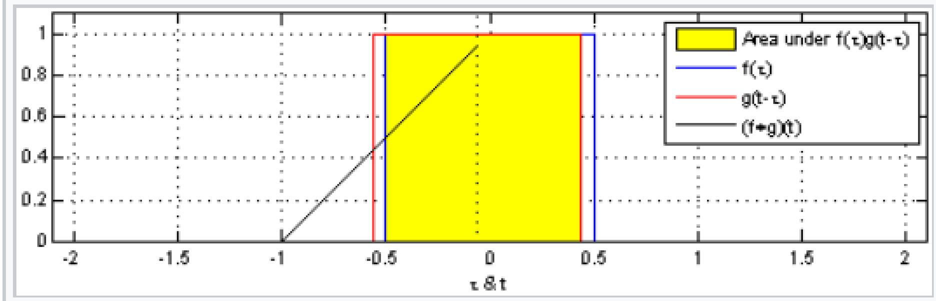
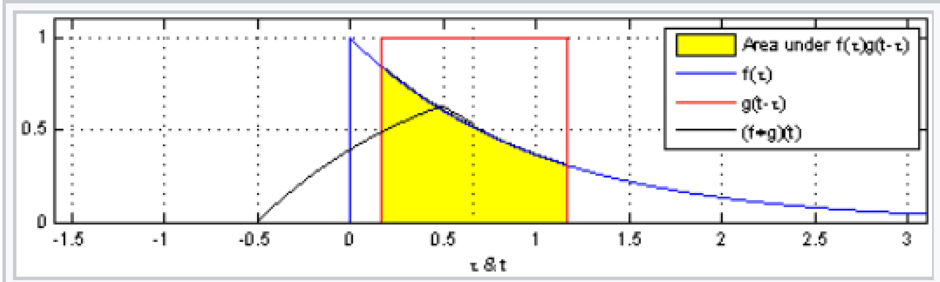
卷积 [编辑]

维基百科，自由的百科全书

在泛函分析中，**卷积**、**叠积**（convolution）、**褶积**或**旋积**，是通过两个函数f和g生成第三个函数的一种数学算子，表征函数f与经过翻转和平移的g的乘积函数所围成的曲边梯形的面积。如果将参加卷积的一个函数看作区间的指示函数，卷积还可以被看作是“移动平均”的推广。



图示两个方形脉冲波的卷积。其中函数"g"首先对$\tau=0$反射，接着平移"t"，成为$g(t-\tau)$。那么重叠部分的面积就相当于"t"处的卷积，其中横坐标代表待变量τ以及新函数$f*g$的自变量"t"。



图示方形脉冲波和指数衰退的脉冲波的卷积（后者可能出现于RC电路中），同样地重叠部分面积就相当于"t"处的卷积。注意到因为"g"是对称的，所以在这两张图中，反射并不会改变它的形状。

目录 [隐藏]

简单介绍 [编辑]

卷积是数学分析中一种重要的运算。设：$f(x)$、$g(x)$是$\mathbb{R}$上的两个可积函数，作积分：

$$\int_{-\infty}^{\infty} f(\tau)g(x-\tau)\,\mathrm{d}\tau$$

可以证明，关于几乎所有的$x\in(-\infty,\infty)$，上述积分是存在的。这样，随着x的不同取值，这个积分就定义了一个新函数$h(x)$，称为函数f与g的卷积，记为$h(x)=(f*g)(x)$。我们可以轻易验证：$(f*g)(x)=(g*f)(x)$，并且$(f*g)(x)$仍为可积函数。这就是说，把卷积代替乘法，$L^1(R^1)$空间是一个代数，甚至是巴拿赫代数。虽然这里为了方便我们假设 $f,g\in L^1(\mathbb{R})$，不过卷积只是运算符号，理论上并不需要对函数 f,g 有特别的限制，虽然常常要求 f,g 至少是可测函数（measurable function）（如果不是可测函数的话，积分可能根本没有意义），至于生成的卷积函数性质会在运算之后讨论。

卷积与傅里叶变换有着密切的关系。例如两函数的傅里叶变换的乘积等于它们卷积后的傅里叶变换，利用此一性质，能简化傅里叶分析中的许多问题。

由卷积得到的函数$f*g$一般要比f和g都光滑。特别当g为具有紧支集的光滑函数，f为局部可积时，它们的卷积$f*g$也是光滑函数。利用这一性质，对于任意的可积函数f，都可以简单地构造出一列逼近于f的光滑函数列f_s，这种方法称为函数的光滑化或正则化。

卷积的概念还可以推广到数列、测度以及广义函数上去。

定义 [编辑]

函数 f,g 是定义在 $\mathbb{R}^n$ 上的可测函数（measurable function），f与g的卷积记作$f*g$，它是其中一个函数翻转并平移后与另一个函数的乘积的积分，是一个对平移量的函数，也就是：

