

# 一种改良的啁啾变换算法及其应用

石 鹏<sup>1)</sup> 刘 强<sup>1)</sup> 曹国威<sup>1)</sup> 李永平<sup>2)†</sup>

1) 中国科学技术大学物理系, 合肥 230026)

2) 中国科学技术大学中国科学院量子信息重点实验室, 合肥 230026)

(2008 年 9 月 22 日收到, 2008 年 12 月 5 日收到修改稿)

快速啁啾算法引入两次快速傅里叶变换(FFT)及一个解析高斯核, 计算复杂度低于卷积算法. 通过对啁啾算法实现过程进行的改进, 避免了该算法在实现过程中存在的一些问题, 比如输出窗口小、信号丢失、计算复杂度稍大等缺点. 把算法用于简单的可求得解析解的系统并与之做比较. 对高斯函数, 最大误差通常在  $10^{-15}$  数量级, 而对矩形函数, 由于受 FFT 算法计算精度的影响, 误差在  $10^{-3}$  数量级, 但这并不影响算法的性能. 最后把算法用于一种典型的标量衍射系统及分数傅里叶变换的计算, 获得了很好的结果.

关键词: 快速啁啾算法, 啁啾 Z 变换, 菲涅耳变换, 分数傅里叶变换

PACC: 4230K, 4230, 4230V

## 1. 引 言

啁啾变换广泛应用于计算信号物理系统的响应, 比如一阶光学系统的标量衍射<sup>[1-7]</sup>、全息透镜<sup>[8]</sup>以及菲涅耳变换<sup>[9]</sup>和分数傅里叶变换等. 菲涅耳变换是标量衍射中最广为研究的课题; 已经有很多的算法用于计算菲涅耳衍射, 比如用快速傅里叶变换(FFT)和卷积方法<sup>[10]</sup>, 以及使用矩阵变换的离散傅里叶变换(DFT)方法<sup>[11, 12]</sup>. 同样, 大量的特别的算法用于实现分数傅里叶变换<sup>[13-19]</sup>, 分数傅里叶变换可以用于描述光束传播<sup>[6]</sup>、成像<sup>[20-22]</sup>、衍射<sup>[23]</sup>以及信号和图像处理<sup>[24-32]</sup>. 本文发展一种快速啁啾算法, 用于统一对菲涅耳衍射及分数傅里叶变换进行数值计算.

本文从啁啾 Z 变换算法<sup>[1]</sup>出发, 对算法的变换及执行过程进行一些改进, 用来计算啁啾变换. 算法引入两次傅里叶变换和一个高斯解析核, 计算复杂度比卷积算法要低. 这种算法同样必须满足 Nyquist 定理, 但它并不要求信号空间( $x$  空间)和响应空间( $u$  空间)的抽样点数相同, 因此不但可放大信号中任意感兴趣的数据部分, 而且能查看一些啁啾系统, 比如近场衍射的精细结构.

## 2. 算法介绍

### 2.1. 啁啾变换的离散化

任意信号函数  $f(x)$  的啁啾变换可以表示成

$$g_{\text{ch}}^p(u) = C^p \{f(x)\} = \int_{-\infty}^{+\infty} f(x) B_{\text{ch}}^p(u, x) dx, \quad (1)$$

其核形式为

$$B_{\text{ch}}^p(u, x) = A(p) \exp \left[ -2\pi i \sum_{l=1}^N (\alpha_l x_l^2 + \beta_l x_l u_l + \gamma_l u_l^2) \right] \quad (u, x \in R^N). \quad (2)$$

不失一般性, 这里把核表示成

$$B_{\text{ch}}^p(u, x) = A(p) \exp \left[ i \sum_{l=1}^N (\alpha_l x_l^2 + \beta_l x_l u_l + \gamma_l u_l^2) \right] \quad (u, x \in R^N), \quad (3)$$

其中  $A(p)$  为核的振幅, 一般为复数;  $p = (\alpha, \beta, \gamma)$  为传播参数, 是  $N$  维矢量. 以上变换描述了信号从  $x$  空间(信号空间)到  $u$  空间(响应空间)的变换. 为了简化, 这里仅讨论一维情况. 但这些结果都可以推广到多维情况.

对于不能求得解析解的函数  $f(x)$  的啁啾变换, 必须先离散化. 为简化起见, 假设抽样是等间距的,

† 通讯联系人. E-mail: liyp@ustc.edu.cn

信号空间的抽样间隔为  $\delta x$  ,啁啾空间的间隔为  $\delta u$  . 信号空间  $x$  和啁啾空间  $u$  的抽样点数分别为  $N$  和  $M$  ,一般情况下并不要求  $N$  和  $M$  相等. 对方程 (1) 进行离散化有

$$\begin{aligned} x &\rightarrow n\delta x \quad (n = 0, 1, 2, \dots, N-1), \\ u &\rightarrow m\delta u \quad (m = 0, 1, 2, \dots, M-1), \end{aligned}$$

于是可得

$$\begin{aligned} g_m^p &= g_{ch}^p(m\delta u) \\ &\approx \sum_{n=0}^{N-1} f_n B_{ch}^p(m, n) \delta x, \end{aligned} \quad (4)$$

