

Computing with Harmonic Functions

Sheldon Axler

13 November 2015

@Sheldon Axler

Getting Started

Harmonic Function Theory

This document is the manual for HFT10.m (version 10.00), a *Mathematica* package for computing with harmonic functions. The HFT10.m package allows the user to make calculations that would take a prohibitive amount of time if done without a computer. For example, the Poisson integral of any polynomial can be computed exactly.

Earlier versions of this software focused on algorithms arising from the material in the book *Harmonic Function Theory* [ABR] by Sheldon Axler, Paul Bourdon, and Wade Ramey. That book is still the source for many of the algorithms used in the HFT10.m package, but the goal of the package has expanded to include additional symbolic manipulations involving harmonic functions. The HFT10.m package can be used without the *Harmonic Function Theory* book, but the debt to the book is reflected in the initials chosen for the name of the package.

This document, the HFT10.m package, and the HFT10.nb notebook are intended for use with *Mathematica* versions 7 through 10 (the package will probably also work with later versions of *Mathematica*).

This document (which is available in both nb and pdf formats) and the HFT10.m package and the HFT10.nb notebook that serves as the source file for the HFT10.m package are available electronically free of charge. The most recent versions are available at [http : // www.axler.net / HFT_Math.html](http://www.axler.net/HFT_Math.html). New features are added periodically, so check for new versions occasionally.

Comments, suggestions, and bug reports about this document, the HFT10.m package, or the book *Harmonic Function Theory* are welcome; please send them via e-mail to axler@sfsu.edu.

Work on some of the algorithms used in this package was partially supported by research grants to the author from the National Science Foundation.

Loading the Package

To begin a session with the HFT10.m package, first start *Mathematica*. Then enter the command shown below, making an appropriate modification to indicate the directory or folder in which HFT10.m is located on your computer. Or, after typing <<, you can use the File Path command, located on the Insert menu, to paste the full pathname of HFT10.m into your input cell:

In[1]:=

```
<< "C:\\Box Sync\\math\\publications\\HFT-Mathematica\\HFT10.m"  
HFT10.m, version 10.00, 12 November 2015; for use with Mathematica  
versions 7 through 10 (and probably later versions of Mathematica).
```

The HFT10.m *Mathematica* package is
designed for computing with harmonic functions.

Documentation for the use of this package and information about the
algorithms used in it is available in the document titled *Computing
with Harmonic Functions*, which is available in both nb and pdf formats.

The most recent version of this HFT10.m package and its documentation
Computing with Harmonic Functions are available at <http://www.axler.net>.

For additional information about harmonic functions, see the
book *Harmonic Function Theory* (second edition), by Sheldon
Axler, Paul Bourdon, and Wade Ramey, published by Springer.

This package is copyrighted by
Sheldon Axler but is distributed without charge.

Comments, suggestions, and bug reports
should be sent by electronic mail to axler@sfsu.edu.

* You can now use the functions in this package.

After loading the HFT10.m package, you can use the functions described in the rest of this document. The nb version of this document is a live *Mathematica* notebook, meaning that you can and should evaluate and modify the input cells in this notebook. The pdf version of this document is a static document. The nb and pdf versions of this document and the HFT package are revised periodically with the addition of new features; new versions are posted at http://www.axler.net/HFT_Math.html.

On-line Help

This document provides the only complete documentation for using the HFT10.m package. However, users can be reminded about the basic use and syntax of each function created by HFT10.m by entering ? followed by the function name:

In[2]:=

```
? volume
```

volume[n] gives the volume of the unit ball in n-dimensional real Euclidean space.

The *Mathematica* statement `Names["HFT`*"]` will produce a complete list of the functions created by HFT10.m:

In[3]:=

```
Names["HFT`*"]
```

Out[3]=

```
{annulus, antiLaplacian, basisH, bergmanKernel, bergmanKernelH,
 bergmanProjection, biDirichlet, dimension, dimHarmonic, dirichlet,
 divergence, expandNorm, exteriorNeumann, exteriorSphere, gradient,
 harmonicConjugate, harmonicDecomposition, hilbertSchmidt, homogeneous,
 integrateBall, integrateSphere, jacobian, kelvin, kelvinH,
 laplacian, multiple, neumann, norm, normalD, partial, poissonKernel,
 poissonKernelH, quadratic, reflection, region, schwarz, setDimension,
 singularity, southPole, surfaceArea, taylor, togetherness, trace,
 turnOff, turnOn, volume, zeroToZero, zonalHarmonic,  $\delta$ ,  $\Delta$ ,  $\Phi$ }
```

The naming scheme for functions introduced by this package is that function names are generally spelled out in full and begin with a lower-case letter, as in **laplacian**. This is a major change from HFT9 and previous versions of the package, where function names began with an upper-case letter. This change was made because new versions of *Mathematica* have introduced new functions beginning with an upper-case letter, which have clashed with names already used by the HFT package. With this change (meaning that HFT functions now begin with a lower-case letter), this issue has disappeared.

Functions in the HFT10 package with a name formed from more than one word begin with a lower-case letter but then have the first letter of additional words begin with an upper-case letter, as in **surfaceArea**.

Calculus in \mathbf{R}^n

norm

norm[**x**] is the Euclidean norm of a vector x . When using StandardForm for output (which is the default in *Mathematica*), this package will use $\|\mathbf{x}\|$ instead of **norm**[**x**]. When using StandardForm for input (which is the default in *Mathematica*), you can use either $\|\mathbf{x}\|$ or **norm**[**x**]:

```
In[4]:=  $\|\mathbf{5\ x}\| - \mathbf{norm}[2\ \mathbf{x}]^2$ 
```

```
Out[4]=  $5\ \|\mathbf{x}\| - 4\ \|\mathbf{x}\|^2$ 
```

To type $\|\mathbf{x}\|$, type escape-key **l** | | escape-key (the three entries between the two escape-keys are one lower-case l followed by two |), then **x**, then escape-key **r** | | escape-key (the three entries between the two escape-keys are one lower-case r followed by two |). To help you remember, l is associated with the left norm bracket and r is associated with the right norm bracket. Even though $\|\mathbf{x}\|$ looks better, it may be easier to type **norm**[**x**].

expandNorm

expandNorm expands norms of sums (and differences) by replacing expressions of the form $\|x + y\|$ with $(\|x\|^2 + \|y\|^2 + x \cdot y)^{1/2}$. **expandNorm** is often useful for verifying identities. For example, to prove the Symmetry Lemma (1.13 in [ABR]), which states that

$$\left\| \frac{y}{\|y\|} - \|y\| x \right\| = \left\| \frac{x}{\|x\|} - \|x\| y \right\|$$

for all nonzero $x, y \in \mathbf{R}^n$, we subtract the left-hand side of the alleged equality from the right-hand side, hoping that the result equals 0:

$$\text{In[5]:= } \left\| \frac{\mathbf{x}}{\|\mathbf{x}\|} - \frac{\|\mathbf{x}\| \mathbf{y}}{\|\mathbf{y}\|} \right\| - \left\| \frac{\mathbf{y}}{\|\mathbf{y}\|} - \frac{\|\mathbf{y}\| \mathbf{x}}{\|\mathbf{x}\|} \right\|$$

$$\text{Out[5]= } \left\| \frac{\mathbf{x}}{\|\mathbf{x}\|} - \frac{\mathbf{y} \|\mathbf{x}\|}{\|\mathbf{y}\|} \right\| - \left\| \frac{\mathbf{y}}{\|\mathbf{y}\|} - \frac{\mathbf{x} \|\mathbf{y}\|}{\|\mathbf{x}\|} \right\|$$

The last output does not look like 0, so we apply `expandNorm` to it. The symbol `%` used below is the *Mathematica* abbreviation for the last output:

$$\text{In[6]:= } \text{expandNorm}[\%]$$

$$\text{Out[6]= } 0$$

Because the last output equals 0, the Symmetry Lemma has been proved.

partial

`partial[f, xj]` is the partial derivative of f with respect to x_j . Here x denotes a vector in Euclidean space, and x_j is the j^{th} coordinate of x . The advantage of using `partial` rather than the built-in *Mathematica* function `D` or `∂` is that `partial` knows that x_j is the j^{th} coordinate of x , and furthermore `partial` can deal with norms and dot products. Note that the dot product of b and x is denoted by `b.x`:

$$\text{In[7]:= } \text{partial}[\|\mathbf{a} + \mathbf{x}\|^{b.x}, x_5]$$

$$\text{Out[7]= } \text{Log}[\|\mathbf{a} + \mathbf{x}\|] \|\mathbf{a} + \mathbf{x}\|^{b.x} b_5 + b.x \|\mathbf{a} + \mathbf{x}\|^{-2+b.x} (a_5 + x_5)$$

A subscript can be typed in *Mathematica* by pressing control-dash. Return to normal mode by pressing control-space. A superscript or exponent can be typed by pressing control-6. Instead of entering `xj`, you can type `x[j]`.

For taking multiple partial derivatives, `partial` uses the same syntax as the built-in *Mathematica* function `D`. For example, the following command computes the partial derivative of $\|x\|$ once with respect to the second coordinate, twice with respect to the first coordinate, and three times with

respect to the fourth coordinate:

```
In[8]:= partial[||x||, x2, {x1, 2}, {x4, 3}]
```

```
Out[8]:= -\frac{1}{||x||^{11}} 15 (3 ||x||^4 x2 x4 - 21 ||x||^2 x1^2 x2 x4 - 7 ||x||^2 x2 x4^3 + 63 x1^2 x2 x4^3)
```

Δ (laplacian)

$\Delta_x [f]$ is the Laplacian of f with respect to x :

```
In[9]:=  $\Delta_x[x_3 ||x||]$ 
```

```
Out[9]:= \frac{x_3 + dimension[x] x_3}{||x||}
```

To type Δ , type escape-key D escape-key. Instead of Δ , you can use **laplacian**; thus **laplacian_x[f]** is the same as $\Delta_x [f]$.

Note that in the last result, the Laplacian depends upon `dimension[x]`, which is the dimension of the Euclidean space in which x lives. For work in \mathbf{R}^n , **setDimension[x, n]** should be used to set the dimension of x equal to n ; here n can be a symbol or a specific integer, say 8. Be sure to use the **setDimension** function rather than changing the value of `dimension[x]` directly. We illustrate this procedure by showing that for each ζ with $|\zeta| = 1$, the Poisson kernel $P(\cdot, \zeta)$ is harmonic (in other words, what follows is a proof of Proposition 1.18 of [ABR]):

```
In[10]:= setDimension[x, n]
```

setDimension::Vector :

x will be considered to be a vector in n-dimensional real Euclidean space.

```
In[11]:=  $\Delta_x[(1 - ||x||^2) / ||x - \zeta||^n]$ 
```

```
Out[11]:= 4 n (-x.\zeta + ||x||^2) ||x - \zeta||^{-2-n} - 2 n ||x - \zeta||^{-n} +
(1 - ||x||^2) (-(-1 + n) n ||x - \zeta||^{-2-n} - n (1 + n) ||x - \zeta||^{-4-n} (2 x.\zeta - ||x||^2 - ||\zeta||^2))
```

```
In[12]:= expandNorm[%] /. ||ξ|| → 1
```

```
Out[12]= 0
```

Thus the Laplacian of $P(\cdot, \zeta)$ is 0, and hence $P(\cdot, \zeta)$ is harmonic, as claimed.

If x and y are symbols, then $\Delta_{x,y}[f]$ gives the Laplacian of f with respect to the vector (x, y) , where x is thought of as a vector, y is thought of as a real variable, and (x, y) is thought of as a vector in a Euclidean space whose dimension is one more than the dimension of x . This format is often useful when working with functions defined on the upper half-space, as in Chapter 7 of [ABR]. Note that if we are thinking of (x, y) as an element of \mathbf{R}^n , then x should be defined to have dimension $n - 1$. We illustrate the use of this format of the Δ function by showing that for each $t \in \mathbf{R}^{n-1}$, the Poisson kernel $P_H(\cdot, t)$ for the upper half-space is harmonic (see page 145 of [ABR] for another proof of this):

```
In[13]:= setDimension[x, n - 1]
```

```
setDimension::Vector :
```

```
x will be considered to be a vector in -1 + n-dimensional real Euclidean space.
```

```
In[14]:= Δx,y[y / (||x - t||2 + y2)n/2]
```

```
Out[14]= -  $\frac{1}{\| -t + x \|^2}$   

 $n y (y^2 + \| -t + x \|^2)^{-2 - \frac{n}{2}} (2 t \cdot x - \| t \|^2 - \| x \|^2 + \| -t + x \|^2) (-y^2 + \| -t + x \|^2 + n \| -t + x \|^2)$ 
```

```
In[15]:= expandNorm[%]
```

```
Out[15]= 0
```

Thus the Laplacian of $P_H(\cdot, t)$ is 0, and hence $P_H(\cdot, t)$ is harmonic, as claimed.

The expression $\Delta_v[f]$, where v is a list of explicit coordinates, gives the Laplacian of f with respect to the coordinates in the list v . For example, below we compute the Laplacian of $x^2 y^3 z^4$ with respect to the usual coordinates x, y , and z of \mathbf{R}^3 :

```
In[16]:=  $\Delta_{\{x,y,z\}} [x^2 y^3 z^4]$ 
```

```
Out[16]=  $12 x^2 y^3 z^2 + 6 x^2 y z^4 + 2 y^3 z^4$ 
```

Note that in the example above, x is no longer being used as a vector but as the first coordinate in \mathbf{R}^3 . Because x has been used in a list of coordinates, the software no longer treats x as a vector.