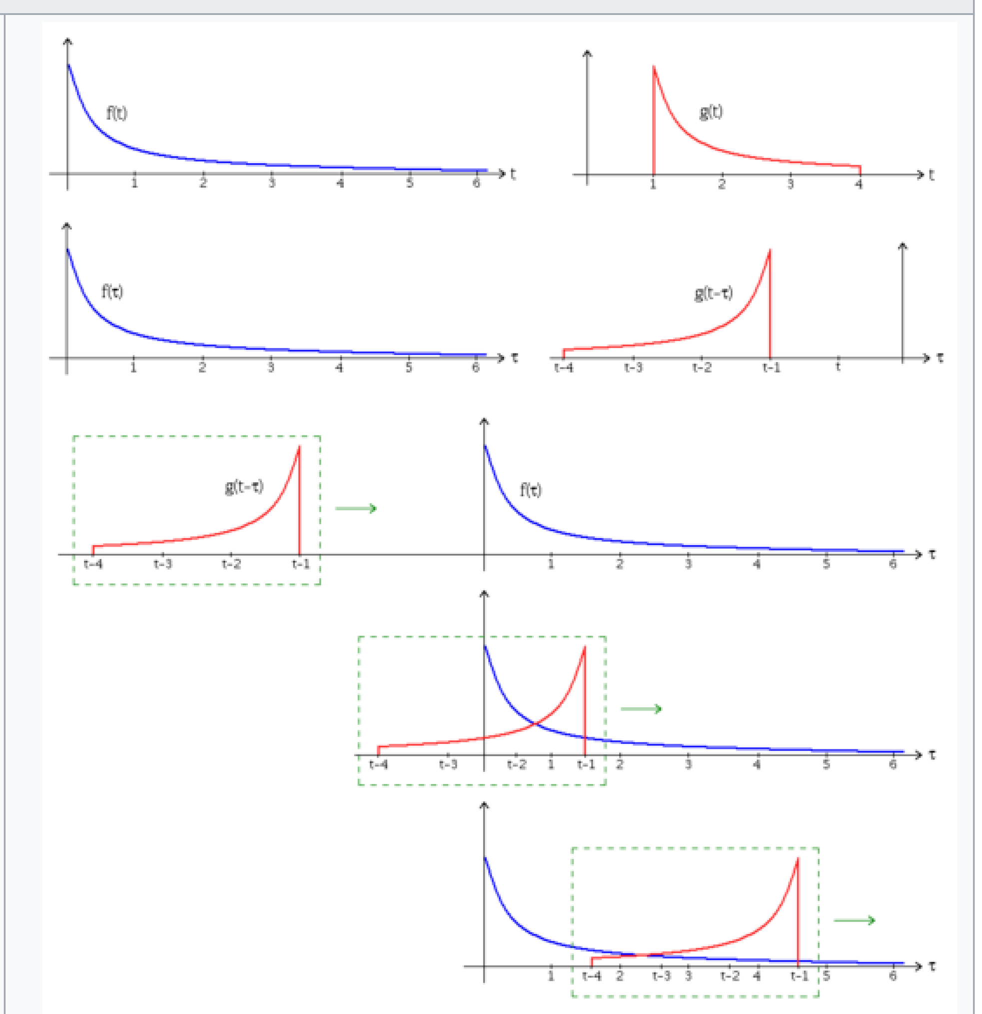
$$(f*g)(t)\ \stackrel{\mathrm{def}}{=}\ \int_{\mathbb{R}^n} f(\tau)g(t-\tau)\,d\tau.$$

如果函数不是定义在 $\mathbb{R}^n$ 上，可以把函数定义域以外的值都规定成零，这样就变成一个定义在 $\mathbb{R}^n$ 上的函数。

图解卷积

1. 已知两函数f(t)和g(t)。右图第一行两图分别为f(t)和g(t)。
2. 首先将两个函数都用τ来表示，从而得到f(τ)和g(τ)。将函数g(τ)向右移动t个单位，得到函数g(τ-t)的图像。将g(τ-t)翻转至纵轴另一侧，得到g(-(τ-t))即g(t-τ)的图像。右图第二行两图分别为f(τ)和g(t-τ)。
3. 由于t非常数（实际上是时间变量），当时间变量（以下简称“时移”）取不同值时，$g(t-\tau)$能沿着τ轴“滑动”。右图第三四五行可理解为“滑动”。
4. 让τ从-∞滑动到+∞。两函数交会时，计算交会范围中两函数乘积的积分值。换句话说，我们是在计算一个滑动的的加权总和(weighted-sum)。也就是使用$g(t-\tau)$当做加权函数，来对$f(\tau)$取加权值。

最后得到的波形（未包含在此图中）就是f和g的卷积。如果f（t）是一个单位脉冲，我们得到的乘积就是g（t）本身，称为冲激响应。



离散卷积 [编辑]

对于定义在整数 $\mathbb{Z}$ 上的函数 f,g，卷积定义为

$$(f*g)[n]\ \stackrel{\mathrm{def}}{=}\ \sum_{m=-\infty}^{\infty} f[m]g[n-m]$$
$$=\sum_{m=-\infty}^{\infty} f[n-m]\,g[m].$$

这里一样把函数定义域以外的值当成零，所以可以扩展函数到所有整数上（如果本来不是的话）。

当$g[n]$ 的支撑集（support）为有限长度M，上式会变成有限和：

$$(f*g)[n]=\sum^{M} f[n-m]g[m].$$

$m=-M$

计算离散卷积的方法 [编辑]

计算卷积$f[n] * g[n]$有三种主要的方法，分别为

1. 直接计算（Direct Method）
2. 快速傅里叶变换（FFT）
3. 分段卷积（sectioned convolution）

方法1是直接利用定义来计算卷积，而方法2和3都是用到了FFT来快速计算卷积。也有不需要用到FFT的作法，如使用数论变换。

方法1：直接计算 [编辑]

