卷积的快速定限求解

罗 永 光

提　要　本文提出了卷积的快速定限表，分析了该表的特点，叙述了该表的拟订方法和用该表求解卷积的步骤。用该表求解卷积比之用历来的折迭图解定限方法要简单省事而快捷得多。文中还在该表的基础上提出了一种在卷积运算中避免容易产生的疏忽出错的有效措施。

一、引　　言

在卷积的运算中，正确地确定积分（或求和）的上下限及其定义域是十分关键的步骤，稍不留意就会出错。以连续函数的卷积为例，历来进行此项工作都是在 τ 轴上，借助图解，按照“两函数在 t 轴的定义域——τ 轴折迭——逐步沿 τ 轴移动——逐段判断并返回 t 轴定出定义域——逐段定限求积分结果”这一程式完成的。在文献中虽也见过对这一问题的总结归纳，但都离不开上述从 t 轴到 τ 轴，再从 τ 轴转回 t 轴的程式，求解过程依然繁复冗长，且描述不直观，依然难于从两函数的定义域直接洞察各段积分的上下限和定义域[1,2]。本文将提出一种新的程式以克服上述的不便。

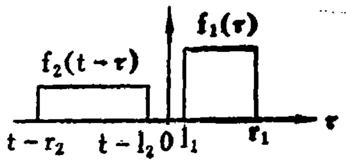
二、卷积积分的快速定限表

考虑两个有始有终函数相卷积的一般情况。如果我们采用以下的约定，就将使定限工作带上很强的规律性：设 $r_2-l_2\geqslant r_1-l_1$，其中 $[l_1,r_1]$ 和 $[l_2,r_2]$ 分别为 $f_1(t)$ 和 $f_2(t)$ 的定义域，并定义 τ 的不定积分为：

$$f(\tau)=\int f_1(\tau)f_2(t-\tau)d\tau \tag{1}$$

此即卷积积分未定限之前的形式。此时折迭图解定限情况如图1所示。

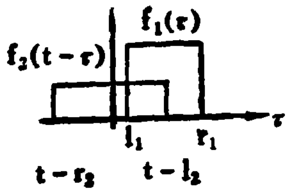
本文1982年3月31日收到



(1) 当 $t-l_2\leqslant l_1$

即 $t\leqslant l_1+l_2$ 时

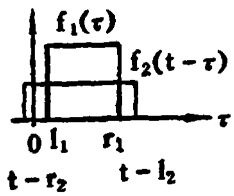
$f(t)=0$



(2) 当 $l_1\leqslant t-l_2\leqslant r_1$

即 $l_1+l_2\leqslant t\leqslant r_1+l_2$ 时

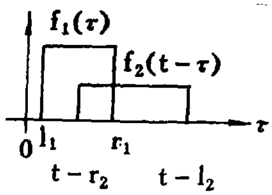
$$f(t)=\int_{l_1}^{t-l_2} f_1(\tau)f_2(t-\tau)d\tau$$



(3) 当 $t-r_2\leqslant l_1$ 且 $t-l_2\geqslant r_1$

即 $r_1+l_2\leqslant t\leqslant l_1+r_2$ 时

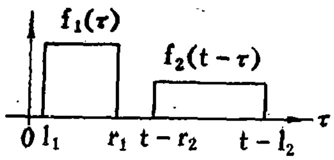
$$f(t)=\int_{l_1}^{r_1} f_1(\tau)f_2(t-\tau)d\tau$$



(4) 当 $l_1\leqslant t-r_2\leqslant r_1$

即 $l_1+r_2\leqslant t\leqslant r_1+r_2$时

$$f(t)=\int_{t-r_2}^{r_1} f_1(\tau)f_2(t-\tau)d\tau$$



(5) 当 $t-r_2\geqslant r_1$

即 $t\geqslant r_1+r_2$ 时

$f(t)=0$

图 1 一般情况下卷积积分限的图解确定

由图1，我们可归纳出卷积积分的快速定限表，如表 1。

卷积积分的快速定限表 **表 1**

卷积积分	0		$\int_{l_1}^{t-l_2}$		$\int_{l_1}^{r_1}$		$\int_{t-r_2}^{r_1}$	0
$f(t)$ ——		|		|		|		| → t
定 义 域		$t_1=l_1+l_2$		$t_2=r_1+l_2$		$t_3=l_1+r_2$		$t_4=r_1+r_2$

