ME201/MTH281ME400/CHE400 Fourier Transforms in *Mathematica* *Mathematica* 7

■ 1. Introduction

This notebook has two goals: to give examples of Fourier transforms of common functions, and to illustrate the use of the *Mathematica* commands FourierTransform and InverseFourierTransform. One of the things that we will learn is that even a package as well-developed as *Mathematica* can at times give a very wrong answer.

The form of answers obtained in symbolic calculations such as those done here may be different in different versions of *Mathematica*. The present notebook was executed with *Mathematica* 7.01. For some calculations, you may get results which appear to be different if you are using an earlier version of *Mathematica*.

Mathematica has considerable ability to handle Fourier transforms analytically. The code that does this is not part of the kernel. Rather, it is a package which is loaded on request. The command to load the Fourier transform package is

```
<< "FourierSeries`"
```

The package is now loaded and ready for use. In addition to analytical Fourier transforms, this package handles Fourier Sine and Cosine transforms, some aspects of Fourier series, and numerical calculation of both Fourier transforms and Fourier series coefficients. In this notebook, we consider only analytical determinations of Fourier transforms and inverse Fourier transforms.

In the literature, there are many variations in the definition of the Fourier transform and its inverse. *Mathematica* lets us specify which we wish to use. This is done by an option called FourierParameters. Setting this option to {a,b} produces the following Fourier transform pair:

$$\tilde{f}(\mathrm{k}) = \sqrt{\frac{|\mathrm{b}|}{(2\pi)^{1-\mathrm{a}}}} \int_{-\infty}^{\infty} e^{i\mathrm{bkx}} f(x)\, dx ,$$

$$\text{and } f(x) = \sqrt{\frac{|\mathrm{b}|}{(2\pi)^{1+\mathrm{a}}}} \int_{-\infty}^{\infty} e^{-i\mathrm{bkx}} \tilde{f}(k)\, dk .$$

Mathematica uses the default setting of {0,1}. In this course (and in the textbook by Haberman), the definitions used correspond to {1,-1}:

$$\tilde{f}(\mathrm{k}) = \int_{-\infty}^{\infty} e^{-i\mathrm{kx}} f(x)\, dx \ ,$$

$$\text{and } f(x) = \frac{1}{(2\pi)} \int_{-\infty}^{\infty} e^{i\mathrm{kx}} \tilde{f}(k)\, dk \ .$$

Let's do one Fourier transform as an example. We take the transform of $1/(1 + x^2)$, and then we reconstruct the function by taking the inverse Fourier transform, in both cases using the *Mathematica* default for the parameters.

```
f[x_] := 1/(1 + x^2)
```

```
ft[k_] = FourierTransform[f[x], x, k]
```

$e^{-\mathrm{Abs}[k]} \sqrt{\frac{\pi}{2}}$

```
InverseFourierTransform[ft[k], k, x]
```

$\frac{1}{1 + x^2}$

Everything works, but the formulas look a little different than they did in class. That's because of the use of the parameter settings {0,1} by *Mathematica*. We repeat the calculation using our class parameter settings:

```
ft[k_] = FourierTransform[f[x], x, k, FourierParameters → {1, -1}]
```

$e^{-\mathrm{Abs}[k]} \pi$

```
InverseFourierTransform[ft[k], k, x, FourierParameters → {1, -1}]
```

$\frac{1}{1 + x^2}$

In the rest of this notebook, we will use the parameter settings {1,-1}, so we set them once and for all.

```
SetOptions[FourierTransform, FourierParameters → {1, -1}];
```

```
SetOptions[InverseFourierTransform, FourierParameters → {1, -1}];
```

Now we check that this has taken effect by repeating our original transform pair, this time without explicitly setting options.

```
ft[k_] = FourierTransform[f[x], x, k]
```

$e^{-\mathrm{Abs}[k]} \pi$

```
InverseFourierTransform[ft[k], k, x]
```

$$\frac{1}{1+x^2}$$

2. General Rules

There are a number of general rules for Fourier transforms. We look briefly at some of them here, and experiment to see which ones *Mathematica* knows. We start with the derivative rule, using $\mathcal{F}$ to denote a Fourier transform. As we showed in class

$\mathcal{F}\{f'\} = i k\, \mathcal{F}\{f\}$.

We see if *Mathematica* knows this rule.

```
Clear[f];
```

```
FourierTransform[f'[x], x, k]
```

```
ⅈ k FourierTransform[f[x], x, k]
```

We see that *Mathematica* does know the derivative rule.

