# 3. The Gaussian kernel

*Of all things, man is the measure.*
Protagoras the Sophist (480-411 B.C.)

```
<< MathVisionTools`;
```



## 3.1 The Gaussian kernel

The Gaussian (better Gaußian) kernel is named after Carl Friedrich Gauß (1777-1855), a brilliant German mathematician.

```
Import["Gauss10DM.gif", ImageSize → 280]
```


Figure 3.1 The Gaussian kernel is apparent on the old German banknote of DM 10,- where it is depicted next to its famous inventor when he was 55 years old.
See also: http://scienceworld.wolfram.com/biography/Gauss.html.

The Gaussian kernel is defined in 1-D, 2D and N-D respectively as

$$G_{1D}(x;\sigma) = \frac{1}{\sqrt{2\pi}\,\sigma}\, e^{-\frac{x^2}{2\sigma^2}},\; G_{2D}(x,y;\sigma) = \frac{1}{2\pi\sigma^2}\, e^{-\frac{x^2+y^2}{2\sigma^2}},\; G_{ND}(\vec{x};\sigma) = \frac{1}{\left(\sqrt{2\pi}\,\sigma\right)^N}\, e^{-\frac{|\vec{x}|^2}{2\sigma^2}}$$

The $\sigma$ determines the *width* of the Gaussian kernel.

In statistics, when we consider the Gaussian probability density function it is called the *standard deviation*, and the square of it, $\sigma^2$, the *variance*. We will refer to $\sigma$ as the *inner scale* or shortly *scale*.

The scale can only take positive values, $\sigma > 0$.

In the process of observation $\sigma$ can never become zero. This would  mean making an observation through an infinitesimally small aperture, which is impossible.

The scale-dimension is *not* just another spatial dimension, as we will thoroughly discuss in the remainder of this book.

The *half width at half maximum* ($\sigma = 2\sqrt{2\ln 2}$) is often used to approximate $\sigma$, but it is somewhat larger:

```
Unprotect[gauss];
gauss[x_, σ_] := 1/(σ Sqrt[2 π]) E^(-x^2/(2 σ^2));
Solve[gauss[x, σ]/gauss[0, σ] == 1/2, x]
```

**FrontEndVision Version 2.0 for *Mathematica* 6**

```
Solve::ifun :
  Inverse functions are being used by Solve, so some solutions may not be found; use
        Reduce for complete solution information. ≫
```

```
{{x → -σ Sqrt[2 Log[2]]}, {x → σ Sqrt[2 Log[2]]}}
```

```
% // N
```

```
{{x → -1.17741 σ}, {x → 1.17741 σ}}
```

## 3.2 Normalization

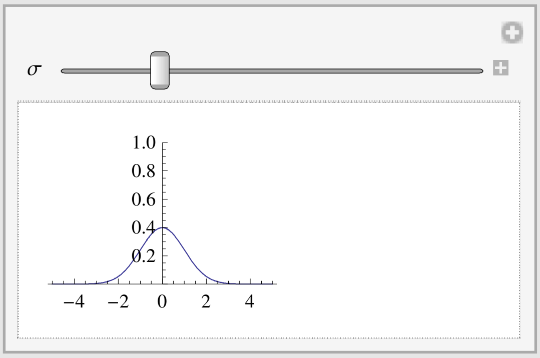
The term $\frac{1}{\sqrt{2\pi}\,\sigma}$ in front of the one-dimensional Gaussian kernel is the normalization constant. It comes from the fact that the integral over the exponential function is not unity: $\int_{-\infty}^{\infty} e^{-x^2/2\sigma^2}\,dx = \sqrt{2\pi}\,\sigma$. With the normalization constant this Gaussian kernel is a *normalized* kernel, i.e. its integral over its full domain is unity for every $\sigma$.

```
Integrate[e^(-x^2/(2 σ^2)), {x, -∞, ∞}, Assumptions → {σ > 0}]
```

```
Sqrt[2 π] σ
```

Increasing the $\sigma$ of the kernel reduces the amplitude substantially. Here are the graphs of the normalized kernels for $.2 < \sigma < 4$ plotted on the same axes:

```
Manipulate[Plot[1/(σ Sqrt[2 π]) Exp[-x^2/(2 σ^2)],
  {x, -5, 5}, PlotRange -> {0, 1}], {{σ, 1}, .2, 4}]
```




The normalization ensures that the average graylevel of the image remains the same when we blur the image with this kernel. This is known as *average grey level invariance*.

