

# Beam Pattern Response Functions and Times of Arrival for Earthbound Interferometers

Warren G. Anderson and John T. Whelan (The University of Texas at Brownsville),  
Patrick R. Brady and Jolien D. E. Creighton (University of Wisconsin - Milwaukee),  
David Chin and Keith Riles (University of Michigan)

First Version March 30, 2000; Last Revised July 18, 2002

## — Preliminaries

### — Notes on usage

Hopefully everything one needs in order to calculate response functions and times of arrival is contained in this worksheet, along with intermediate results which one can use to check calculations. There are a few results in this worksheet that are so long and unweildy that we chose to suppress the output by using a colon (:) to end the input statement rather than the usual semicolon (;). We always warn you that this is the case by putting (results suppressed) into the preceding text. If you need to see these results, just change the colon to a semicolon.

### — Technical asides

#### — *LIGO technical document LIGO-T980044-08-E*

Much of this document is based upon LIGO technical document [LIGO-T980044-08-E](#). There are, however, several definitions that are left to the reader to determine, presumably because there is only one clear choice. We have made those definitions obvious here, to wit:

- the LIGO document refers to a surface, a point, and a line through the point perpendicular to the surface. There can in principle be an arbitrary number of such lines. Clearly, they mean the line passing through the surface whose intersection with the surface is closest to the point. It can be shown that this line always intersects the surface perpendicularly.
- the term local horizontal at a point is undefined. We have taken this to mean the plane through that point which is parallel to the tangent to the surface at the intersection point mentioned above.
- they give values for angles of vectors "above the local horizontal". Again, there are an infinite number of such angles. Clearly, they mean the angle of smallest magnitude.

We have also corrected one typographical error in the document - the denominator of the equation for  $R[\phi]$  should be the square root of the value quoted.

Finally, the document also refers to an eccentricity,  $\epsilon$ . However, it is a quantity derived from

two other quoted quantities,  $a$  and  $b$  ( $[1 - \epsilon^2] = \frac{b^2}{a^2}$ ), and is therefore unnecessary, so we

have not used it.

### Note about GMST

In the [Times of Arrival](#) section, one encounters the following situation: the vector describing the location of the detector depends on the GMST at which the wave arrives at the detector. However, the whole purpose of this section is to calculate the difference between the GMST at which the wave arrives at the detector and the GMST at which it arrives at the center of the earth. This could be dealt with by inverting the functions of GMST, at the cost of computational complexity. However, the dependence of the time of arrival on this difference between the two GMST's is clearly a higher order effect, and we will ignore it. Note however that this may not be the case for the response functions themselves. The earth can rotate by approximately 0.3 arcseconds during the time it takes light to travel from the center of the earth to it's limb. This might have a significant effect on the response functions for some applications.

### Housekeeping duties

Getting started. [restart](#) for a clean slate. Load the [linalg](#) and [tensor](#) packages to give us matrix and tensor handling capabilities.

```
> restart; with(linalg):with(tensor):  
Warning, new definition for norm  
Warning, new definition for trace
```

### Earth Model WGS-84

It is convenient to first specify an earth model with which to relate the detector position and orientation to the gravitational wave propagation direction and polarization. The earth model used here is the same one used in LIGO technical document [LIGO-T980044-08-E](#); earth model WGS-84. The reference figure of this model is the surface of ( $\Sigma$ ) an oblate ellipsoid with semi-major axis

$a = 6378137$  m, semi-minor axis  $b = .6356752314 \cdot 10^7$  m. The document also refers to the

eccentricity,  $\epsilon$ . However, it is a quantity derived from  $a$  and  $b$  (i.e.  $[1 - \epsilon^2] = \frac{b^2}{a^2}$ ), and it is

therefore unnecessary. The set of relevant parameters is therefore (in units of meters):

```
> WGS84_params := {a = 6378137, b = 6356752.314};  
  
WGS84_params := { a=6378137, b=.6356752314 107 }
```

In WGS-84, one specifies the position of any point in space  $\mathbf{x}$  in terms of the point  $\sigma(\mathbf{x})$ , which is defined to be the closest point on the surface  $\Sigma$  to  $\mathbf{x}$ . The coordinates of  $\mathbf{x}$  are then specified by  $l$  and  $\lambda$ , the North latitude and East longitude of  $\sigma(\mathbf{x})$  respectively, and by the height  $h$ , which is the distance from  $\sigma(\mathbf{x})$  to  $\mathbf{x}$  along the outward normal to  $\Sigma$ . *N.B. the line segment connecting any point in space to the nearest point on any smooth surface intersects the surface orthogonally.*

## Coordinate Frames

There are at least four coordinate frames which are useful in discussing the detection of gravitational waves:

- source frame, which is chosen to provide a simple description of the source
- the [wave propagation frame](#), which provides a simple description of the waves propagating from the source to the earth
- the [earth fixed frame](#), which provides a standard earthbound coordinate frame in which to describe the waves
- the [detector frame](#), which is adapted to the particular geometry of the detector which is to detect the waves

This worksheet is primarily concerned with the latter three frames. Each frame is an orthogonal Cartesian frame. It is assumed that the spatial curvature of space-time near the earth is sufficiently small that Euclidean translations may be used to relate frames at the earth's center and its surface.

### Wave Propagation Frame

The wave propagation frame is chosen to be compatible with the definition of [Will and Wiseman](#). The origin of this frame is taken to be the centroid of the earth. The  $z$ -axis points along the line joining the origin to the source, and points *away* from the source. The  $x$ -axis will typically be chosen to make the transition from the source frame to the wave propagation frame convenient. For the purposes of this document, the  $x$ -axis can be considered to be an arbitrary vector chosen in the plane orthogonal to the  $z$ -axis. The  $y$ -axis is then chosen to complete a right-handed coordinate system. The coordinates in the wave propagation frame are denoted by  $(x_w, y_w, z_w)$ . The unit vectors are:

```
> e_wx:=create([-1],array(1..3,[1,0,0]));e_wy:=create([-1],array(1..3,[0,1,0]));e_wz:=create([-1],array(1..3,[0,0,1]));
e_wx := table([
  index_char=[ -1 ]
  compts=[ 1, 0, 0 ]
])
e_wy := table([
  index_char=[ -1 ]
  compts=[ 0, 1, 0 ]
])
e_wz := table([
  index_char=[ -1 ]
  compts=[ 0, 0, 1 ]
])
```

### Earth Fixed Frame

The earth fixed frame is defined in LIGO technical document [LIGO-T980044-08-E](#). The origin

of this frame is the centroid of the earth. The  $z$ -axis points from the origin to the North pole  $\{ l = 90 \text{ N}, \lambda = 0 \}$ . The  $x$ -axis points from the origin to the intersection of the earth's equator and prime meridian  $\{ 0, 0 \}$ . The  $y$ -axis is chosen to complete a right-handed coordinate system  $\{ 0, 90 \text{ E} \}$ . Note that this coordinate system rotates with earth (with respect to the fixed background stars). Its relationship to the wave propagation frame therefore changes as a periodic function of time, with a period of one sidereal day. The coordinates in the earth fixed frame are denoted by