- 作法：利用卷积的定义

$$y[n] = f[n] * g[n] = \sum_{m=0}^{M-1} f[n-m]g[m]$$

- 若$f[n]$和$g[n]$皆为实数信号，则需要MN个乘法。
- 若$f[n]$和$g[n]$皆为更一般性的复数信号，不使用复数乘法的快速算法，会需要$4MN$个乘法;但若使用复数乘法的快速算法，则可简化至$3MN$个乘法。
  因此，使用定义直接计算卷积的复杂度为$O(MN)$。

方法2：快速傅里叶变换（FFT） [编辑]

- 概念：由于两个离散信号在时域（time domain）做卷积相当于这两个信号的离散傅里叶变换在频域（frequency domain）做相乘：
  $$y[n] = f[n] * g[n] \leftrightarrow Y[f] = F[f]G[f]$$
  ，可以看出在频域的计算较简单。
- 作法：因此这个方法即是先将信号从时域转成频域：
  $$F[f] = DFT_P(f[n]), G[f] = DFT_P(g[n])$$
  ，于是
  $$Y[f] = DFT_P(f[n])DFT_P(g[n])$$
  ，最后再将频域信号转回时域，就完成了卷积的计算：
  $$y[n] = IDFT_P DFT_P(f[n])DFT_P(g[n])$$
  总共做了2次DFT和1次IDFT。
- 特别注意DFT和IDFT的点数P要满足$P \geq M+N-1$。
- 由于DFT有快速算法FFT，所以运算量为$O(P\log_2 P)$
- 假设P点DFT的乘法量为a，$f[n]$和$g[n]$为一般性的复数信号，并使用复数乘法的快速算法，则共需要$3a+3P$个乘法。

方法3：分段卷积（sectioned convolution） [编辑]

- 概念：将$f[n]$切成好几段，每一段分别和$g[n]$做卷积后，再将结果相加。
- 作法：先将$f[n]$切成每段长度为L的区段（$L > M$），假设共切成S段：

  $$f[n](n=0,1,\ldots,N-1) \to f_1[n], f_2[n], f_3[n], \ldots, f_S[n] (S = \left\lceil \frac{N}{L} \right\rceil)$$

  Section 1: $f_1[n] = f[n], n = 0,1,\ldots,L-1$

  Section 2: $f_2[n] = f[n+L], n = 0,1,\ldots,L-1$

  $\vdots$

  Section r: $f_r[n] = f[n+(r-1)L], n = 0,1,\ldots,L-1$

  $\vdots$

  Section S: $f_S[n] = f[n+(S-1)L], n = 0,1,\ldots,L-1$

  ，$f[n]$为各个section的和

  $$f[n] = \sum_{r=1}^{S} f_r[n+(r-1)L]。$$

  因此，

  $$y[n] = f[n] * g[n] = \sum_{r=1}^{S}\sum_{m=0}^{M-1} f_r[n+(r-1)L-m]g[m],$$

  每一小段作卷积则是采用方法2，先将时域信号转到频域相乘，再转回时域：

  $$y[n] = IDFT(\sum_{r=1}^{S}\sum_{m=0}^{M-1} DFT_P(f_r[n+(r-1)L-m])DFT_P(g[m])), P \geq M+L-1。$$

- 总共只需要做P点FFT $2S+1$次，因为$g[n]$只需要做一次FFT。
- 假设P点DFT的乘法量为a，$f[n]$和$g[n]$为一般性的复数信号，并使用复数乘法的快速算法，则共需要$(2S+1)a+3SP$个乘法。
- 运算量：$\frac{N}{L}3(L+M-1)[\log_2(L+M-1)+1]$
- 运算复杂度：$O(N)$，和N呈线性，较方法2小。
- 分为 Overlap-Add 和 Overlap-Save 两种方法。

分段卷积: Overlap-Add

欲做$x[n] * h[n]$的分段卷积分，$x[n]$ 长度为 N，$h[n]$ 长度为 M，

Step 1: 将$x[n]$每 L 分成一段

Step 2: 再每段 L 点后面添加 $M-1$ 个零，变成长度 $L+M-1$

Step 3: 把 $h[n]$ 添加 $L-1$个零，变成长度 $L+M-1$的 $h'[n]$

Step 4: 把每个 $x[n]$ 的小段和 $h'[n]$ 做快速卷积，也就是$IDFT_{L+M-1}\{DFT_{L+M-1}(x[n])DFT_{L+M-1}(h'[n])\}$，每小段会得到长度 $L+M-1$ 的时域信号

Step 5: 放置第 i 个小段的起点在位置 $L \times i$ 上, $i = 0,1,\ldots,\lceil\frac{N}{L}\rceil - 1$