## 3.3 Cascade property, selfsimilarity

The *shape* of the kernel remains the same, irrespective of the $\sigma$. When we *convolve* two Gaussian kernels we get a new wider Gaussian with a variance $\sigma^2$ which is the sum of the variances of the constituting Gaussians:

$$g_{\text{new}}(\vec{x}; \sigma_1^2 + \sigma_2^2) = g_1(\vec{x}; \sigma_1^2) \otimes g_2(\vec{x}; \sigma_2^2).$$

```
σ =.; Simplify[
 ∫_{-∞}^{∞} gauss[α, σ1] gauss[α - x, σ2] dα, {σ1 > 0, σ2 > 0}]
```

$$\frac{e^{-\frac{x^2}{2(\sigma_1^2+\sigma_2^2)}}}{\sqrt{2\pi}\sqrt{\sigma_1^2+\sigma_2^2}}$$

This phenomenon, i.e. that a new function emerges that is similar to the constituting functions, is called *self-similarity*.

The Gaussian is a *self-similar function*.

Convolution with a Gaussian is a linear operation, so a convolution with a Gaussian kernel followed by a convolution with again a Gaussian kernel is equivalent to convolution with the broader kernel.

This is often used to convolve with large kernels *faster*.

Note that the *squares* of $\sigma$ add, not the $\sigma$'s themselves.

The *variances* add.

We can concatenate as many small blurring steps as we want to create a larger blurring step. With analogy to a cascade of waterfalls spanning the same height as the total waterfall, this phenomenon is also known as the *cascade smoothing property*.

Othe famous examples of self-similar functions are *fractals*. This shows the famous Mandelbrot fractal:

```
cMandelbrot =
  Compile[{{c, _Complex}}, -Length[FixedPointList[
      #1^2 + c &, c, 50, SameTest → (Abs[#2] > 2.` &)]]];
RasterPlot[-Table[cMandelbrot[a + b i], 
   {b, -1.1`, 1.1`, 0.0114`}, {a, -2.`, 0.5`, 0.0142`}],
 Mesh → False, AspectRatio → Automatic, Frame → False,
 ColorFunction → Hue, ImageSize → 170]
```

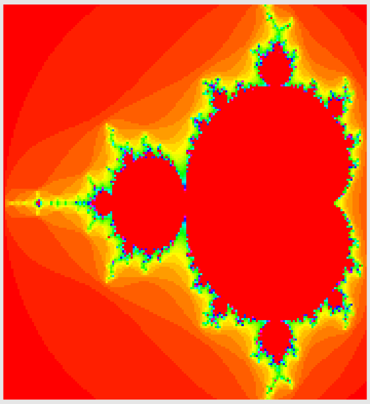


Figure 3.2 The Mandelbrot fractal is a famous example of a self-similar function. Source: www.mathforum.org. See also mathworld .wolfram.com/MandelbrotSet.html.

## 3.4 The scale parameter

In order to avoid the summing of squares, one often uses the following parametrization: $2\sigma^2 \to t$, so the Gaussian kernel gets a particularly short form. In $N$ dimensions:

$$G_{\mathrm{ND}}(\vec{x}, t) = \frac{1}{(\pi t)^{N/2}} e^{-\frac{x^2}{t}}.$$

It is this *t* that emerges in the diffusion equation

$$\frac{\partial L}{\partial t} = \frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2} + \frac{\partial^2 L}{\partial z^2}.$$

It is often referred to as 'scale' (like in: differentiation to scale, $\frac{\partial L}{\partial t}$), but a better name is *variance*.

We introduce the *dimensionless* spatial parameter, $\tilde{x} = \frac{x}{\sigma\sqrt{2}}$.

We say that we have *reparametrized* the *x*-axis. Now the Gaussian kernel becomes:

$g_n(\tilde{x}; \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\tilde{x}^2}$, or $g_n(\tilde{x}; t) = \frac{1}{(\pi t)^{N/2}} e^{-\tilde{x}^2}$.