$(x_e, y_e, z_e)$ . The unit vectors are:

```
> e_ex:=create([-1],array(1..3,[1,0,0]));e_ey:=create([-1],array(1..
  3,[0,1,0]));e_ez:=create([-1],array(1..3,[0,0,1]));

e_ex := table([
  index_char=[ -1 ]
  compts=[ 1, 0, 0 ]
])
e_ey := table([
  index_char=[ -1 ]
  compts=[ 0, 1, 0 ]
])
e_ez := table([
  index_char=[ -1 ]
  compts=[ 0, 0, 1 ]
])
```

## Detector Frame

We define the detector frame in terms of cardinal compass points at the position of the detector on the earth's surface. The definition of the detector frame assumes the oblate ellipsoidal [earth model WGS-84](#). We begin with some preliminary definitions. Let  $\sigma(\mathbf{x})$  be the on the reference ellipsoid's surface ( $\Sigma$ ) nearest to the point  $\mathbf{x}$ . Let  $\Lambda(\mathbf{x})$  be the tangent plane to  $\Sigma$  at  $\sigma(\mathbf{x})$ . Define the *local horizontal* at  $\mathbf{x}$  to be the plane containing  $\mathbf{x}$  which is parallel to  $\Lambda(\mathbf{x})$ . Within the local horizontal, the directions North and East are inherited from their definitions in  $\Lambda(\mathbf{x})$ . Note that there is no ambiguity in this since the local horizontal is parallel to  $\Lambda(\mathbf{x})$ .  $\sigma(\mathbf{x})$  is the point on the surface of the ellipsoid that is closest to the corner of the interferometer (this will be unique if the corner is near enough the surface). The origin of the detector frame is then the corner of the interferometer. The  $x$ - $y$  plane is defined to be the local horizontal. The  $x$ -axis is chosen to point due East in this plane, and the  $y$ -axis to point due North, so that  $x$  and  $y$  are orthogonal. The  $z$ -axis points along the outward normal to the earth's surface. The coordinates in the detector frame are denoted by  $(x_d, y_d, z_d)$ . The unit vectors are:

```
> e_dx:=create([-1],array(1..3,[1,0,0]));e_dy:=create([-1],array(1..
```

```

3,[0,1,0]));e_dz:=create([-1],array(1..3,[0,0,1]));
e_dx := table([
  index_char=[ -1 ]
  compts=[ 1, 0, 0 ]
])
e_dy := table([
  index_char=[ -1 ]
  compts=[ 0, 1, 0 ]
])
e_dz := table([
  index_char=[ -1 ]
  compts=[ 0, 0, 1 ]
])

```

## Gravitational Wave Tensor Components

The components of the wave are naturally given in the [wave propagation frame](#). As a first step, define the polarization tensors  $e_{plus}$  and  $e_{cross}$  to be:

```

> e_plus:=lin_com(1,prod(e_wx,e_wx),-1,prod(e_wy,e_wy));
e_plus := table([
  index_char=[ -1, -1 ]
  compts= $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ 
])
> e_cross:=lin_com(1,prod(e_wx,e_wy),1,prod(e_wy,e_wx));
e_cross := table([
  index_char=[ -1, -1 ]
  compts= $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ 
])

```

The gravitational wave tensor is a linear combination of  $e_{plus}$  and  $e_{cross}$

```

> h_tensor:=lin_com(h_plus,e_plus,h_cross,e_cross);
h_tensor := table([
  index_char=[ -1, -1 ]

```

$$compts = \begin{bmatrix} h\_plus & h\_cross & 0 \\ h\_cross & -h\_plus & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

1)

## The Transformation from Wave Propagation to Earth Fixed Frames

We will need to write an arbitrary vector in the [wave propagation frame](#) in terms of [earth fixed frame](#) unit vectors. The transformation between the wave propagation and earth fixed frames that will allow us to do this is carried out using Euler angle rotations.

### Definition of the Euler Angles

First, some initial definitions. "Counterclockwise about an axis" means in a counterclockwise direction as viewed from that axis while facing the origin. The line of nodes is defined to be the line of intersection of the  $x_e - y_e$  plane and the  $x_w - y_w$  plane. If one draws a circle about the origin on the  $x_e - y_e$  plane, there will be two points of intersection between the circle and the line of nodes, and these are called nodes. Draw a circle in the  $x_w - y_w$  plane about the origin and passing through the nodes. Draw tangent vectors to the circle pointing counterclockwise about the  $z_w$ -axis. At one node, the tangent vector thus drawn will have a positive  $z_e$  component, at the other the tangent vector will have a negative  $z_e$  component. The former is called the ascending node, and the half line from the origin through the ascending node is called the line of ascending nodes. The Euler angles are defined with respect to the axes of the two frames and the line of ascending nodes. They are:

$\Psi$  - the angle from the line of ascending nodes to the positive  $x_w$ -axis counterclockwise about the positive  $z_w$ -axis.

$\Theta$  - the smallest angle between the positive  $z_e$ -axis and the positive  $z_w$ -axis. It is always positive.

$\Phi$  - the angle from the positive  $x_e$ -axis to the line of ascending nodes counterclockwise about the  $z_e$ -axis.

### The Rotation Matrices

The three rotation matrices are:

```
> R[Psi]:=array(1..3,1..3,[[cos(Psi),sin(Psi),0],[-sin(Psi),cos(Psi),0],[0,0,1]]);
```

$$R_{\Psi} := \begin{bmatrix} \cos(\Psi) & \sin(\Psi) & 0 \\ -\sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```

> R[Theta]:=array(1..3,1..3,[[1,0,0],[0,cos(Theta),sin(Theta)],[0,-sin(Theta),cos(Theta)]]);

R_Θ := [ 1      0      0
         0      cos(Θ)  sin(Θ)
         0     -sin(Θ)  cos(Θ) ]

> R[Phi]:=array(1..3,1..3,[[cos(Phi),sin(Phi),0],[-sin(Phi),cos(Phi),0],[0,0,1]]);

R_Φ := [ cos(Φ)  sin(Φ)  0
        -sin(Φ)  cos(Φ)  0
         0        0      1 ]

```

The full rotation matrix which transforms wave propagation frame vectors in the fixed earth frame is:

```

> R[Euler]:=multiply(R[Psi],multiply(R[Theta],R[Phi]));