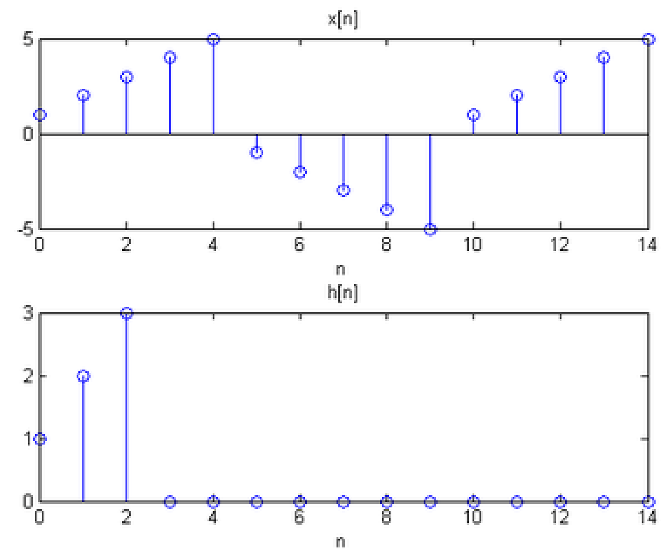
Step 6: 会发现在每一段的后面 $M-1$ 点有重叠，将所有点都相加起来，顾名思义 Overlap-Add，最后得到结果

举例来说:

$x[n] = [1,2,3,4,5,-1,-2,-3,-4,-5,1,2,3,4,5]$, 长度 $N = 15$

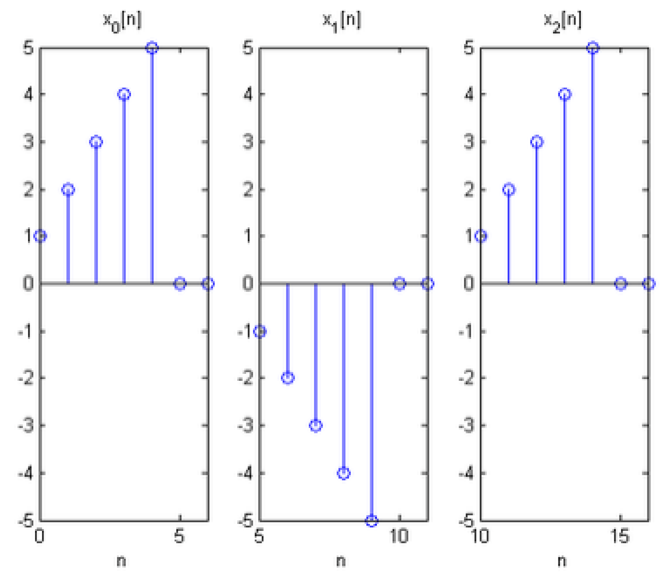
$h[n]=[1,2,3]$, 长度 $M=3$

令 $L=5$



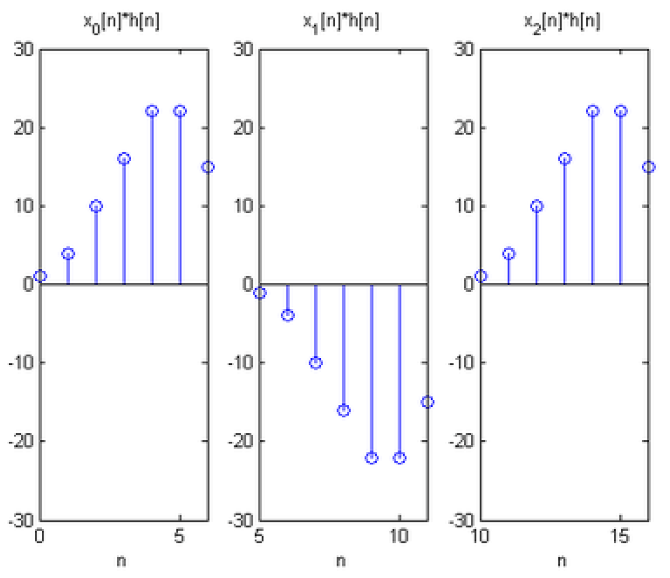
x[n]和h[n]

令 $L=5$ 切成三段，分别为 $x_0[n],x_1[n],x_2[n]$, 每段填 $M-1$ 个零，并将 $h[n]$ 填零至长度 $L+M-1$



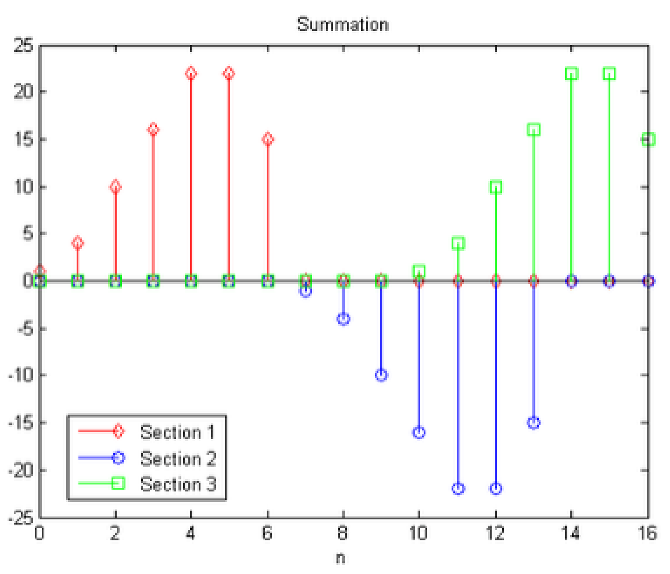
分段x[n]