If m is a positive integer, then $\Delta_{\mathbf{x}}^m$ can be used, with the same syntax as Δ_x , to evaluate the m^{th} power of the Laplacian. For example, here we find the biLaplacian (the Laplacian of the Laplacian) of $1/\|x\|$ on \mathbf{R}^8 :

```
In[17]:= setDimension[x, 8]
```

```
setDimension::Vector :
```

```
x will be considered to be a vector in 8-dimensional real Euclidean space.
```

```
In[18]:=  $\Delta_{\mathbf{x}}^2 [1 / \|\mathbf{x}\|]$ 
```

```
Out[18]=  $\frac{45}{\|\mathbf{x}\|^5}$ 
```

gradient

`gradient[f, x]` is the gradient of f with respect to the vector x :

```
In[19]:= gradient[a . x + x_k + \|x\|, x]
```

```
Out[19]=  $a + \frac{x}{\|x\|} + \delta_k$ 
```

In the last output, δ_k denotes the vector that equals 1 in the k^{th} coordinate and 0 in the other coordinates. To type δ , type escape-key d escape-key.

normalD

normalD[*f*, *z*] equals the outward normal derivative of *f* as a function of *z*, where we also think of *z* as a point on the unit sphere in a Euclidean space.

In the section of Chapter 1 of [ABR] titled *The Poisson Kernel for the Ball*, the formula for the Poisson kernel was derived by showing that $P(x, z)$ equals the normal derivative (with respect to *z*) of

$$\left(\|z - x\|^{2-n} - \|x\|^{2-n} \left\| z - \frac{x}{\|x\|^2} \right\|^{2-n} \right) / (2 - n).$$

The computation of that normal derivative is a bit complicated and was left to the reader. Now we show how this computation can be done easily by a computer. We begin by taking the normal derivative of the function above:

```
In[20]:= normalD [ ( \|z - x\|^{2-n} - \|x\|^{2-n} \|z - \frac{x}{\|x\|^2}\|^{2-n} ) / (2 - n) , z ]
```

```
Out[20]= \frac{1}{2 - n} \left( (2 - n) (1 - x \cdot z) \| -x + z \|^{-n} - (-2 + n) \|x\|^{-n} (x \cdot z - \|x\|^2) \left\| z - \frac{x}{\|x\|^2} \right\|^{-n} \right)
```

Now we ask the computer to replace $\left\| z - \frac{x}{\|x\|^2} \right\|$ in the last output with $\|z - x\| / \|x\|$; the Symmetry Lemma (1.13 of [ABR]) implies that the two expressions are equal:

```
In[21]:= % /. \|z - \frac{x}{\|x\|^2}\| -> \|z - x\| / \|x\|
```

```
Out[21]= \frac{1}{2 - n} \left( (2 - n) (1 - x \cdot z) \| -x + z \|^{-n} + (-2 + n) \|x\|^{-n} (-x \cdot z + \|x\|^2) \left( \frac{\| -x + z \|}{\|x\|} \right)^{-n} \right)
```

Applying the *Mathematica* function **PowerExpand** to the last output gives the desired result:

```
In[22]:= PowerExpand [%]
```

```
Out[22]= - (-1 + \|x\|^2) \| -x + z \|^{-n}
```

The last output completes the derivation of the formula for the Poisson kernel for the ball.

Matrices

Matrix dimensions

Mathematica uses `.` to represent matrix multiplication as well as dot product. When using this package, you must use the `setDimension` command to tell *Mathematica* whenever a symbol is being thought of as a matrix. To think of A as an m -by- n matrix, enter the command `setDimension[A, {m, n}]`; here m and n can be symbols or specific positive integers such as 7 and 3. For the examples in this section, we want to think of A as a square matrix, so we make the two dimensions of A equal:

```
In[23]:= setDimension[A, {n, n}]
```

```
setDimension::Matrix : A will be considered to be a matrix of size n-by-n.
```

```
In[24]:= gradient[||A.x||, x]
```

```
Out[24]=
```

$$\frac{(A.x) . A}{||A.x||}$$

To interpret input or output involving products of matrices and vectors, think of vectors as either column vectors or row vectors, whichever makes sense in context. Thus in the input above, x is obviously a vector, because we are taking the gradient of some function with respect to x . For $A.x$ to make sense as a vector in \mathbf{R}^n in the input above, we must think of x as a column vector (an n -by-1 matrix). To interpret $(A.x).A$ in the output above, we again think of x as a column vector (an n -by-1 matrix), so that $A.x$ makes sense as an element of \mathbf{R}^n . Because $A.x$ is then multiplied by the n -by- n matrix A , we must think of $A.x$ as a row vector (an 1-by- n matrix), so that $(A.x).A$ is a vector, as expected here (because it is the gradient of a function).

In the example above we did not enter the command `setDimension[x, n]` to tell *Mathematica* that x is a vector (although it would not have hurt to do so). The HFT10.m package can almost always use the context to distinguish scalars from vectors (if both interpretations make sense, the package assumes that a symbol represents a scalar). However, you must always explicitly use the `setDimension` command to tell *Mathematica* which objects are matrices.

The expression **dimension[A]** gives the dimensions of a matrix A . However, be sure to use the **setDimension** function rather than changing the value of **dimension[A]** directly.

Rows and columns

If A is a matrix, then $A_{j,}$ denotes the j^{th} row of A and $A_{.,j}$ denotes the j^{th} column of A :

```
In[25]:= partial[||A.x|| + 5 ||x.A||, x3]
```

```
Out[25]= 
$$\frac{(A.x).A_{.,3}}{\|A.x\|} + \frac{5(x.A).A_{3,.}}{\|x.A\|}$$

```

Note that in the input above, x is thought of as a column vector in the term $A.x$ and as a row vector in the term $x.A$. To interpret $(A.x).A_{.,3}$ in the output above, read from left to right and think of x as a column vector, so that $A.x$ is a vector; then $(A.x).A_{.,3}$ is just the dot product of two vectors, so that it is a number, as expected here. Similarly, to interpret $(x.A).A_{3,.}$ in the output above, read from left to right and think of x as a row vector, so that $x.A$ is a vector; then $(x.A).A_{3,.}$ is just the dot product of two vectors, so that it is a number, as expected here

Hilbert Schmidt norm

hilbertSchmidt[A] is the Hilbert-Schmidt norm of a matrix A , which is the square root of the sum of the squares of all entries of A :

```
In[26]:= hilbertSchmidt[4 IdentityMatrix[n]]
```

```
Out[26]= 4 n
```

```
In[27]:= Δx[||x.A||]
```

```
Out[27]= 
$$\frac{1}{\|x.A\|^3} \left( \text{hilbertSchmidt}[A]^2 \|x.A\|^2 - \|x.A.\text{Transpose}[A]\|^2 \right)$$

```

Trace

trace [A] is the trace of a square matrix A :

```
In[28]:= trace[-5 IdentityMatrix[n] ]
```

```
Out[28]= -5 n
```

divergence

divergence [f, x] is the divergence of f with respect to the vector x :

```
In[29]:= setDimension[A, {n, n}]
```

setDimension::Matrix : A will be considered to be a matrix of size n-by-n.

```
In[30]:= divergence[A.x / ||x||, x]
```

```
Out[30]= -  $\frac{x \cdot (A.x)}{\|x\|^3} + \frac{\text{trace}[A]}{\|x\|}$ 
```

jacobian

jacobian [f, x] is the Jacobian derivative of f with respect to x . Here f should be a function of x taking values in some Euclidean space, and the Jacobian derivative is the usual matrix consisting of partial derivatives of the coordinates of f :

```
In[31]:= setDimension[x, n]
```

setDimension::Vector :

x will be considered to be a vector in n-dimensional real Euclidean space.

```
In[32]:= jacobian[x / ||x||^2, x]
```

```
Out[32]= -  $\frac{2 \text{Transpose}[\{x\}] \cdot \{x\}}{\|x\|^4} + \frac{\text{IdentityMatrix}[n]}{\|x\|^2}$ 
```

Although x can be thought of as a row vector or a column vector, depending on the context, if we were using explicit coordinates we would represent x in *Mathematica* by something like $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$. Thus $\{\mathbf{x}\}$ would be $\{\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}\}$, which is a 1-by- n matrix. Thus **Transpose** $[\{\mathbf{x}\}]$ is an n -by-1 matrix, and the product `Transpose[{x}] . {x}`, which appears in the last output, is an n -by- n matrix, as expected.

homogeneous

homogeneous $[u, m, \mathbf{x}]$ is the term of degree m in the homogeneous expansion of u at the origin; here u is thought of as a function of the vector x :

```
In[33]:= setDimension[x, 3]
```

```
setDimension::Vector :  
x will be considered to be a vector in 3-dimensional real Euclidean space.
```

```
In[34]:= homogeneous[(1 - ||x||^2) / Cos[x1]^3/2, 4, x]
```

```
Out[34]=  $\frac{1}{32} (-11 x_1^4 - 24 x_1^2 x_2^2 - 24 x_1^2 x_3^2)$ 
```

To find homogeneous expansions about points other than the origin, use **Homogeneous** $[u, m, \mathbf{x}, \mathbf{b}]$, which gives the term of degree m in the homogeneous expansion of u , as a function of x , about b :

```
In[35]:= homogeneous[x1 x2 + x34, 2, x, b]
```

togetherness::Off : togetherness has been turned off. The command turnOn[togetherness] will turn it back on.

```
Out[35]= (-b1 + x1) (-b2 + x2) + 6 b32 (-b3 + x3)2
```

The first time in each *Mathematica* session that you use **homogeneous** to find an expansion about a point other than the origin, you may see a message that `togetherness` has been turned off. The togetherness subsection later in this document provides an explanation for this message.

taylor

taylor[u, m, x] is the sum of all terms of degree at most m in the Taylor series expansion of u at the origin; here u is thought of as a function of the vector x :

```
In[36]:= taylor[x12 / (6 + x2 + Cos[x3]), 4, x]
```

```
Out[36]=  $\frac{x_1^2}{7} - \frac{1}{49} x_1^2 x_2 + \frac{1}{343} x_1^2 x_2^2 + \frac{1}{98} x_1^2 x_3^2$ 
```

To find Taylor series expansions about points other than the origin, use **taylor[u, m, x, b]**, which gives the sum of all terms of degree at most m in the Taylor series expansion of u , as a function of x , about b :

```
In[37]:= taylor[1 + x1 x2 + x12, 2, x, b]
```

```
Out[37]= 1 + b12 + b1 b2 + (2 b1 + b2) (-b1 + x1) + (-b1 + x1)2 + b1 (-b2 + x2) + (-b1 + x1) (-b2 + x2)
```

The first time in each *Mathematica* session that you use **Taylor** to find an expansion about a point other than the origin, you may see a message that `togetherness` has been turned off. The togetherness subsection later in this document provides an explanation for this message. For now, we turn `togetherness` back on so that further output in this document will be displayed nicely.

```
In[38]:= turnOn[togetherness]
```

```
togetherness::On : togetherness has been turned
on. The command turnOff[togetherness] will turn it back off.
```

Nonexplicit functions

Our examples so far have involved only concretely defined functions. But functions in symbolic form can also be used with each of the differentiation commands in this package. Here, for example, is how to find the gradient of the function whose value at x is $f((g(3x))^2)$:

```
In[39]:= gradient[f[g[3 x]^2], x]
```

```
Out[39]= 6 g[3 x] gradient[g][3 x] f'[g[3 x]^2]
```

Here `gradient[g][3x]` denotes the gradient of g , evaluated at $3x$. For the example above to make sense, we (and the computer) must think of g as a real-valued function on some Euclidean space (where x lives) and f as a function from \mathbf{R} to \mathbf{R} . Let's find the Laplacian of the same function:

```
In[40]:= Δx[f[g[3 x]^2]]
```

```
Out[40]= 18 (||gradient[g][3 x]||^2 + g[3 x] Δ[g][3 x]) f'[g[3 x]^2] +
36 g[3 x]^2 ||gradient[g][3 x]||^2 f''[g[3 x]^2]
```

$\Delta[g][3x]$ denotes, of course, the Laplacian of g , evaluated at $3x$.