其中

$$\begin{aligned} B_{ch}^p(m, n) &= A(\mathbf{p}) \exp\{i[\alpha(n\delta x) + \beta nm\delta x\delta u \\ &\quad + \gamma(m\delta u)]\}, \end{aligned} \quad (5)$$

其中  $f_n = f(n\delta x)$  ,且有  $B_{ch}^p(m, n) = B_{ch}^p(m\delta u, n\delta x)$  . 可以看出,函数  $f(x)$  的啁啾变换相当于函数  $f(x) = f(x) \exp(i\alpha x^2)$  的傅里叶变换,然后再乘以一个二次相位因子.故可以表示成

$$\begin{aligned} g_{ch}^p(u) &\propto g(\eta) = F\{f(x)\} \\ &= F\{f(x) \exp(i\alpha x^2)\}, \end{aligned} \quad (6)$$

其中  $\eta = \beta u$  ,  $F\{\cdot\}$  表示傅里叶变换. 这是一种使用相当广泛的计算方法. 根据抽样定理,若  $f(x)$  是带限的,那么信号空间  $x$  的抽样间隔需要满足<sup>[30, 33]</sup>  $\delta x \leq \frac{1}{2\eta_m} = \frac{1}{2\beta u_m}$  ,在这个约束下面,可以计算一个给定函数的啁啾变换. 由于这种算法已经有很多的介绍,这里就不作详细的讨论了.

### 2.2. 改良的快速啁啾算法

本算法主要思想基于啁啾 Z 变换算法<sup>[1, 34, 35]</sup>.

$$\begin{aligned} f'_L(n) &= \begin{cases} f'_n\left(n - \frac{(L-N)}{2}\right) & \left(\frac{(L-N)}{2} \leq n < \frac{(L+N)}{2}\right), \\ 0 & \left(0 \leq n < \frac{(L-N)}{2}, \frac{(L+N)}{2} \leq n < L\right). \end{cases} \end{aligned} \quad (12)$$

计算 (10) 式解析核得到

$$\begin{aligned} \hat{B}_L^p(m) &= F\left\{\exp\left(-i\frac{\beta}{2}n^2\delta x\delta u\right)\right\} \\ &= \sqrt{-\frac{i\pi}{\beta\delta x\delta u}} \exp\left[\left(-i\frac{2\pi^2}{\beta\delta x\delta u}\right)\left(\frac{m}{L}\right)^2\right] \\ &\quad (m = 0, 1, 2, 3, \dots, L-1). \end{aligned} \quad (13)$$

计算  $f'_L(n)$  的傅里叶变换,然后与解析核相乘,再求傅里叶逆变换.

对方程 (5) 中指数项的第二项做变换,

$$nm\delta x\delta u = -\frac{1}{2}[(n-m)^2 - n^2 - m^2] \delta x\delta u \quad (7)$$

于是方程 (5) 变换为

$$\begin{aligned} g_{ch}^p(m\delta u) &= A(\mathbf{p}) \sum_{n=0}^{N-1} f(n\delta x) \exp\left[i\left(\alpha + \frac{\beta}{2}\frac{\delta u}{\delta x}\right)(n\delta x)^2\right] \\ &\quad \times \exp\left[-i\frac{\beta}{2}(n-m)^2\delta x\delta u\right] \\ &\quad \times \exp\left[i\left(\gamma + \frac{\beta}{2}\frac{\delta x}{\delta u}\right)(m\delta u)^2\right] \delta x, \end{aligned} \quad (8)$$

表示成卷积形式为

$$\begin{aligned} g_{ch}^p(m\delta u) &= A(\mathbf{p}) \left\{f'_n(n\delta x) \otimes \exp\left(-i\frac{\beta}{2}n^2\delta x\delta u\right)\right\} \\ &\quad \times \exp\left[i\left(\gamma + \frac{\beta}{2}\frac{\delta x}{\delta u}\right)(m\delta u)^2\right], \end{aligned} \quad (9)$$

其中符号  $\otimes$  表示卷积. 于是 (8) 式可以表示成

$$\begin{aligned} g_{ch}^p(m\delta u) &= A(\mathbf{p}) F^{-1}\left\{F\{f'_n(n\delta x)\}\right\} \\ &\quad \times F\left\{\exp\left(-i\frac{\beta}{2}n^2\delta x\delta u\right)\right\} \\ &\quad \times \exp\left[i\left(\gamma + \frac{\beta}{2}\frac{\delta x}{\delta u}\right)(m\delta u)^2\right] \end{aligned} \quad (10)$$

其中

$$\begin{aligned} f'_n(n\delta x) &= f(n\delta x) \exp\left[i\left(\alpha + \frac{\beta}{2}\frac{\delta u}{\delta x}\right)(n\delta x)^2\right] \\ &\quad (n = 0, 1, \dots, N-1), \end{aligned} \quad (11)$$

即任意信号的啁啾变换都可以用 (10) 式计算出来.

### 2.3. 算法的执行

下面分两种情况进行讨论.