Another useful rule is the shifting rule. Although we did not derive this in class, it is easy to derive by a simple translation of the variable of integration in the integral defining the Fourier transform. The rule is

$\mathcal{F}\{f\,(x-a)\} = e^{-i k a}\, \mathcal{F}\{f(x)\}$.

Another shifting rule is

$\mathcal{F}\{e^{i x a} f\,(x\,)\} = \tilde{f}(k - a)$.

There are similar shift rules for the inverse transform.

Let's see if *Mathematica* knows any of these rules.

```
Clear[f];
```

```
FourierTransform[f[x - a], x, k]
```

```
FourierTransform[f[-a + x], x, k]
```

It appears as though *Mathematica* doesn't know this rule. We try the other shifting rule.

```
FourierTransform[ⅇ^(ⅈ x a) f[x], x, k]
```

```
FourierTransform[ⅇ^(ⅈ a x) f[x], x, k]
```

Mathematica doesn't know this rule either.

We try one last time, by seeing if *Mathematica* knows a shifting rule for the inverse transform.

```
InverseFourierTransform[f[k - a], k, x]

InverseFourierTransform[f[-a + k], k, x]
```

Again *Mathematica* does not know the rule.

The last general rule we consider is the convolution theorem. As we showed in class,

$\mathcal{F}^{-1}\{\tilde{f}(k)\,\tilde{g}(k)\,\} = \int_{-\infty}^{\infty} f(x')\,g(x - x')\,dx'$.

We see if *Mathematica* knows this by looking at the inverse transform of the product of two functions of *k*.

```
InverseFourierTransform[f[k] g[k], k, x]

InverseFourierTransform[f[k] g[k], k, x]
```

We see that *Mathematica* does not know the convolution theorem.

■ 3. Generalized Functions

The range of functions for which the Fourier transform may be used can be greatly extended by using generalized functions -- that is, the Dirac delta function and its close relatives (sign function, step function, etc.). We experiment here to see if *Mathematica* knows these functions, and if it can deal with their Fourier transforms. We start with the basic completeness relation which we derived in class:

$\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\,k\,(x-a)}\,dk = \delta(x - a)$.

We try this in *Mathematica*.

$$\frac{1}{2\,\pi}\int_{-\infty}^{\infty} e^{i\,k\,(x-a)}\,dk$$

— Integrate::idiv : Integral of $e^{i\,k\,(-a+x)}$ does not converge on $\{-\infty, \infty\}$. ≫

$$\frac{\int_{-\infty}^{\infty} e^{i\,k\,(-a+x)}\,dk}{2\,\pi}$$

A failure. Let's ask the same question, but now in the context of a Fourier transform. The Fourier transform of e^{iax} is $2\pi\delta(k - a)$. We see if *Mathematica* knows this.

$\texttt{FourierTransform}[e^{i\,a\,x}, x, k]$

```
2 π DiracDelta[a - k]
```

Success! Because the delta function is an even function of its argument, the two forms are equivalent. A special case of interest is $a = 0$, so that the function being transformed is the constant 1.

```
FourierTransform[1, x, k]
```

2 π DiracDelta[k]

We try another example. The Fourier transform of Cos[a x] is easily shown to be $\pi(\delta(k+a)+\delta(k-a))$. *Mathematica* says

```
FourierTransform[Cos[a x], x, k]
```

π DiracDelta[a − k] + π DiracDelta[a + k]

which agrees.

Now we see if *Mathematica* knows how to take the Fourier transform of a delta function.

```
FourierTransform[DiracDelta[x], x, k]
```

1

This also works if we put a shift in the argument.

```
FourierTransform[DiracDelta[x - a], x, k]
```

Cos[a k] − ⅈ Sin[a k]

A related function is the sign function, which is -1 for negative x and 1 for positive x. In *Mathematica* the sign function is denoted by Sign[x]. Because the delta function is the derivative of one-half the sign function, we conclude from the above transform result for the delta function that the transform of the sign function should be 2/ⅈk = -2ⅈ/k. Let's see if *Mathematica* knows this.

```
FourierTransform[Sign[x], x, k]
```

$-\frac{2 i}{k}$

Exactly as we predicted. Finally we look at the unit step function, denoted by UnitStep[x] in *Mathematica*. Because

UnitStep[x] = $\frac{1}{2}+\frac{1}{2}$Sign[x] ,

we predict from our above results that the Fourier transform of UnitStep[x] will be

$\mathcal{F}$\{UnitStep[x]\} $=\pi\delta(k)+\frac{1}{i k}$.