All our differentiation commands (**partial**, **Δ**, **gradient**, **normalD**, **divergence**, **Jacobian**) and **homogeneous** and **taylor** can be used with non-explicit functions. For example, if working in \mathbf{R}^2 , to find the term of degree 3 in the homogeneous expansion of $f((g(3x))^2)$ about the origin, enter the commands shown below. In the output below you will see terms such as `partial{1,1,2}[g][0]`, which denotes the partial derivative of g , twice with respect to the first variable and once with respect to the second variable, evaluated at 0:

```
In[41]:= setDimension[x, 2]
```

setDimension::Vector :

x will be considered to be a vector in 2-dimensional real Euclidean space.

```
In[42]:= homogeneous[f[g[3 x]^2], 3, x]
```

```
Out[42]= 9 x1^3 (6 g[0] f''[g[0]^2] partial_{1}[g][0]^3 + 4 g[0]^3 f^{(3)}[g[0]^2] partial_{1}[g][0]^3 +
3 f'[g[0]^2] partial_{1}[g][0] partial_{1,1}[g][0] + 6 g[0]^2 f''[g[0]^2]
partial_{1}[g][0] partial_{1,1}[g][0] + g[0] f'[g[0]^2] partial_{1,1,1}[g][0]) +
27 x1^2 x2 (6 g[0] f''[g[0]^2] partial_{1}[g][0]^2 partial_{2}[g][0] +
4 g[0]^3 f^{(3)}[g[0]^2] partial_{1}[g][0]^2 partial_{2}[g][0] +
f'[g[0]^2] partial_{2}[g][0] partial_{1,1}[g][0] +
2 g[0]^2 f''[g[0]^2] partial_{2}[g][0] partial_{1,1}[g][0] +
2 f'[g[0]^2] partial_{1}[g][0] partial_{1,2}[g][0] + 4 g[0]^2 f''[g[0]^2]
partial_{1}[g][0] partial_{1,2}[g][0] + g[0] f'[g[0]^2] partial_{1,1,2}[g][0]) +
27 x1 x2^2 (6 g[0] f''[g[0]^2] partial_{1}[g][0] partial_{2}[g][0]^2 +
4 g[0]^3 f^{(3)}[g[0]^2] partial_{1}[g][0] partial_{2}[g][0]^2 +
2 f'[g[0]^2] partial_{2}[g][0] partial_{1,2}[g][0] +
4 g[0]^2 f''[g[0]^2] partial_{2}[g][0] partial_{1,2}[g][0] +
f'[g[0]^2] partial_{1}[g][0] partial_{2,2}[g][0] + 2 g[0]^2 f''[g[0]^2]
partial_{1}[g][0] partial_{2,2}[g][0] + g[0] f'[g[0]^2] partial_{1,2,2}[g][0]) +
9 x2^3 (6 g[0] f''[g[0]^2] partial_{2}[g][0]^3 + 4 g[0]^3 f^{(3)}[g[0]^2] partial_{2}[g][0]^3 +
3 f'[g[0]^2] partial_{2}[g][0] partial_{2,2}[g][0] + 6 g[0]^2 f''[g[0]^2]
partial_{2}[g][0] partial_{2,2}[g][0] + g[0] f'[g[0]^2] partial_{2,2,2}[g][0])
```

Vector-valued functions

Suppose we compute the partial derivative with respect to x_1 of the function that takes x to $f(g(h(x)^2))$:

```
In[43]:= partial[f[g[h[x]^2]], x1]
```

```
Out[43]= 2 h[x] f'[g[h[x]^2]] g'[h[x]^2] partial_{1}[h][x]
```

The example above illustrates a general principle: This package assumes that all functions are real valued, unless told otherwise. Thus the last output is correct if we are thinking of g as a real-valued

function. If we want to think of g as taking values in \mathbf{R}^n , we must first enter the command `setDimension[g[_], n]`, which instructs the computer that g is a function with range in \mathbf{R}^n :

In[44]:=

```
setDimension[g[_], n]
```

```
setDimension::VectorValuedFunction :
```

```
g will be considered to be a function taking values in n-dimensional Euclidean space.
```

In[45]:=

```
partial[f[g[h[x]^2]], x1]
```

Out[45]=

```
2 gradient[f][g[h[x]^2]] . (h[x] g'[h[x]^2] partial_{(1)}[h][x])
```

Note that the last output involves the dot product of two vectors: the gradient of f , evaluated at $g(h(x)^2)$, and the derivative of g , evaluated at $h(x)^2$. The last two outputs should be compared.

volume

`volume[n]` is the volume (unnormalized) of the unit ball in \mathbf{R}^n :

In[46]:=

```
volume[4]
```

Out[46]=

$$\frac{\pi^2}{2}$$

`volume` is computed using the formula given by Exercise 6 in Appendix A of [ABR].

surfaceArea

`surfaceArea[n]` is the surface area (unnormalized) of the unit sphere in \mathbf{R}^n :

```
In[47]:= surfaceArea[57]
```

```
Out[47]= 
$$\frac{536\,870\,912\,\pi^{28}}{8\,687\,364\,368\,561\,751\,199\,826\,958\,100\,282\,265\,625}$$

```

surfaceArea [**n**] is computed by multiplying the formula for **volume** [**n**] by n ; see A.2 in Appendix A of [ABR].

integrateSphere

integrateSphere [**f**, **x**] equals the integral of f , with respect to normalized surface area measure, over the unit sphere in the Euclidean space defined by x . Here f should be a polynomial function of x :

```
In[48]:= setDimension[x, n]
```

setDimension::Vector :

x will be considered to be a vector in n -dimensional real Euclidean space.

```
In[49]:= integrateSphere[x1^2 x2^4 x3^6, x]
```

```
Out[49]= 
$$\frac{45}{n(2+n)(4+n)(6+n)(8+n)(10+n)}$$

```

integrateSphere is computed by using the results in Section 3 of Hermann Weyl's paper [W].

integrateBall

integrateBall [**f**, **x**] equals the integral of f , with respect to (unnormalized) volume measure, over the unit ball in the Euclidean space defined by x . Here f should be a function of x and $\|x\|$ that is a polynomial in x and a function in $\|x\|$ for which *Mathematica* can find an explicit antiderivative:

```
In[50]:= setDimension[x, 7]
```

setDimension::Vector :

x will be considered to be a vector in 7-dimensional real Euclidean space.

```
In[51]:= integrateBall[x1^2 x2^4 / (1 + ||x||), x]
```

```
Out[51]= 
$$\frac{16 \pi^3 \left( -\frac{18107}{27720} + \text{Log}[2] \right)}{3465}$$

```

The following expression finds the norm of $x_1 x_2^4$ in the space $L^2(B, (1 - \|x\|^2) dV)$, where B is the unit ball in \mathbf{R}^7 and dV is (unnormalized) volume measure on B .

```
In[52]:= 
$$\sqrt{\text{integrateBall}[(x_1 x_2^4)^2 (1 - \|x\|^2), x]}$$

```

```
Out[52]= 
$$\frac{4}{3} \sqrt{\frac{2}{692835}} \pi^{3/2}$$

```

integrateBall is computed by converting to polar coordinates and then using the function **integrateSphere** (see [R], Chapter 8, Exercise 6).

Boundary Value Problems

Dirichlet problems

The Dirichlet problem on the sphere

dirichlet[p , x] is the solution to the standard Dirichlet problem: find the harmonic function on the unit ball in the Euclidean space defined by x that equals p on the unit sphere. Thus **dirichlet**[p , x] is the Poisson integral of p as a function of x . Here p must be a polynomial function of x :

```
In[53]:= setDimension[x, 5]
```

```
setDimension::Vector :
```

```
x will be considered to be a vector in 5-dimensional real Euclidean space.
```

```
In[54]:= dirichlet[x14 x22, x]
```

```
Out[54]= 
$$\frac{1}{15015} \left( 143 - 273 \|x\|^2 + 165 \|x\|^4 - 35 \|x\|^6 + 910 x_1^2 - \right.$$


$$1540 \|x\|^2 x_1^2 + 630 \|x\|^4 x_1^2 + 1155 x_1^4 - 1155 \|x\|^2 x_1^4 + 455 x_2^2 -$$


$$\left. 770 \|x\|^2 x_2^2 + 315 \|x\|^4 x_2^2 + 6930 x_1^2 x_2^2 - 6930 \|x\|^2 x_1^2 x_2^2 + 15015 x_1^4 x_2^2 \right)$$

```

The last output is a harmonic function on \mathbf{R}^5 that equals $x_1^4 x_2^2$ on the unit sphere. To check this, first we take the Laplacian of the last output, and then we evaluate the last output on the unit sphere:

```
In[55]:= Δx [%]
```

```
Out[55]= 0
```

```
In[56]:= %% /. ||x|| → 1
```

```
Out[56]= x14 x22
```

Thus we have indeed solved the specified Dirichlet problem.

The solution to the Dirichlet problem on the sphere is computed by using the algorithm described in [AR].

The Dirichlet problem on the exterior of the sphere

The **region** option allows the user to solve Dirichlet problems on regions other than the sphere. Currently supported values for **region** are **Sphere**, **exteriorSphere**, **annulus**, and **quadratic**. The default value of **region** is **Sphere**; thus **dirichlet[p, x, region → Sphere]** is the same as **dirichlet[p, x]**.

dirichlet[p, x, region → exteriorSphere] gives the solution to the standard exterior Dirichlet problem: find the harmonic function on the exterior (including ∞) of the unit ball in

the Euclidean space defined by x that equals p on the unit sphere. This function is the exterior Poisson integral of p as a function of x (see Theorem 4.11 of [ABR]). Here p must be a polynomial function of x :

```
In[57]:= dirichlet[x1^4 x2^2, x, region -> exteriorSphere]
```

```
Out[57]= 
$$\frac{1}{15015 \|x\|^{15}} \left( -35 \|x\|^6 + 165 \|x\|^8 - 273 \|x\|^{10} + 143 \|x\|^{12} + 630 \|x\|^4 x_1^2 - \right.$$


$$1540 \|x\|^6 x_1^2 + 910 \|x\|^8 x_1^2 - 1155 \|x\|^2 x_1^4 + 1155 \|x\|^4 x_1^4 + 315 \|x\|^4 x_2^2 -$$


$$\left. 770 \|x\|^6 x_2^2 + 455 \|x\|^8 x_2^2 - 6930 \|x\|^2 x_1^2 x_2^2 + 6930 \|x\|^4 x_1^2 x_2^2 + 15015 x_1^4 x_2^2 \right)$$

```

The last output is a harmonic function on the exterior of the unit ball in \mathbf{R}^5 (because the dimension of x was set to 5 earlier) that equals $x_1^4 x_2^2$ on the unit sphere.

The solution to the Dirichlet problem on the exterior of the sphere is computed by taking the Kelvin transform of the solution that would have been obtained without the **region** \rightarrow **exteriorSphere** option; see Chapter 4 of [ABR].

The Dirichlet problem on annular regions

dirichlet[{**p**, **q**}, **x**, **region** \rightarrow **annulus**[**r**, **s**]] gives the solution to the annular Dirichlet problem: find the harmonic function on the annular region with inner radius r and outer radius s that equals p on the sphere of radius r and equals q on the sphere of radius s . Here p and q must be polynomial functions of x :

```
In[58]:= dirichlet[{x1^3, x3^2}, x, region -> annulus[1, 4]]
```

```
Out[58]= 
$$-\frac{1024}{315} \left( -1 + \frac{1}{\|x\|^3} \right) + \frac{1024 \left( -\frac{1}{1024} + \frac{1}{\|x\|^5} \right) x_1}{2387} +$$


$$\frac{(-262144 + \|x\|^9) (3 \|x\|^2 x_1 - 7 x_1^3)}{1835001 \|x\|^9} - \frac{16384 \left( -1 + \frac{1}{\|x\|^7} \right) \left( -\frac{\|x\|^2}{5} + x_3^2 \right)}{16383}$$

```

Now we check that the last output is harmonic on \mathbf{R}^5 (because the dimension of x was set to 5 earlier), that it equals x_1^3 on the sphere of radius 1, and that it equals x_3^2 on the sphere of radius 4:

```
In[59]:= Δx [%]
```

```
Out[59]= 0
```

```
In[60]:= %% /. ||x|| → 1
```

```
Out[60]= x13
```

```
In[61]:= %%% /. ||x|| → 4
```

```
Out[61]= x32
```

If $p = q$, then the shortcut `dirichlet[p, x, region → annulus[r, s]]` can be used instead of `dirichlet[{p, q}, x, region → annulus[r, s]]`.

The solution to the Dirichlet problem on an annulus is computed by using the techniques described in Chapter 10 of [ABR].

The Dirichlet problem on quadratic surfaces

`dirichlet[p, x, region → quadratic[b, c, d]]` gives the harmonic polynomial that equals p on the quadratic surface

$$b.x^2 + c.x + d = 0.$$

Here b and c must be lists whose length equals the dimension of x , and p must be a polynomial function of x . For example, here we find the harmonic polynomial that equals $x_1^3 x_3^2$ on the elliptic paraboloid given by the equation

$$2x_1^2 + 3x_2^2 = 4x_3 + 1$$

in \mathbf{R}^3 :

```
In[62]:= setDimension[x, 3]
```

```
setDimension::Vector :
```

```
x will be considered to be a vector in 3-dimensional real Euclidean space.
```

```
In[63]:= dirichlet[x1^3 x3^2, x, region -> quadratic[{2, 3, 0}, {0, 0, -4}, -1]]
```

```
Out[63]= x1^3 x3^2 + 1/21627 (-1 + 2 x1^2 + 3 x2^2 - 4 x3) (-5123 x1 - 405 x1^3 + 1053 x1 x2^2 - 9684 x1 x3 - 7209 x1 x3^2)
```

The last output is of the form $x_1^3 x_3^2 + (2x_1^2 + 3x_2^2 - 4x_3 - 1)f$, where f is a polynomial. Thus the last output obviously equals $x_1^3 x_3^2$ on the elliptic paraboloid given by the equation $2x_1^2 + 3x_2^2 = 4x_3 + 1$. Thus to verify that the last output is correct, we only need check that it is harmonic:

```
In[64]:= Δx [%]
```

```
Out[64]= 0
```

The solution to the Dirichlet problem on a quadratic surface is computed by using the algorithm developed in [AGV]. That paper also shows that if each coordinate of b is nonnegative and at least one coordinate of b is nonzero, and if the quadratic surface in question is nontrivial in the sense that there exists $x \in \mathbf{R}^n$ such that

$$b \cdot x^2 + c \cdot x + d < 0,$$

then the polynomial produced by this algorithm is the unique harmonic polynomial that equals p on the quadratic surface in question.