快速定限表有如下的规律

1. 卷积结果的定义域一般地分为五段，由四个断点将 t 轴切割而成，这四个断点完全由两函数的定义域决定，有如下的关系：

t_1＝两函数的左时限之和

t_4＝两函数的右时限之和

$t_2-t_1=t_4-t_3=f_1(t)$ 的时长

t_4-t_1＝两函数时长之和

t_3-t_2＝两函数时长之差

$t_4-t_2=t_3-t_1=f_2(t)$ 的时长

如 $f_1(t)$ 的时长$=f_2(t)$ 的时长，则 t_2，t_3 两点重合，中间一段消失，卷积结果只出现四

段。

2．在卷积表达式中 $f_1(\cdot)$ 和 $f_2(\cdot)$ 函数括号内的值分别为 τ 和 $t-\tau$，而在积分限中，则出现 $f_1(t)$ 的时限的就为其时限值，出现 $f_2(t)$ 的时限的，则为 t －其时限值。在 τ 和函数的时限值之间存在着有趣的对应关系。

3．最边上的两段结果都为零。中间一段积分的上下限分别为 $f_1(t)$ 的上下时限，右边一段积分的上限和左边一段积分的下限也分别为 $f_1(t)$ 的上下时限。其余两积分限则是 $t-f_2(t)$ 的时限值，其左时限出现在左边一段，而其右时限则出现在右边一段。

4．在左边一段，积分限中只出现两函数的左时限，而在右边一段，则只出现两函数的右时限。

三、快速定限求解卷积积分的步骤

如 $f(t)=f_{长}(t)*f_{短}(t)$，其中长短的意思是 $f_{长}(t)$ 的时长 $\geqslant f_{短}(t)$ 的时长，则其快速定限求解可按下列步骤进行：

1．求出 τ 的不定积分 $f(\tau)=\int f_{短}(\tau)f_{长}(t-\tau)d\tau$

2．将 $f_{长}(t)$ 的两个时限值分别与 $f_{短}(t)$ 的两个时限值相加，并将得出的四个数从小到大依次地排列在水平画出的 t 轴上。于是 t 轴被分为五段，这就是五段卷积结果的定义域。

3．在最边上的两段上部写出卷积结果——零。在其余三段上部分别写出卷积积分，其中中间一段的上下限分别为 $f_{短}(t)$ 的上下时限；右边一段的上限与中间一段同，其下限为 $t-f_{长}(t)$ 的右时限；左边一段的下限与中间一段同，其上限为 $t-f_{长}(t)$ 的左时限。于是快速定限表制订完毕。

4．按 1 得出的 $f(\tau)$ 的函数形式和快速定限表中的上下限，分别求出中间三段的结果。

如熟练了，则可将 3、4 两步合并，在 3 中即可直接写出最终结果。

在函数为分段定义的場合，则可将一函数的每一段与另一函数的每一段按以上步骤进行卷积，然后再分段汇总求和。下面以例说明：

例 1　已知

$$f_1(t)=\begin{cases}\phi_1(t) & 3\leqslant t\leqslant 6\\ \phi_2(t) & 6\leqslant t\leqslant 7\\ 0 & 其余\end{cases} \tag{2}$$

$$f_2(t)=\begin{cases}\psi(t) & 0\leqslant t\leqslant 2\\ 0 & 其余\end{cases} \tag{3}$$

求 $f(t)=f_1(t)*f_2(t)$

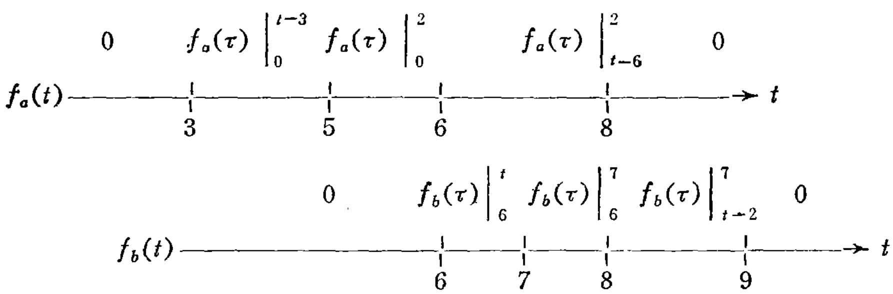
解　第一步，在本题中，$\phi_1(t)$ 的时长大于 $\psi(t)$ 的时长，而 $\phi_2(t)$ 的时长小于 $\psi(t)$ 的时长，故应求出下列的不定积分：