If we walk along the spatial axis in footsteps expressed in $\sigma$'s, all kernels are of equal size or 'width'. (but due to the normalization constraint not necessarily of the same amplitude).

We now have a 'natural' size of footstep to walk over the spatial coordinate: a unit step in $x$ is now $\sigma\sqrt{2}$, so in more blurred images we make bigger steps.

We call this basic Gaussian kernel the *natural* Gaussian kernel $g_n(\tilde{x};\sigma)$.

The new coordinate $\tilde{x} = \frac{x}{\sigma\sqrt{2}}$ is called the *natural coordinate*.

It eliminates the scale factor $\sigma$ from the spatial coordinates, i.e. it makes the Gaussian kernels similar, despite their different inner scales. We will encounter natural coordinates many times hereafter.

The spatial extent of the Gaussian kernel ranges from $-\infty$ to $+\infty$, but in practice it has negligible values for x larger then a few (say 5) $\sigma$. The numerical value at $x=5\sigma$, and the area under the curve from $x=5\sigma$ to infinity (recall that the total area is 1):

```
gauss[5, 1] // N
```

```
1.48672×10^-6
```

```
Integrate[gauss[x, 1], {x, 5, Infinity}] // N
```

```
2.86652×10^-7
```

In the limit of blurring to infinity, the image becomes homogenous in intensity: the average intensity of the image.

## 3.5 Relation to generalized functions

The Gaussian kernel is the physical equivalent of the *mathematical point*. It is not strictly local, like the mathematical point, but *semi-local*. It has a *Gaussian weighted extent*, indicated by its inner scale $\sigma$.

The mathematical functions involved are the *generalized functions*, i.e. the Delta-Dirac function, the Heaviside function and the error function. In the next section we study these functions in detail.

When we take the limit as the inner scale goes down to zero (remember that $\sigma$ can only take positive values for a physically realistic system), we get the mathematical delta function, or Dirac delta function, $\delta(x)$.

This function is everywhere zero except in $x = 0$, where it has infinite amplitude and zero width, its area is unity.

$$\lim_{\sigma\downarrow 0}\left(\frac{1}{\sqrt{2\pi}\,\sigma}\,e^{-\frac{x^2}{2\sigma^2}}\right) = \delta(x).$$

$\delta(x)$ is called the *sampling function* in mathematics. It is assumed that $f(x)$ is continuous at $x = a$:

```
Simplify[∫_{-∞}^{∞} DiracDelta[x - a] f[x] dx, a ∈ Reals]
```

```
f[a]
```

The *sampling property of derivatives* of the Dirac delta function is shown below:

$$\int_{-\infty}^{\infty} \texttt{D[DiracDelta[x], \{x, 2\}] f[x]} \, dx$$

```
f''[0]
```

The delta function was originally proposed by the eccentric Victorian mathematician Oliver Heaviside (1880-1925, see also [Pickover1998]). Story goes that mathematicians called this function a "monstrosity", but it did work! Around 1950 physicist Paul Dirac (1902-1984) gave it new light. Mathematician Laurent Schwartz (1915-2002) proved it in 1951 with his famous "theory of distributions" (we discuss this theory in chapter 8). And today it's called "the Dirac delta function".

The integral of the Gaussian kernel from $-\infty$ to $x$ is a well known function as well. It is the *error function*, or *cumulative* Gaussian function, and is defined as:

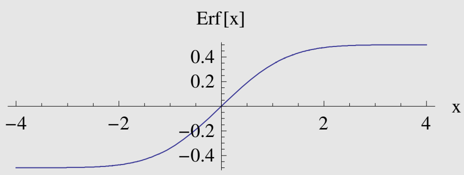
$$\sigma =.;\ \texttt{err}[x\_, \sigma\_] = \int_0^x \frac{1}{\sigma\sqrt{2\pi}} \texttt{Exp}\left[-\frac{y^2}{2\sigma^2}\right] dy$$

$$\frac{1}{2}\,\texttt{Erf}\left[\frac{x}{\sqrt{2}\,\sigma}\right]$$

The *y* in the integral above is just a dummy integration variable, and is integrated out. The *Mathematica* error function is **Erf[x]**.