Using the notation above, if c consists of all 0's, then it may be omitted. For example, here we find the harmonic polynomial that equals $x_1^4 x_3^2$ on the ellipsoid given by the equation $2x_1^2 + 3x_2^2 + 4x_3^2 = 1$:

```
In[65]:= dirichlet[x1^4 x3^2, x, region -> quadratic[{2, 3, 4}, -1]]
```

```
Out[65]= x1^4 x3^2 + 1/39309793479 (-1 + 2 x1^2 + 3 x2^2 + 4 x3^2) (-52529359 - 511000770 x1^2 - 801788415 x1^4 + 102950118 x2^2 + 945182934 x1^2 x2^2 - 67605435 x2^4 - 64713579 x3^2 - 5766993996 x1^2 x3^2 + 204109461 x2^2 x3^2 + 342331356 x3^4)
```

Generalized Dirichlet problems

The generalized Dirichlet problem asks for a function with specified boundary values on some surface and with specified Laplacian inside that surface. Thus if the Laplacian is required to be 0, then this is

the same as the usual Dirichlet problem. The generalized Dirichlet problem with specified Laplacian equal to q is solved by adding the option $\Delta \rightarrow \mathbf{q}$ to the **dirichlet** function; here q must be a polynomial. For example, we can find the function that equals $x_1^3 x_2^2$ on the unit sphere in \mathbf{R}^3 and whose Laplacian equals $x_2^2 x_3$:

```
In[66]:= dirichlet[x1^3 x2^2, x, Δ → x2^2 x3]
```

```
Out[66]= 
$$\frac{1}{1260} (108 x_1 - 168 \|x\|^2 x_1 + 60 \|x\|^4 x_1 + 140 x_1^3 - 140 \|x\|^2 x_1^3 + 420 x_1 x_2^2 - 420 \|x\|^2 x_1 x_2^2 + 1260 x_1^3 x_2^2 - 9 x_3 + 14 \|x\|^2 x_3 - 5 \|x\|^4 x_3 - 70 x_2^2 x_3 + 70 \|x\|^2 x_2^2 x_3)$$

```

We can check that output above correctly solves the generalized Dirichlet problem by verifying that the Laplacian of the last output indeed equals $x_2^2 x_3$ and that the last output restricted to the unit sphere indeed equals $x_1^3 x_2^2$:

```
In[67]:= Δx [%]
```

```
Out[67]= x2^2 x3
```

```
In[68]:= %% /. ||x|| → 1
```

```
Out[68]= x1^3 x2^2
```

The Δ option can be used combined with the **region** option to solve the generalized Dirichlet problem on any region for which this package can solve the usual Dirichlet problem. The order of the options does not matter. Here we find the function on \mathbf{R}^3 whose Laplacian equals x_2^2 and that equals $x_1^4 x_3^2$ on the ellipsoid $2 x_1^2 + 3 x_2^2 + 4 x_3^2 - 1 = 0$:

```
In[69]:= dirichlet[x1^4 x3^2, x, Δ → x2^2, region → quadratic[{2, 3, 4}, -1]]
```

```
Out[69]= 
$$x_1^4 x_3^2 + \frac{1}{157239173916} (-1 + 2 x_1^2 + 3 x_2^2 + 4 x_3^2) (82719889 - 2354066130 x_1^2 - 3207153660 x_1^4 + 3779085195 x_2^2 + 3780731736 x_1^2 x_2^2 - 270421740 x_2^4 - 680540064 x_3^2 - 23067975984 x_1^2 x_3^2 + 816437844 x_2^2 x_3^2 + 1369325424 x_3^4)$$

```

The form of the last output shows, without calculation, that it obviously equals $x_1^4 x_3^2$ on the ellipsoid $2x_1^2 + 3x_2^2 + 4x_3^2 - 1 = 0$. Thus to verify that we have correctly solved this generalized Dirichlet problem, we only need to check that the Laplacian equals x_2^2 :

```
In[70]:= Δx [%]
```

```
Out[70]= x22
```

The generalized Dirichlet problem on some region requires finding a function u that equals p on the boundary of the region and such that $\Delta u = q$. This problem is solved as follows: First find any function v such that $\Delta v = q$ (see the **antiLaplacian** section later in this document for the computation of this step). Then solve the usual Dirichlet problem for the specified region with boundary function $p - v$, getting a harmonic function w such that w equals $p - v$ on the boundary of the region. Now set $u = w + v$, getting the desired function u that equals p on the boundary of the region and such that $\Delta u = q$.

Dirichlet problems with explicit coordinates

So far we have worked with symbolic vectors, which we have usually called x or z . Vectors can also be described by giving explicit coordinates in the form of a list. Both formats (symbolic vectors and explicit lists) work with all functions in this package. For example, suppose we are working in \mathbf{R}^3 and want to use x , y , and z instead of x_1 , x_2 , and x_3 . Here is how to find the Poisson integral of $x^3 y z^2$:

```
In[71]:= dirichlet[x3 y z2, {x, y, z}]
```

```
Out[71]= 
$$\frac{1}{231} (11 x y + 3 x^3 y - 14 x^5 y - 18 x y^3 - 7 x^3 y^3 + 7 x y^5 + 45 x y z^2 + 161 x^3 y z^2 - 49 x y^3 z^2 - 56 x y z^4)$$

```

To check that the last output is correct, first we verify that it is harmonic by checking that its Laplacian equals 0:

```
In[72]:= Δ{x,y,z} [%]
```

```
Out[72]= 0
```

Now we check that the difference between the claimed solution to the Dirichlet problem equals $x^3 y z^2$ on the unit sphere by verifying that the difference between these two functions, when restricted to the unit sphere, equals 0:

```
In[73]:= x^3 y z^2 - %% /. z -> Sqrt[1 - x^2 - y^2]
```

```
Out[73]= 0
```

Here is another example using explicit coordinates. In this example, we find the function on \mathbf{R}^3 whose Laplacian equals $y^2 z$ and that equals $x^3 y z^2$ on the ellipsoid $2x^2 + 3y^2 + 4z^2 - 1 = 0$:

```
In[74]:= Dirichlet[x^3 y z^2, {x, y, z}, Region -> Quadratic[{2, 3, 4}, -1], Delta -> y^2 z^3]
```

```
Out[74]= x^3 y z^2 + (-1 + 2 x^2 + 3 y^2 + 4 z^2)
  ( - 961 x y / 150 917 - 223 x^3 y / 7943 + 93 x y^3 / 7943 + 6 128 862 523 z / 154 091 469 268 302 - 424 717 558 x^2 z / 4 532 102 037 303 +
    44 102 x^4 z / 734 657 487 + 3 416 529 117 y^2 z / 6 042 802 716 404 - 448 445 x^2 y^2 z / 489 771 658 - 1 320 573 y^4 z / 979 543 316 -
    571 x y z^2 / 7943 + 1 235 669 309 z^3 / 18 128 408 149 212 - 213 085 x^2 z^3 / 1 469 314 974 + 8 956 053 y^2 z^3 / 979 543 316 - 297 101 z^5 / 734 657 487 )
```

Obviously the last output equals $x^3 y z^2$ on the ellipsoid $2x^2 + 3y^2 + 4z^2 - 1 = 0$. We now check that its Laplacian is the desired function:

```
In[75]:= Delta[{x, y, z}][%]
```

```
Out[75]= y^2 z^3
```

harmonicDecomposition

harmonicDecomposition[p , \mathbf{x}] gives the decomposition of p into a sum of harmonic polynomials (on the Euclidean space defined by x) times even powers of $\|x\|$. Here p must be a polynomial function of x :

```
In[76]:= setDimension[x, n]
```

setDimension::Vector :

x will be considered to be a vector in n-dimensional real Euclidean space.

```
In[77]:= harmonicDecomposition[x4, x]
```

```
Out[77]= {{ {  $\frac{3 \|x\|^4}{(2+n)(4+n)} - \frac{6 \|x\|^2 x_1^2}{4+n} + x_1^4, 0$  }, {  $\frac{6(-\|x\|^2 + n x_1^2)}{n(4+n)}, 2$  }, {  $\frac{3}{n(2+n)}, 4$  } }
```

The output given by **harmonicDecomposition**[**p**, **x**] consists of a list of pairs. The first entry in each pair is a harmonic function, the second entry is the power of $\|x\|$ by which the first entry should be multiplied so that the sum of the resulting terms equals p . For example, in the last output $\frac{3\|x\|^4}{(2+n)(4+n)} - \frac{6\|x\|^2 x_1^2}{4+n} + x_1^4$ and $\frac{6(-\|x\|^2 + n x_1^2)}{n(4+n)}$ and $\frac{3}{n(2+n)}$ are harmonic functions of x ; furthermore

$$x_1^4 = \frac{3\|x\|^4}{(2+n)(4+n)} - \frac{6\|x\|^2 x_1^2}{4+n} + x_1^4 + \frac{6(-\|x\|^2 + n x_1^2)}{n(4+n)} \|x\|^2 + \frac{3}{n(2+n)} \|x\|^4.$$

For a proof of the existence and uniqueness of the harmonic decomposition, see [ABR], Theorem 5.7. **harmonicDecomposition** is computed using the algorithm described in Section 2 of [AR].

antiLaplacian

antiLaplacian[**u**, **x**] gives an anti-Laplacian of u with respect to x ; this is a function whose Laplacian with respect to x equals u . Here u must be a polynomial function of x or a sum of terms, each of which is a polynomial in x times a function of $\|x\|$:

```
In[78]:= setDimension[x, 5]
```

setDimension::Vector :

x will be considered to be a vector in 5-dimensional real Euclidean space.

```
In[79]:= antiLaplacian[x1^2 x2 ||x||^3 Log[||x||], x]
```

```
Out[79]=
```

$$\frac{(-19 ||x||^7 + 84 \text{Log}[||x||] ||x||^7) x_2}{49392} - \frac{1}{34300} \frac{(-19 ||x||^5 + 70 \text{Log}[||x||] ||x||^5) (||x||^2 x_2 - 7 x_1^2 x_2)}{}$$

The function given by the last output has Laplacian equal to $x_1^2 x_2 ||x||^3 \log(||x||)$ on \mathbf{R}^5 , as we can easily verify:

```
In[80]:= Δx [%]
```

```
Out[80]=
```

$$\text{Log}[||x||] ||x||^3 x_1^2 x_2$$

Note that finding the anti-Laplacian of even a fairly simple function such as $x_1^2 x_2 ||x||^3 \log(||x||)$ led to a solution containing five-digit integers.

Here is another interesting example and its verification. Note in the example below that the anti-Laplacian is unbounded near 0, even though $x_1 x_2 \cos(||x||)$ is bounded near 0. However, this behavior is not surprising because $\cos(||x||)$ is not smooth at 0:

```
In[81]:= antiLaplacian[x1 x2 Cos[||x||], x]
```

```
Out[81]=
```

$$\frac{1}{||x||^7} (-\text{Cos}[||x||] ||x|| (-5760 + 960 ||x||^2 - 48 ||x||^4 + ||x||^6) + 8 (-720 + 360 ||x||^2 - 30 ||x||^4 + ||x||^6) \text{Sin}[||x||]) x_1 x_2$$

```
In[82]:= Δx [%]
```

```
Out[82]=
```

$$\text{Cos}[||x||] x_1 x_2$$

To understand how **antiLaplacian** is computed, suppose we want to find an anti-Laplacian of $p(x)f(||x||)$, where p is a polynomial on \mathbf{R}^n and f is a continuous function on $(0, \infty)$. Applying **harmonicDecomposition** to p , using linearity, and replacing $f(||x||)$ with an even power of $||x||$ times $f(||x||)$, we see that we only need to find anti-Laplacians of functions of the form $q(x)f(||x||)$,

where q is a harmonic polynomial on \mathbf{R}^n homogeneous of degree m . To find an anti-Laplacian of $q(x)f(\|x\|)$, suppose h is a twice-differentiable function on $(0, \infty)$. The Laplacian of $q(x)h(\|x\|)$ equals

$$q(x) \frac{(2m+n-1)h'(\|x\|) + \|x\| h''(\|x\|)}{\|x\|},$$

where we have used the product rule for Laplacians (see 1.19 in [ABR]) and Exercise 29, Chapter 1 of [ABR]. We want to make the expression above equal $q(x)f(\|x\|)$, so we need only solve the differential equation