$$f_a(\tau)=\int\phi_1(t-\tau)\psi(\tau)d\tau \tag{4}$$

$$f_b(\tau)=\int\phi_2(\tau)\psi(t-\tau)d\tau \tag{5}$$

第二步，制订快速定限表如表 2 所示：

例 1 的快速定限求解 **表 2**



第三步，由表 2 分段汇总求和得：

$$f(t)=\begin{cases} 0 & t\leqslant 3 \\ f_a(\tau)\Big|_0^{t-3} & 3\leqslant t\leqslant 5 \\ f_a(\tau)\Big|_0^{2} & 5\leqslant t\leqslant 6 \\ f_a(\tau)\Big|_{t-6}^{2}+f_b(\tau)\Big|_6^{t} & 6\leqslant t\leqslant 7 \\ f_a(\tau)\Big|_{t-6}^{2}+f_b(\tau)\Big|_6^{7} & 7\leqslant t\leqslant 8 \\ f_b(\tau)\Big|_{t-2}^{7} & 8\leqslant t\leqslant 9 \\ 0 & t\geqslant 9 \end{cases} \tag{6}$$

到此已实际上完成了卷积运算。如题中 $\phi_1(t)$、$\phi_2(t)$ 和 $\psi(t)$的函数形式明确给出，则可在快速定限表及 (6) 式中就写出实际结果。

由上述可看出，卷积的快速定限求解有如下的特点：

1. 三集中，即集中求不定积分，集中确定积分限和定义域，集中代限求结果。有很强的系统性和整体预见性。从而克服了折迭图解方法逐步重复进行判断，定限、解积分（也是先求不定积分后代限）等手续的烦琐性和逐步走着瞧的盲目性。

2. 定限工作模式化，事先心中有数。比如，卷积一般地是分为五段，如少了一段或数段，就意味着两函数时长相等或函数时限中有为∞或—∞者，否则就表明有遗漏。而如怀疑定限有错，则可按积分限出现的规律在快速定限表中去查找。因此是一种容易检错的方法。

3. 利于使用计算机编程计算。快速定限表一旦制订好，计算程序的初置段就可定出。余下的工作，在不定积分能解析得出的場合，仅仅是分段调用代限子程序，而在不定积分不能解析得出的場合，也仅仅是分段代限并调用定积分子程序。这一特点在离散卷积中具有更大的意义。

关于快速定限求解与历来的折迭图解法的比较，可参见附录。

四、时长无限的函数之卷积积分

当函数具有无限时长时，情况稍特殊。以下列出，以供参考。

1. 两因果性函数的卷积

因果性函数的卷积定限表　　**表 3**

设 $f_1(t)$ 和 $f_2(t)$ 的定义域分别为 $[t_1,\infty)$ 和 $[t_2,\infty]$。有

$$0 \qquad \int_{t_1}^{t-t_2} f_1(\tau)f_2(t-\tau)d\tau \text{ 或 } \int_{t_2}^{t-t_1} f_1(t-\tau)f_2(\tau)d\tau$$

t_1+t_2　　t

其积分上限总是 t －时限的形式，而下限以时限值单独出现。此表符合“二”所述的规律。

2. 两反因果性函数的卷积

反因果性函数卷积的定限表　　**表 4**

设 $f_1(t)$ 和 $f_2(t)$ 的定义域分别为 $(-\infty,t_1]$ 和 $(-\infty,t_2]$，其 定 限表亦只分两段。

$$\int_{t-t_1}^{t_2} f_1(t-\tau)f_2(\tau)d\tau \text{ 或 } \int_{t-t_2}^{t_1} f_1(\tau)f_2(t-\tau)d\tau \qquad 0$$

t_1+t_2　　t

其积分上限都以时限值单独出现，而下限总是 τ －时限的形式。此表亦符合“二”所述规律。

3. 因果性函数和反因果性函数的卷积

因函数与反因函数卷积的定限表　　**表 5**

设 $f_{因}(t)$ 和 $f_{反因}(t)$ 的定义域分别为 $[t_{因},\infty)$ 和 $(-\infty,\ t_{反因}]$。这种情况较特殊，其定限如下：

$$\int_{-\infty}^{t-t_{因}} f_{因}(t-\tau)f_{反因}(\tau)d\tau \qquad \int_{t_{反因}}^{t-t_{因}} f_{因}(t-\tau)f_{反因}(\tau)d\tau$$