In our integral of the Gaussian function we need to do the reparametrization $x \to \frac{x}{\sigma\sqrt{2}}$. Again we recognize the natural coordinates. The factor $\frac{1}{2}$ is due to the fact that integration starts halfway, in $x = 0$.

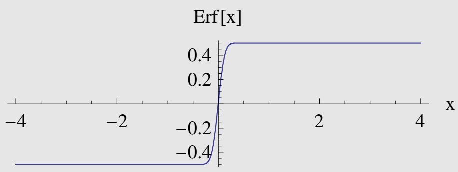
$$\sigma = 1.\grave{};\ \texttt{Plot}\left[\frac{1}{2}\,\texttt{Erf}\left[\frac{x}{\sigma\sqrt{2}}\right],\right.$$
$$\texttt{\{x, -4, 4\}, AspectRatio} \to 0.3\grave{},$$
$$\left.\texttt{AxesLabel} \to \texttt{\{"x", "Erf[x]"\}, ImageSize} \to 300\right]$$



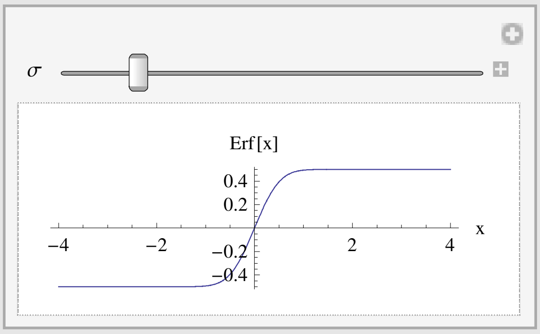

For inner scale $\sigma \downarrow 0$, we get in the limiting case the *Heavyside function* or *unitstep function*.

The derivative of the Heavyside function is the Delta-Dirac function, just as the derivative of the error function of the Gaussian kernel.

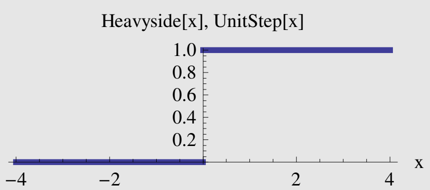
```
σ = 0.1`; Plot[1/2 Erf[x/(σ √2)],
 {x, -4, 4}, AspectRatio → 0.3`,
 AxesLabel → {"x", "Erf[x]"}, ImageSize → 270]
```




```
Manipulate[Plot[1/2 Erf[x/(σ √2)], {x, -4, 4},
  AspectRatio → 0.3`, AxesLabel → {"x", "Erf[x]"},
  ImageSize → 270], {{σ, .4}, .1, 2}]
```




```
Plot[UnitStep[x], {x, -4, 4},
 DisplayFunction → $DisplayFunction, AspectRatio → 0.3`,
 AxesLabel → {"x", "Heavyside[x], UnitStep[x]"},
 PlotStyle → Thickness[0.015`], ImageSize → 270]
```



## 3.6 Separability

The Gaussian kernel for dimensions higher than one, say $N$, can be described as a regular product of $N$ one-dimensional kernels.

Example: $g_{2D}(x, y; \sigma_1^2 + \sigma_2^2) = g_{1D}(x; \sigma_1^2)\, g_{1D}(y; \sigma_2^2)$

where the space in between is the product operator.

Because higher dimensional Gaussian kernels are regular products of one-dimensional Gaussians, they are called *separable*. We will use quite often this property of *separability* because of speed.

```
GraphicsRow[{Plot[gauss[x, σ = 1], {x, -3, 3}],
  Plot3D[gauss[x, σ = 1] gauss[y, σ = 1],
   {x, -3, 3}, {y, -3, 3}]}, ImageSize → 440]
```

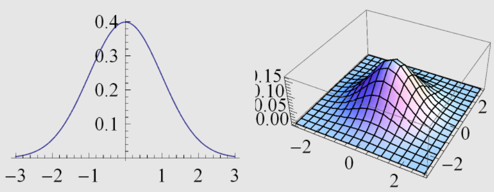



Figure 3.3 A product of Gaussian functions gives a higher dimensional Gaussian function. This is a consequence of the separability.