对于  $M \geq N$  ,可令  $L = M$  .

把  $f'_n(n)$  变换成大小为  $L$  的矩阵

$$\begin{aligned} C(l) &= F^{-1}\{F\{f'_L(n)\}\hat{B}_L^p(m)\} \\ &\quad (l = 0, 1, 2, \dots, L-1). \end{aligned} \quad (14)$$

把  $C(l)$  与后面的相位因子相乘,可以得到

$$\begin{aligned} g_l^p &= A(\mathbf{p}) C(l) \exp\left[i\left(\gamma + \frac{\beta}{2}\frac{\delta x}{\delta u}\right)(l\delta u)^2\right] \\ &\quad (l = 0, 1, 2, \dots, L-1). \end{aligned} \quad (15)$$

对于  $N > M$  ,令  $L = N$  .

令  $f'_L(n) = f'_n(n)$  . 根据 (10) 式同样可计算解

析核为

$$\hat{B}_L^p(m) = F \left\{ \exp \left( -i \frac{\beta}{2} n^2 \delta x \delta u \right) \right\} \\ = \sqrt{-\frac{i\pi}{\beta \delta x \delta u}} \exp \left[ \left( -i \frac{2\pi^2}{\beta \delta x \delta u} \right) \left( \frac{m}{L} \right)^2 \right] \\ (m = 0, 1, 2, 3, \dots, L-1). \quad (16)$$

计算  $f_L^p(n)$  的傅里叶变换, 然后与解析核相乘, 再求它们的傅里叶逆变换.

$$C(l) = F^{-1} \{ F \{ f_L^p(n) \} \hat{B}_L^p(m) \} \\ (l = 0, 1, 2, \dots, L-1). \quad (17)$$

舍掉  $C(l)$  前面和后面的  $\frac{L-M}{2}$  个数据. 并且与后面的相位因子相乘得到

$$g_l^p = A(p) C \left( l + \frac{L-M}{2} \right) \exp \left[ i \left( \gamma + \frac{\beta}{2} \frac{\delta x}{\delta u} \right) \right] \\ \times \left( l + \frac{L-M}{2} \right)^2 (\delta u)^2 \\ (l = 0, 1, 2, \dots, M-1). \quad (18)$$

$g_l^p$  即是  $f(x)$  的啁啾变换数值解.

在算法执行中, 我们希望  $\hat{B}_L^p(m)$  上的抽样点足够密集从而能够对核获得更好的近似. 实际上, 当满足条件

$$\left( \frac{2\pi}{\beta \delta x \delta u} \right) \left( \frac{1}{L} \right)^2 \ll 1, \quad (19)$$

就能够使抽样点足够密集. 它与 Nyquist 定理是算法在执行时需要满足的两个基本要求.

## 2.4. 与 Z 变换算法及快速啁啾算法的比较

通过对 Z 变换算法<sup>[1]</sup>及快速啁啾算法<sup>[36]</sup>的研究发现, 算法同样需要满足 Nyquist 定理, 但由于算法在执行时舍弃了一些数据, 因而它的表达式发生了变化, 为  $2u_m \leq M(\beta L' \delta x)$ , 其中  $L' = M + N$ ,  $2u_m$  为响应面窗口的最大范围. 而对于改良的快速啁啾算法, 并没有强制舍弃数据, 所以在 Nyquist 定理下的表达式仍为  $2u_m \leq \frac{1}{\beta \delta x}$ . 即对于同样的条件, 要获得好的结果评价, 改良的快速啁啾算法所能考察的响应空间的范围要大. 比如假设  $M = N$ , 可以看到 Z 变换算法的  $u_m$  只有改良的啁啾算法  $u_m$  的  $\frac{1}{2}$ . 在实际应用中, 如衍射光学器件设计及一些物理响应系统的计算中, 要求响应空间尽量大, 改良的快速啁啾算法能获得更好的评价. 同时由于快速啁啾算法的计算窗口面积较小, 且舍弃了部分数据, 而在进行下次变换或迭代时又被强制令为零, 很容

易引起信号失真, 大大地降低了计算的准确度. 而改良的算法能够有效地避免这种情况的发生.

使用这种算法时, 计算复杂度下降为  $2L[\log(L) + O(1)]$ , 比原有的  $2L'[\log(L') + O(1)]$  要小. 同时这种算法可以提高计算精确度, 且可以实现原有快速啁啾算法及 Z 变换算法所能实现的计算, 当  $M = N$  时相应于常规的 FFT, 因此这种算法应比快速啁啾算法更有效且具有广泛的应用.

## 3. 改良的快速啁啾算法的应用

### 3.1. 算法数值计算准确度测试

#### 3.1.1. 对高斯函数的测试

为了测试算法的性能, 引入一个比例因子,  $\xi = \frac{\delta u}{\delta x}$ , 一般有  $\xi > 0$ . 对于  $0 < \xi \leq 1$  表示精细的放大度. 反之, 当  $\xi > 1$  时, 表示精细的缩小度.