$t_{因}+t_{反因}$　　t

或：

$$\int_{t-t_{反因}}^{t_{因}} f_{因}(\tau)f_{反因}(t-\tau)d\tau \qquad \int_{t-t_{反因}}^{\infty} f_{因}(\tau)f_{反因}(t-\tau)d\tau$$

$t_{因}+t_{反因}$　　t

4. 半时限函数与无时限函数的卷积

$f_{因}(t)$ 和 $f_{反因}(t)$ 合称半时限函数。设 $f_{无}(t)$ 的定义域为 $(-\infty,\infty)$。半时限函数与无时限函数的卷积是无时限的。其积分形式为：

$$\int_{t_{因}}^{\infty} f_{因}(\tau) f_{无}(t-\tau)\,d\tau$$

和

$$\int_{-\infty}^{t_{反因}} f_{反因}(\tau) f_{无}(t-\tau)\,d\tau$$

很明显，两个无时限函数卷积亦为无时限函数，其积分上下限为 $\pm\infty$。

以上结果容易由图解导出，其过程从略。其中 1 的情况最常见。对于因果性函数，通常的记法都含有阶跃函数作为因子。就是不用快速定限表，其积分限也容易定出。然而正因为如此，在演算中就往往会疏忽其定义域的限制而产生错误。如

例 2 求 $f(t)=f_1(t)*f_2(t)$

其中 $f_1(t)=U(t-a)$ (7)

$f_2(t)=U(t-b)+2U(t-c)$ (8)

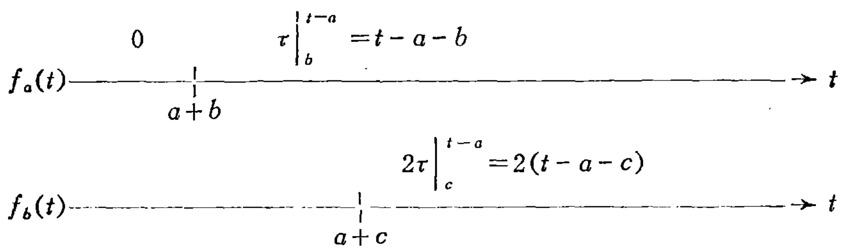
$b<c$

就往往会出现如下的错误：

$$\begin{aligned} f(t)&=\int_{-\infty}^{\infty} U(t-\tau-a)\,[U(\tau-b)+2U(\tau-c)]\,d\tau \\ &=\int_{b}^{t-a} d\tau+\int_{c}^{t-a} d\tau \\ &=t-b-a+2(t-c-a) \end{aligned} \tag{9}$$

用快速定限表即能有效地避免这种错误。如表 6 。这里 $f_a(\tau)=\tau$，$f_b(\tau)=2\tau$。

例 2 的快速定限卷积法 表 6



分段汇总求和得：

$$f(t)=\begin{cases} 0 & t<a+b \\ t-a-b & a+b\leqslant t\leqslant a+c \\ 3t-3a-b-2c & t\geqslant a+c \end{cases} \tag{10}$$

如不用快速定限解法，例 2 的正确解法应为：

$$\begin{aligned} f(t)&=\int_{-\infty}^{\infty} U(t-a-\tau)\,[U(t-b)+2U(\tau-c)]\,d\tau \\ &=\left[\int_{b}^{t-a} d\tau\right] U(t-a-b)+\left[\int_{c}^{t-a} 2d\tau\right] U(t-a-c) \\ &=(t-a-b)U(t-a-b)+2(t-a-c)U(t-a-c) \end{aligned} \tag{11}$$

比较起来，快速定限解法和上式的直接解析解法结果一样，简便程度都差不多。对于习惯了后者的人来说，也许是宁用后者。但必须记住，一定要在实际的积分定限式之后再乘上一个阶跃函数的因子。这个阶跃函数的写法是由积分结果的定义域决定的。这里就存在着一个应该由定出的积分限再恢复出结果的定义域的问题。下面的判定准则对此问题将会有所帮助。