We get considerable speed improvement when implementing numerical *separable convolution*. The convolution with a 2D (or better: N-dimensional) Gaussian kernel can be replaced by a cascade of 1D convolutions, making the process much more efficient because convolution with the 1D kernels requires far fewer multiplications.

## 3.7 Relation to binomial coefficients

The Gaussian function emerges is in expansions of powers of polynomials:

```
Manipulate[Expand[(x + y)^n], {{n, 10}, 1, 100, 1}]
```

n

$x^{10} + 10\, x^9\, y + 45\, x^8\, y^2 + 120\, x^7\, y^3 + 210\, x^6\, y^4 + 252\, x^5\, y^5 + 210\, x^4\, y^6 + 120\, x^3\, y^7 + 45\, x^2\, y^8 + 10\, x\, y^9 + y^{10}$

The coefficients of this expansion are the *binomial coefficients* $\binom{n}{m}$ ('n over m'):

```
Manipulate[ListPlot[Table[Binomial[nr, n], {n, 1, nr}],
  PlotStyle → {PointSize[0.015`]},
  AspectRatio → 0.3`], {{nr, 30}, 5, 100, 1}]
```

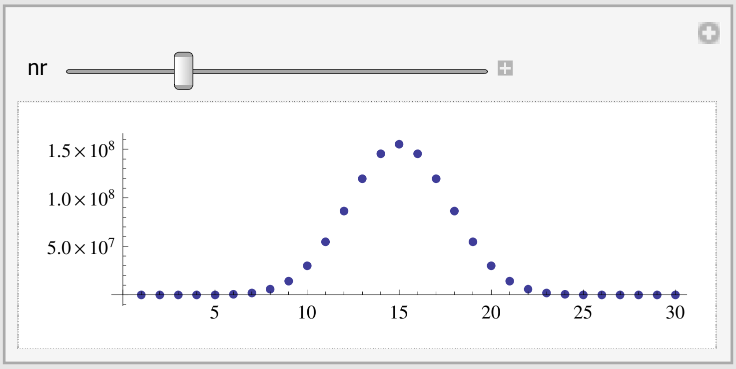
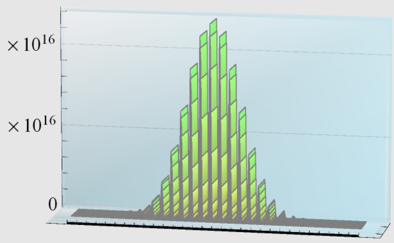



Figure 3.4 Binomial coefficients approximate a Gaussian distribution for increasing order.

And here in two dimensions:

```
BarChart3D[Table[Binomial[30, n] Binomial[30, m],
  {n, 1, 30}, {m, 1, 30}], ImageSize → 180]
```




## 3.8 The Fourier transform of the Gaussian kernel

The basis functions of the Fourier transform $\mathcal{F}$ are the sinusoidal functions $e^{i\omega x}$.

```
ExpToTrig[E^(I ω x)]
```

```
Cos[x ω] + i Sin[x ω]
```

The definitions for the Fourier transform and its inverse are:

- the *Fourier transform*:

$$F(\omega) = \mathcal{F}\{f(x)\} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x)\, e^{i\omega x}\, dx$$

- the *inverse Fourier transform*:

$$\mathcal{F}^{-1}\{F(\omega)\} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(\omega)\, e^{-i\omega x}\, d\omega$$

```
σ = .; ℱgauss[ω_, σ_] =
  Simplify[1/Sqrt[2 π] Integrate[1/(σ Sqrt[2 π]) Exp[-x^2/(2 σ^2)] Exp[I ω x],
     {x, -∞, ∞}], {σ > 0, Im[σ] == 0}]
```

$$\frac{e^{-\frac{1}{2}\sigma^2\omega^2}}{\sqrt{2\pi}}$$

The Fourier transform is a standard *Mathematica* command:

```
Simplify[FourierTransform[gauss[x, σ], x, ω], σ > 0]
```

