and, introducing the shorthand $g(z) = f(z)/z^{m+1}$ for the integrand, the trapezoidal rule approximation to the integral is

$$\begin{aligned}
\oint g(z)\,\mathrm{d}z &\simeq \sum_{k=0}^{N-1} \tfrac{1}{2}\big[g(z_{k+1}) + g(z_k)\big]\big[\mathrm{e}^{\mathrm{i}2\pi(k+1)/N} - \mathrm{e}^{\mathrm{i}2\pi k/N}\big] \\
&= \tfrac{1}{2}\Bigg[\sum_{k=0}^{N-1} g(z_{k+1})\,\mathrm{e}^{\mathrm{i}2\pi(k+1)/N} - \sum_{k=0}^{N-1} g(z_k)\,\mathrm{e}^{\mathrm{i}2\pi k/N} \\
&\qquad - \sum_{k=0}^{N-1} g(z_{k+1})\,\mathrm{e}^{\mathrm{i}2\pi k/N} + \sum_{k=0}^{N-1} g(z_k)\,\mathrm{e}^{\mathrm{i}2\pi(k+1)/N}\Bigg].
\end{aligned}$$

Noting that $z_N = z_0$, the first two sums inside the brackets cancel each other in their entirety, and the remaining two sums are equal except for trivial phase factors, so the entire expression simplifies to

$$\begin{aligned}
\oint g(z)\,\mathrm{d}z &\simeq \tfrac{1}{2}\big[\mathrm{e}^{\mathrm{i}2\pi/N} - \mathrm{e}^{-\mathrm{i}2\pi/N}\big]\sum_{k=0}^{N-1} g(z_k)\,\mathrm{e}^{\mathrm{i}2\pi k/N} \\
&\simeq \frac{2\pi\mathrm{i}}{N}\sum_{k=0}^{N-1} f(z_k)\,\mathrm{e}^{-\mathrm{i}2\pi km/N},
\end{aligned}$$

where we have used the definition of $g(z)$ again. Combining this result with the Cauchy formula, we then have

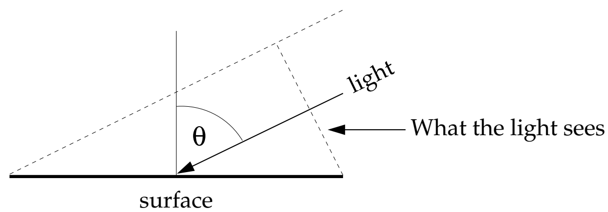
$$\left(\frac{\mathrm{d}^m f}{\mathrm{d}z^m}\right)_{z=0} \simeq \frac{m!}{N}\sum_{k=0}^{N-1} f(z_k)\,\mathrm{e}^{-\mathrm{i}2\pi km/N}.$$

Write a program to calculate the first twenty derivatives of $f(z) = \mathrm{e}^{2z}$ at $z = 0$ using this formula with $N = 10000$. You will need to use the version of the `exp` function from the `cmath` package, which can handle complex arguments. You may also find the function `factorial` from the `math` package useful; it calculates factorials of integer arguments.

The correct value for the mth derivative in this case is easily shown to be 2^m, so it should be straightforward to tell if your program is working—the results should be powers of two, 2, 4, 8, 16, 32, etc. You should find that it is possible to get reasonably accurate results for all twenty derivatives rapidly using this technique. If you use standard difference formulas for the derivatives, on the other hand, you will find that you can calculate only the first three or four derivatives accurately before the numerical errors become so large that the results are useless. In this case, therefore, the Cauchy formula gives the better results.

The sum $\sum_k f(z_k)\,\mathrm{e}^{\mathrm{i}2\pi km/N}$ that appears in the formula above is known as the *discrete Fourier transform* of the complex samples $f(z_k)$. There exists an elegant technique for evaluating the Fourier transform for many values of m simultaneously, known as the *fast Fourier transform*, which could be useful in cases where the direct evaluation of the formula is slow. We will study the fast Fourier transform in detail in Chapter 7.