因果性函数的卷积结果的定义域应满足以下的关系：

$y(t)\triangleq$ 定出的积分上限 $-$ 定出的积分下限 $\geqslant 0$

其正确性可在快速定限表中得到验证。按此准则，应该在每一个定了限的积分式之后都乘以相应的 $U[y(t)]$。比如在 (11) 式中，第一个积分相应的 $y(t)=t-a-b$，而第二个积分相应的 $y(t)=t-a-c$。

这个方法对于两个反因果性函数的卷积也适用。

五、离散卷积的快速定限求解和因果性序列卷积的定义域判定

类比连续卷积的情况，可以得出离散卷积的快速定限解法。设 $f_1(k)$ 的定义域为 $l_1\leqslant k\leqslant r_1$，$f_2(k)$ 的定义域为 $l_2\leqslant k\leqslant r_2$，且 $r_2-l_2\geqslant r_1-l_1$，则 $f(k)=f_1(k)*f_2(k)$ 的快速定限解法如下：

令不定限求和式 $f(i)\triangleq\sum\limits_i f_1(i)f_2(k-i)$　(12)

定限求和式 $f(i)\Big|_x^y\triangleq\sum\limits_{i=x}^{y}f_1(i)f_2(k-i)$　(13)

则有：

离散卷积的快速定限表　**表 7**

$f(k)$						k
结果	0	$f(i)\Big\vert_{l_1}^{k-l_2}$	$f(i)\Big\vert_{l_1}^{r_1}$	$f(i)\Big\vert_{k-r_2}^{r_1}$	0	
定义域	l_1+l_2	r_1+l_2	l_1+r_2	r_1+r_2		

表 7 同样可由图解导出，此处从略。

对于分段定义的序列，也可与连续情况类似地拆成几个序列之和。如

例 3　求 $f(k)=f_1(k)*f_2(k)$，其中

$$f_1(k)=\begin{cases}k & 1\leqslant k\leqslant 7\\ 7 & k=7,8\\ 0 & \text{其余}\end{cases}\tag{14}$$

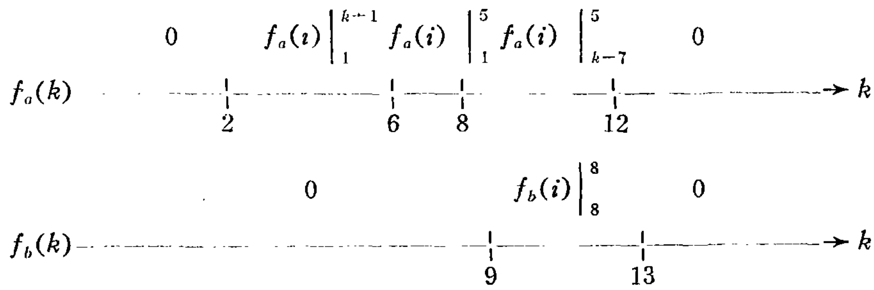
$$f_2(k)=\begin{cases}2 & 1\leqslant k\leqslant 5\\ 0 & \text{其余}\end{cases}\tag{15}$$

解　在把 $f_1(k)$ 这样的序列拆成几个序列之和时，必须注意各段交接点的处理。如其中$k=7$的点，既可属于第一段，又可属于第二段，却不能同时计入二者之中，否则会出现重复计算错误。这与卷积积分不同，在那里重要的是区间，单个的点对积分没有贡

献，同时划在两段中也行。而离散卷积中重要的是点，每个点对求和都将作出贡献。这里把$k=7$的点划入第一段，有：

$$f_a(i)\Big|_x^y=\sum_{i=x}^{y}(k-i)2=2(y-x+1)\left(k-\frac{y+x}{2}\right)\tag{16}$$

$$f_b(i)\Big|_x^y=\sum_{i=x}^{y}7\times 2=14(y-x+1)\tag{17}$$



$$f(k)=\begin{cases}0 & k<2\\ k(k-1) & 2\leqslant k\leqslant 6\\ 5(2k-6) & 6\leqslant k\leqslant 8\\ (13-k)(k+2)+14 & 9\leqslant k\leqslant 12\\ 14 & k=13\\ 0 & k>13\end{cases}\tag{16}$$