$$\frac{e^{-\frac{1}{2}\sigma^2\omega^2}}{\sqrt{2\pi}}$$

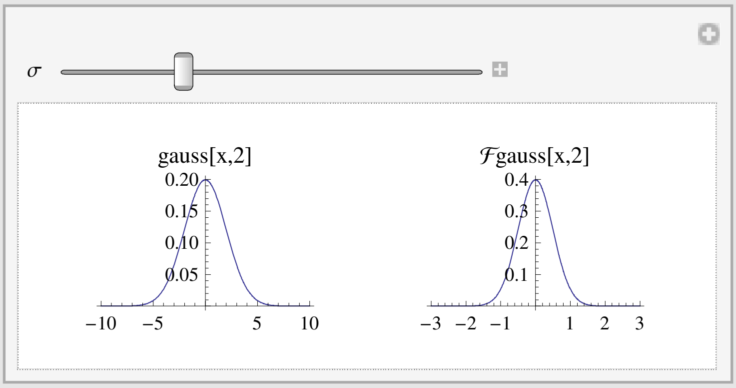
Note that different communities (mathematicians, computer scientists, engineers) have different definitions for the Fourier transform. See the *Mathematica* help function.

In this book we consistently use the default definition.

So the Fourier transform of the Gaussian function is again a Gaussian function, but now of the frequency $\omega$. The Gaussian function is the *only* function with this property. Note that the scale $\sigma$ now appears as a *multiplication* with the frequency.

A smaller kernel in the spatial domain gives a wider kernel in the Fourier domain, and vice versa.

```
Manipulate[GraphicsRow[
  {Plot[gauss[x, σ], {x, -10, 10}, PlotRange → All,
    PlotLabel → "gauss[x," <> ToString[σ] <> "]"],
Plot[ℱgauss[ω, σ], {ω, -3, 3}, PlotRange → All,
    PlotLabel → "ℱgauss[x," <> ToString[σ] <> "]"]}],
 {{σ, 2}, .5, 6, .2}]
```



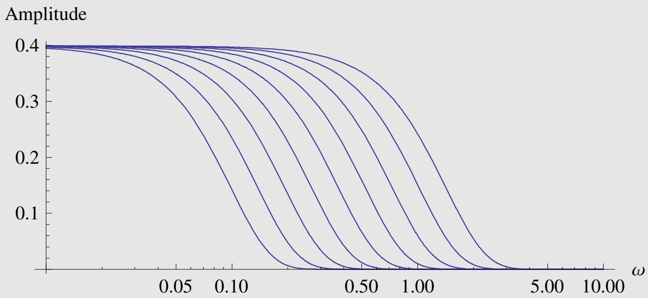
There are many names for the Fourier transform $\mathcal{F}\, g(\omega; \sigma)$ of $g(x; \sigma)$: when the kernel $g(x; \sigma)$ is considered to be the point spread function, $\mathcal{F}\, g(\omega; \sigma)$ is referred to as the *modulation transfer function*.

When the kernel $g(x;\sigma)$ is considered to be a signal, $\mathcal{F}\, g(\omega; \sigma)$ is referred to as the *spectrum*.

When applied to a signal, it operates as a lowpass *filter*.

```
scales = N[Table[ⅇ^(t/3), {t, 0, 8}]]
spectra =
  (LogLinearPlot[ℱgauss[ω, #1], {ω, 0.01`, 10}] &) /@
   scales;
Show[spectra, AspectRatio → 0.4`, PlotRange → All,
 AxesLabel → {"ω", "Amplitude"}, ImageSize → 300]
```

```
{1., 1.39561, 1.94773, 2.71828, 3.79367, 5.29449, 7.38906, 10.3123, 14.3919}
```




## 3.9 Central limit theorem

We see in the paragraph above the relation with the *central limit theorem*: <u>any</u> positive repetitive operator goes in the limit to a Gaussian function.

```
f[x_] :=
  UnitStep[1 / 2 + x] + UnitStep[1 / 2 - x] - 1 // Simplify;
g[x_] := UnitStep[1 / 2 + x] + UnitStep[1 / 2 - x] - 1 //
   Simplify;
```

```
Plot[f[x], {x, -3, 3}, ImageSize → 140]
```

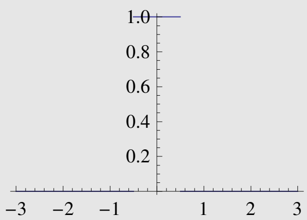


Figure 3.5 The analytical blockfunction is a combination of two Heavyside unitstep functions.