R_Euler :=
[ cos(Ψ) cos(Φ) - sin(Ψ) cos(Θ) sin(Φ) ,
  cos(Ψ) sin(Φ) + sin(Ψ) cos(Θ) cos(Φ) , sin(Ψ) sin(Θ) ]
[ -sin(Ψ) cos(Φ) - cos(Ψ) cos(Θ) sin(Φ) ,
  -sin(Ψ) sin(Φ) + cos(Ψ) cos(Θ) cos(Φ) , cos(Ψ) sin(Θ) ]
[ sin(Θ) sin(Φ) , -sin(Θ) cos(Φ) , cos(Θ) ]

```

## Sky Angles

While the Euler angles are useful for defining the rotation matrix, the gravitational wave community has historically used angles specifying the sky position rather than the Euler angles above. We will recalculate in terms of these angles now.

### Definition of Sky Angles

The three angles in terms of which the response functions are usually calculated are the altitude  $\theta$ , the azimuth  $\phi$ , and the polarization angle  $\psi$ . Altitude and azimuth specify the position of the gravitational wave source on the sky with respect to the [earth fixed coordinates](#), i.e. they define a vector in the earth fixed frame that points toward the source. The polarization angle  $\psi$  contains information about the polarization of the wave. Altitude and azimuth might best be understood by their simple relationship to declination  $\delta$  and right ascension  $\alpha$ . The relationship is

```

> sky_to_celestial:={phi=alpha-GMST,theta=Pi/2-delta};

sky_to_celestial := { φ = α - GMST, θ =  $\frac{1}{2} \pi - \delta$  }

```

where GMST, the Greenwich Mean Sidereal Time of the observation, allows us to relate a frame fixed with respect to the celestial sphere to the rotating earth fixed frame.

### Relationship Between Sky Angles and Euler Angles

The relationship between the sky angles and the [Euler angles](#) is also particularly simple. Recall that  $\Theta$  measures the angle between the  $z_e$ -axis and the  $z_w$ -axis, which points *away* from the

source. Thus, the angle  $\theta$  between the  $z_e$  and the direction to the source must be given by  $\theta = \pi - \Theta$ . Likewise, recall that  $\Phi$  is the angle from the  $x_e$ -axis to the line of ascending nodes.

Furthermore, observe that the projection of the  $z_w$ -axis onto the  $x_e - y_e$  plane is  $\frac{\pi}{2}$  clockwise from the line of ascending nodes about the positive  $z_e$ -axis. Thus,  $\phi$ , the angle from the  $x_e$ -axis to the line of ascending nodes about the positive  $z_e$ -axis, is given by  $\phi = \Phi + \frac{\pi}{2}$ . Finally, since the Euler angle  $\Psi$  encodes all the necessary polarization information already, we take it to be the polarization angle, i.e.  $\psi = \Psi$ . The complete coordinate transformation is therefore

```
> Euler_to_sky := {Theta=Pi-theta, Phi=phi-Pi/2, Psi=psi};
      Euler_to_sky := { Theta=pi - theta, Phi=phi - 1/2 pi, Psi=psi }
```

## The Rotation Matrices in Terms of Sky Angles

It is now a straightforward matter to express the rotation matrices in terms of the sky angles. They are:

```
> R[psi] := map2(subs, Euler_to_sky, R[Psi]);
      R_psi := [ cos(psi)  sin(psi)  0
                -sin(psi)  cos(psi)  0
                 0         0         1 ]

> R[theta] := map(expand, map2(subs, Euler_to_sky, R[Theta]));
      R_theta := [ 1  0  0
                  0 -cos(theta)  sin(theta)
                  0 -sin(theta) -cos(theta) ]

> R[phi] := map(expand, map2(subs, Euler_to_sky, R[Phi]));
      R_phi := [ sin(phi)  -cos(phi)  0
                 cos(phi)  sin(phi)  0
                  0         0         1 ]

> R[sky] := map(expand, map2(subs, Euler_to_sky, R[Euler]));
      R_sky :=
      [ cos(psi) sin(phi) - sin(psi) cos(theta) cos(phi) ,
        -cos(psi) cos(phi) - sin(psi) sin(phi) cos(theta) , sin(psi) sin(theta) ]
      [ -sin(psi) sin(phi) - cos(psi) cos(theta) cos(phi) ,
        sin(psi) cos(phi) - cos(psi) sin(phi) cos(theta) , cos(psi) sin(theta) ]
      [ -sin(theta) cos(phi) , -sin(theta) sin(phi) , -cos(theta) ]
```

## The Gravitational Wave Tensor in the Earth Fixed Frame

In order to find detector response functions, it is convenient to express the [gravitational wave vector](#)

in terms of the [earth fixed frame](#) coordinate basis. To do this, we first create a triad to hold the earth fixed frame vectors

```
[ > triad[e]:=vector([e_e_x,e_e_y,e_e_z]);
      triad_e := [ e_e_x, e_e_y, e_e_z ]
```

We then transform then multiply by the rotation matrix  $R_{Euler}$  to get the triad in the [wave propagation frame](#)

```
[ > triad[w]:=multiply(R[sky],triad[e]);
      triad_w := [ ( cos(ψ) sin(φ) - sin(ψ) cos(θ) cos(φ) ) e_e_x
                  + ( -cos(ψ) cos(φ) - sin(ψ) sin(φ) cos(θ) ) e_e_y
                  + sin(ψ) sin(θ) e_e_z,
                  ( -sin(ψ) sin(φ) - cos(ψ) cos(θ) cos(φ) ) e_e_x
                  + ( sin(ψ) cos(φ) - cos(ψ) sin(φ) cos(θ) ) e_e_y
                  + cos(ψ) sin(θ) e_e_z,
                  -sin(θ) cos(φ) e_e_x - sin(θ) sin(φ) e_e_y - cos(θ) e_e_z ]
```

The individual wave propagation frame vectors can now be written in the earth fixed frame. They are

```
[ > ind:=[x,y,z]:
      for c1 from 1 to 3 do
          e_w.(ind[c1]):=lin_com(coeff(triad[w][c1],e_e_x),e_ex,coeff(triad[w][c1],e_e_y),e_ey,coeff(triad[w][c1],e_e_z),e_ez);
      od; c1:='c1':
      e_wx := table([
          index_char=[ -1 ]
          compts=[ cos(ψ) sin(φ) - sin(ψ) cos(θ) cos(φ),
                  -cos(ψ) cos(φ) - sin(ψ) sin(φ) cos(θ), sin(ψ) sin(θ) ]
          ])
      e_wy := table([
          index_char=[ -1 ]
          compts=[ -sin(ψ) sin(φ) - cos(ψ) cos(θ) cos(φ),
                  sin(ψ) cos(φ) - cos(ψ) sin(φ) cos(θ), cos(ψ) sin(θ) ]
          ])
      e_wz := table([
          index_char=[ -1 ]
          compts=[ -sin(θ) cos(φ), -sin(θ) sin(φ), -cos(θ) ]
          ])
```

Now we can recalculate the [e\\_plus](#) and [e\\_cross](#) in the earth fixed frame ([results suppressed](#)),

```
[ > 'tensor/lin_com/simp':= proc(x) simplify(x, trig) end:
      e_plus:=lin_com(1,prod(e_wx,e_wx),-1,prod(e_wy,e_wy)):
```

```

[ e_cross:=lin_com(1,prod(e_wx,e_wy),1,prod(e_wy,e_wx)):

```

And finally, the gravitational wave tensor in the earth fixed frame ([results suppressed](#)).