对于一些特殊的啁啾变换系统, 如高斯函数的傅里叶变换, 以及矩孔、圆孔的夫琅禾费衍射等, 可以求得变换的解析解. 对于高斯函数的傅里叶变换, 有  $\alpha = \gamma = 0$ , 为了便于比较, 我们取  $x$  面计算窗口大小  $R_x = N\delta x = \sqrt{N}$ , 高斯光束的半高全宽为  $a^2 = \sqrt{\pi N}(2R_x)$ , 抽样点数  $M = 2N = 1024$ , 高斯函数形式为  $f(x) = \exp\left(-\left(\frac{x}{a}\right)^2\right)$ , 其中  $x \in \left[-\frac{R_x}{2}, \frac{R_x}{2}\right]$ . 归一化的误差可以表示为

$$\frac{\left| g_k^p - F \left\{ \exp \left( -\frac{x^2}{a^2} \right) \right\} \right|}{\max |g_k^p|}, \quad \text{用来描述数值结果与解析解之间的误差. 对于不同的 } \xi, \text{ 啁啾变换与相应解析解的误差极大值}$$

$$e_{\max} = \max \left( \frac{\left| g_k^p - F \left\{ \exp \left( -\frac{x^2}{a^2} \right) \right\} \right|}{\max |g_k^p|} \right) \quad \text{示于图 1(a)}$$

中, 一个特定的  $\xi$  值 ( $\xi = 0.560$ ) 的啁啾变换的误差曲线图 1(b) 中. 由图可以看出, 对于  $0.04 \leq \xi \leq 1.67$  相应于窗口的放大倍数为  $0.6 \sim 25$ ,  $e_{\max} \sim 10^{-15}$ , 优于文献 [36] 中算法的精确度, 可以获得很高的计算精确度.

#### 3.1.2. 对矩形函数的测试

同样可以使用一个矩形函数  $\text{rec}(x/a)$  来验证

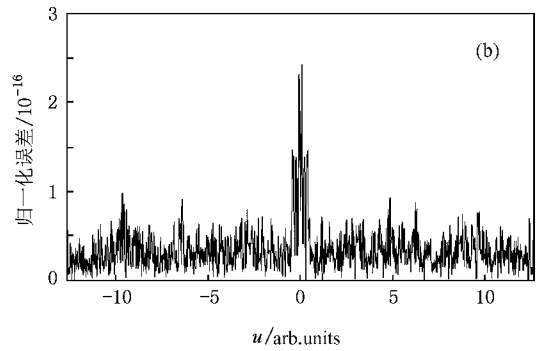
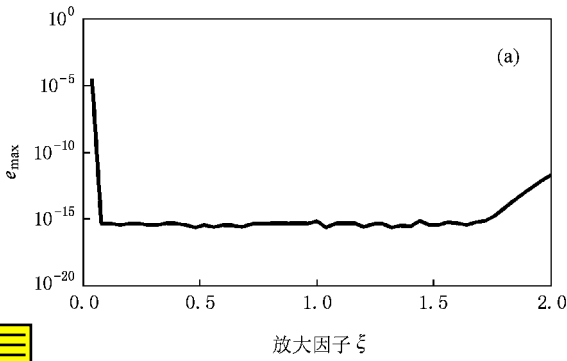


图 1 在不同比例因子  $\xi$  下用改良的快速啁啾算法求解高斯函数的傅里叶变换与解析解之间的  $e_{\max}$  和  $\xi = 0.560$  时的误差曲线 (a) 不同  $\xi$  下的  $e_{\max}$  (b)  $\xi = 0.560$  时的误差曲线

算法的精确度. 它的傅里叶变换可以表示为  $F\{\text{rec}(x/a)\} = \text{sinc}(2\pi au)/\pi u$ . 为了选取  $R_x = N \times \delta x = \sqrt{N}$ , 矩孔的孔径  $a = R_x/4$ ,  $M = 2N =$

1024 (其他参数的意义同前面高斯函数), 图 2(a) 描绘了对于不同  $\xi$  的最大数值误差  $e_{\max}$ , 图 2(b) 描绘了一个特定  $\xi$  值 ( $\xi = 0.560$ ) 下的误差曲线.

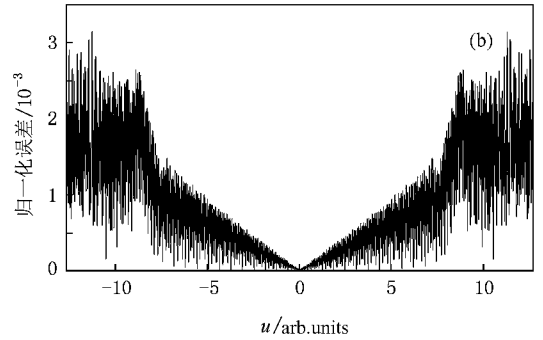
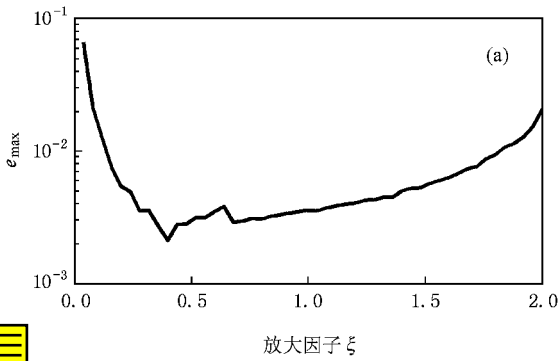