We see what *Mathematica* gives for this:

```
FourierTransform[UnitStep[x], x, k]
```

$-\frac{i}{k}+\pi$ DiracDelta[k]

Again exactly as predicted.

4. Examples of Fourier Transforms

4.1 Exponential Functions

The simplest exponential, e^{ax}, does not have a Fourier transform because it blows up at +∞ for $a > 0$ or -∞ for $a < 0$. A simple exponential function which does have a Fourier transform for $a > 0$ is

```
f[x_] := e^(-a Abs[x])
```

```
ft[k_] = FourierTransform[f[x], x, k]
```

$$\frac{2\,a}{a^2 + k^2}$$

We recover our original function with the inverse Fourier transform:

```
InverseFourierTransform[ft[k], k, x]
```

$$\frac{1}{2} e^{-a x} \left(\left(-1 + e^{2 a x}\right) \text{Sign}[x]\ (-1 + \text{Sign}[\text{Abs}[\text{Re}[a]]]) + 2 \left(e^{2 a x} \text{HeavisideTheta}[-x\ \text{Sign}[\text{Re}[a]]] + \text{HeavisideTheta}[x\ \text{Sign}[\text{Re}[a]]]\right) \text{Sign}[\text{Re}[a]]\right)$$

We simplify this by telling *Mathematica* that $a > 0$.

```
Simplify[%, a > 0]
```

$$e^{a x}\ \text{HeavisideTheta}[-x] + e^{-a x}\ \text{HeavisideTheta}[x]$$

The function HeavisideTheta is 1 for positive argument, 0 for negative argument, and not defined for zero argument:

```
HeavisideTheta[-1]
```

0

```
HeavisideTheta[1]
```

1

```
HeavisideTheta[0]
```

HeavisideTheta[0]

It differs from the function UnitStep in that Unitstep is defined to be 1 for zero argument:

```
UnitStep[-1]
```

0

```
UnitStep[1]
```

1

```
UnitStep[0]
```

1

In any case our answer for the inverse Fourier transform is equivalent to our original function. We can also do inversion in one step by incorporating the assumption a > 0 into the inversion command:

```
InverseFourierTransform[ft[k], k, x, Assumptions → a > 0]
```

$e^{a x}$ HeavisideTheta[-x] + $e^{-a x}$ HeavisideTheta[x]

We can also look at a function which vanishes for $x < 0$ and is $e^{-ax}$for $x > 0$. This is most easily described by using the unit step function.

```
f[x_] := UnitStep[x] e^(-a x)
```

```
ft[k_] = FourierTransform[f[x], x, k]
```

$$\frac{1}{a + i k}$$

Again we check this by finding the inverse.

```
Simplify[InverseFourierTransform[ft[k], k, x], a > 0]
```

$e^{-a x}$ HeavisideTheta[x]

Finally we look at a "left-half" exponential, which is $e^{ax}$for $x < 0$ and is 0 for $x > 0$.

```
f[x_] := UnitStep[-x] e^(a x)
```

```
ft[k_] = FourierTransform[f[x], x, k]
```

$$\frac{1}{a - i k}$$

This is correct. Let's invert this to get the original function.

```
InverseFourierTransform[ft[k], k, x]
```

$\frac{1}{2} e^{a x} (\text{Sign}[x] (-1 + \text{Sign}[\text{Abs}[\text{Re}[a]]]) + 2\, \text{HeavisideTheta}[-x\, \text{Sign}[\text{Re}[a]]]\, \text{Sign}[\text{Re}[a]])$

There is an interesting point here. The original Fourier transform makes sense only for $a > 0$, but we did not give that information to *Mathematica*, and we got the correct transform anyway. However, on the inversion, we see that *Mathematica* has considered the possibility that a may be complex. If we add to the inversion command an assumption that $a > 0$, we will get the simple answer we are looking for.

```
InverseFourierTransform[ft[k], k, x, Assumptions → a > 0]
```

$e^{a x}\, \text{HeavisideTheta}[-x]$

This is our original function.