```

[ > 'tensor/lin_com/simp' := proc(x) collect(simplify(x,
  trig),{h_plus,h_cross}) end:
[ h_tensor:=lin_com(h_plus,e_plus,h_cross,e_cross):

```

To verify, calculate the response functions for a detector at the center of the earth with arms along the  $x_e$  and  $y_e$  axes. Since  $h\_tensor$  has already been written in the detector frame above, the

strain  $h$  is just given by  $h\_tensor_{1,1} - h\_tensor_{2,2}$ .  $F\_plus$  and  $F\_cross$  are then simply the coefficients of  $h\_plus$  and  $h\_cross$  respectively.

```

[ > F_plus:=coeff(get_compts(h_tensor)[1,1]/2-get_compts(h_tensor)[2,2]/2,
  h_plus);
  F_cross:=coeff(get_compts(h_tensor)[1,1]/2-get_compts(h_tensor)[2,2]/2,
  h_cross);

```

```

F_plus := cos(ψ)² - 2 cos(ψ)² cos(φ)²
        - 4 cos(ψ) sin(φ) sin(ψ) cos(θ) cos(φ) + cos(θ)² cos(φ)²
        - 2 cos(θ)² cos(φ)² cos(ψ)² - 1/2 + cos(φ)² - 1/2 cos(θ)²
        + cos(θ)² cos(ψ)²
F_cross := -cos(ψ) sin(ψ) + 2 cos(ψ) sin(ψ) cos(φ)²
          - 4 cos(ψ)² sin(φ) cos(θ) cos(φ) + 2 cos(θ) cos(φ) sin(φ)
          + 2 sin(ψ) cos(θ)² cos(φ)² cos(ψ) - sin(ψ) cos(θ)² cos(ψ)

```

This is a bit messy, so let's simplify

```

[ > tmp1:=expand(cos(2*phi))=cos(2*phi):
  tmp2:=expand(cos(2*psi))=cos(2*psi):
  tmp3:=expand(sin(2*phi))=sin(2*phi):
  tmp4:=expand(sin(2*psi))=sin(2*psi):
  F_plus:=algsubs(tmp1,algsubs(tmp2,algsubs(tmp3,algsubs(tmp4,F_plus)))));
  F_cross:=algsubs(tmp1,algsubs(tmp2,algsubs(tmp3,algsubs(tmp4,F_cross)))));

```

```

F_plus := -cos(θ) sin(2ψ) sin(2φ)
          - 1/2 cos(θ)² cos(2ψ) cos(2φ) - 1/2 cos(2ψ) cos(2φ)
F_cross := 1/2 cos(θ)² sin(2ψ) cos(2φ) + 1/2 sin(2ψ) cos(2φ)
          - cos(θ) sin(2φ) cos(2ψ)

```

This is exactly what [Anderson, Brady, Creighton and Flanagan](#) get!

## Location and Arm Orientation Vectors for Detectors

The primary advantage of the [earth fixed frame](#) is that one can conveniently express all the vectors describing the detector arms in it, and these vectors, once determined, are fixed for all time. Our goal, therefore, is to find for each detector three vectors:

- a *location vector* describing the position of the corner station and
- two *arm orientation vectors* describing the orientation of the arms.

These can then be used with the [gravitational wave tensor in the earth fixed frame](#) to obtain the strain produced by the gravitational wave anywhere. Information which can be used to construct these vectors for each detector is (mostly) contained within the data frames (not to be confused with coordinate frames) produced by the detector. However, this information is not specified in terms of the earth fixed coordinates. The information used to reconstruct the location vector is given in terms of the [earth model WGS-84](#) coordinates. The information needed for the arm orientation vectors is encoded in a set of arm orientation angles.

### Arm orientation angle definitions

There are four arm orientation angles,  $\psi_1$  and  $\psi_2$  which are found in data frames, and  $\omega_1$  and  $\omega_2$  which are not in the data frames at the present time. We note in passing that while three angles are sufficient to specify the orientation of any frame (and hence of the detector arms), we will follow [LIGO-T980044-08-E](#) in using a four-angle system. The four orientation angles most easily defined in the [detector frame](#). They are defined to be:

- $\psi_1$ : the angle North of East of the projection of one arm (arm 1) onto the local horizontal.
- $\psi_2$ : the angle North of East of the projection of the other arm (arm 2) onto the local horizontal.
- $\omega_1$ : the angle of smallest magnitude between arm 1 and the local horizontal. Angles above (below) the horizontal are +ve (-ve).
- $\omega_2$ : the angle of smallest magnitude between arm 2 and the local horizontal. Angles above (below) the horizontal are +ve (-ve).

We furthermore require the transformation from [earth model WGS-84](#) coordinates to earth fixed coordinates,

### Transformation from earth model WGS-84 coordinates to earth fixed coordinates.

This transformation from  $(l, \lambda, h)$ , the [earth model WGS-84](#) coordinates to the [fixed earth frame](#) coordinates  $(x_e, y_e, z_e)$  is described in [LIGO-T980044-08-E](#). The transformation is

```
> M2E_x:=x[e]=(R+h)*cos(l)*cos(lambda);
M2E_y:=y[e]=(R+h)*cos(l)*sin(lambda);
M2E_z:=z[e]=(b^2*R/a^2+h)*sin(l);
earthModel_to_earthFixed:=[M2E_x,M2E_y,M2E_z];

M2E_x := x_e = ( R + h ) cos( l ) cos( lambda )
```

$$M2E_y := y_e = (R+h) \cos(l) \sin(\lambda)$$

$$M2E_z := z_e = \left( \frac{b^2 R}{a^2} + h \right) \sin(l)$$

$$earthModel\_to\_earthFixed := \begin{bmatrix} x_e = (R+h) \cos(l) \cos(\lambda), \\ y_e = (R+h) \cos(l) \sin(\lambda), z_e = \left( \frac{b^2 R}{a^2} + h \right) \sin(l) \end{bmatrix}$$

where  $R$  is the local radius of curvature, defined as

```
> Rdef:=R=a^2/sqrt(a^2*cos(l)^2+b^2*sin(l)^2);
```

$$Rdef := R = \frac{a^2}{\sqrt{a^2 \cos(l)^2 + b^2 \sin(l)^2}}$$

and the [detector frame](#) basis vectors expressed in terms of the [earth fixed frame](#) basis.

## Detector frame basis vectors in the earth fixed frame

The components of the [detector frame](#) basis vectors in the [earth fixed frame](#) basis are can be calculated in terms of the derivatives of the transformation from earth model WGS-84 coordinates to earth fixed coordinates (i.e. the Jacobian of the transformation). For the  $\lambda$  and  $h$  derivatives, which give us the components of  $e_{d_x}$  and  $e_{d_z}$  respectively, this is relatively trivial. The components of  $e_{d_y}$  are considerably more difficult. First, we calculate the components of  $e_{d_x}$  in the earth fixed frame.

### $\lambda$ derivatives and $e_{d_x}$

First the derivatives with respect to  $\lambda$ .