这里由于$f_1(k)$的第二段只有一点，故与之卷积就只有一个求和段，这与連续情况下与δ函数的卷积相类似。

还需指出一点：連续卷积结果的定义域时长为两函数时长之和，而离散卷积的定义域是点数，令$n_1=r_1-l_1+1$和$n_2=r_2-l_2+1$分别表示两序列的点数，则卷积结果的点数$n=(r_1+r_2)-(l_1+l_2)+1=n_1+n_2-1$。典型的情况如两个$N$点序列卷积结果具有$2N-1$点。这与連续情况略有差异。

对于因果性序列的卷积，也有与因果性函数卷积相类似的定义域恢复准则，只需将其中的$y(t)$换成$y(k)$即可。即应有：

$y(k)\triangleq$定出的求和上限－定出的求和下限$\geqslant 0$

避免出现类似于(9)式的错误的最好办法也是在定了求和限后立即乘上相应的$U[y(k)]$。如

$$\begin{aligned}f(k)&=k\cdot U(k-a)*[U(k-b)+U(k-c)]\\&=\left[\sum_{i=a}^{k-b}i\right]\cdot U(k-b-a)+\left[\sum_{i=a}^{k-c}i\right]\cdot U(k-c-a)\\&=[(k-b-a+1)(k-b+a)/2]\cdot U(k-b-a)\\&\quad+[(k-c-a+1)(k-c+a)/2]\cdot U(k-c-a)\end{aligned}\tag{17}$$

六、结　论

历来对于卷积运算，在选择正确的积分限时，画出折迭图解草图几乎是一种唯一可行的方法，因而要把“卷积的数学运算具体化是相当困难的”[1]。本文有鉴于此，提出了卷积的快速定限表。此表很有规律，容易掌握，易于拟订。有了此表，就无需经过折迭图解的手续，即可以直截了当地洞察各段卷积的积分限和定义域，这比布赖姆所描述的方法要直观简便得多。用此表求解卷积，步骤简单分明，容易检错，利于使用计算机编程计算，远比用历来的图解法省事、快捷。

文中还对容易定限因而容易疏忽定义域而出错的卷积运算提供了避免出错的有效措施。

以上两内容构成了由定义域求积分限以及由积分限恢复定义域的正逆两个方面。

以上结果对于工程计算，特别是对于教学实践，有着较大的实用价值。

附　录　卷积的两种定限解法的比较

为简洁并便于比较起见，此处就以文献 [2] 例 5.4-2 为例。其折迭图解定限解法可参阅 [2] 的 PP48—51。其中

$$f_1(t)=\begin{cases}1 & -2\leqslant t\leqslant 2\\ 0 & \text{其余}\end{cases}\tag{18}$$

$$f_2(t)=\begin{cases}\dfrac{1}{2} & 0\leqslant t\leqslant 2\\ 0 & \text{其余}\end{cases}\tag{19}$$

以下进行快速定限求解。按 $f_1(t)$ 和 $f_2(t)$ 时长间的关系，应作下述不定积分：

$$\begin{aligned}f(\tau)&=\int f_1(t-\tau)f_2(\tau)\,d\tau\\&=\frac{1}{2}\tau\end{aligned}\tag{20}$$

这里略写了积分常数，因为马上就要代积分限相消。其表解如下：

$f(t)$	$t<-2$	-2 ~ 0	0 ~ 2	2 ~ 4	$t>4$
	0	$\left.\dfrac{\tau}{2}\right\|_0^{t-2}=\dfrac{1}{2}(t+1)$	$\left.\dfrac{\tau}{2}\right\|_0^{2}=1$	$\left.\dfrac{\tau}{2}\right\|_{t-2}^{2}=\dfrac{1}{2}(4-t)$	0

这就是此例之快速定限解法的全过程。两相比较，快速定限解法比历来的图解法简单多了。至少是可少画 5 幅草图，少解 8 个不等式。当两函数均分段定义时，快速定限求解就更简便，这里不再赘述。

参 考 文 献

[1] E.O.布赖姆、快速富里叶变换，柳群译
上海科学技术出版社　1979年3月第1版 PP57-64

[2] 吴大正，信号与线性网络分析　下册
人民教育出版社　1980年6月第1版 PP46-52

The Fast Solution of Convolution by Use of a Table

Luo Yong-guang

Abstract

In this paper the table to fast determine the limit of convolution is proposed, the feature of the table is analyzed, the method of formation of the table and the steps of solution of convolution by use of the table are discribed. For solution of convolution the method proposed here is much faster and much more convenient than the classic graphical method. On the basis of the table the effective measure to avoid the mistakes in solution of convolution is proposed.