4.2 Combinations of Exponential and Trigonometric Functions

Here we consider a few examples of products of trig functions and exponentials.

```
f[x_] := Cos[b x] e^(-a Abs[x])
```

```
ft[k_] = FourierTransform[f[x], x, k]
```

$$\frac{a}{a^2 + b^2 - 2 b k + k^2} + \frac{a}{a^2 + (b + k)^2}$$

Let's use the inversion command to recover our original function. We include the assumptions that a and b are real, and that $a > 0$.

```
InverseFourierTransform[ft[k], k,
 x, Assumptions → {a > 0, {a, b } ∈ Reals}]
```

$$\frac{1}{2} e^{-(a+i b) x} \left(1 + e^{2 i b x}\right) \left(e^{2 a x}\, \text{HeavisideTheta}[-x] + \text{HeavisideTheta}[x]\right)$$

It is not hard to show that this is equivalent to our original function. Here's what happens if we don't include the assumptions in the inversion:

```
InverseFourierTransform[ft[k], k, x]
```

$$\frac{1}{4} e^{-(a+i b) x} \Big(\text{Sign}[x] \big(-(-1+e^{2 a x})(1+e^{2 i b x}) + (-1+e^{2 (a+i b) x})\, \text{Sign}[\text{Abs}[\text{Im}[b]-\text{Re}[a]]] + (e^{2 a x}-e^{2 i b x})\, \text{Sign}[\text{Abs}[\text{Im}[b]+\text{Re}[a]]]\big) - 2\, \text{HeavisideTheta}[-x\, \text{Sign}[\text{Im}[b]-\text{Re}[a]]]\, \text{Sign}[\text{Im}[b]-\text{Re}[a]] - 2 e^{2 (a+i b) x}\, \text{HeavisideTheta}[x\, \text{Sign}[\text{Im}[b]-\text{Re}[a]]]\, \text{Sign}[\text{Im}[b]-\text{Re}[a]] + 2 e^{2 a x}\, \text{HeavisideTheta}[-x\, \text{Sign}[\text{Im}[b]+\text{Re}[a]]]\, \text{Sign}[\text{Im}[b]+\text{Re}[a]] + 2 e^{2 i b x}\, \text{HeavisideTheta}[x\, \text{Sign}[\text{Im}[b]+\text{Re}[a]]]\, \text{Sign}[\text{Im}[b]+\text{Re}[a]]\Big)$$

Now we have to do the simplifying ourselves.

```
Simplify[%, {a, b} ∈ Reals]
```

$$\frac{1}{4} e^{-(a+i b) x} (1+e^{2 i b x}) \big(2 e^{2 a x}\, \text{HeavisideTheta}[-x\, \text{Sign}[a]]\, \text{Sign}[a] + 2\, \text{HeavisideTheta}[x\, \text{Sign}[a]]\, \text{Sign}[a] + (-1+e^{2 a x})\, \text{Sign}[x]\, (-1+\text{Sign}[\text{Abs}[a]])\big)$$

```
Simplify[%, a > 0]
```

$$\frac{1}{2} e^{-(a+i b) x} (1+e^{2 i b x}) \big(e^{2 a x}\, \text{HeavisideTheta}[-x] + \text{HeavisideTheta}[x]\big)$$

Let's also derive this transform by the shifting rule. We define

```
g[x_] := e^(-a Abs[x])
```

The Fourier transform of g is

```
gt[k_] = FourierTransform[g[x], x, k]
```

$$\frac{2 a}{a^2 + k^2}$$

According to the shifting rule, the Fourier transform of f should be $\mathcal{F}\left\{\frac{1}{2}\left(e^{i bx} + e^{-i bx}\right) g(x)\right\}$ which is

```
1/2 (gt[k - b] + gt[k + b])
```

$$\frac{1}{2}\left(\frac{2 a}{a^2 + (-b+k)^2} + \frac{2 a}{a^2 + (b+k)^2}\right)$$

We check this by computing the difference.

```
Simplify[% - ft[k]]
```

0

It checks. The sine function times the exponential will give us a similar result.

```
f[x_] := Sin[b x] ⅇ^(-a Abs[x])
```

```
ft[k_] = FourierTransform[f[x], x, k]
```

$$-\frac{4\,i\,a\,b\,k}{\left(a^2+b^2-2\,b\,k+k^2\right)\,\left(a^2+(b+k)^2\right)}$$

We invert this to recover our original function.

```
InverseFourierTransform[ft[k],
 k, x, Assumptions → {a > 0, {a, b} ∈ Reals}]
```

$$-\frac{1}{2}\,i\,e^{-(a+i\,b)\,x}\,\left(-1+e^{2\,i\,b\,x}\right)\,\left(e^{2\,a\,x}\,\text{HeavisideTheta}[-x]+\text{HeavisideTheta}[x]\right)$$

This is equivalent to our original function.