```
> M2E_derivs[lambda] := [ diff(rhs(M2E_x),lambda),
  diff(rhs(M2E_y),lambda), diff(rhs(M2E_z),lambda) ];
M2E_derivs_lambda :=
  [ -(R+h) cos(l) sin(lambda), (R+h) cos(l) cos(lambda), 0 ]
```

This gives an un-normalized vector. The normalization factor is:

```
> e_dx_norm:=simplify(sqrt(algsubs(sin(lambda)^2+cos(lambda)^2=1,
  factor(M2E_derivs[lambda][1]^2+
  M2E_derivs[lambda][2]^2+M2E_derivs[lambda][3]^2)),symbolic);
  e_dx_norm := cos(l) (R+h)
```

The components of the detector frame  $x$  basis vector in the earth fixed basis are therefore:

```
> e_d_x:=array([M2E_derivs[lambda][i]/e_dx_norm$i=1..3]);
  e_d_x := [ -sin(lambda), cos(lambda), 0 ]
```

Now the components of  $e_{d_z}$ .

### $h$ derivatives and $e_{d_z}$

Derivatives with respect to  $h$ .

```
[ > M2E_derivs[h] := [ diff(rhs(M2E_x),h), diff(rhs(M2E_y),h),
diff(rhs(M2E_z),h) ];
M2E_derivs_h := [ cos( l ) cos( λ ), cos( l ) sin( λ ), sin( l ) ]
```

This is, in fact normalized:

```
[ > e_dz_norm:=sqrt(simplify(M2E_derivs[h][1]^2+
M2E_derivs[h][2]^2+M2E_derivs[h][3]^2,trig));
e_dz_norm := 1
```

The components of the detector frame  $x$ ; basis vector in the earth fixed basis are therefore:

```
[ > e_d_z:=array([M2E_derivs[h][i]/e_dz_norm$i=1..3]);
e_d_z := [ cos( l ) cos( λ ), cos( l ) sin( λ ), sin( l ) ]
```

We could find the components of  $e_{d_y}$  by taking a cross product of the other two basis vectors, to complete a right handed frame. However, for pedagogy's sake, we calculate it using derivatives as well.

### **1 derivatives and $e_{d_y}$**

First, let's compute the derivatives ([results suppressed](#)).

```
[ > M2E_derivs[l] := [ diff(rhs(subs(Rdef,M2E_x)),l),
diff(rhs(subs(Rdef,M2E_y)),l), diff(rhs(subs(Rdef,M2E_z)),l) ]:
```

Since we will be normalizing anyway, we can always rescale. When algebraic functions are involved, one will tend to get repeated multiplicative factors. A simple way to remove and identify these is to consider ratios of components. We do so to get simple forms for the  $x$  and  $y$  components of  $e_{d_y}$ .

```
[ > Lx:=numer(normal(M2E_derivs[l][1]/M2E_derivs[l][2]));Ly:=denom(
normal(M2E_derivs[l][1]/M2E_derivs[l][2]));
Lx := cos( λ )
Ly := sin( λ )
```

Now remove the same factor from the  $z$  component.

```
[ > Lz:=algsubs(sin(l)^2+cos(l)^2=1,normal(M2E_derivs[l][3]/M2E_der
ivs[l][1])*Lx);
Lz := -  $\frac{\cos( l )}{\sin( l )}$ 
```

The normalization factor is

```
[ > e_dy_norm:=simplify(sqrt(algsubs(sin(l)^2+cos(l)^2=1,algsubs(si
n(lambda)^2+cos(lambda)^2=1,normal(Lx^2+Ly^2+Lz^2))))),symbolic)
;
e_dy_norm :=  $\frac{1}{\sin( l )}$ 
```

Up to a sign (which we may have lost by rescaling by removing a negative multiplicative factor above), the vector is

```
[ > e_d_y_tmp:=array([Lx/e_dy_norm,Ly/e_dy_norm,Lz/e_dy_norm]);
e_d_y_tmp := [ cos( λ ) sin( l ), sin( λ ) sin( l ), -cos( l ) ]
```

We can check the sign by computing ( $e_{d_x} \times e_{d_y}$ ) and seeing whether we get  $e_{d_z}$  or  $-e_{d_z}$ .

```

> crossprod(e_d_x,e_d_y_tmp);
[ -cos( l ) cos( λ ), -cos( l ) sin( λ ),
  -sin( λ )2 sin( l ) - cos( λ )2 sin( l ) ]

```

This is  $-e_d_z$ , so  $e_d_y$  is actually given by

```

> e_d_y:=array([-e_d_y_tmp[1],-e_d_y_tmp[2],-e_d_y_tmp[3]]);
e_d_y := [ -cos( λ ) sin( l ), -sin( λ ) sin( l ), cos( l ) ]

```

## Location vector

The location vector  $\Lambda$  for a detector is given simply by the [WGS-84 to earth fixed coordinate transformation](#),

```

> Lambda:=[rhs(M2E_x),rhs(M2E_y),rhs(M2E_z)];
Λ := [ ( R+h ) cos( l ) cos( λ ), ( R+h ) cos( l ) sin( λ ),
       (  $\frac{b^2 R}{a^2} + h$  ) sin( l ) ]

```

where  $R$  is given [above](#), and the parameters  $a$  and  $b$  are also given [above](#).

## Orientation vectors

For each of the two arms, we want to have unit orientation vectors, which will determine the orientation of the arms in terms of the [orientation angles](#). In the detector coordinate frame, the angles  $\psi_{1, 2}$  play the role of azimuthal angles for arms 1 and 2 respectively, while  $\omega_{1, 2}$  are the polar complementary angles to the polar angle. Thus, it is straightforward to see that:

```

> Omega_d[1]:=vector([cos(psi1)*cos(omega1),sin(psi1)*cos(omega1),sin(omega1)]);
Omega_d[2]:=vector([cos(psi2)*cos(omega2),sin(psi2)*cos(omega2),sin(omega2)]);
Omega_d_1 := [ cos( ψ1 ) cos( ω1 ), sin( ψ1 ) cos( ω1 ), sin( ω1 ) ]
Omega_d_2 := [ cos( ψ2 ) cos( ω2 ), sin( ψ2 ) cos( ω2 ), sin( ω2 ) ]
> transpose(evalm(Omega_d[1]));
transpose( [ cos( ψ1 ) cos( ω1 ), sin( ψ1 ) cos( ω1 ), sin( ω1 ) ] )

```

To transform these into the earth fixed frame, we need to use the [detector frame to earth fixed frame transformation](#).