$$\frac{(2m+n-1)h'(t) + t h''(t)}{t} = f(t).$$

A solution to this equation is given by

$$h(t) = \int_a^t s^{1-2m-n} \int_b^s r^{2m+n-1} f(r) dr ds$$

for any choice of the constants a and b . **antiLaplacian** uses the formula above with $b = 0$, although if a singularity is encountered at 0 it automatically makes a different choice. If you know there is a singularity at 0, you can save some time by using the option **singularity** \rightarrow **0**, as illustrated by the following example. The two timing tests below show a large speed-up when using the option **singularity** \rightarrow **0**:

```
In[83]:= Timing[antiLaplacian[x1^2 x2 Log[||x||]^10, x]]
```

```
Out[83]= {4.0625,  $\frac{1}{44\,432\,481\,042\,432}$ 
(1 098 190 285 675 ||x||4 - 4 392 027 139 500 Log[||x||] ||x||4 + 8 780 751 264 600 Log[||x||]2
||x||4 - 11 697 759 309 600 Log[||x||]3 ||x||4 + 11 675 463 962 400 Log[||x||]4 ||x||4 -
9 300 239 544 960 Log[||x||]5 ||x||4 + 6 139 962 259 200 Log[||x||]6 ||x||4 -
3 431 153 157 120 Log[||x||]7 ||x||4 + 1 628 505 285 120 Log[||x||]8 ||x||4 -
636 708 833 280 Log[||x||]9 ||x||4 + 176 319 369 216 Log[||x||]10 ||x||4) x2 -
 $\frac{1}{15\,977\,453\,554\,216}$  (449 365 877 986 725 ||x||2 - 898 731 726 943 050 Log[||x||] ||x||2 +
898 731 567 275 850 Log[||x||]2 ||x||2 - 599 153 792 737 500 Log[||x||]3 ||x||2 +
299 575 286 391 150 Log[||x||]4 ||x||2 - 119 826 572 605 740 Log[||x||]5 ||x||2 +
39 935 697 292 260 Log[||x||]6 ||x||2 - 11 399 995 035 000 Log[||x||]7 ||x||2 +
2 835 967 995 630 Log[||x||]8 ||x||2 - 613 066 399 660 Log[||x||]9 ||x||2 +
103 749 698 404 Log[||x||]10 ||x||2) (||x||2 x2 - 7 x12 x2)}
```

```
In[84]:= Timing[antiLaplacian[x1^2 x2 Log[||x||]^10, x, singularity -> 0]]
```

```
Out[84]= {0.03125,  $\frac{1}{44\,432\,481\,042\,432}$ 
(1\,098\,190\,285\,675 ||x||^4 - 4\,392\,027\,139\,500 Log[||x||] ||x||^4 + 8\,780\,751\,264\,600 Log[||x||]^2
||x||^4 - 11\,697\,759\,309\,600 Log[||x||]^3 ||x||^4 + 11\,675\,463\,962\,400 Log[||x||]^4 ||x||^4 -
9\,300\,239\,544\,960 Log[||x||]^5 ||x||^4 + 6\,139\,962\,259\,200 Log[||x||]^6 ||x||^4 -
3\,431\,153\,157\,120 Log[||x||]^7 ||x||^4 + 1\,628\,505\,285\,120 Log[||x||]^8 ||x||^4 -
636\,708\,833\,280 Log[||x||]^9 ||x||^4 + 176\,319\,369\,216 Log[||x||]^10 ||x||^4) x2 -
 $\frac{1}{15\,977\,453\,554\,216}$  (449\,365\,877\,986\,725 ||x||^2 - 898\,731\,726\,943\,050 Log[||x||] ||x||^2 +
898\,731\,567\,275\,850 Log[||x||]^2 ||x||^2 - 599\,153\,792\,737\,500 Log[||x||]^3 ||x||^2 +
299\,575\,286\,391\,150 Log[||x||]^4 ||x||^2 - 119\,826\,572\,605\,740 Log[||x||]^5 ||x||^2 +
39\,935\,697\,292\,260 Log[||x||]^6 ||x||^2 - 11\,399\,995\,035\,000 Log[||x||]^7 ||x||^2 +
2\,835\,967\,995\,630 Log[||x||]^8 ||x||^2 - 613\,066\,399\,660 Log[||x||]^9 ||x||^2 +
103\,749\,698\,404 Log[||x||]^10 ||x||^2) (||x||^2 x2 - 7 x1^2 x2)}
```

In the case where \mathbf{p} is a polynomial (so that there are no terms such as $\mathbf{Log}[\|\mathbf{x}\|]$ or other functions of $\|\mathbf{x}\|$ as in the examples above), then $\mathbf{antiLaplacian}[\mathbf{p}, \mathbf{x}]$ is not computed using the procedure described above. Instead, the anti-Laplacian is computed using an iterative procedure that at each step reduces the degree of some of the coordinates and increases the degree of other coordinates until eventually the problem reduces to finding an anti-Laplacian of an expression of the form x_k^m or $x_j x_k^m$, both of which are easily computed. This procedure always produces a polynomial for the anti-Laplacian. Here is an example:

```
In[85]:= antiLaplacian[x1^2 x2^5, x]
```

```
Out[85]=  $\frac{1}{42} x_1^2 x_2^7 - \frac{x_2^9}{1512}$ 
```

The anti-Laplacian of a given function is never unique. However, each polynomial has a unique anti-Laplacian that is a polynomial multiple of $\|\mathbf{x}\|^2$, where x is the variable. The option **multiple** \rightarrow **norm²** produces the unique anti-Laplacian that is a polynomial multiple of $\|\mathbf{x}\|^2$:

```
In[86]:= antiLaplacian[x1^2 x2^5, x, multiple -> norm^2]
```

Out[86]=
$$\frac{1}{302328} (-9 \|x\|^8 x_2 + 156 \|x\|^6 x_1^2 x_2 + 104 \|x\|^6 x_2^3 - 2340 \|x\|^4 x_1^2 x_2^3 - 234 \|x\|^4 x_2^5 + 7956 \|x\|^2 x_1^2 x_2^5)$$

Future versions of this software will allow addition options for finding an anti-Laplacian than is a multiple of some functions other than $\|x\|^2$.

neumann

neumann [**p**, **x**] is the solution to the standard Neumann problem: find the harmonic function on the unit ball in the Euclidean space defined by x whose outward normal derivative on the unit sphere equals p and whose value at the origin equals 0. Here p must be a polynomial function of x . Furthermore, the integral of p over the unit sphere with respect to surface area measure must equal 0 (Green's identity shows that this condition is necessary for the existence of a solution to the standard Neumann problem):

In[87]:= `setDimension[x, 4]`

setDimension::Vector :

x will be considered to be a vector in 4-dimensional real Euclidean space.

In[88]:= `neumann[x1^6 x2, x]`

Out[88]=
$$\frac{x_2}{32} + \frac{1}{48} (-\|x\|^2 x_2 + 6 x_1^2 x_2) + \frac{3 (3 \|x\|^4 x_2 - 48 \|x\|^2 x_1^2 x_2 + 80 x_1^4 x_2)}{1120} + \frac{1}{784} (-\|x\|^6 x_2 + 30 \|x\|^4 x_1^2 x_2 - 120 \|x\|^2 x_1^4 x_2 + 112 x_1^6 x_2)$$

In[89]:= `Δ_x [%]`

Out[89]= 0

```
In[90]:= normalD[%%, x]
```

```
Out[90]:= x16 x2
```

`neumann [p, x]` is computed by using the algorithm discussed in [AR].

exteriorNeumann

`exteriorNeumann [p, x]` is the solution to the standard exterior Neumann problem: find the harmonic function on the exterior of the unit ball in the Euclidean space defined by x whose outward normal derivative (with respect to the exterior of the unit ball) on the unit sphere equals p and whose limit at ∞ equals 0. Here p must be a polynomial function of x . If the dimension of the Euclidean space equals 2, then the integral of p over the unit circle with respect to arc length measure must equal 0:

```
In[91]:= setDimension[x, 5]
```

```
setDimension::Vector :
```

```
x will be considered to be a vector in 5-dimensional real Euclidean space.
```

```
In[92]:= exteriorNeumann[x16 x2, x]
```

```
Out[92]= 
$$\frac{5 x_2}{924 \|x\|^5} - \frac{15 (\|x\|^2 x_2 - 7 x_1^2 x_2)}{2002 \|x\|^9} + \frac{\|x\|^4 x_2 - 18 \|x\|^2 x_1^2 x_2 + 33 x_1^4 x_2}{264 \|x\|^{13}} +$$


$$\frac{-\frac{1}{143} \|x\|^6 x_2 + \frac{3}{13} \|x\|^4 x_1^2 x_2 - \|x\|^2 x_1^4 x_2 + x_1^6 x_2}{10 \|x\|^{17}}$$

```

```
In[93]:= Δx [%]
```

```
Out[93]= 0
```

```
In[94]:= normalD[%, x]
```

```
Out[94]= -x1^6 x2
```

Note that the normal derivative above is the negative of our desired function. This result is correct because the function `normalD` takes the outward normal with respect to the unit ball, but for the exterior Neumann problem the outward normal points in the opposite direction.

biDirichlet

`biDirichlet[p, x]` is the solution to the standard biDirichlet problem: find the biharmonic function on the unit ball in the Euclidean space defined by x that equals p on the unit sphere and whose normal derivative on the unit sphere equals 0. (A function is called biharmonic if the Laplacian of its Laplacian equals 0.) Here p must be a polynomial function of x :

```
In[95]:= setDimension[x, 3]
```

```
setDimension::Vector :
```

```
x will be considered to be a vector in 3-dimensional real Euclidean space.
```

```
In[96]:= biDirichlet[x1^4 x2^3, x]
```

```
Out[96]= 
$$\frac{1}{30030} (1287 x_2 - 4524 \|x\|^2 x_2 + 5922 \|x\|^4 x_2 - 3420 \|x\|^6 x_2 + 735 \|x\|^8 x_2 + 13650 x_1^2 x_2 - 40530 \|x\|^2 x_1^2 x_2 + 40110 \|x\|^4 x_1^2 x_2 - 13230 \|x\|^6 x_1^2 x_2 + 24255 x_1^4 x_2 - 48510 \|x\|^2 x_1^4 x_2 + 24255 \|x\|^4 x_1^4 x_2 + 2275 x_2^3 - 6755 \|x\|^2 x_2^3 + 6685 \|x\|^4 x_2^3 - 2205 \|x\|^6 x_2^3 + 48510 x_1^2 x_2^3 - 97020 \|x\|^2 x_1^2 x_2^3 + 48510 \|x\|^4 x_1^2 x_2^3 + 135135 x_1^4 x_2^3 - 105105 \|x\|^2 x_1^4 x_2^3)$$

```

To check that the last output is correct, first we take its Laplacian squared to check that we have a biharmonic function:

```
In[97]:=  $\Delta_x^2$  [%]
```

```
Out[97]= 0
```

Next we check that the outward normal derivative on the unit sphere indeed equals 0:

```
In[98]:= normalD [%%, x]
```

```
Out[98]= 0
```

Finally, we check that we have the correct boundary values:

```
In[99]:= %%% /. ||x||  $\rightarrow$  1
```

```
Out[99]=  $x_1^4 x_2^3$ 
```

biDirichlet[**p**, **x**] is computed by using the algorithm discussed in [AR].

Spherical Harmonics

basisH

basisH[**m**, **x**] gives a vector space basis of $H_m(\mathbf{R}^n)$, the space of homogeneous harmonic polynomials on \mathbf{R}^n of degree m , where n is the dimension of x . The user must first set the dimension of x to a positive integer value. Wrapping the *Mathematica* command **TableForm** around **basisH**[**m**, **x**] will produce an output display that is more readable than the usual list format:

```
In[100]:= setDimension[x, 3]
```

```
setDimension::Vector :
```

```
  x will be considered to be a vector in 3-dimensional real Euclidean space.
```

```
In[101]:= TableForm[basisH[4, x]]
```

```
Out[101]//TableForm=
```

```

3 ||x||4 - 30 ||x||2 x22 + 35 x24
3 ||x||2 x2 x3 - 7 x23 x3
||x||4 - 5 ||x||2 x22 - 5 ||x||2 x32 + 35 x22 x32
3 ||x||2 x2 x3 - 7 x2 x33
3 ||x||4 - 30 ||x||2 x32 + 35 x34
3 ||x||2 x1 x2 - 7 x1 x23
||x||2 x1 x3 - 7 x1 x22 x3
||x||2 x1 x2 - 7 x1 x2 x32
3 ||x||2 x1 x3 - 7 x1 x33

```

An optional third argument for **basisH** will produce a basis of $H_m(\mathbf{R}^n)$ that is orthonormal with respect to the inner product determined by the third argument. Two inner products are already defined for possible use. One of them will be generated by using **Ball** for the third argument, giving the inner product on $L^2(B, dV)$ (here B denotes the open unit ball in Euclidean space and V denotes volume measure on B). Thus **basisH[4, x, Ball]** produces a basis of $H_4(\mathbf{R}^3)$ that is orthonormal with respect to the inner product on $L^2(B, dV)$, where B is the open unit ball in \mathbf{R}^3 (the 3 comes from the dimension of x , which was set to 3 above), as shown below:

```
In[102]:= TableForm[basisH[4, x, Ball]]
```

```
Out[102]//TableForm=
```

$$\begin{aligned} & \frac{3}{16} \sqrt{\frac{11}{\pi}} \left(3 \|x\|^4 - 30 \|x\|^2 x_2^2 + 35 x_2^4 \right) \\ & \frac{3}{4} \sqrt{\frac{55}{2\pi}} \left(3 \|x\|^2 x_2 x_3 - 7 x_2^3 x_3 \right) \\ & \frac{3}{8} \sqrt{\frac{55}{\pi}} \left(\|x\|^4 - 8 \|x\|^2 x_2^2 + 7 x_2^4 - 2 \|x\|^2 x_3^2 + 14 x_2^2 x_3^2 \right) \\ & \frac{3}{4} \sqrt{\frac{385}{2\pi}} \left(3 \|x\|^2 x_2 x_3 - 3 x_2^3 x_3 - 4 x_2 x_3^3 \right) \\ & \frac{3}{16} \sqrt{\frac{385}{\pi}} \left(\|x\|^4 - 2 \|x\|^2 x_2^2 + x_2^4 - 8 \|x\|^2 x_3^2 + 8 x_2^2 x_3^2 + 8 x_3^4 \right) \\ & \frac{3}{4} \sqrt{\frac{55}{2\pi}} \left(3 \|x\|^2 x_1 x_2 - 7 x_1 x_2^3 \right) \\ & \frac{3}{4} \sqrt{\frac{55}{\pi}} \left(\|x\|^2 x_1 x_3 - 7 x_1 x_2^2 x_3 \right) \\ & \frac{3}{4} \sqrt{\frac{385}{2\pi}} \left(\|x\|^2 x_1 x_2 - x_1 x_2^3 - 4 x_1 x_2 x_3^2 \right) \\ & \frac{3}{4} \sqrt{\frac{385}{\pi}} \left(\|x\|^2 x_1 x_3 - x_1 x_2^2 x_3 - 2 x_1 x_3^3 \right) \end{aligned}$$