We calculate analytically the convolution integral

```
h1 = Integrate[f[x] g[x - x1], {x, -∞, ∞}]
```

$$\begin{cases} 1 - x1 & 0 < x1 < 1 \\ 1 + x1 & -1 < x1 \leq 0 \\ 0 & \text{True} \end{cases}$$

```
Plot[h1, {x1, -3, 3}, PlotRange → All, ImageSize → 150]
```

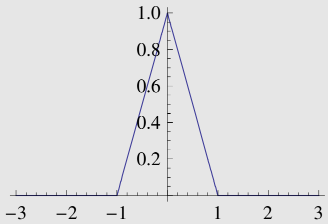



Figure 3.6 One times a convolution of a blockfunction with the same blockfunction gives a triangle function.

The next convolution is this function convolved with the block function again:

```
h2 = Integrate[(h1 /. x1 -> x) g[x - x1], {x, -∞, ∞}]
```

$$\begin{cases} \frac{1}{2} & x1 == \frac{1}{2} \\ \frac{1}{4}\left(3 - 4\, x1^2\right) & -\frac{1}{2} < x1 < \frac{1}{2} \\ \frac{1}{8}\left(9 - 12\, x1 + 4\, x1^2\right) & \frac{1}{2} < x1 < \frac{3}{2} \\ \frac{1}{8}\left(9 + 12\, x1 + 4\, x1^2\right) & -\frac{3}{2} < x1 \leq -\frac{1}{2} \\ 0 & \text{True} \end{cases}$$

We see that we get a result that begins to look more towards a Gaussian:

```
Plot[{h2, gauss[x1, 0.5`]}, {x1, -3, 3}, PlotRange → All,
  PlotStyle → {Blue, Dashing[{0.02, 0.02}]},
  ImageSize → 300]
```

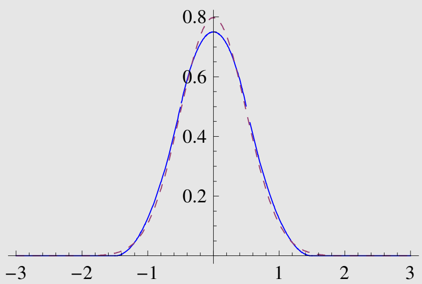



Figure 3.7 Two times a convolution of a blockfunction with the same blockfunction gives a function that rapidly begins to look like a Gaussian function. A Gaussian kernel with $\sigma = 0.5$ is drawn (dotted) for comparison.

Task 3.0 Show the central limit theorem in practice for a number of other arbitrary kernels.

## 3.10 Anisotropy

```
GradientFieldPlot[-gauss[x, 1] gauss[y, 1], {x, -3, 3},
  {y, -3, 3}, PlotPoints → 20, ImageSize → 140]
```

```
GradientFieldPlot[-0.159155 2.71828^(-0.5 x^2-0.5 y^2),
  {x, -3, 3}, {y, -3, 3}, PlotPoints → 20, ImageSize → 140]
```

Figure 3.8 The slope of an isotropic Gaussian function is indicated by arrows here. There are circularly symmetric, i.e. in all directions the same, from which the name *isotropic* derives. The arrows are in the direction of the *normal* of the intensity landscape, and are called *gradient vectors*.

The Gaussian kernel as specified above is *isotropic*, which means that the behaviour of the function is in any *direction* the same.

It is of no use to speak of isotropy in 1-D. When the standard deviations in the different dimensions are not equal, we call the Gaussian function *anisotropic*. An example is the pointspreadfunction of an *astigmatic* eye, where differences in curvature of the cornea/lens in different directions occur. This show an anisotropic Gaussian with anisotropy ratio of 2 $(\sigma_x/\sigma_y = 2)$:

```
Unprotect[gauss]; gauss[x_, y_, σx_, σy_] := e^(-(x^2/(2 σx^2) + y^2/(2 σy^2)))/(2 π σx σy);
σx = 2; σy = 1;
p1 = DensityPlot[gauss[x, y, σx, σy], {x, -10, 10},
   {y, -10, 10}, PlotPoints → 50, PlotRange -> All];
p2 = Plot3D[gauss[x, y, σx, σy], {x, -10, 10},
   {y, -10, 10}, PlotRange -> All, Axes -> False];
p3 = ContourPlot[gauss[x, y, σx, σy],
   {x, -5, 5}, {y, -10, 10}, PlotRange -> All];
GraphicsRow[{p1, p2, p3}, ImageSize → 400]
```