图 2 在不同比例因子  $\xi$  下用改良的快速啁啾算法求解的矩形函数  $\text{rec}(x/a)$  的傅里叶变换与解析解之间的  $e_{\max}$  和  $\xi = 0.560$  时的误差曲线 (a) 不同  $\xi$  下的  $e_{\max}$  (b)  $\xi = 0.560$  时的误差曲线

对于  $0.04 \leq \xi \leq 2$ , 相应于窗口的放大倍数为 0.5—25, 此时的  $e_{\max} \sim 10^3$ . 与文献 [36] 中的计算结果相比较, 虽然误差有所增大, 但是这里增大了响应空间的计算窗口, 且根据文献中对 FFT 精确度的计算, 可以发现这个误差主要是因为 FFT 的计算误差引起的, 它不会影响算法的性能和有效性.

在 (20) 式中,  $\lambda$  表示波长,  $k = 2\pi/\lambda$  表示波矢. 这个公式可以与菲涅耳变换联系起来. 它可以表示成一种特殊情况下的啁啾变换, 相应的参数可以写成

$$p = (\alpha, \beta, \gamma) = \frac{k}{2B} (A, -2, D),$$

其中

$$A(p) = \frac{-i}{B\lambda}. \tag{22}$$

### 3.2. 菲涅耳衍射计算

对于一般的菲涅耳光学系统, 可以用柯林斯公式表示<sup>[2, 6, 37]</sup>, 其核可以表示成

$$B_{\text{FH}}^M(u, v) = \frac{-i}{B\lambda} \exp\left[\frac{ik}{2B}(Ax^2 - 2xu + Du^2)\right] \tag{20}$$

它与射线传播矩阵<sup>[2, 37, 38]</sup>相对应,

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \quad (\det M = \pm 1). \tag{21}$$

使用上面的参数, 我们可以用啁啾算法计算抽样点数分别为  $N$  和  $M$  的菲涅耳衍射. 把算法应用于矩孔的近场衍射, 矩孔函数为  $\text{rec}(x/a, y/b)$ , 其中  $a = 200 \mu\text{m}$ ,  $b = 50 \mu\text{m}$ , 考查透镜后不同位置处的衍射问题. 入射孔径与透镜间距离为  $l_1 = 10^{-8} \mu\text{m}$ ; 透镜与观察面间距离为  $l_2$ , 在 0 到焦距  $f$  间变化,

焦距  $f = 5.0 \times 10^3 \mu\text{m}$ , 波长为  $\lambda = 0.850 \mu\text{m}$ . 可以得到系统的射线矩阵元素为  $A = 1 - l_2/f$ ,  $B = l_1 A + l_2$ ,  $C = -1/f$ ,  $D = 1 - l_1/f$ . 用啁啾变换计算的结果如图 3 所示. 取抽样点数  $M = 2N = 1024$ . 由于  $x$  与  $y$  方向是可以分离的, 因此可以变换为两个

一维的 ( $x$  与  $y$  方向) 啁啾变换相乘. 把计算结果与文献 [36] 得到的结果相比较, 发现是一致的. 从图中可以看出, 本算法增大了输出窗口, 可以看到更高级次的衍射级, 避免了信号失真, 且可以获得很好的效果.

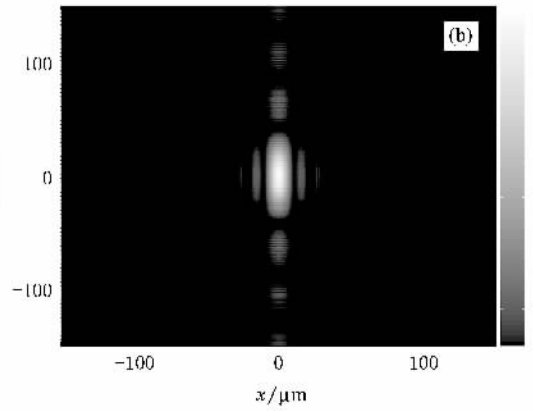
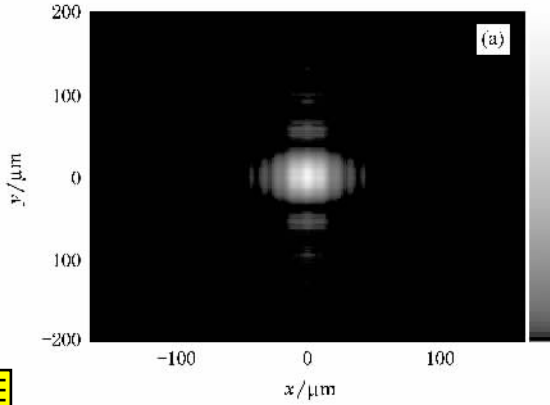