The other inner product that is already defined is generated by using **Sphere** as the third argument, giving the inner product on $L^2(S, d\sigma)$, where S denotes the unit sphere in Euclidean space and σ denotes normalized surface area measure on S . Thus **basisH[4, x, Sphere]** produces a basis of $H_4(\mathbf{R}^3)$ that is orthonormal with respect to the inner product on $L^2(S, d\sigma)$, where S is the open unit sphere in \mathbf{R}^3 (the 3 comes from the dimension of x , which was set to 3 above), as shown below:

```
In[103]:= TableForm[basisH[4, x, Sphere]]
```

```
Out[103]//TableForm=
```

$$\begin{aligned} & \frac{3}{8} \left(3 \|x\|^4 - 30 \|x\|^2 x_2^2 + 35 x_2^4 \right) \\ & \frac{3}{2} \sqrt{\frac{5}{2}} \left(3 \|x\|^2 x_2 x_3 - 7 x_2^3 x_3 \right) \\ & \frac{3}{4} \sqrt{5} \left(\|x\|^4 - 8 \|x\|^2 x_2^2 + 7 x_2^4 - 2 \|x\|^2 x_3^2 + 14 x_2^2 x_3^2 \right) \\ & \frac{3}{2} \sqrt{\frac{35}{2}} \left(3 \|x\|^2 x_2 x_3 - 3 x_2^3 x_3 - 4 x_2 x_3^3 \right) \\ & \frac{3}{8} \sqrt{35} \left(\|x\|^4 - 2 \|x\|^2 x_2^2 + x_2^4 - 8 \|x\|^2 x_3^2 + 8 x_2^2 x_3^2 + 8 x_3^4 \right) \\ & \frac{3}{2} \sqrt{\frac{5}{2}} \left(3 \|x\|^2 x_1 x_2 - 7 x_1 x_2^3 \right) \\ & \frac{3}{2} \sqrt{5} \left(\|x\|^2 x_1 x_3 - 7 x_1 x_2^2 x_3 \right) \\ & \frac{3}{2} \sqrt{\frac{35}{2}} \left(\|x\|^2 x_1 x_2 - x_1 x_2^3 - 4 x_1 x_2 x_3^2 \right) \\ & \frac{3}{2} \sqrt{35} \left(\|x\|^2 x_1 x_3 - x_1 x_2^2 x_3 - 2 x_1 x_3^3 \right) \end{aligned}$$

More generally, the optional third argument for **basisH** can be any inner product defined on polynomials on \mathbf{R}^n . This inner product should itself be a function of three arguments, giving the inner product of the first two arguments; the third argument specifies the variable. For example, the following can be used to find an orthonormal basis in the weighted space $L^2(B, (1 - \|x\|^2)dV)$.

```
In[104]:= innerProduct[f_, g_, x_] := integrateBall[f g (1 - \|x\|^2), x]
```

```
In[105]:= TableForm[basisH[4, x, innerProduct]]
```

```
Out[105]//TableForm=
```

$$\begin{aligned} & \frac{3}{16} \sqrt{\frac{143}{2\pi}} \left(3 \|x\|^4 - 30 \|x\|^2 x_2^2 + 35 x_2^4 \right) \\ & \frac{3}{8} \sqrt{\frac{715}{\pi}} \left(3 \|x\|^2 x_2 x_3 - 7 x_2^3 x_3 \right) \\ & \frac{3}{8} \sqrt{\frac{715}{2\pi}} \left(\|x\|^4 - 8 \|x\|^2 x_2^2 + 7 x_2^4 - 2 \|x\|^2 x_3^2 + 14 x_2^2 x_3^2 \right) \\ & \frac{3}{8} \sqrt{\frac{5005}{\pi}} \left(3 \|x\|^2 x_2 x_3 - 3 x_2^3 x_3 - 4 x_2 x_3^3 \right) \\ & \frac{3}{16} \sqrt{\frac{5005}{2\pi}} \left(\|x\|^4 - 2 \|x\|^2 x_2^2 + x_2^4 - 8 \|x\|^2 x_3^2 + 8 x_2^2 x_3^2 + 8 x_3^4 \right) \\ & \frac{3}{8} \sqrt{\frac{715}{\pi}} \left(3 \|x\|^2 x_1 x_2 - 7 x_1 x_2^3 \right) \\ & \frac{3}{4} \sqrt{\frac{715}{2\pi}} \left(\|x\|^2 x_1 x_3 - 7 x_1 x_2^2 x_3 \right) \\ & \frac{3}{8} \sqrt{\frac{5005}{\pi}} \left(\|x\|^2 x_1 x_2 - x_1 x_2^3 - 4 x_1 x_2 x_3^2 \right) \\ & \frac{3}{4} \sqrt{\frac{5005}{2\pi}} \left(\|x\|^2 x_1 x_3 - x_1 x_2^2 x_3 - 2 x_1 x_3^3 \right) \end{aligned}$$

To produce an orthonormal basis of $H_m(S)$, the space of spherical harmonics on the unit sphere S with respect to the inner product on $L^2(S, d\sigma)$, use the optional third argument **Sphere** and in the result replace $\|x\|$ with 1:

```
In[106]:= TableForm[basisH[4, x, Sphere]] /. ||x|| -> 1
```

```
Out[106]//TableForm=
```

$$\begin{aligned} & \frac{3}{8} (3 - 30 x_2^2 + 35 x_2^4) \\ & \frac{3}{2} \sqrt{\frac{5}{2}} (3 x_2 x_3 - 7 x_2^3 x_3) \\ & \frac{3}{4} \sqrt{5} (1 - 8 x_2^2 + 7 x_2^4 - 2 x_3^2 + 14 x_2^2 x_3^2) \\ & \frac{3}{2} \sqrt{\frac{35}{2}} (3 x_2 x_3 - 3 x_2^3 x_3 - 4 x_2 x_3^3) \\ & \frac{3}{8} \sqrt{35} (1 - 2 x_2^2 + x_2^4 - 8 x_3^2 + 8 x_2^2 x_3^2 + 8 x_3^4) \\ & \frac{3}{2} \sqrt{\frac{5}{2}} (3 x_1 x_2 - 7 x_1 x_2^3) \\ & \frac{3}{2} \sqrt{5} (x_1 x_3 - 7 x_1 x_2^2 x_3) \\ & \frac{3}{2} \sqrt{\frac{35}{2}} (x_1 x_2 - x_1 x_2^3 - 4 x_1 x_2 x_3^2) \\ & \frac{3}{2} \sqrt{35} (x_1 x_3 - x_1 x_2^2 x_3 - 2 x_1 x_3^3) \end{aligned}$$

basisH is computed by using Theorem 5.25 of [ABR]. If the optional third argument is present in **basisH**, then the Gram-Schmidt procedure is used to produce an orthonormal basis with respect to the inner product determined by the third argument.

zonalHarmonic

zonalHarmonic[*m*, **x**, **y**] is the extended zonal harmonic $Z_m(x, y)$ as defined by 8.7 of [ABR]:

```
In[107]:= setDimension[x, n]
```

```
setDimension::Vector :
```

x will be considered to be a vector in n-dimensional real Euclidean space.

```
In[108]:= zonalHarmonic[5, x, y]
```

$$\text{Out[108]} = \frac{1}{120} n (2+n) (4+n) (6+n) (8+n) (x \cdot y)^5 - \frac{1}{12} n (2+n) (4+n) (8+n) (x \cdot y)^3 \|x\|^2 \|y\|^2 + \frac{1}{8} n (2+n) (8+n) x \cdot y \|x\|^4 \|y\|^4$$

To obtain the zonal harmonic $Z_m(x, y)$ as a homogeneous polynomial in x of degree m with pole z on the unit sphere (as in Theorem 5.38 of [ABR]), use **zonalHarmonic**[m , x , y] and in the result replace $\|z\|$ with 1:

```
In[109]:= setDimension[x, 7]
```

setDimension::Vector :

x will be considered to be a vector in 7-dimensional real Euclidean space.

```
In[110]:= zonalHarmonic[6, x, z] /. \|z\| -> 1
```

$$\text{Out[110]} = \frac{357}{16} (143 (x \cdot z)^6 - 143 (x \cdot z)^4 \|x\|^2 + 33 (x \cdot z)^2 \|x\|^4 - \|x\|^6)$$

To obtain the zonal harmonic $Z_m(x, z)$ as an element of $H_m(S)$ as originally defined in Chapter 5 of [ABR], use **zonalHarmonic**[m , x , z] and in the result replace $\|x\|$ and $\|z\|$ with 1:

```
In[111]:= zonalHarmonic[8, x, z] /. {\|x\| -> 1, \|z\| -> 1}
```

$$\text{Out[111]} = \frac{693}{128} (7 - 364 (x \cdot z)^2 + 2730 (x \cdot z)^4 - 6188 (x \cdot z)^6 + 4199 (x \cdot z)^8)$$

dimHarmonic

dimHarmonic[m , n] is the vector space dimension of $H_m(\mathbf{R}^n)$, the space of harmonic polynomials on \mathbf{R}^n homogeneous of degree m :

```
In[112]:= dimHarmonic[12, 100]
```

```
Out[112]= 3 901 030 682 812 965
```

dimHarmonic is computed by using Proposition 5.8 of [ABR].

Inversion and the Kelvin Transform

reflection

reflection $[\mathbf{x}]$ is the reflection of a vector x with respect to the unit sphere. Thus the reflection of x equals $\frac{x}{\|x\|^2}$. The argument of **reflection** can be either an expression representing a vector or a list of coordinates for a vector:

```
In[113]:= reflection[2 x + y]
```

```
Out[113]= 
$$\frac{2 x + y}{\|2 x + y\|^2}$$

```

```
In[114]:= reflection[{1, -2, 5, 11}]
```

```
Out[114]= 
$$\left\{ \frac{1}{151}, -\frac{2}{151}, \frac{5}{151}, \frac{11}{151} \right\}$$

```

reflection $[\mathbf{x}, \text{Sphere}[\mathbf{c}, \mathbf{r}]]$ is the reflection of a vector x with respect to the sphere centered at c with radius r . Here each of \mathbf{x} and \mathbf{c} can be either an expression representing a vector or a list of coordinates for a vector:

```
In[115]:= reflection[{2, 4, 5}, Sphere[{3, 1, 6}, 7]]
```

```
Out[115]=  $\left\{-\frac{16}{11}, \frac{158}{11}, \frac{17}{11}\right\}$ 
```

```
In[116]:= reflection[2 x + y, Sphere[z, 3]]
```

```
Out[116]=  $z + \frac{9(2x + y - z)}{\|2x + y - z\|^2}$ 
```

reflection[**x**, **Sphere**[**c**, **r**]] is computed using equation 4.13 in [ABR].

reflection[**x**, **Hyperplane**[**b**, **t**]] is the reflection of a vector x with respect to the hyperplane consisting of those vectors whose inner product with b equals t . In other words, if n denotes the dimension of x (which must equal the dimension of b), then the reflection is taken with respect to the hyperplane $\{y \in \mathbf{R}^n: b \cdot y = t\}$. Here each of **x** and **b** can be either an expression representing a vector or a list of coordinates for a vector:

```
In[117]:= reflection[x, Hyperplane[{1, 4, 5}, 7]]
```

```
Out[117]=  $\left\{\frac{1}{21}(7 + 20x_1 - 4x_2 - 5x_3), \frac{1}{21}(28 - 4x_1 + 5x_2 - 20x_3), \frac{1}{21}(35 - 5x_1 - 20x_2 - 4x_3)\right\}$ 
```

```
In[118]:= reflection[{x1, x2}, Hyperplane[{4, 5}, 7]]
```

```
Out[118]=  $\left\{\frac{1}{41}(56 + 9x_1 - 40x_2), \frac{1}{41}(70 - 40x_1 - 9x_2)\right\}$ 
```

reflection[**x**, **Hyperplane**[**b**, **t**]] is computed using Exercise 9 on page 71 of [ABR].

kelvin

kelvin[**u**, **x**] is the Kelvin transform of u , thought of as a function of x :

```
In[119]:= setDimension[x, n]
```

```
setDimension::Vector :
```

```
x will be considered to be a vector in n-dimensional real Euclidean space.
```

```
In[120]:= kelvin[(x.y)^2 + x3^5 ||x||, x]
```

```
Out[120]= ||x||^-9-n ((x.y)^2 ||x||^7 + x3^5)
```

Phi

$\Phi[z]$ is the modified inversion introduced on page 154 of [ABR]. The modified inversion Φ is defined by

$$\Phi(z) = 2 \frac{z - \text{southPole}}{\|z - \text{southPole}\|^2} + \text{southPole},$$

where **southPole** is the south pole $(0, 0, \dots, 0, -1)$. The modified inversion Φ is useful for translating questions about harmonic functions on balls to questions about harmonic functions on half-spaces, and vice versa.