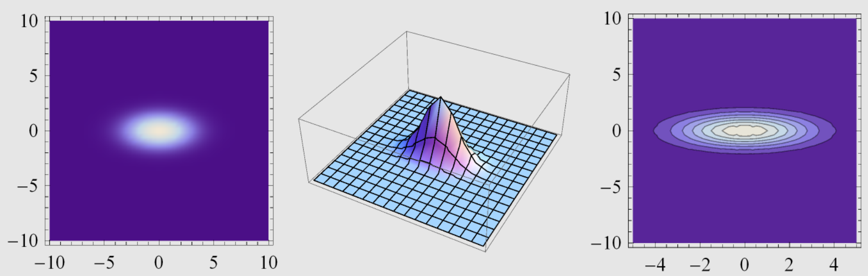



Figure 3.9 An anisotropic Gaussian kernel with anisotropy ratio $\sigma_x/\sigma_y = 2$ in three appearances. Left: **DensityPlot**, middle: **Plot3D**, right: **ContourPlot**.

## 3.11 The diffusion equation

The Gaussian function is the solution of several differential equations. It is the solution of $\frac{dy}{dx} = \frac{y(\mu - x)}{\sigma^2}$, because $\frac{dy}{y} = \frac{(\mu - x)}{\sigma^2} dx$, from which we find by integration $\ln\left(\frac{y}{y_0}\right) = -\frac{(\mu - x)^2}{2\sigma^2}$ and thus $y = y_0 e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

It is the solution of the linear diffusion equation, $\frac{\partial L}{\partial t} = \frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2} = \Delta L$.

```
hessian2D = ( Lxx  Lxy
              Lxy  Lyy ); Tr[hessian2D]
```

$L_{xx} + L_{yy}$

```
hessian3D = ( Lxx  Lxy  Lxz
              Lyx  Lyy  Lyz
              Lzx  Lyz  Lzz ); Tr[hessian3D]
```

$L_{xx} + L_{yy} + L_{zz}$

The diffusion equation $\frac{\partial u}{\partial t} = \Delta u$ is one of some of the most famous differential equations in physics. It is often referred to as the *heat equation*. It belongs in the row of other famous equations like the Laplace equation $\Delta u = 0$, the wave equation $\frac{\partial^2 u}{\partial t^2} = \Delta u$ and the Schrödinger equation $\frac{\partial u}{\partial t} = i \Delta u$.

The diffusion equation $\frac{\partial u}{\partial t} = \Delta u$ is a *linear* equation. It consists of just linearly combined derivative terms, no nonlinear exponents or functions of derivatives.

The diffused entity is the intensity in the images. The role of time is taken by the variance $t = 2\sigma^2$. The intensity is diffused over time (in our case over scale) in all directions in the same way (this is called *isotropic*). E.g. in 3D one can think of the example of the intensity of an inkdrop in water, diffusing in all directions.

The diffusion equation can be derived from physical principles: the luminance can be considered a *flow*, that is pushed away from a certain location by a force equal to the gradient. The divergence of this gradient gives how much the total entity (luminance in our case) diminishes with time.

```
<< "VectorAnalysis`"
SetCoordinates[Cartesian[x, y, z]];
```

```
Div[ Grad[L[x, y, z]]]
```

$L^{(0,0,2)}[x, y, z] + L^{(0,2,0)}[x, y, z] + L^{(2,0,0)}[x, y, z]$

A very important feature of the diffusion process is that it satisfies a *maximum principle* [Hummel1987b]: the amplitude of local maxima are always decreasing when we go to coarser scale, and vice versa, the amplitude of local minima always increase for coarser scale.

This argument was the principal reasoning in the derivation of the diffusion equation as the generating equation for scale-space by Koenderink [Koenderink1984a].