```

> for C1 from 1 to 2 do
  Omega_(C1) :=
  evalm(Omega_d[C1][1]*e_d_x+Omega_d[C1][2]*e_d_y+Omega_d[C1][3]*e_d_z)
od;C1:='C1':
Omega_1 := [ -cos( ψ1 ) cos( ω1 ) sin( λ )
            -sin( ψ1 ) cos( ω1 ) cos( λ ) sin( l ) + sin( ω1 ) cos( l ) cos( λ ),

```

```

cos(ψ1) cos(ω1) cos(λ) - sin(ψ1) cos(ω1) sin(λ) sin(l)
+ sin(ω1) cos(l) sin(λ),
sin(ψ1) cos(ω1) cos(l) + sin(ω1) sin(l) ]
Omega_2 := [ -cos(ψ2) cos(ω2) sin(λ)
- sin(ψ2) cos(ω2) cos(λ) sin(l) + sin(ω2) cos(l) cos(λ),
cos(ψ2) cos(ω2) cos(λ) - sin(ψ2) cos(ω2) sin(λ) sin(l)
+ sin(ω2) cos(l) sin(λ),
sin(ψ2) cos(ω2) cos(l) + sin(ω2) sin(l) ]

```

These arm direction unit vectors allow us to construct the *response tensor* in the earth-fixed frame. As a matrix, this is one-half the difference between the matrices generated by taking the outer product of each arm vector with itself. ([results suppressed](#), but available below under "[Location and Orientation vectors for known detectors](#)")

```

> Omega_1m:=convert(Omega_1,matrix):
Omega_2m:=convert(Omega_2,matrix):
Omega_1M:=transpose(Omega_1m): Omega_2M:=transpose(Omega_2m):
response_tensor:=evalm((Omega_1m &* Omega_1M-Omega_2m &*
Omega_2M)/2):

```

## Location and Orientation vectors for known detectors

First, put all vectors into a list.

```

> Det_vecs:=[Lambda,convert(Omega_1,list),convert(Omega_2,list),eval
(response_tensor)];
Det_vecs:=subs(WGS84_params, subs(Rdef,Det_vecs)):

```

$$\text{Det\_vecs} := \left[ \left[ \begin{array}{l} (R+h) \cos(l) \cos(\lambda), (R+h) \cos(l) \sin(\lambda), \\ \left( \frac{b^2 R}{a^2} + h \right) \sin(l) \end{array} \right], \left[ \begin{array}{l} -\cos(\psi_1) \cos(\omega_1) \sin(\lambda) \\ -\sin(\psi_1) \cos(\omega_1) \cos(\lambda) \sin(l) + \sin(\omega_1) \cos(l) \cos(\lambda), \\ \cos(\psi_1) \cos(\omega_1) \cos(\lambda) - \sin(\psi_1) \cos(\omega_1) \sin(\lambda) \sin(l) \\ + \sin(\omega_1) \cos(l) \sin(\lambda), \\ \sin(\psi_1) \cos(\omega_1) \cos(l) + \sin(\omega_1) \sin(l) \end{array} \right], \left[ \begin{array}{l} -\cos(\psi_2) \cos(\omega_2) \sin(\lambda) - \sin(\psi_2) \cos(\omega_2) \cos(\lambda) \sin(l) \\ + \sin(\omega_2) \cos(l) \cos(\lambda), \cos(\psi_2) \cos(\omega_2) \cos(\lambda) \\ - \sin(\psi_2) \cos(\omega_2) \sin(\lambda) \sin(l) + \sin(\omega_2) \cos(l) \sin(\lambda), \\ \sin(\psi_2) \cos(\omega_2) \cos(l) + \sin(\omega_2) \sin(l) \end{array} \right], \left[ \begin{array}{l} \frac{1}{2} \\ \frac{1}{2} \end{array} \right] \left[ \begin{array}{l} -\cos(\psi_1) \cos(\omega_1) \sin(\lambda) - \sin(\psi_1) \cos(\omega_1) \cos(\lambda) \sin(l) \\ \cos(\psi_1) \cos(\omega_1) \cos(\lambda) - \sin(\psi_1) \cos(\omega_1) \sin(\lambda) \sin(l) \\ + \sin(\omega_1) \cos(l) \sin(\lambda) \\ \sin(\psi_1) \cos(\omega_1) \cos(l) + \sin(\omega_1) \sin(l) \end{array} \right] \right]$$

$$\begin{aligned}
& + \sin(\omega_1) \cos(l) \cos(\lambda) \Big)^2 - \frac{1}{2} (-\cos(\psi_2) \cos(\omega_2) \sin(\lambda) \\
& - \sin(\psi_2) \cos(\omega_2) \cos(\lambda) \sin(l) + \sin(\omega_2) \cos(l) \cos(\lambda) \Big)^2, \\
& \frac{1}{2} (-\cos(\psi_1) \cos(\omega_1) \sin(\lambda) - \sin(\psi_1) \cos(\omega_1) \cos(\lambda) \sin(l) \\
& + \sin(\omega_1) \cos(l) \cos(\lambda) \Big) (\cos(\psi_1) \cos(\omega_1) \cos(\lambda) \\
& - \sin(\psi_1) \cos(\omega_1) \sin(\lambda) \sin(l) + \sin(\omega_1) \cos(l) \sin(\lambda) \Big) - \frac{1}{2} \\
& (-\cos(\psi_2) \cos(\omega_2) \sin(\lambda) - \sin(\psi_2) \cos(\omega_2) \cos(\lambda) \sin(l) \\
& + \sin(\omega_2) \cos(l) \cos(\lambda) \Big) (\cos(\psi_2) \cos(\omega_2) \cos(\lambda) \\
& - \sin(\psi_2) \cos(\omega_2) \sin(\lambda) \sin(l) + \sin(\omega_2) \cos(l) \sin(\lambda) \Big), \frac{1}{2} \\
& (-\cos(\psi_1) \cos(\omega_1) \sin(\lambda) - \sin(\psi_1) \cos(\omega_1) \cos(\lambda) \sin(l) \\
& + \sin(\omega_1) \cos(l) \cos(\lambda) \Big) \\
& (\sin(\psi_1) \cos(\omega_1) \cos(l) + \sin(\omega_1) \sin(l) \Big) - \frac{1}{2} ( \\
& -\cos(\psi_2) \cos(\omega_2) \sin(\lambda) - \sin(\psi_2) \cos(\omega_2) \cos(\lambda) \sin(l) \\
& + \sin(\omega_2) \cos(l) \cos(\lambda) \Big) \\
& (\sin(\psi_2) \cos(\omega_2) \cos(l) + \sin(\omega_2) \sin(l) \Big) \Big] \\
& \left[ \frac{1}{2} (-\cos(\psi_1) \cos(\omega_1) \sin(\lambda) - \sin(\psi_1) \cos(\omega_1) \cos(\lambda) \sin(l) \right. \\
& + \sin(\omega_1) \cos(l) \cos(\lambda) \Big) (\cos(\psi_1) \cos(\omega_1) \cos(\lambda) \\
& - \sin(\psi_1) \cos(\omega_1) \sin(\lambda) \sin(l) + \sin(\omega_1) \cos(l) \sin(\lambda) \Big) - \frac{1}{2} \\
& (-\cos(\psi_2) \cos(\omega_2) \sin(\lambda) - \sin(\psi_2) \cos(\omega_2) \cos(\lambda) \sin(l) \\
& + \sin(\omega_2) \cos(l) \cos(\lambda) \Big) (\cos(\psi_2) \cos(\omega_2) \cos(\lambda) \\
& - \sin(\psi_2) \cos(\omega_2) \sin(\lambda) \sin(l) + \sin(\omega_2) \cos(l) \sin(\lambda) \Big), \frac{1}{2} \\
& (\cos(\psi_1) \cos(\omega_1) \cos(\lambda) - \sin(\psi_1) \cos(\omega_1) \sin(\lambda) \sin(l) \\
& + \sin(\omega_1) \cos(l) \sin(\lambda) \Big)^2 - \frac{1}{2} (\cos(\psi_2) \cos(\omega_2) \cos(\lambda) \\
& - \sin(\psi_2) \cos(\omega_2) \sin(\lambda) \sin(l) + \sin(\omega_2) \cos(l) \sin(\lambda) \Big)^2, \\
& \frac{1}{2} (\cos(\psi_1) \cos(\omega_1) \cos(\lambda) - \sin(\psi_1) \cos(\omega_1) \sin(\lambda) \sin(l)
\end{aligned}$$

$$\begin{aligned}
& + \sin(\omega_1) \cos(l) \sin(\lambda) ) \\
& ( \sin(\psi_1) \cos(\omega_1) \cos(l) + \sin(\omega_1) \sin(l) ) - \frac{1}{2} ( \\
& \cos(\psi_2) \cos(\omega_2) \cos(\lambda) - \sin(\psi_2) \cos(\omega_2) \sin(\lambda) \sin(l) \\
& + \sin(\omega_2) \cos(l) \sin(\lambda) ) \\
& ( \sin(\psi_2) \cos(\omega_2) \cos(l) + \sin(\omega_2) \sin(l) ) \Big] \\
& \left[ \frac{1}{2} ( -\cos(\psi_1) \cos(\omega_1) \sin(\lambda) - \sin(\psi_1) \cos(\omega_1) \cos(\lambda) \sin(l) \right. \\
& + \sin(\omega_1) \cos(l) \cos(\lambda) ) \\
& ( \sin(\psi_1) \cos(\omega_1) \cos(l) + \sin(\omega_1) \sin(l) ) - \frac{1}{2} ( \\
& -\cos(\psi_2) \cos(\omega_2) \sin(\lambda) - \sin(\psi_2) \cos(\omega_2) \cos(\lambda) \sin(l) \\
& + \sin(\omega_2) \cos(l) \cos(\lambda) ) \\
& ( \sin(\psi_2) \cos(\omega_2) \cos(l) + \sin(\omega_2) \sin(l) ) , \frac{1}{2} ( \\
& \cos(\psi_1) \cos(\omega_1) \cos(\lambda) - \sin(\psi_1) \cos(\omega_1) \sin(\lambda) \sin(l) \\
& + \sin(\omega_1) \cos(l) \sin(\lambda) ) \\
& ( \sin(\psi_1) \cos(\omega_1) \cos(l) + \sin(\omega_1) \sin(l) ) - \frac{1}{2} ( \\
& \cos(\psi_2) \cos(\omega_2) \cos(\lambda) - \sin(\psi_2) \cos(\omega_2) \sin(\lambda) \sin(l) \\
& + \sin(\omega_2) \cos(l) \sin(\lambda) ) \\
& ( \sin(\psi_2) \cos(\omega_2) \cos(l) + \sin(\omega_2) \sin(l) ) , \\
& \frac{1}{2} ( \sin(\psi_1) \cos(\omega_1) \cos(l) + \sin(\omega_1) \sin(l) )^2 \\
& \left. - \frac{1}{2} ( \sin(\psi_2) \cos(\omega_2) \cos(l) + \sin(\omega_2) \sin(l) )^2 \right] \Big]
\end{aligned}$$

Now the detectors:

### **Reference to make sure we're getting it right (REF)**

Let's try a detector with arms due east and due south at 0 degrees longitude and 0 degrees latitude ... should be `[[ a, 0, 0 ], [ 0, 1, 0 ], [ 0, 0, 1 ]]`.