Here is a proof of that $\Phi(\Phi(z)) = z$, as claimed in Proposition 7.18 (a) of [ABR]:

```
In[121]:= Phi[z]
```

```
setDimension::Vector : southPole will be considered
```

```
to be a vector in dimension[z]-dimensional real Euclidean space.
```

```
Out[121]= southPole + \frac{2(-southPole + z)}{\| -southPole + z \|^2}
```

```
In[122]:= Phi[%]
```

```
Out[122]= z
```

Another, and often more convenient, form of Φ is given by $\Phi[x, y]$. Here x is a vector, y is a real number, and thus (x, y) is a vector in a Euclidean space whose dimension is one more than the

dimension of x .

Here is a proof of the assertion of Exercise 6 in Chapter 7 of [ABR], which asserts that

$$1 - \|\Phi(x, y)\|^2 = \frac{4y}{\|x\|^2 + (y+1)^2} :$$

In[123]:= `$\Phi[x, y]$`

Out[123]= $\left\{ \frac{2x}{(1+y)^2 + \|x\|^2}, -1 + \frac{2(1+y)}{(1+y)^2 + \|x\|^2} \right\}$

In[124]:= `$1 - \|\% \|^2$`

Out[124]= $\frac{4y}{1 + 2y + y^2 + \|x\|^2}$

kelvinH

`kelvinH[u, z]` is the modified Kelvin transform defined to be equal to

$$2^{(n-2)/2} \|z - \text{southPole}\|^{2-n} u(\Phi(z)).$$

This modified Kelvin transform is defined on page 155 of [ABR]. Here is a proof of the assertion made there that the modified Kelvin transform is its own inverse:

In[125]:= `setDimension[z, n]`

setDimension::Vector :

z will be considered to be a vector in n -dimensional real Euclidean space.

In[126]:= `kelvinH[u[z], z]`

Out[126]= $2^{-1 + \frac{n}{2}} \| -\text{southPole} + z \|^2 - n u \left[\text{southPole} + \frac{2(-\text{southPole} + z)}{\| -\text{southPole} + z \|^2} \right]$

In[127]:= `kelvinH[%, z]`

Out[127]=
$$\left(\frac{1}{\|-\text{southPole} + z\|} \right)^{-n} \|-\text{southPole} + z\|^{-n} u[z]$$

In[128]:= `PowerExpand[%]`

Out[128]= $u[z]$

Another form of `kelvinH` is given by `kelvinH[u, x, y]`. Here x is a vector, y is a real variable, and u is a function of (x, y) , which is a vector in a Euclidean space whose dimension is one more than the dimension of x :

In[129]:= `setDimension[x, n - 1]`

setDimension::Vector :

x will be considered to be a vector in $-1 + n$ -dimensional real Euclidean space.

In[130]:= `kelvinH[z[4], z]`

Out[130]=
$$2^{-1 + \frac{n}{2}} \|-\text{southPole} + z\|^{-n} \left(-2 \text{southPole}_4 + \|-\text{southPole} + z\|^2 \text{southPole}_4 + 2 z_4 \right)$$

The output above includes terms of the form `southPole4`, which denotes the fourth coordinate of `southPole`. Of course `southPole4` equals either 0 or -1, but since the dimension of z has been set to equal the symbol n , the computer has no way of determining whether or not n equals 4. With the input `kelvinH[x[4], x, y]` there is no ambiguity; x_4 cannot be the last coordinate, because the last coordinate is denoted by y , and thus the fourth coordinate of `southPole` is 0:

In[131]:= `kelvinH[x[4], x, y]`

Out[131]=
$$2^{n/2} \left((1 + y)^2 + \|x\|^2 \right)^{-n/2} x_4$$

Kernels

poissonKernel

poissonKernel[**x**, **y**] is the extended Poisson kernel $P(x,y)$ for the unit ball as defined by Equation 6.21 of [ABR]:

```
In[132]:= setDimension[x, n]
```

setDimension::Vector :

x will be considered to be a vector in n-dimensional real Euclidean space.

```
In[133]:= poissonKernel[x, y]
```

```
Out[133]= (1 - ||x||^2 ||y||^2) (1 - 2 x.y + ||x||^2 ||y||^2)^(-n/2)
```

The usual Poisson kernel $P(x, \zeta)$, where ζ lies on the unit sphere, can be obtained by replacing $\|\zeta\|$ with 1:

```
In[134]:= poissonKernel[x, \zeta] /. ||\zeta|| -> 1
```

```
Out[134]= (1 - ||x||^2) (1 - 2 x.\zeta + ||x||^2)^(-n/2)
```

poissonKernelH

poissonKernelH[**z**, **w**] is the extended Poisson kernel $P_H(z, w)$ for the upper half-space H , as defined by Equation 8.22 of [ABR]:

```
In[135]:= setDimension[z, 5]
```

setDimension::Vector :

z will be considered to be a vector in 5-dimensional real Euclidean space.

```
In[136]:= poissonKernelH[z, w]
```

$$\text{Out[136]= } \frac{3 (w_5 + z_5)}{4 \pi^2 (-2 w \cdot z + \|w\|^2 + \|z\|^2 + 4 w_5 z_5)^{5/2}}$$

The usual Poisson kernel $P_H(z, w)$ for the upper half-space, where we think of $w \in \partial H$, can be obtained by replacing the last coordinate of w with 0:

```
In[137]:= % /. w5 -> 0
```

$$\text{Out[137]= } \frac{3 z_5}{4 \pi^2 (-2 w \cdot z + \|w\|^2 + \|z\|^2)^{5/2}}$$

poissonKernelH[x, y, t, u] equals $P_H((x, y), (t, u))$ and thus is another format for the extended Poisson kernel on the upper half-space. Here x and t denote vectors, and y and u denote nonnegative numbers. Note that if we want to work in the upper half-space of \mathbf{R}^n , then the dimension of x should be set to $n - 1$:

```
In[138]:= setDimension[x, n - 1]
```

setDimension::Vector :

x will be considered to be a vector in $-1 + n$ -dimensional real Euclidean space.

```
In[139]:= poissonKernelH[x, y, t, u]
```

$$\text{Out[139]= } \frac{2 (u + y) \left((u + y)^2 + \|-t + x\|^2 \right)^{-n/2}}{n \text{ volume}[n]}$$

The usual Poisson kernel $P_H((x, y), t)$ for the upper half-space, where $t \in \partial H$, can be setting the last argument in this format to 0:

```
In[140]:= poissonKernelH[x, y, t, 0]
```

$$\text{Out[140]= } \frac{2 y \left(y^2 + \|-t + x\|^2 \right)^{-n/2}}{n \text{ volume}[n]}$$

bergmanKernel

bergmanKernel [**x**, **y**] is the reproducing kernel for the harmonic Bergman space of the unit ball:

In[141]:=

```
setDimension[x, 10]
```

setDimension::Vector :

x will be considered to be a vector in 10-dimensional real Euclidean space.

In[142]:=

```
bergmanKernel[x, y]
```

Out[142]=

$$\frac{12 (10 + 8 (-3 + \mathbf{x} \cdot \mathbf{y}) \|\mathbf{x}\|^2 \|\mathbf{y}\|^2 + 6 \|\mathbf{x}\|^4 \|\mathbf{y}\|^4)}{\pi^5 (1 - 2 \mathbf{x} \cdot \mathbf{y} + \|\mathbf{x}\|^2 \|\mathbf{y}\|^2)^6}$$

bergmanKernel is computed using Theorem 8.13 of [ABR].

bergmanKernelH

bergmanKernelH [**z**, **w**] is the reproducing kernel for the harmonic Bergman space of the upper half-space:

In[143]:=

```
setDimension[z, 10]
```

setDimension::Vector :

z will be considered to be a vector in 10-dimensional real Euclidean space.

In[144]:=

```
bergmanKernelH[z, w]
```

Out[144]=

$$\frac{(48 (2 \mathbf{w} \cdot \mathbf{z} - \|\mathbf{w}\|^2 - \|\mathbf{z}\|^2 + 16 w_{10} z_{10} + 10 (w_{10}^2 + z_{10}^2)))}{(\pi^5 (2 \mathbf{w} \cdot \mathbf{z} - \|\mathbf{w}\|^2 - \|\mathbf{z}\|^2 - 4 w_{10} z_{10})^6)}$$

bergmanKernelH [**x**, **y**, **t**, **u**] is another format for the reproducing kernel for the harmonic Bergman space of the upper half-space. Here *x* and *t* denote vectors, *y* and *u* denote

nonnegative numbers, and the kernel is evaluated at $((x, y), (t, u))$. Note that if we want to work in the upper half-space of \mathbf{R}^n , then the dimension of x should be set to $n - 1$:

In[145]:=

```
setDimension[x, n - 1]
```

setDimension::Vector :

x will be considered to be a vector in $-1 + n$ -dimensional real Euclidean space.

In[146]:=

```
bergmanKernelH[x, y, t, u]
```

Out[146]=

$$\frac{1}{n \text{ volume}[n]} 4 \left((-1 + n) (u + y)^2 - \|-t + x\|^2 \right) \left((u + y)^2 + \|-t + x\|^2 \right)^{-1 - \frac{n}{2}}$$

bergmanKernelH is computed by using Theorem 8.24 of [ABR].

Miscellaneous

bergmanProjection

bergmanProjection $[u, \mathbf{x}]$ is the orthogonal projection of u onto $b^2(B)$, the harmonic Bergman space of the unit ball (see Chapter 8 of [ABR]). Here u must be a polynomial function of x :

In[147]:=

```
setDimension[x, 5]
```

setDimension::Vector :

x will be considered to be a vector in 5-dimensional real Euclidean space.

In[148]:=

```
bergmanProjection[x1^5 x2^3, x]
```

Out[148]=

$$\begin{aligned} & \frac{3 x_1 x_2}{143} - \frac{3}{221} \|x\|^6 x_1 x_2 + \frac{2}{17} \|x\|^4 x_1^3 x_2 - \frac{3}{17} \|x\|^2 x_1^5 x_2 + \frac{1}{17} \|x\|^4 x_1 x_2^3 - \\ & \frac{10}{17} \|x\|^2 x_1^3 x_2^3 + x_1^5 x_2^3 + \frac{1}{17} \left(-\|x\|^2 x_1 x_2 + 2 x_1^3 x_2 + x_1 x_2^3 \right) + \frac{1}{2717} \\ & \left(135 \|x\|^4 x_1 x_2 - 660 \|x\|^2 x_1^3 x_2 + 429 x_1^5 x_2 - 330 \|x\|^2 x_1 x_2^3 + 1430 x_1^3 x_2^3 \right) \end{aligned}$$

bergmanProjection is computed by using Corollary 8.15 of [ABR].

harmonicConjugate

harmonicConjugate [**u**, **x**, **y**] is the harmonic conjugate of u on \mathbf{R}^2 , where the coordinates in \mathbf{R}^2 are denoted by x, y and u is a harmonic function of (x, y) :

```
In[149]:= harmonicConjugate[15 x^2 y + 12 x^3 y - 5 y^3 - 12 x y^3, x, y]
```

```
Out[149]= -5 x^3 - 3 x^4 + 15 x y^2 + 18 x^2 y^2 - 3 y^4
```

harmonicConjugate is computed by using Exercise 11 in Chapter 1 of [ABR].

schwarz

schwarz [**x**] is the maximum of $|u(x)|$, where u ranges over all harmonic functions on the unit ball (in the Euclidean space whose dimension is determined by x) with $u(0) = 0$ and $|u| < 1$ on the ball. Thus, in the notation used in Theorem 6.24 of [ABR] to present the Harmonic Schwarz Lemma, **schwarz** [**x**] equals $U(\|x\| \mathbf{N})$. The user must first set the dimension of the Euclidean space to a positive integer value:

```
In[150]:= setDimension[x, 5]
```

setDimension::Vector :

x will be considered to be a vector in 5-dimensional real Euclidean space.

```
In[151]:= schwarz[x]
```

```
Out[151]= 
$$\frac{\left(2 + 4 \|x\|^2 + 2 \|x\|^4 - 2 \sqrt{1 + \|x\|^2} - 3 \|x\|^2 \sqrt{1 + \|x\|^2} + 3 \|x\|^4 \sqrt{1 + \|x\|^2} + 2 \|x\|^6 \sqrt{1 + \|x\|^2}\right)}{\left(2 \|x\|^3 (1 + \|x\|^2)^2\right)}$$

```

schwarz [**x**] is computed by evaluating the Poisson integral that defines $U(\|x\| \mathbf{N})$, iterating an

appropriate formula from Appendix A of [ABR]. The computation of **schwarz [x]** may take a while (this computation involves a complicated integration), so be patient.

togetherness

When not using the HFT10.m software package, *Mathematica* might tell the user that the result of a calculation is $t^2 + (1 - t)(1 + t)$. Of course the user will easily recognize that the last quantity equals 1, but with more complicated results the user will need to apply appropriate *Mathematica* commands (such as **Simplify**, **Expand**, or **Together**) to get the result in simplest form. Applying **Simplify** to all *Mathematica* output is poor practice, because *Mathematica* can take a long time to realize that some expressions cannot be simplified. For almost all output generated by the HFT10.m package, applying **Together** to the result and all its subexpressions puts it in simplest form. Thus the HFT10.m package redefines the *Mathematica* variable **\$Post** so that for each output X , *Mathematica* will display either X or the result of applying **Together** to X and all its subexpressions, whichever is shorter (if the user had defined **\$Post** before starting the HFT10.m package, the HFT10.m version of **\$Post** is composed with the user's **\$Post**). This procedure is invisible to the user and usually requires only a short amount of computer time. If you want to turn off this feature, use the command **turnOff [togetherness]** (if the user had defined **\$Post** before starting the HFT10.m package, this restores the user's version of **\$Post**). The command **turnOn [togetherness]** turns this feature back on again.

