为指数，以产生种种断折图形，或者说字符状图形．不过，α 这个指数也可以看作是一个单纯的符号或记号．当计算机或智能机械遇到这个符号时，它就立即检查 α 下面的底数的绝对值是否小于 1．如果小于 1，就判定该项消失，变为零；如果大于 1，则该项变为无穷大，使得此方程式在有穷数域内无法成立．所以说，α 这个数是不奇特的．它是一个很有用的数学符号，可以用它来写方程，也可以将它编入计算机程序内．

六、一些易表达的字符方程

为了帮助建立简单而明确的概念，先讲解一些简例：

1）加号方程

$$(x/1.8)^{\alpha}+(y/3)^{\alpha}=xy \tag{7}$$

1）注：本文内的 α 最好不要称为无穷大偶数，而应该看作是一个极限，即 $\alpha=\lim 2n$，式中 n 代表正整数．

在 $|x| < 1.8$ 和 $|y| < 3$ 的矩形范围内，$0 = xy$，即 $x = 0$ 或 $y = 0$．所得图形是 $x = 0$ 及 $y = 0$，相当于一个加号．

2）减号方程和等号方程

可以取横轴的一段为减号，方程如下：

$$(x/1.8)^\alpha = y \tag{8}$$

显然，当 $|x| < 1.8$ 时，$y = 0$．

分析结果表明，从实用上说，我们所得到的图形就可以作为一个减号．它是横轴的一段，长 2×1.8．

同理，还可列出等号方程如下：

$$(x/1.8)^\alpha = y(y + 1) \tag{9}$$

3）短竖线方程．

可以取纵轴的一段为这个字符，其方程如下：

$$(y/3)^\alpha = x \tag{10}$$

4）正、反斜线方程、乘号方程和字符✱方程

/——正斜线方程：　$(x/1.8)^\alpha = y - x$　(11)

\\——反斜线方程：　$(x/1.8)^\alpha = y + x$　(12)

×——乘号方程：　$(x/1.8)^\alpha = (y - x)(y + x) = y^2 - x^2$

✱字符方程：　$\left(\dfrac{x}{1.8}\right)^\alpha + \left(\dfrac{y}{1.56}\right)^\alpha = y(y^2 - 3x^2)$　(13)

5）字符◇方程

将正方形方程

$$(x/a)^\alpha + (y/a)^\alpha = 1 \tag{14}$$

进行相当于旋转 45° 的坐标变换，得：

$$\left(\frac{x + y}{\sqrt{2}\,a}\right)^\alpha + \left(\frac{x - y}{\sqrt{2}\,a}\right)^\alpha = 1 \tag{15}$$

若要求适合所选的字符区，则列出：

$$\left(\frac{x + y}{1.8}\right)^\alpha + \left(\frac{x - y}{1.8}\right)^\alpha = 1 \tag{16}$$

6）除号方程

可以取字符◇的图形为切割范围，并从 $x = 0$ 和 $x = \pm 1.7$ 这三条水平线切割出一个除号而得：

$$\left(\frac{x + y}{1.8}\right)^\alpha + \left(\frac{x - y}{1.8}\right)^\alpha = x(x + 1.7)(x - 1.7) \tag{17}$$

7）符号＋－×÷／＼✱◇下降半行

我们所设的横轴恰巧穿过各大写字母和各无根小写字母的中心，很好．但是对于稍矮的数字来说，有些欠妥当．既然 10 个数字比大写字母矮一行，那么，常位于数字间的上述各符号也应相应地下降半行．办法是在上述各方程式内，以 $(y + 0.5)$ 代替 y．

8）．百分比号方程

$$\left(\frac{x}{1.8}\right)^{\alpha} = (y - 1.25x + 1.5)[(x + 0.8)^2 + (y + 0.2)^2 - 0.8^2][(x - 0.8)^2 + (y + 2.8)^2 - 0.8^2] \tag{18}$$

9) 左右圆括号方程

$$(x/0.8)^{\alpha} = (x \mp 3.8)^2 + (y + 0.5)^2 - 4.5^2 \tag{19}$$

10) 小于号方程和大于号方程

$$(x/1.8)^{\alpha} = (y \mp 0.43x - 0.2)(y \pm 0.43x + 1.2) \tag{20}$$

11) Z 和 z 的方程

这两字符由直线组成，左右两边砍齐．例如 Z 方程：

$$\left(\frac{x}{1.8}\right)^{\alpha} = \left(y - \frac{3.8}{1.8}x\right)(y - 3.8)(y + 3.8) \tag{21}$$