```

> params_REF := [
    l           = 0,
    lambda      = 0,
    h           = 0,
    psi1        = 0,
    psi2        = Pi/2,

```

```

    omega1 = 0,
    omega2 = 0
];

params_REF := [ l=0, lambda=0, h=0, psi1=0, psi2=1/2 pi, omega1=0, omega2=0 ]

> DV_REF:=evalf(subs(params_REF,Det_vecs),12);

DV_REF := [ [ .637813700002 107, 0, 0 ], [ 0, 1., 0 ],

[ 0, -.510338076868 10-11, 1. ],

[ 0          0          0
  0          .500000000000    .255169038434 10-11
  0          .255169038434 10-11    -.500000000000 ] ]

```

### **LIGO-Hanford Observatory (LHO)**

Reference:

William Althouse, Larry Jones, Albert Lazzarini (1999)

"Determination of Global and Local Coordinate Axes for the LIGO Sites"

LIGO-T980044-08-E

```

> params_LHO:= [
    h          = 142.554,
    l          = (46 + 27/60 + 18.528/3600)*Pi/180,
    lambda     = (240 + 35/60 + 32.4343/3600)*Pi/180,
    psi1       = 125.9994*Pi/180,
    psi2       = 215.9994*Pi/180,
    omega1     = -6.195e-4,
    omega2     = 1.25e-5
];

params_LHO := [ h=142.554, l=.2580841482 pi, lambda=1.336624127 pi,
    psi1=.6999966667 pi, psi2=1.199996667 pi, omega1=-.0006195,
    omega2=.0000125 ]

> DV_LHO:=evalf(subs(params_LHO,Det_vecs),12);

DV_LHO := [
[ -.216141492611 107, -.383469517823 107, .460035022722 107 ],
[ -.223892661573, .799830627548, .556904878171 ],
[ -.913978185537, .0260940390132, -.404923421764 ],

```

$$\begin{bmatrix} -.392614099866 & -.0776134127892 & -.247389044862 \\ -.0776134127892 & .319524066946 & .227997832878 \\ -.247389044862 & .227997832878 & .0730900329195 \end{bmatrix}$$

### **LIGO-Livingston Observatory (LLO)**

Reference:

William Althouse, Larry Jones, Albert Lazzarini (1999)

"Determination of Global and Local Coordinate Axes for the LIGO Sites"

LIGO-T980044-08-E

```

> params_LLO := [
    h          = -6.574,
    l          = (30 + 33/60 + 46.4196/3600)*Pi/180,
    lambda     = (269 + 13/60 + 32.7346/3600)*Pi/180,
    psi1      = 197.7165*Pi/180,
    psi2      = 287.7165*Pi/180,
    omega1    = -3.121e-4,
    omega2    = -6.107e-4
];

params_LLO := [ h=-6.574, l=.1697938574 pi, lambda=1.495698664 pi,
    psi1=1.098425000 pi, psi2=1.598425000 pi, omega1=-.0003121,
    omega2=-.0006107 ]

> DV_LLO:=evalf(subs(params_LLO,Det_vecs),12);