将每一段做$IDFT_{L+M-1}\{DFT_{L+M-1}(x[n])DFT_{L+M-1}(h'[n])\}$



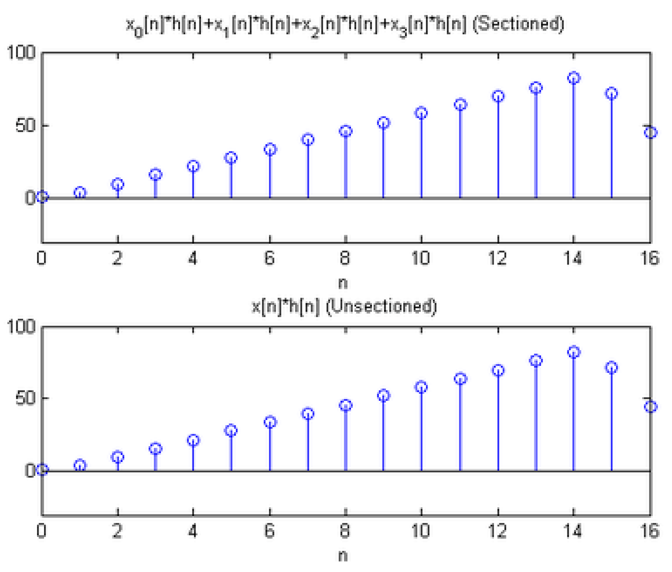
分段运算结果

若将每小段摆在一起，可以注意到第一段的范围是 $0\sim6$ ，第二段的范围是 $5\sim11$，第三段的范围是 $10\sim16$，三段的范围是有重叠的



合并分段运算结果

最后将三小段加在一起，并将结果和未分段的卷积做比较，上图是分段的结果，下图是没有分段并利用快速卷积所算出的结果，验证两者运算结果相同。



结果比较图

分段卷积: Overlap-Save

欲做$x[n]*h[n]$的分段卷积分，$x[n]$ 长度为 N，$h[n]$ 长度为 M，

Step 1: 将 $x[n]$ 前面填 $M-1$ 个零

Step 2: 第一段 $i=0$, 从新的 $x[n]$ 中 $L\times i-(M-1)\times i$ 取到 $L\times(i+1)-(M-1)\times i-1$ 总共 L 点当做一段，因此每小段会重复取到前一小段的 $M-1$ 点，取到新的一段全为零为止

Step 3: 把 $h[n]$ 添加 $L-M$ 个零，变成长度 L 的 $h'[n]$

Step 4: 把每个 $x[n]$ 的小段和 $h'[n]$ 做快速卷积，也就是$IDFT_L\{DFT_L(x[n])DFT_L(h'[n])\}$，每小段会得到长度 L 的时域信号

Step 5: 对于每个 i 小段，只会保留末端的 $L-(M-1)$ 点，因此得名 Overlap-Save

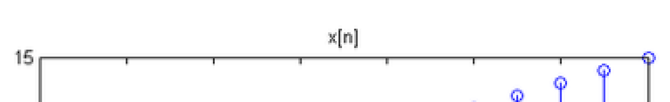
Step 6: 将所有保留的点合再一起，得到最后结果

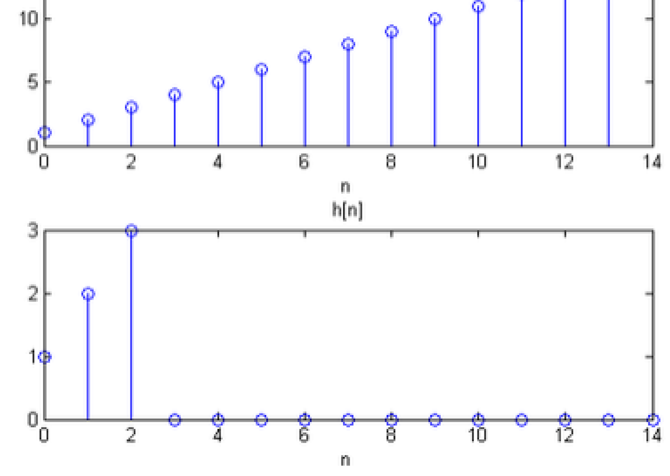
举例来说:

$x[n]=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]$, 长度 $N=15$

$h[n]=[1,2,3]$, 长度 $M=3$

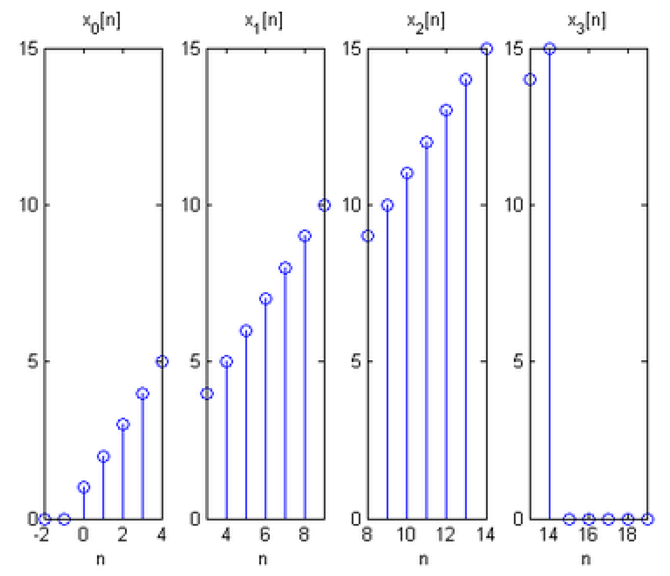
令 $L=7$





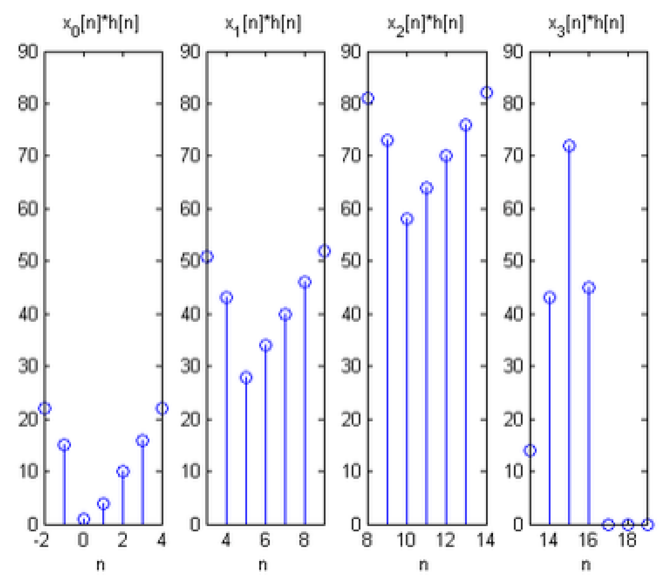
x[n]和h[n]

将 $x[n]$ 前面填 $M-1$ 个零以后，按照 Step 2 的方式分段，可以看到每一段都重复上一段的 $M-1$ 点



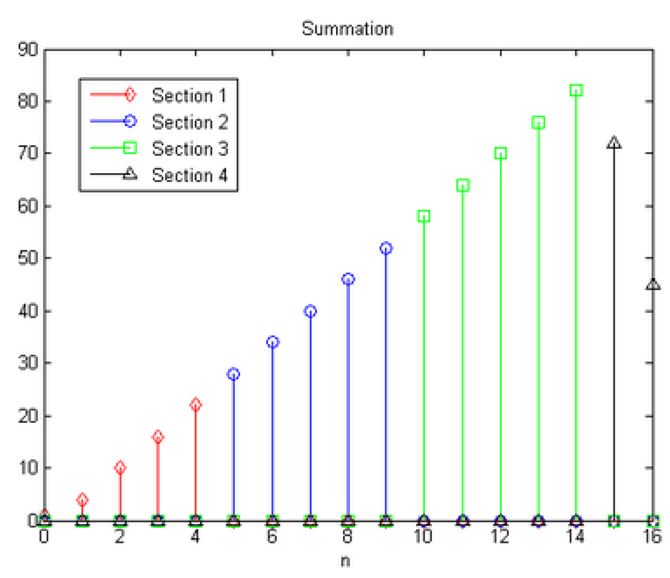
分段x[n]

再将每一段做 $IDFT_L\{DFT_L(x[n])DFT_L(h'[n])\}$ 以后可以得到



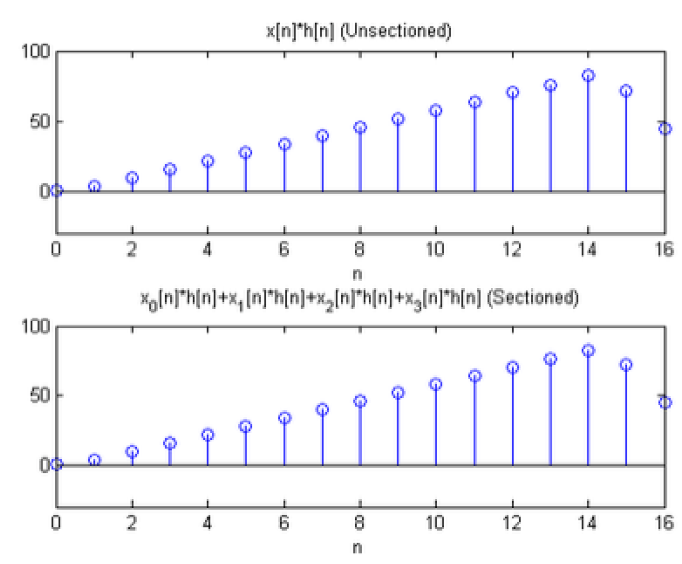
分段运算结果

保留每一段末端的 $L-(M-1)$ 点，摆在一起以后，可以注意到第一段的范围是 $0\sim4$，第二段的范围是 $5\sim9$，第三段的范围是 $10\sim14$，第四段的范围是 $15\sim16$，四段的范围是没有重叠的



合并分段运算结果

将结果和未分段的卷积做比较，下图是分段的结果，上图是没有分段并利用快速卷积所算出的结果，验证两者运算结果相同。



结果比较图

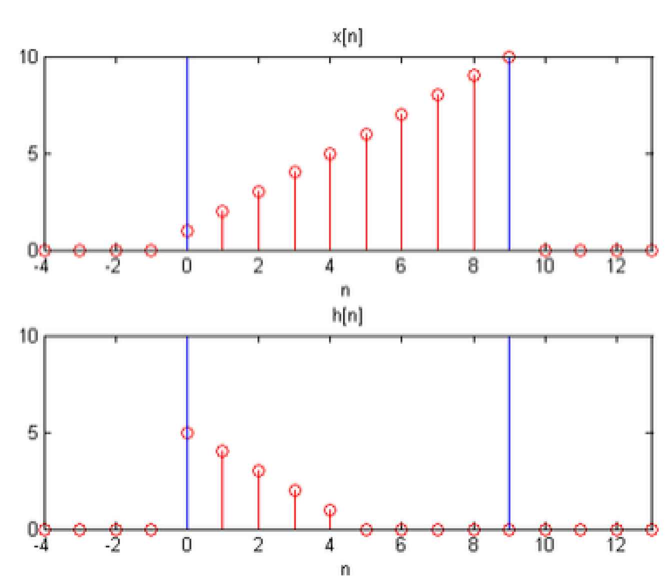
至于为什么要把前面 $M-1$ 丢掉?