■ 4.3 Combinations of Algebraic and Exponential Functions

We look at a few simple examples of products of polynomials and exponential functions. We start with a quadratic polymomial times the absolute value exponential.

```
f[x_] := (α + β x + γ x^2) ⅇ^(-a Abs[x])
```

```
ft[k_] = FourierTransform[f[x], x, k]
```

$$\frac{2\,a\,\left(a^4\,\alpha+k^2\,\left(k^2\,\alpha-2\,i\,k\,\beta-6\,\gamma\right)+2\,a^2\,\left(k^2\,\alpha-i\,k\,\beta+\gamma\right)\right)}{\left(a^2+k^2\right)^3}$$

Now we see if *Mathematica* is smart enough to invert this to give us our original function.

```
InverseFourierTransform[ft[k], k, x, Assumptions → {a > 0}]
```

$$e^{-a\,x}\,(\alpha+x\,(\beta+x\,\gamma))\,\left(e^{2\,a\,x}\,\text{HeavisideTheta}[-x]+\text{HeavisideTheta}[x]\right)$$

It is easy to see that this is equivalent to our original function.

We try one more example of this type -- *x* times the "right-hand" exponential. That is, we try

```
f[x_] := x UnitStep[x] ⅇ^(-a x)
```

```
ft[k_] = FourierTransform[f[x], x, k]
```

$$\frac{1}{(a + i k)^2}$$

We invert this to recover our original function.

```
InverseFourierTransform[ft[k], k, x, Assumptions → {a > 0}]
```

$e^{-a x}$ x HeavisideTheta[x]

Success.

■ 4.4 Combinations of Algebraic, Trigonometric and Exponential Functions

We can also find the Fourier transforms of products of exponential, algebraic and trig functions. By way of example, we look at one such transform. The function to be transformed is

```
f[x_] := x^2 Cos[b x] e^(-a Abs[x])
```

```
ft[k_] = FourierTransform[f[x], x, k]
```

$$\left(4 \left(a^9 - 6 a^5 b^4 - 8 a^3 b^6 - 3 a b^8 + 36 a^5 b^2 k^2 + 24 a^3 b^4 k^2 - 12 a b^6 k^2 - 6 a^5 k^4 + 24 a^3 b^2 k^4 + 30 a b^4 k^4 - 8 a^3 k^6 - 12 a b^2 k^6 - 3 a k^8\right)\right) \Big/ \left(\left(a^2 + b^2 - 2 b k + k^2\right)^3 \left(a^2 + b^2 + 2 b k + k^2\right)^3\right)$$

Now we try to recover our original function with the inverse transform.

```
InverseFourierTransform[ft[k], k, x]
```

0

It wouldn't be unreasonable to call this the low point of this notebook! After a great deal of thought, *Mathematica* has just told us that our function is identically zero! This is a good reminder that the truth of a result is not established by the fact that the result was generated by a computer. The problem now is that we don't know which results to believe from the commands FourierTransform and InverseFourierTransform.

Let's be a little stubborn about this example and see if we can get the correct inverse transform by using contour integration, with *Mathematica*'s help on the residue calculations. We start by seeing if we can simplify the expression for the transform.

```
ftmod[k_] = FullSimplify[ft[k]]
```

$$a\left(-\frac{6}{\left(a^2+(b-k)^2\right)^2}-\frac{6}{\left(a^2+(b+k)^2\right)^2}+8\,a^2\left(\frac{1}{\left(a^2+(b-k)^2\right)^3}+\frac{1}{\left(a^2+(b+k)^2\right)^3}\right)\right)$$

We see that the transform has third order poles at two points in the upper half k-plane ($k = \pm b + ia$), and third order poles at two points in the lower half k-plane ($k = \pm b - ia$). The inversion integral is an integral along the entire real k axis of

```
func[k_] = e^(i k x) ftmod[k] / (2 π)
```

$$\frac{a\,e^{i k x}\left(-\frac{6}{\left(a^2+(b-k)^2\right)^2}-\frac{6}{\left(a^2+(b+k)^2\right)^2}+8\,a^2\left(\frac{1}{\left(a^2+(b-k)^2\right)^3}+\frac{1}{\left(a^2+(b+k)^2\right)^3}\right)\right)}{2\pi}$$

For $x > 0$, we can close in the upper half plane, and the integral will be equal to the $2\pi i$ times the sum of the residues at the poles in the upper half plane.

```
2 π i (Residue[func[k], {k, b + i a}] + Residue[func[k], {k, -b + i a}])
```

$$2\,i\,\pi\left(-\frac{i\,e^{-a x-i b x}\,x^2}{4\pi}-\frac{i\,e^{-a x+i b x}\,x^2}{4\pi}\right)$$

```
FullSimplify[%]
```

$$\frac{1}{2}\,e^{-(a+i b)\,x}\left(1+e^{2 i b x}\right)x^2$$

This is equivalent to our original function for $x > 0$.