图 3 不同比例因子  $\xi$  下孔径后面不同距离的菲涅耳衍射图样 (a)  $l_2 = 4000 \mu\text{m}$ ,  $\xi = 0.1650$ ; (b)  $l_2 = 5000 \mu\text{m}$ ,  $\xi = 0.1500$

### 3.3. 分数傅里叶变换的计算

连续分数傅里叶变换可以用来描述光在一种渐

变折射率介质中传播的横向分布<sup>[24, 38]</sup>. 对于  $\kappa$  阶的分数傅里叶变换, 核可以表示成  $B_{\text{FrFT}}^\kappa$ <sup>[39-44]</sup>,

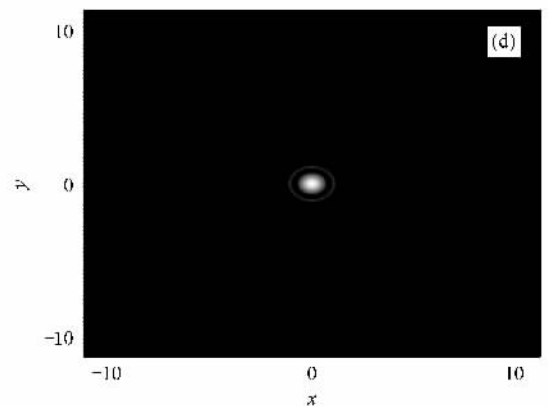
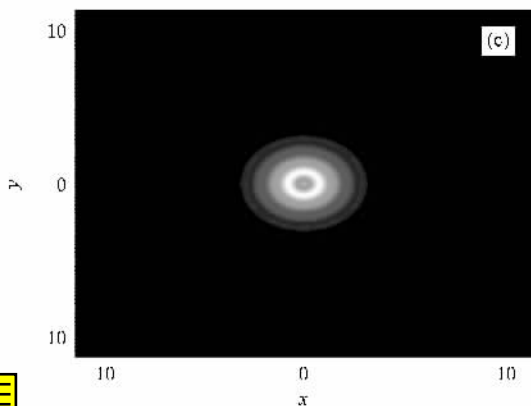
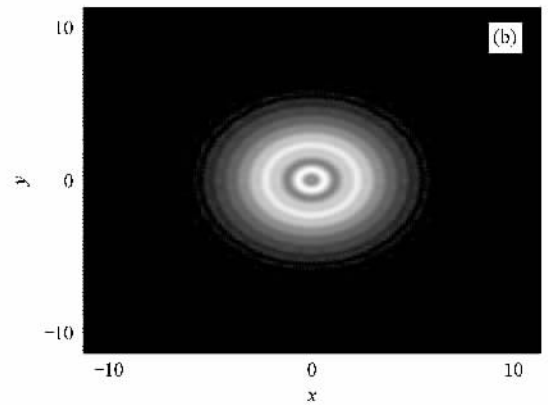
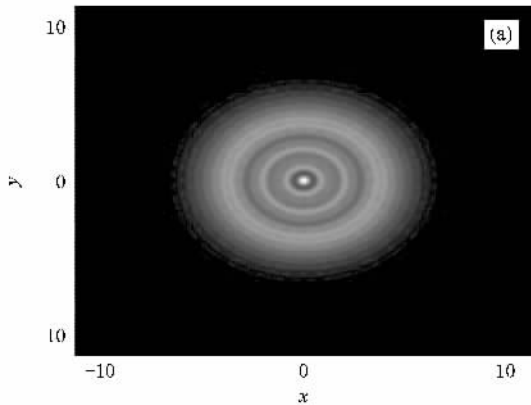


图 4 圆孔的分数傅里叶变换图样 (a)  $\kappa = 0.4$  (b)  $\kappa = 0.6$  (c)  $\kappa = 0.8$ , (d)  $\kappa = 1.0$

$$B_{\text{FFT}}^{\kappa} = \frac{\exp[-\kappa(\phi_s - \phi)/2]}{(2\pi|\sin(\phi)|)^{1/2}} \times \exp\left\{i\left[\frac{(u^2 + x^2)}{2}\cot\phi - ux\csc\phi\right]\right\} \quad (\kappa \neq 2n), \quad (23)$$

其中  $\phi = (\pi/2)\kappa$ ,  $\phi_s = (\pi/2)\text{sgn}(\sin\phi)$ ,  $n$  为一个整数. 当  $\kappa = 1, 3, 5, \dots$  时, 分数傅里叶变换退化为常规的傅里叶变换. 把它与啁啾变换相比较得到相应的啁啾变换的参数为

$$p = (\alpha, \beta, \gamma) = \left(\frac{\cot\phi}{2}, -\csc\phi, \frac{\cot\phi}{2}\right), \quad (24)$$

$$A(p) = \frac{\exp[-\kappa(\phi_s - \phi)/2]}{(2\pi|\sin(\phi)|)^{1/2}}, \quad (25)$$

下面把啁啾算法应用于二维圆孔衍射系统, 设圆孔的大小为  $0.1R_x$ ,  $x$  和  $u$  空间都是规范化的,  $N = M = 512$ , 取计算窗口  $R_x = N\delta x = R_u = N\delta u = \sqrt{N}$ , 对分数阶  $\kappa = 0.4, 0.6, 0.8$  和  $1.0$  进行模拟计算. 计算得到的光强如图 4 所示. 把计算结果与文献 [11, 36] 相比较, 发现结果大体一致, 图 4(d) 所示就是艾里斑. 由于可以增加响应空间的抽样点数, 因而可以得到更准确的结果.