以下以一例子来阐述:

$x[n]=[1,2,3,4,5,6,7,8,9,10]$, 长度 $L=10$，

$h[n]=[1,2,3,4,5]$, 长度 $M=5$,

第一条蓝线代表 y 轴，而两条蓝线之间代表长度 L，是在做快速卷积时的周期



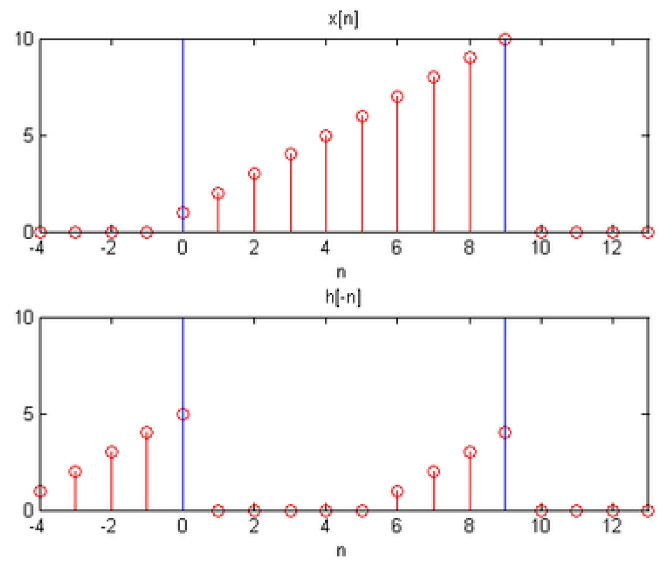
x[n]和h[n]

当在做快速卷积时 $IDFT_L\{DFT_L(x[n])DFT_L(h'[n])\}$，是把信号视为周期 L，在时域上为循环卷积分,

而在一开始前 $M-1$ 点所得到的值，是 $h[0],h[6],h[7],h[8],h[9]$ 和 $x[0],x[6],x[7],x[8],x[9]$ 内积的值,

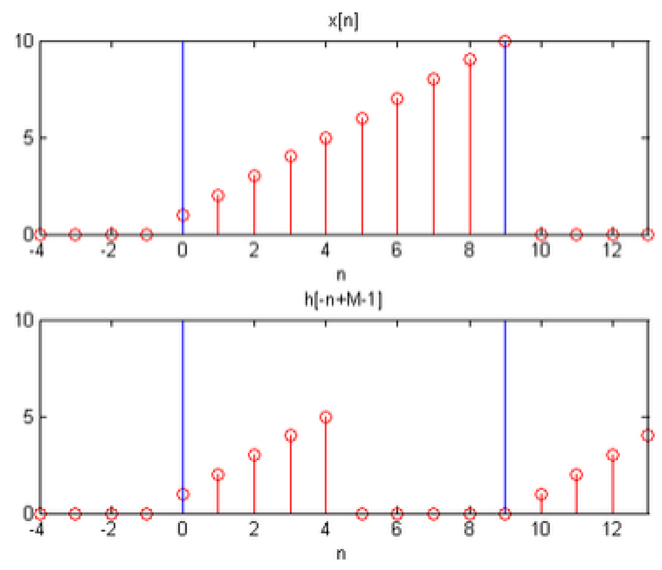
然而 $h[6],h[7],h[8],h[9]$ 这 $M-1$ 个值应该要为零，以往在做快速卷积时长度为 $L+M-1$ 时不会遇到这些问题,

而今天因为在做快速卷积时长度为 L 才会把这 $M-1$ 点算进来，因此我们要丢弃这 $M-1$ 点内积的结果



循环卷积

为了要丢弃这 $M-1$ 点内积的结果，位移 $h[-n]$ $M-1$ 点，并把位移以后内积合的值才算有效。



位移以后内积

应用时机 [编辑]

以上三种方法皆可用来计算卷积，其差别在于所需总体乘法量不同。基于运算量以及效率的考量，在计算卷积时，通常会选择所需总体乘法量较少的方法。

以下根据$f[n]$和$g[n]$的长度（N, M）分成5类，并列出适合使用的方法：

1. M为一非常小的整数 - 直接计算
2. $M \ll N$ - 分段卷积
3. $M \approx N$ - 快速傅里叶变换
4. $M \gg N$ - 分段卷积
5. N为一非常小的整数 - 直接计算

基本上，以上只是粗略的分类。在实际应用时，最好还是算出三种方法所需的总乘法量，再选择其中最有效率的方法来计算卷积。

例子 [编辑]

Q1：当$N = 2000, M = 17$，适合用哪种方法计算卷积？

Ans：

方法1：所需乘法量为$3MN = 102000$

方法2：$P \geq M + N - 1 = 2016$，而2016点的DFT最少乘法数$a = 12728$，所以总乘法量为$3(a+P) = 44232$

方法3：

若切成8块（$S = 8$），则$L = 250, P \geq M + L - 1 = 266$。选$P = 288$，则总乘法量为$(2S+1)a + 3SP = 26632$，比方法1和2少了很多。