12) N V W X 和 v w x 的方程

这七字符也都由直线组成，上下两端砍齐．例如 X 方程：

$$\left(\frac{y}{3.8}\right)^{\alpha} = \left(y - \frac{3.8}{1.8}x\right)\left(y + \frac{3.8}{1.8}x\right) \tag{22}$$

七、字符区为矩形的一般字符方程

在领会上述特例的基础上，再进行一般性的了解．一般的矩形框架方程是

$$(x/a)^{\alpha} + (y/b)^{\alpha} = 1 \tag{23}$$

这矩形框架宽 $2a$，高 $2b$．现将这式内的 1 改为一般函数，并写成乘积形式：

$$\left(\frac{x}{a}\right)^{\alpha} + \left(\frac{y}{b}\right)^{\alpha} = \prod_{k=1}^{N} f_k(x, y) \tag{24}$$

这就是矩形字符一般方程．

证明如下：我们首先研究下列范围

$$|x| < a, \quad |y| < b$$

这式成立时，(24) 式就变成为：

$$\prod_{k=1}^{N} f_k(x, y) = 0 \tag{25}$$

于是 $f_1 = 0$ 或 $f_2 = 0$，…或 $f_N = 0$ 成立．可见，(24) 式在所指矩形范围内的图象也就是局限于所指矩形范围内的 (25) 式的图象． 它应包括 $f_1 = 0$ 的图象及 $f_2 = 0$ 的图象及…… $f_N = 0$ 的图象． 这些图象都同时出现，有时互相重叠，它们都出现在所指矩形范围内．在矩形范围外的有穷数域内，则不会出现图象，因为头两项或其中之一会变成为正无穷大实数，于是当 x 和 y 都取有穷值时，(24) 式是不会成立的，但偶尔出现的极个别的孤立点除外．换句话说，在矩形框架外是不会出现任何线条的．（在前面已说过，我们预先教会计算机，在进行自动识别时，对极个别的细微的孤立点不予理睬.）

上述结论是令人鼓舞的．可以说，现在我们已进入到了一个非常自由的境地． 我们可以在一块长 $2a$ 高 $2b$ 的"矩形土地"上，随心所欲地拼凑各种图形，使之符合各种字符的

形状.

八、一些难表达的字符方程

字符难于表达的原因之一是不允许字符左右切齐或上下左右都切齐，例如 Y、A、E、F、J、M、3、☆、e、G. 这时我们可以采取斜直线或种种曲线为字符边界线. 下面是一个典型的例子.

我们以字符M为例来加以说明. 这字符的下边界线应该是向上凸的曲线，而且，这曲线应向左右延伸，否则(例如用抛物线)就无法切割出这样一个字符. 经过一番分析后，确定以下面两式为边界线:

下线: $$y = 2.14\cos\left(\frac{\pi}{3}x\right) - 3.14$$

上线: $$y = 2.14\cos\left(\frac{\pi}{3}x\right) - 3.14 + 7.6 = 2.14\cos\left(\frac{\pi}{3}x\right) + 4.46$$

两边界线之间的区域是

$$\left|y - 2.14\cos\left(\frac{\pi}{3}x\right) - \frac{4.46 - 3.14}{2}\right| < \frac{4.46 + 3.14}{2}$$

$$\left|y - 2.14\cos\left(\frac{\pi}{3}x\right) - 0.66\right| < 3.8$$

即:

$$\left|0.263y - 0.564\cos\left(\frac{\pi}{3}x\right) - 0.174\right| < 1$$

于是，可取得一合适的边界函数，而求得字符M的方程式如下:

$$\left[0.263y - 0.564\cos\left(\frac{\pi}{3}x\right) - 0.174\right]^{\alpha} = (y + 1 - 2.5x)(y + 1 + 2.5x)(x - 1.8)(x + 1.8) \tag{26}$$

按照类似的方法，还可推导出一个精确的五角星的方程式:

$$(y - 0.619)(y - 3.077x - 2)(y + 3.077x - 2)(y - 0.7335x + 0.763) \cdot (y + 0.7335x + 0.763) = [0.586(x^2 + y^2) - 1.341]^{\alpha} \tag{27}$$

九、在规定容差下数据与方程的符合验算

这个问题有普遍意义. 科技工作者在获得一批数据后，若要检验是否符合规律，就得对数据逐一地进行验算. 这验算完全可依靠计算机，但是人必须预先教会计算机.

按字符方程识别字符，也确实需要一个识别准则. 我们不能说，代入方程式完全对就算是识别了. 而应该说，对到何种程度就算作识别. 如果我们对 x 偏差 Δx 加以某数值限制以及对 y 偏差 Δy 加以同数值的限制，这显然是不妥当的. 例如，当字符的笔画接近平行 Y 轴时，偏差 Δy 是允许很大很大的.