If **homogeneous** or **taylor** is used to find an expansion about a point other than the origin, then **togetherness** will be automatically turned off. If this were not done then, for example, the last output in the *Taylor* subsection earlier in this document would be simplified to $1 + x_1 x_2 + x_1^2$, which is not what the user wanted to see when asking for an expansion about b .

x, y, z, ... Instead of x₁, x₂, x₃, ...

Most of the examples in this document use symbolic vectors, which we have usually called x or z . Vectors can also be described by giving explicit coordinates in the form of a list. Both formats (symbolic vectors and explicit lists) work with all functions in this package. Some examples of using explicit coordinates can be found in the subsection earlier in this document titled *Dirichlet problems with explicit coordinates*. Here is another example, using x , y , and z instead of x_1 , x_2 , and x_3 to find a

basis of the space of harmonic polynomials on \mathbf{R}^3 that are homogeneous of degree 4:

```
In[152]:= TableForm[basisH[4, {x, y, z}]]
```

```
Out[152]//TableForm=
```

```

35 y^4 - 30 y^2 (x^2 + y^2 + z^2) + 3 (x^2 + y^2 + z^2)^2
3 x^2 y z - 4 y^3 z + 3 y z^3
x^4 - 3 x^2 y^2 - 4 y^4 - 3 x^2 z^2 + 27 y^2 z^2 - 4 z^4
3 x^2 y z + 3 y^3 z - 4 y z^3
35 z^4 - 30 z^2 (x^2 + y^2 + z^2) + 3 (x^2 + y^2 + z^2)^2
3 x^3 y - 4 x y^3 + 3 x y z^2
x^3 z - 6 x y^2 z + x z^3
x^3 y + x y^3 - 6 x y z^2
3 x^3 z + 3 x y^2 z - 4 x z^3

```

The next command verifies that all the polynomials above are indeed harmonic.

```
In[153]:= Table[Delta[{x,y,z}][%[[j]]], {j, Length[%]}]
```

```
Out[153]= {0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

Again, using explicit coordinate lists instead of symbolic vectors will work with all functions in this package. Each notation has its advantages, depending upon the context. When working in dimensions bigger than 3 or 4, it is usually easier to use symbolic vectors instead of explicit lists of coordinates.

zeroToZero

The command `turnOn[zeroToZero]` converts a numeric output of `0` to an exact output of `0` with no decimal point. For example, *Mathematica* usually gives the result of the following calculation as `0` with a decimal point:

```
In[154]:= Integrate[E^-x^2, {x, 0, infinity}] - 0.5 sqrt(pi)
```

```
Out[154]= 0.
```

The following command changes this behavior:

```
In[155]:= turnOn[ zeroToZero]
```

zeroToZero::On : zeroToZero has been turned on. The command turnOff[zeroToZero] will turn it back off.

Now we do the same integral again, this time getting 0 with no decimal point:

```
In[156]:= Integrate[E^-x^2, {x, 0, ∞}] - 0.5 √π
```

```
Out[156]= 0
```

The command `turnOff[zeroToZero]` restores the original behavior of *Mathematica*:

```
In[157]:= turnOff[ zeroToZero ]
```

zeroToZero::Off : zeroToZero has been turned off. The command turnOn[zeroToZero] will turn it back on.

Changes from Earlier Versions

Changes for HFT10.m, version 10.00

The naming scheme for functions introduced by this package is that function names are generally spelled out in full and begin with a lower-case letter, as in **laplacian**. This is a major change from HFT9 and previous versions of the package, where function names began with an upper-case letter. This change was made because new versions of *Mathematica* have introduced new functions beginning with an upper-case letter, which have clashed with names already used by the HFT package. With this change (meaning that HFT functions now begin with a lower-case letter), this issue has disappeared.

Functions in the HFT10 package with a name formed from more than one word begin with a lower-case letter but then have the first letter of additional words begin with an upper-case letter, as in **surfaceArea**.

annulus, Annulus

The previous option **Annulus** has been replaced by **annulus**.

antiLaplacian, AntiLaplacian

The previous function **AntiLaplacian** has been replaced by **antiLaplacian**. The ability of **antiLaplacian** to detect a singularity at 0 (and thus automatically to use the option **singularity -> 0** without requiring the user to include this option) has been improved.

basisH, BasisH

The previous function **BasisH** has been replaced by **basisH**. The syntax for finding a basis that is orthonormal with respect to volume measure on the ball or normalized surface area measure on the sphere has been changed. Furthermore, a new feature has been added: now a basis can be found that is orthonormal with respect to an arbitrary user-specified inner product.

bergmanKernel, BergmanKernel

The previous function **BergmanKernel** has been replaced by **bergmanKernel**.

bergmanKernelH, BergmanKernelH

The previous function **BergmanKernelH** has been replaced by **bergmanKernelH**.

bergmanProjection, BergmanProjection

The previous function **BergmanProjection** has been replaced by **bergmanProjection**.

biDirichlet, BiDirichlet

The previous function **BiDirichlet** has been replaced by **biDirichlet**.

dimension, Dimension

The previous function **Dimension** has been replaced by **dimension**.

dimHarmonic, DimensionH

The previous function **DimensionH** has been replaced by **dimHarmonic**.

dirichlet, Dirichlet

The previous function **Dirichlet** has been replaced by **dirichlet**.

divergence, Divergence

The previous function **Divergence** has been replaced by **divergence**.

expandNorm, ExpandNorm

The previous function **ExpandNorm** has been replaced by **expandNorm**.

exteriorNeumann, exteriorNeumann

The previous function **ExteriorNeumann** has been replaced by **exteriorNeumann**.

exteriorSphere, ExteriorSphere

The previous option **ExteriorSphere** has been replaced by **exteriorSphere**.

gradient, Grad

The previous function **Grad** has been replaced by **gradient**. The function **Grad** now is a built-in *Mathematica* function with a different meaning.

harmonicConjugate, HarmonicConjugate

The previous function **HarmonicConjugate** has been replaced by **harmonicConjugate**.

hilbertSchmidt, HilbertSchmidt

The previous function **HilbertSchmidt** has been replaced by **hilbertSchmidt**. Also, some additional functionality has been added. For example, **hilbertSchmidt[4 IdentityMatrix[n]]** now evaluates to $4n$.

homogeneous, Homogeneous

The previous function **Homogeneous** has been replaced by **homogeneous**.

IdMatrix

The previous function **IdMatrix** has been deleted. Instead, use the built-in *Mathematica* function **IdentityMatrix**.

integrateBall, IntegrateBall

The previous function **IntegrateBall** has been replaced by **integrateBall**.

integrateSphere, IntegrateSphere

The previous function **IntegrateSphere** has been replaced by **integrateSphere**.

jacobian, J

The previous function **J** has been replaced by **jacobian**.

kelvin, Kelvin

The previous function **Kelvin** has been replaced by **kelvin**.

kelvinH, KelvinM

The previous function **KelvinM** has been replaced by **kelvinH**.

laplacian, Laplacian

The previous function **Laplacian** has been replaced by **laplacian**. The function **Laplacian** now is a built-in *Mathematica* function with a different meaning.

multiple, Multiple

The previous option **Multiple** has been replaced by **multiple**.

neumann, Neumann

The previous function **Neumann** has been replaced by **neumann**.

norm, Norm

The previous function **Norm** has been replaced by **norm**. The function **Norm** now is a built-in *Mathematica* function with a different meaning.

normalD, NormalD

The previous function **NormalD** has been replaced by **normalD**.

NumericQ, NumberQ

All previous uses of **NumberQ** have been replaced by **NumericQ** for greater generality.

orthonormal, Orthonormal

The previous option **Orthonormal** has been replaced by **orthonormal**.

partial, Partial

The previous function **Partial** has been replaced by **partial**.

poissonKernel, PoissonKernel

The previous function **PoissonKernel** has been replaced by **poissonKernel**.

poissonKernelH, PoissonKernelH

The previous function **PoissonKernelH** has been replaced by **poissonKernelH**.

quadratic, Quadratic

The previous option value **Quadratic** has been replaced by **quadratic**.

reflection, Reflection

The previous function **Reflection** has been replaced by **reflection**. Furthermore, for reflections in a sphere, the syntax has been changed to agree with the use of **Sphere**[**c**, **r**] by *Mathematica*, so that the first argument now denotes the center of the sphere and the second argument now denotes the radius of the sphere.

region, Region

The previous option **Region** has been replaced by **region**.

schwarz, Schwarz

The previous function **Schwarz** has been replaced by **schwarz**.

setDimension, SetDimension

The previous function **SetDimension** has been replaced by **setDimension**.

singularity, Singularity

The previous function **Singularity** has been replaced by **singularity**.

southPole, S

The previous symbol **S** has been replaced by **southPole**.

surfaceArea, SurfaceArea

The previous function **SurfaceArea** has been replaced by **surfaceArea**.

taylor, Taylor

The previous function **Taylor** has been replaced by **taylor**.

trace, Tr

The previous function **Tr** has been replaced by **trace**. The function **Tr** now is a built-in *Mathematica* function with a slightly different meaning.

togetherness, Togetherness

The previous function **Togetherness** has been replaced by **togetherness**.

turnOff, TurnOff

The previous function **TurnOff** has been replaced by **turnOff**.

turnOn, TurnOn

The previous function **TurnOn** has been replaced by **turnOn**.

volume, Volume

The previous function **Volume** has been replaced by **volume**. The function **Volume** now is a built-in *Mathematica* function with a different meaning.

zeroToZero, ZeroToZero

The previous function **ZeroToZero** has been replaced by **zeroToZero**.

zonalHarmonic, ZonalHarmonic

The previous function **ZonalHarmonic** has been replaced by **zonalHarmonic**.

Changes for HFT9.m, version 9.00

Grad

Grad is a built-in function in *Mathematica 9*. Thus it has been unprotected, removed, and then redefined to restore previous functionality.

Laplacian

Laplacian is a built-in function in *Mathematica 9*. Thus it has been unprotected, removed, and then redefined to restore previous functionality.

Changes for HFT7.m, version 7.01

Δ (Laplacian)

$\Delta_{x_symbol}[x_ . y_]$ is now correctly coded.

Changes for HFT7.m, version 7.00

δ (delta)

The function **Delta** from earlier versions has been replaced by **δ** to agree better with the usual

mathematical notation. Furthermore, the syntax has changed, with `Delta[j]` replaced by δ_j . See the subsection of this document titled "**gradient**" for an example of the use of δ .

identityMatrix

`identityMatrix[n]` denotes the n -by- n identity matrix.

Both `trace[identityMatrix[n]]` and `hilbertSchmidt[identityMatrix[n]]` will evaluate as `n`.

See the subsection of this document titled "**jacobian**" for another example of the use of `identityMatrix`.

Δ (Laplacian)

The function `Laplacian` from earlier versions has been replaced by `Δ` to agree better with the usual mathematical notation. Furthermore, the syntax has changed, with `Laplacian[f, x]` replaced by `$\Delta_x[f]$` .

See the subsection of this document titled " **Δ (Laplacian)**" for more details.

ZeroToZero

By default, `ZeroToZero` is now off rather than on when HFTm.7 starts. To turn this feature on, use the command `TurnOn[ZeroToZero]`.

See the subsection of this document titled "**ZeroToZero**" for more details.

References

[ABR] Sheldon Axler, Paul Bourdon, and Wade Ramey, *Harmonic Function Theory*, second edition, Springer Graduate Texts in Mathematics, 2001.

[AGV] Sheldon Axler, Pamela Gorkin, and Karl Voss, The Dirichlet problem on quadratic surfaces, *Mathematics of Computation* 73 (2004), 637-651.

[AR] Sheldon Axler and Wade Ramey, Harmonic polynomials and Dirichlet-type problems, *Proceedings of the American Mathematical Society* 123 (1995), 3765-3773.

[R] Walter Rudin, *Real and Complex Analysis*, third edition, McGraw-Hill, 1987.

[W] Hermann Weyl, On the volume of tubes, *American Journal of Mathematics* 61 (1939), 461-472.

Sheldon Axler

Mathematics Department

San Francisco State University

San Francisco, CA 94132, USA

email: axler@sfsu.edu

website: www.axler.net