## 4. 结 论

本文从啁啾  $Z$  变换算法出发, 通过对算法的变

换过程和执行做了一些改进, 构造了一种改良的快速啁啾算法. 此算法包含两次傅里叶变换和一个解析核, 这样减少了计算复杂度(低于卷积算法). 同时从 Nyquist 定理出发, 分析了可以获得良好评价的响应面窗口大小, 增大了响应面的考察范围, 有效地避免了信号的失真, 同时具有比原有的啁啾算法更低的计算复杂度, 且具有相当高的准确度. 这种算法可以解决标量衍射如菲涅耳衍射和近场衍射及分数傅里叶变换的问题, 且在抽样定理限制下, 可以自由选择  $x$  和  $u$  空间的抽样点数, 因此可以查看精细结构, 能放大数据中任何感兴趣的部分.

通过对算法数值结果与解析结果的比较, 可以发现计算具有相当高的精确度, 高斯函数的误差在  $10^{-15}$  数量级, 矩形函数的误差为  $10^{-3}$  数量级, 通过与 FFT 算法的误差相比较可以看出, 误差主要源于 FFT 算法, 因此此算法具有良好的性能及广泛的适应性. 这种算法可以直接变换到多维情况. 把算法应用于两种典型情况, 菲涅耳衍射和分数傅里叶变换与已有计算结果<sup>[15, 36]</sup>比较发现符合得相当好. 相信这种算法能够成为一种求解物理响应问题新的良好的工具.

- |  |  |
|--|--|
| <p>[ 1 ] Rabiner L R 1969 <i>IEEE Trans. Audio Electroacoust.</i> <b>17</b> 86</p> <p>[ 2 ] Collins S A 1970 <i>J. Opt. Soc. Am.</i> <b>60</b> 1168</p> <p>[ 3 ] Abe S, Sheridan J T 1994 <i>Opt. Lett.</i> <b>19</b> 1801</p> <p>[ 4 ] Abe S, Sheridan J T 1995 <i>Opt. Commun.</i> <b>113</b> 385</p> <p>[ 5 ] Abe S, Sheridan J T 1994 <i>J. Phys. A</i> <b>27</b> 4179</p> <p>[ 6 ] Deng X G, Li Y P, Fan D, Qiu Y 1997 <i>Opt. Commun.</i> <b>140</b> 226</p> <p>[ 7 ] Katzir A, Livanos A C, Shellan J B, Yariv A 1977 <i>IEEE J. Quantum Electron.</i> <b>13</b> 296</p> <p>[ 8 ] Forbes G W 1996 <i>J. Opt. Soc. Am. A</i> <b>13</b> 1816</p> <p>[ 9 ] Bihari B, Gan J, Wu L, Liu Y, Tang S, Chen R T 1999 <i>Proc. SPIE</i> <b>3632</b> 123</p> <p>[ 10 ] Kopp C, Meyrueis P 1998 <i>Opt. Commun.</i> <b>158</b> 7</p> <p>[ 11 ] Tucker S B, Ojeda-Castaneda J, Cathey W T 1999 <i>J. Opt. Soc. Am. A</i> <b>16</b> 316</p> <p>[ 12 ] Ariz V, Ojeda-Castaneda J 1995 <i>Opt. Lett.</i> <b>20</b> 118</p> <p>[ 13 ] Deng X, Li Y, Fan D, Qiu Y 1997 <i>Opt. Commun.</i> <b>138</b> 270</p> <p>[ 14 ] Marinho F J, Bernardo L M 1998 <i>J. Opt. Soc. Am. A</i> <b>15</b> 2111</p> <p>[ 15 ] Haman H, de Bougrenet de la Tocnaye J L 1995 <i>J. Opt. Soc. Am. A</i> <b>12</b> 1920</p> | <p>[ 16 ] Dickinson B W, Steigletz K 1982 <i>IEEE Trans. Acoust. Speech Signal Process</i> <b>30</b> 25</p> <p>[ 17 ] Pei S C, Yeh M H 1997 <i>Opt. Lett.</i> <b>22</b> 1047</p> <p>[ 18 ] Garca J, Mas D, Dorsch R G 1996 <i>Appl. Opt.</i> <b>35</b> 7013</p> <p>[ 19 ] Mas D, Garcia J, Ferreira C, Bernardo L M, Marinho F 1999 <i>Opt. Commun.</i> <b>164</b> 233</p> <p>[ 20 ] Lohmann A W 1995 <i>Opt. Commun.</i> <b>115</b> 437</p> <p>[ 21 ] Liu S, Xu J, Zhang Y, Chen L, Li C 1995 <i>Opt. Lett.</i> <b>20</b> 1053</p> <p>[ 22 ] Bernardo L M, Soares O D 1994 <i>J. Opt. Soc. Am. A</i> <b>11</b> 2622</p> <p>[ 23 ] Pellat-Finet P 1994 <i>Opt. Lett.</i> <b>19</b> 1388</p> <p>[ 24 ] Dorsch R G, Lohmann A W, Bitran Y, Mendlovic D, Ozaktas H M 1994 <i>Appl. Opt.</i> <b>33</b> 7599</p> <p>[ 25 ] Mendlovic D, Ozaktas H M 1993 <i>J. Opt. Soc. Am. A</i> <b>10</b> 1875</p> <p>[ 26 ] Ozaktas H M, Mendlovic D 1993 <i>J. Opt. Soc. Am. A</i> <b>10</b> 2522</p> <p>[ 27 ] Granieri S, Trabocchi O, Sicre E 1995 <i>Opt. Commun.</i> <b>119</b> 275</p> <p>[ 28 ] Ozaktas H M, Barshan B, Mendlovic D, Onural L 1994 <i>J. Opt. Soc. Am. A</i> <b>11</b> 547</p> <p>[ 29 ] Mendlovic D, Bitran Y, Dorsh R G, Lohmann A W 1995 <i>J. Opt. Soc. Am. A</i> <b>12</b> 1665</p> |
|--|--|