For $x < 0$, we must close in the lower half plane, and there will be an additional minus sign because the contour integral is now clockwise.

```
-2 π i (Residue[func[k], {k, b - i a}] + Residue[func[k], {k, -b - i a}])
```

$$-2\,i\,\pi\left(\frac{i\,e^{a x-i b x}\,x^2}{4\pi}+\frac{i\,e^{a x+i b x}\,x^2}{4\pi}\right)$$

```
FullSimplify[%]
```

$$\frac{1}{2}\,e^{(a-i b)\,x}\left(1+e^{2 i b x}\right)x^2$$

This is equivalent to our original function for $x < 0$.

4.5 Gaussian and Related Functions

In this section we consider transforms and inverse transforms of the Gaussian and of products of the Gaussian with polynomials or trig functions or exponentials. The basic function is

```
fbase[x_] := e^(-a x^2)
```

We find the Fourier transform of fbase.

```
fbaset[k_] = FourierTransform[fbase[x], x, k]
```

$$\frac{e^{-\frac{k^2}{4a}}\sqrt{\pi}}{\sqrt{a}}$$

Now the inverse to recover our original function:

```
InverseFourierTransform[fbaset[k], k, x]
```

$$e^{-a x^2}$$

Note that *Mathematica* has assumed throughout that $a > 0$ (or more generally that Re[a] > 0).

Now we look at a few variations, starting with powers of x times the original function.

```
ft[k_] = FourierTransform[x fbase[x], x, k]
```

$$-\frac{i\, e^{-\frac{k^2}{4a}}\, k\sqrt{\pi}}{2\, a^{3/2}}$$

```
ft[k_] = FourierTransform[x^2 fbase[x], x, k]
```

$$\frac{e^{-\frac{k^2}{4a}}\left(2\,a - k^2\right)\sqrt{\pi}}{4\, a^{5/2}}$$

```
ft[k_] = FourierTransform[x^3 fbase[x], x, k]
```

$$-\frac{i\, e^{-\frac{k^2}{4a}}\, k\left(6\,a - k^2\right)\sqrt{\pi}}{8\, a^{7/2}}$$

```
ft[k_] = FourierTransform[x^4 fbase[x], x, k]
```

$$\frac{e^{-\frac{k^2}{4a}}\left(12\,a^2 - 12\,a\,k^2 + k^4\right)\sqrt{\pi}}{16\, a^{9/2}}$$

Can *Mathematica* invert these as easily as it calculated them? We try an inversion with the last example.

```
InverseFourierTransform[ft[k], k, x]
```

$e^{-a x^2} x^4$

We see that *Mathematica* handles this class of functions well.

Now we look at the Fourier transform of an exponential times the Gaussian.

```
FourierTransform[e^(b x) fbase[x], x, k, Assumptions → a > 0]
```

$$\frac{e^{\frac{(b-i k)^2}{4 a}} \sqrt{\pi}}{\sqrt{a}}$$

```
InverseFourierTransform[%, k, x, Assumptions → a > 0]
```

$e^{x (b-a x)}$

This pair also was well done, although the calcuations took some time. Let's try a product of the absolute value exponential and a Gaussian.

```
ft[k_] = FourierTransform[e^(-b Abs[x]) fbase[x],
  x, k, Assumptions → {{a, b} ∈ Reals, a > 0}]
```

$$\frac{\sqrt{\pi} \left(e^{\frac{(b-i k)^2}{4 a}} \text{Erfc}\left[\frac{b-i k}{2 \sqrt{a}}\right] + e^{\frac{(b+i k)^2}{4 a}} \text{Erfc}\left[\frac{b+i k}{2 \sqrt{a}}\right]\right)}{2 \sqrt{a}}$$

A formidable transform which took a lot of time to produce.. We try the inversion. First we see if we can simplify the transform.

```
FullSimplify[ft[k], Assumptions → {{a, b} ∈ Reals, a > 0}]
```

$$\frac{\sqrt{\pi} \left(e^{\frac{(b-i k)^2}{4 a}} \text{Erfc}\left[\frac{b-i k}{2 \sqrt{a}}\right] + e^{\frac{(b+i k)^2}{4 a}} \text{Erfc}\left[\frac{b+i k}{2 \sqrt{a}}\right]\right)}{2 \sqrt{a}}$$

A useful simplification. Now we try the inversion.

```
InverseFourierTransform[%, k, x, Assumptions → {{a, b} ∈ Reals, a > 0}]
```

```
$Aborted
```

Unfortunately *Mathematica* gets lost in k-space and we have to abort the calculation. (But maybe it was just about to return an answer -- we'll never know.) One can probably invert this transform with the shifting rule and a good set of tables, but we won't pursue that here.