下面提出一个简单的数据验算法: “二点至八点法”，或简称“八点法”. 大意是，对于

每一数据 (x_K, y_K)，应至少检验表 1 内的头两点，而充其量检验八点，就能判定该数据是否符合方程. 表 1 内的 r 代表容差，即数据点与理想曲线之间的最大容许距离. 本文内各字符方程的容差一律规定为 0.2.

表 1　八点法的各点

1	$x_K + r$	y_K
2	$x_K - r$	y_K
3	x_K	$y_K + r$
4	x_K	$y_K - r$
5	$x_K + r\sqrt{2}$	$y_K + r\sqrt{2}$
6	$x_K - r\sqrt{2}$	$y_K - r\sqrt{2}$
7	$x_K + r\sqrt{2}$	$y_K - r\sqrt{2}$
8	$x_K - r\sqrt{2}$	$y_K + r\sqrt{2}$

这方法要求一切方程都写成为右边等于零的形式，如 $f(x, y) = 0$ 或 $y - \phi(x) = 0$. 当得到任一数据 (x_K, y_K) 后，我们首先将表内头一点代入方程，若发现 $f<0$. 然后将表内第二点代入方程，这时若发现 $f > 0$，即 f 变了号，则检验可停止. 这时肯定 (x_K, y_K) 与理想曲线距离未超过容差 r，即认为这数据是符合方程的. 若八点检验完毕时，f 仍未变号，则肯定该数据不符合方程.

当我们按字符方程进行这种检验时，若发现 f 从无穷大值跳至有穷值，或从有穷值跳至无穷大值. 这时即使 f 变号，也应该认为该数据不符合方程，即只符合字符方程的某些多余的虚图象，而不符合真正的笔画.

例如，我们只要检验表内头两点就马上发现 (1.81, 1.11) 不符合加号方程. 又例如，只需要检验头三点就能肯定数据 (−0.60, 3.40) 符合大写字母 O 的方程.

其实，对于某些离字符笔画相当远的点，就不必徒劳无益地作八点检验了. 例如当识别加号时，在双曲线 $xy = \pm r \times 3 = \pm 0.6$ 外面的点，如刚才提到的点 (1.81, 1.11)，就根本不必检验.

上述方法是比较简单而实用的，比作者原先提出的二准则(包络线准则和附校正项的直角三角形准则)要方便得多.

十、讨　　论

通常我们在进行印刷字符的识别前，先存储其标准字符图形，存储其模板点阵. 每一模板需数百或上千个存储单元. 总字符存储量非常大，因而在信息处理上甚感不便. 而字符方程可直接编成为程序语句，便于处理.

至于按字符方程进行字符标准化问题，不知是否能行. 目前国内外均尚未做到字符统一[1]. 作者希望将来最好能统一地采用既美观又便于拟定其字符方程的字符.

样条函数也许是描述字符的一条途径. 但这种函数只能表示一条相当光滑的无限长曲线. 折断或砍断后的曲线以及互相重迭起来的图象是无法用样条函数表示的.

本文提出了八点法，以便检验数据与标准方程是否符合. 至于进行印刷字识别时，究竟按本文的八点法为好，还是按类比度方法为好，尚值得研究. 当然有了字符方程的程序语句后，也可以计算出输入字符与理想字符之间的类比度.

1) 据国际标准 ISO1073/II 前言称，英、法、德、意、日等十九国投赞成票，而美国投反对票. 又据美国国家标准 ANSI 3.49-1975 (OCR-B) 附录 A 称，欧洲国家的 OCR-B 与美国的 OCR-B 是不同的，且美国的 OCR-B 也有变更. 我国字符统一使用问题曾在 1978 年 4 月初的无锡会议上讨论过. 最后并没有取得一致的意见.

本文提出的字符方程是单一方程. 写出一个三角形或一个五角星的单一方程，这在数学上尚属首次. 这对数学工作者是否有一定参考意义?

本文只介绍了一些基本内容. 各种字符方程究竟如何拟定，还有待进一步研究，许多问题的解决还要靠大家帮助.

FIRST EXPLORATION FOR CHARACTER EQUATIONS

JIANG KALIN

(Shenyang Institute of Automation, Academia Sinica)

Abstract

This paper proposes an α-exponential technique for the establishment of equations of various character-like patterns, and indicates the possibility to formulize a single equation representing approximately a common character. Finally, a method for conformity check of data and equation is given, and some other questions are discussed.