但是若要找到最少的乘法量，必须依照以下步骤

(1)先找出L:解$L:\dfrac{\partial \frac{N}{L} 3(L+M-1)[\log_2(L+M-1)+1]}{\partial L} = 0$

(2)由$P \geq L + M - 1$算出点数在P附近的DFT所需最少的乘法量，选择DFT的点数

(3)最后由$L = P + 1 - M$算出L_{opt}

因此，

(1)由运算量对L的偏微分为0而求出$L = 85$

(2)$P \geq L + M - 1 = 101$，所以选择101点DFT附近点数乘法量最少的点数$P = 96$或$P = 120$。

(3-1)当 $P = 96 \rightarrow a = 280, L = P + 1 - M = 80 \rightarrow S = 25$，总乘法量为$(2S+1)a + 3SP = 21480$。

(3-2)当 $P = 120 \rightarrow a = 380, L = P + 1 - M = 104 \rightarrow S = 20$，总乘法量为$(2S+1)a + 3SP = 22780$。

由此可知，切成20块会有较好的效率，而所需总乘法量为21480。

- 因此，当$N = 2000, M = 17$，所需总乘法量：分段卷积<快速傅里叶变换<直接计算。故，此时选择使用分段卷积来计算卷积最适合。

Q2：当$N = 1024, M = 3$，适合用哪种方法计算卷积？

Ans：

方法1：所需乘法量为$3MN = 9216$

方法2：$P \geq M + N - 1 = 1026$，选择1026点DFT附近点数乘法量最少的点数，$\rightarrow P = 1152, a = 7088$。

因此，所需乘法量为$3(a+P) = 24342$

方法3：

(1)由运算量对L的偏微分为0而求出$L = 5$

(2)$P \geq L + M - 1 = 7$，所以选择7点DFT附近点数乘法量最少的点数$P = 8$或$P = 6$或$P = 4$。

(3-1)当 $P = 8 \rightarrow a = 4, L = P + 1 - M = 6 \rightarrow S = 171$，总乘法量为$(2S+1)a + 3SP = 5476$。

(3-2)当$P = 6 \rightarrow a = 4, L = P + 1 - M = 4 \rightarrow S = 256$，总乘法量为$(2S+1)a + 3SP = 6660$。

(3-3)当$P = 4 \rightarrow a = 0, L = P + 1 - M = 2 \rightarrow S = 512$，总乘法量为$(2S+1)a + 3SP = 6144$。

由此可知，切成171块会有较好的效率，而所需总乘法量为5476。

- 因此，当$N = 1024, M = 3$，所需总乘法量：分段卷积<直接计算<快速傅里叶变换。故，此时选择使用分段卷积来计算卷积最适合。
- 虽然当M是个很小的正整数时，大致上适合使用直接计算。但实际上还是将3个方法所需的乘法量都算出来，才能知道用哪种方法可以达到最高的效率。

Q3：当$N = 1024, M = 600$，适合用哪种方法计算卷积？

Ans：

方法1：所需乘法量为$3MN = 1843200$

方法2：$P \geq M + N - 1 = 1623$，选择1026点DFT附近点数乘法量最少的点数，$\rightarrow P = 2016, a = 12728$。

因此，所需乘法量为$3(a+P) = 44232$

方法3：

(1)由运算量对L的偏微分为0而求出$L = 1024$

(2)$P \geq L + M - 1 = 1623$，所以选择1623点DFT附近点数乘法量最少的点数$P = 2016$。

(3)当$P = 2016 \rightarrow a = 12728, L = P + 1 - M = 1417 \rightarrow S = 1$，总乘法量为$(2S + 1)a + 3SP = 44232$。

由此可知，此时切成一段，就跟方法2一样，所需总乘法量为44232。

- 因此，当$N = 1024, M = 600$，所需总乘法量：快速傅里叶变换 = 分段卷积<直接计算。故，此时选择使用分段卷积来计算卷积最适合。

多元函数卷积 [编辑]

按照翻转、平移、积分的定义，还可以类似的定义多元函数上的积分：

$$(f * g)(t_1, t_2, \cdots, t_n) = \int\int\cdots\int f(\tau_1, \tau_2, \cdots, \tau_n) g(t_1 - \tau_1, t_2 - \tau_2, \cdots, t_n - \tau_n,)\, d\tau_1 d\tau_2 \cdots d\tau_n$$

性质 [编辑]

各种卷积算子都满足下列性质：

交换律

$$f * g = g * f$$

结合律

$$f * (g * h) = (f * g) * h$$

分配律

$$f * (g + h) = (f * g) + (f * h)$$

数乘结合律

$$a(f * g) = (af) * g = f * (ag)$$

其中a为任意实数（或复数）。

微分定理

$$\mathcal{D}(f * g) = \mathcal{D}f * g = f * \mathcal{D}g$$

其中D f表示f的微分，如果在离散域中则是指差分算子，包括前向差分与后向差分两种：

- 前向差分：$\mathcal{D}^+ f(n) = f(n + 1) - f(n)$
- 后向差分：$\mathcal{D}^- f(n) = f(n) - f(n - 1)$