Now let's look at a product of the Gaussian and a trig function.

```
FourierTransform[Cos[b x] fbase[x],
 x, k, Assumptions → {{a, b} ∈ Reals, a > 0}]
```

$$\frac{\sqrt{\pi}\left(\text{Cosh}\left[\frac{(b-k)^2}{4a}\right]+\text{Cosh}\left[\frac{(b+k)^2}{4a}\right]-\text{Sinh}\left[\frac{(b-k)^2}{4a}\right]-\text{Sinh}\left[\frac{(b+k)^2}{4a}\right]\right)}{2\sqrt{a}}$$

We use TrigToExp to convert the hyperbolics to exponentials, to simplify the result.

```
TrigToExp[%]
```

$$\frac{e^{-\frac{(b-k)^2}{4a}}\sqrt{\pi}}{2\sqrt{a}}+\frac{e^{-\frac{(b+k)^2}{4a}}\sqrt{\pi}}{2\sqrt{a}}$$

So far so good. Now we try the inverse.

```
InverseFourierTransform[%, k, x, Assumptions → {{a, b} ∈ Reals, a > 0}]
```

$$\frac{1}{2}e^{-x\,(i\,b+a\,x)}\left(1+e^{2\,i\,b\,x}\right)$$

This is equivalent to our original function.

4.6 Top Hat, Sinc, and Truncated Trigonometric Functions

We consider a few more, mostly simple, examples. We start with the top hat function of amplitude a and range [-b,b]:

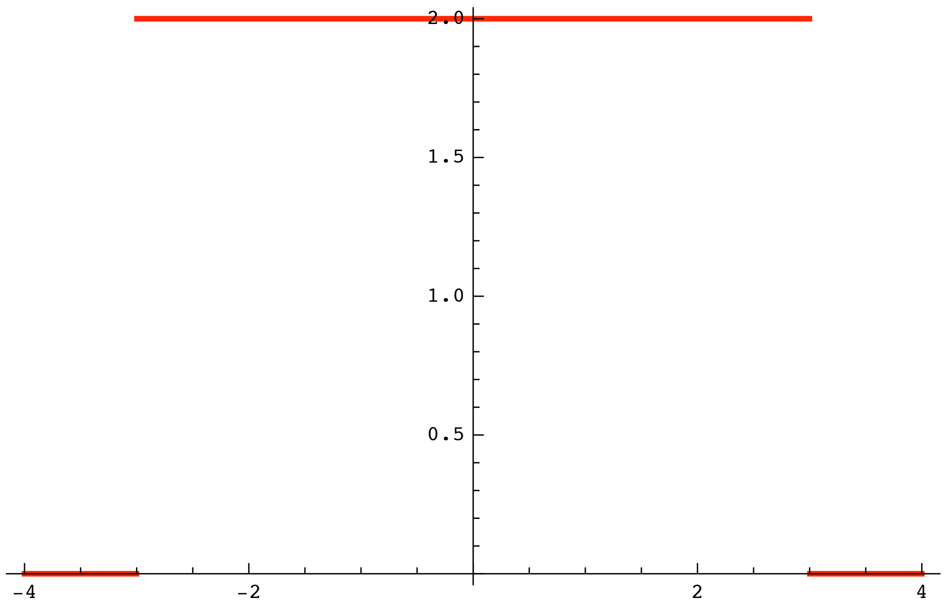
```
f[x_] := a (UnitStep[x + b] + UnitStep[b - x] - 1)
```

We check our definition by assigning values to *a* and *b* and plotting the function.

```
test[x_] = f[x] /. a → 2 /. b → 3
```

```
2 (-1 + UnitStep[3 - x] + UnitStep[3 + x])
```

```
Plot[test[x], {x, -4, 4},
 PlotStyle → {RGBColor[1, 0, 0], Thickness[0.006]}]
```



Now we calculate the Fourier transform.

```
ft[k_] = FourierTransform[f[x], x, k, Assumptions → b > 0]
```

$$\frac{2\,a\,\mathrm{Sin}[b\,k]}{k}$$

Let's do the inversion.

```
InverseFourierTransform[%, k, x, Assumptions → b > 0]
```

$$2\,a\left(\frac{1}{4}\,\mathrm{Sign}[b-x]+\frac{1}{4}\,\mathrm{Sign}[b+x]\right)$$

This is equivalent to our original function.