5.23 Image processing and the STM: When light strikes a surface, the amount falling per unit area depends not only on the intensity of the light, but also on the angle of incidence. If the light makes an angle θ to the normal, it only "sees" $\cos\theta$ of area per unit of actual area on the surface:



So the intensity of illumination is $a\cos\theta$, if a is the raw intensity of the light. This simple physical law is a central element of 3D computer graphics. It allows us to calculate how light falls on three-dimensional objects and hence how they will look when illuminated from various angles.

Suppose, for instance, that we are looking down on the Earth from above and we see mountains. We know the height of the mountains $w(x,y)$ as a function of position in the plane, so the equation for the Earth's surface is simply $z = w(x,y)$, or equivalently $w(x,y) - z = 0$, and the normal vector $\mathbf{v}$ to the surface is given by the gradient of $w(x,y) - z$ thus:

$$\mathbf{v} = \nabla[w(x,y) - z] = \begin{pmatrix} \partial/\partial x \\ \partial/\partial y \\ \partial/\partial z \end{pmatrix} [w(x,y) - z] = \begin{pmatrix} \partial w/\partial x \\ \partial w/\partial y \\ -1 \end{pmatrix}.$$

Now suppose we have light coming in represented by a vector $\mathbf{a}$ with magnitude equal to the intensity of the light. Then the dot product of the vectors $\mathbf{a}$ and $\mathbf{v}$ is

$$\mathbf{a} \cdot \mathbf{v} = |\mathbf{a}|\,|\mathbf{v}| \cos\theta,$$

where θ is the angle between the vectors. Thus the intensity of illumination of the surface of the mountains is

$$I = |\mathbf{a}| \cos\theta = \frac{\mathbf{a}\cdot\mathbf{v}}{|\mathbf{v}|} = \frac{a_x(\partial w/\partial x) + a_y(\partial w/\partial y) - a_z}{\sqrt{(\partial w/\partial x)^2 + (\partial w/\partial y)^2 + 1}}.$$

Let's take a simple case where the light is shining horizontally with unit intensity, along a line an angle ϕ counter-clockwise from the east-west axis, so that $\mathbf{a} = (\cos\phi, \sin\phi, 0)$. Then our intensity of illumination simplifies to

$$I = \frac{\cos\phi\,(\partial w/\partial x) + \sin\phi\,(\partial w/\partial y)}{\sqrt{(\partial w/\partial x)^2 + (\partial w/\partial y)^2 + 1}}.$$

If we can calculate the derivatives of the height $w(x,y)$ and we know ϕ we can calculate the intensity at any point.

a) In the on-line resources you'll find a file called `altitude.txt`, which contains the altitude $w(x, y)$ in meters above sea level (or depth below sea level) of the surface of the Earth, measured on a grid of points (x, y). Write a program that reads this file and stores the data in an array. Then calculate the derivatives $\partial w/\partial x$ and $\partial w/\partial y$ at each grid point. Explain what method you used to calculate them and why. (Hint: You'll probably have to use more than one method to get every grid point, because awkward things happen at the edges of the grid.) To calculate the derivatives you'll need to know the value of h, the distance in meters between grid points, which is about 30 000 m in this case.[7]

b) Now, using your values for the derivatives, calculate the intensity for each grid point, with $\phi = 45°$, and make a density plot of the resulting values in which the brightness of each dot depends on the corresponding intensity value. If you get it working right, the plot should look like a relief map of the world—you should be able to see the continents and mountain ranges in 3D. (Common problems include a map that is upside-down or sideways, or a relief map that is "inside-out," meaning the high regions look low and *vice versa*. Work with the details of your program until you get a map that looks right to you.)

c) There is another file in the on-line resources called `stm.txt`, which contains a grid of values from scanning tunneling microscope measurements of the (111) surface of silicon. A scanning tunneling microscope (STM) is a device that measures the shape of surfaces at the atomic level by tracking a sharp tip over the surface and measuring quantum tunneling current as a function of position. The end result is a grid of values that represent the height of the surface as a function of position and the data in the file `stm.txt` contain just such a grid of values. Modify the program you just wrote to visualize the STM data and hence create a 3D picture of what the silicon surface looks like. The value of h for the derivatives in this case is around $h = 2.5$ (in arbitrary units).

[7] It's actually not precisely constant because we are representing the spherical Earth on a flat map, but $h = 30\,000$ m will give reasonable results.