- [ 30 ] Mendlovic D , Zalevsky Z , Dorsh R G , Bitran Y , Lohmann A W , Ozaktas H M 1995 *J. Opt. Soc. Am. A* **12** 2424
- [ 31 ] Ozaktas H M , Mendlovic D 1995 *J. Opt. Soc. Am. A* **12** 743
- [ 32 ] Pellat-Finet P , Bonnet G 1994 *Opt. Commun.* **111** 141
- [ 33 ] Lohmann A W , Dorsch R G , Mendlovic D , Zalevsky Z , Ferria C 1996 *J. Opt. Soc. Am. A* **13** 470
- [ 34 ] Roose S , Bricchau B , Stijns E W 1993 *Opt. Commun.* **97** 312
- [ 35 ] Rabiner L R , Gold B 1975 *Theory and Applications of Digital Signal Processing* ( Englewood , Cliffs , New Jersey : Prentice-Hall ) p41
- [ 36 ] Deng X G , Bihari B , Gan J H , Zhao F , Chen Ray T 2000 *J. Opt. Soc. Am. A* **17** 762
- [ 37 ] Béleuger P A 1991 *Opt. Lett.* **16** 196
- [ 38 ] Yariv A 1985 *Optical Electronics* ( New York : CBC College Publishing ) p17
- [ 39 ] Karasik Y B 1994 *Opt. Lett.* **19** 769
- [ 40 ] Wu P , Lü B D 2005 *Acta Phys. Sin.* **54** 658 ( in Chinese ) [ 吴平、吕百达 2005 物理学报 **54** 658 ]
- [ 41 ] Chen B X , Li M , Zhang A J 2007 *Acta Phys. Sin.* **56** 4535 ( in Chinese ) [ 陈宝信、李明、张爱菊 2007 物理学报 **56** 4535 ]
- [ 42 ] Ran Q W , Yuan L , Tan L Y , Ma J , Wang J 2004 *Chin. Phys.* **13** 178
- [ 43 ] Wu P , Lü B D , Chen T L 2005 *Chin. Phys.* **14** 1130
- [ 44 ] Yuan L 2008 *Chin. Phys.* **17** 170

## An improved fast algorithm for chirp transforms and its applications

Shi Peng<sup>1)</sup> Liu Qiang<sup>1)</sup> Cao Guo-Wei<sup>1)</sup> Li Yong-Ping<sup>2)†</sup>

<sup>1</sup> *Department of Physics , University of Science and Technology of China , Hefei 230026 , China*

<sup>2</sup> *Key Laboratory of Quantum Information of Chinese Academy of Sciences , University of Science and Technology of China , Hefei 230026 , China*

( Received 22 September 2008 ; revised manuscript received 5 December 2008 )

### Abstract

A fast algorithm for chirp Z-transforms is improved form chirp Z-transform , which is developed by using two fast Fourier transforms and an analytical Gaussian kernel. Its computational complexity is less than a fast convolution algorithm. However , there are some problems when the algorithm is implemented , such as the discarding of the data , the smallness of the response domain , the bigness of the computational complexity and so on. To avoid the problems mentioned above , we make a change on the implementing of the algorithm in this paper. Then we compare the numerical results of some chirp systems with the analytical ones. The accuracy of Fourier transforms of Gaussian function is higher than the  $10^{-15}$  order for most cases , and the accuracy of Fourier transforms of rectangle function is about the  $10^{-3}$  order , which is essentially limited by the accuracy of the fast Fourier transform. Fianially this algorithm is used to calculate some typical systems of scalar diffraction and fractional-order Fourier transforms , and the results are in good agreement with other published results in the literatures.

**Keywords** : fast algorithm for chirp transform , chirp Z-transform , Fresnel transform , fractional Fourier transform

**PACC** : 4230K , 4230 , 4230V