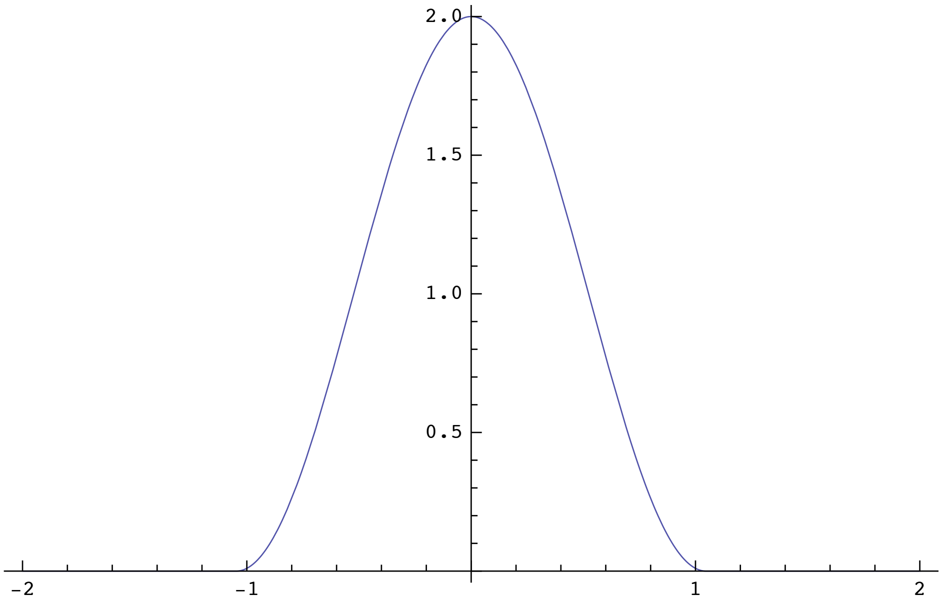
Now let's look at a truncated trig function -- essentially one peak of the shifted cosine. The function is defined by

$$f[x\_] := \frac{a}{4}\,(\mathrm{Sign}[\pi/b-x]+\mathrm{Sign}[\pi/b+x])\,(1+\mathrm{Cos}[b\,x])$$

Let's check our definition by plotting it for particular values of a and b.

```
a = 2; b = 3;
```

```
Plot[f[x], {x, -2, 2}]
```



Looks OK. Now we clear the values of *a* and *b* and find the Fourier transform.

```
Clear[a, b];
ft[k_] = FourierTransform[f[x], x, k]
```

$$\frac{2\,a\,b^2\,\text{Sin}\left[\frac{k\,\pi}{b}\right]\,\text{UnitStep}\left[\frac{2\,\pi}{b}\right]}{2\,b^2\,k-2\,k^3}$$

We try the inverse.

```
InverseFourierTransform[ft[k], k, x, Assumptions → b > 0]
```

$$\frac{1}{8\,\text{Sign}[b]}\,a\,(\text{Cos}[b\,x]-i\,\text{Sin}[b\,x])\,\Big(\text{Sign}[\pi-b\,x]\,(1+\text{Cos}[b\,x]+i\,\text{Sin}[b\,x])^2+\text{Sign}[\pi+b\,x]\,(2\,\text{Cos}[b\,x]+\text{Sign}[b]+\text{Cos}[2\,b\,x]\,\text{Sign}[b]+2\,i\,\text{Sin}[b\,x]+i\,\text{Sign}[b]\,\text{Sin}[2\,b\,x])\Big)\,\text{UnitStep}\left[\frac{1}{b}\right]$$

Unfortunately, *Mathematica* has ignored the assumption b > 0 in carrying out the inversion. We impose that assumption directly on the answer.

```
FullSimplify[%, Assumptions -> b > 0]
```

$$\frac{1}{4}\,a\,(1+\text{Cos}[b\,x])\,(\text{Sign}[\pi-b\,x]+\text{Sign}[\pi+b\,x])$$

Now we have our original function back.

As our final example in this section, we look at the sinc function, defined by

```
sinc[x_] := Sin[b x] / x
```

We ask *Mathematica* for the Fourier transform.

```
sinct[k_] = FourierTransform[sinc[x], x, k]
```

$$\frac{1}{2} \pi \, \text{Sign}[b - k] + \frac{1}{2} \pi \, \text{Sign}[b + k]$$

This is the correct result -- the Fourier transform of the sinc function is the top hat function.

Now the inversion.

```
InverseFourierTransform[sinct[k], k, x, Assumptions → b > 0]
```

0

On that sorry note, we end this notebook.