DV_LLO := [

[ -74276.0534461, -.549628371972 107, .322425701722 107 ],
[ -.954574121757, -.141580771885, -.262189113245 ],
[ .297741568158, -.487910336926, -.820544612871 ],
[ .411280856259   .140210264909   .247294591145
.140210264909   -.109005690956   -.181615630748
.247294591145   -.181615630748   -.302275165304 ] ]

```

### **VIRGO Interferometer (VIRGO)**

Reference:

Benoit Mours, E-mail: mours@lapp.in2p3.fr

```

> params_VIRGO := [
    h          = 51.884,
    l          = (43 + 37/60 + 53.0921/3600)*Pi/180,
    lambda     = (10 + 30/60 + 16.1878/3600)*Pi/180,
    psi1      = 70.5674*Pi/180,
    psi2      = 160.5674*Pi/180,
    omega1    = 0,
    omega2    = 0
];

```

```

params_VIRGO := [ h=51.884, l=.2423967471  $\pi$ ,  $\lambda$ =.05835831450  $\pi$ ,
   $\psi$ 1=.3920411111  $\pi$ ,  $\psi$ 2=.8920411111  $\pi$ ,  $\omega$ 1=0,  $\omega$ 2=0 ]
> DV_VIRGO:=evalf(subs(params_VIRGO,Det_vecs),12);

DV_VIRGO := [
  [ .454637409863 107, 842989.697467, .437857696275 107 ],
  [ -.700458214851, .208489486220, .682561662696 ],
  [ -.053792553751, -.969081805482, .240804517079 ],
  [ .243874035956      -.0990837792218      -.232576216925 ],
  [ -.0990837792218    -.447825839926      .187833103275 ],
  [ -.232576216925     .187833103275      .203951803968 ] ]

```

### **GEO-600 Interferometer (GEO600)**

Reference:

<http://www.geo600.uni-hannover.de/geo600/project/location.html>

```

> params_GEO600:= [
  h      = 114.425,
  l      = (52 + 14/60 + 42.528/3600)*Pi/180,
  lambda = (9 + 48/60 + 25.894/3600)*Pi/180,
   $\psi$ 1    = 21.6117*Pi/180,
   $\psi$ 2    = 115.9431*Pi/180,
   $\omega$ 1 = 0,
   $\omega$ 2 = 0
];

params_GEO600 := [ h=114.425, l=.2902508148  $\pi$ ,
   $\lambda$ =.05448440432  $\pi$ ,  $\psi$ 1=.1200650000  $\pi$ ,  $\psi$ 2=.6441283333  $\pi$ ,  $\omega$ 1=0,
   $\omega$ 2=0 ]
> DV_GEO600:=evalf(subs(params_GEO600,Det_vecs),12);

DV_GEO600 := [
  [ .385630994953 107, 666598.956352, .501964141692 107 ],
  [ -.445306769034, .866513541301, .225513113130 ],
  [ -.626057567788, -.552186095145, .550583724918 ],
  [ -.0968249798190      -.365782314548      .122137295952 ],
  [ -.365782314548      .222968116793      .249717421691 ],
  [ .122137295952       .249717421691      -.126143136976 ] ]

```

### **TAMA-300 Interferometer (TAMA300)**

Reference:

Masa-Katsu Fujimoto (1995) (unpublished), E-mail: [fujimoto@gravity.mtk.nao.ac.jp](mailto:fujimoto@gravity.mtk.nao.ac.jp)

```

> params_TAMA300:=[
  h      = 90,
  l      = (35 + 40/60 + 35.6/3600)*Pi/180,
  lambda = (139 + 32/60 + 9.8/3600)*Pi/180,
  psi1   = 180*Pi/180,
  psi2   = 270*Pi/180,
  omega1 = 0,
  omega2 = 0
];

params_TAMA300 := [ h=90, l=.1982030864 π, λ=.7752003083 π,
  ψ1=π, ψ2=3/2 π, ω1=0, ω2=0 ]

> DV_TAMA300:=evalf(subs(params_TAMA300,Det_vecs),12);

DV_TAMA300 := [
  [-.394640898771 107, .336625903242 107, .369915069189 107 ],
  [.648969406115, .760814504283, -.167956993920 10-12 ],
  [-.443713768767, .378484715231, -.812322233933 ],
  [.112139690741 .330842108212 -.180219279936]
  [.330842108212 .217794015133 .153725774693]
  [-.180219279936 .153725774693 -.329933705871] ]

```

### **Caltech-40 Interferometer (CIT40)**

Reference:

B. Allen, "Gravitational Wave Detector Sites," gr-qc/9607075 (1996).

```

> params_CIT40:=[
  h      = 0,
  l      = 34.17*Pi/180,
  lambda = -118.13*Pi/180,
  psi1   = 270*Pi/180,
  psi2   = 0*Pi/180,
  omega1 = 0,
  omega2 = 0
];

params_CIT40 := [ h=0, l=.1898333333 π, λ=-.6562777778 π,
  ψ1=3/2 π, ψ2=0, ω1=0, ω2=0 ]

> DV_CIT40:=evalf(subs(params_CIT40,Det_vecs),12);

```

```
DV_CIT40 := [
    [ -.249064958399 107, -.465869968229 107, .356206411337 107 ],
    [ -.264803316317, -.495308185286, -.827374767114 ],
    [ .881880123832, -.471473697241, 0 ],
    [ -.353795878239    .273471266285    .109545791085 ],
    [ .273471266285    .011521375611    .204902747226 ],
    [ .109545791085    .204902747226    .342274502629 ] ]
```

### MPQ-30 (MPQ30)

Reference:

B. Allen, "Gravitational Wave Detector Sites," gr-qc/9607075 (1996).

```
> params_MPQ30 := [
    h      = 0,
    l      = 35.57*Pi/180,
    lambda = 139.47*Pi/180,
    psi1   = 132*Pi/180,
    psi2   = 225*Pi/180,
    omega1 = 0,
    omega2 = 0
];

params_MPQ30 := [ h=0, l=.1976111111 pi, lambda=.7748333333 pi,
    psi1 = 11/15 pi, psi2 = 5/4 pi, omega1=0, omega2=0 ]

> DV_MPQ30 := evalf(subs(params_MPQ30, Det_vecs), 12);

DV_MPQ30 := [
    [ -.394770376318 107, .337523393974 107, .368948807262 107 ],
    [ .763397142331, .227664424303, .604478050046 ],
    [ .146878798953, .804743690014, -.575164508468 ],
    [ .280600907669    .0277992921345    .272968144106 ],
    [ .0277992921345    -.297890658262    .300239078089 ],
    [ .272968144106    .300239078089    .017289750593 ] ]
```

### TAMA-20 (TAMA20)

Reference:

B. Allen, "Gravitational Wave Detector Sites," gr-qc/9607075 (1996).

```
> params_TAMA20 := [
    h      = 0,
```

```

    l      = 35.68*Pi/180,
    lambda = 139.54*Pi/180,
    psi2   = 45*Pi/180,
    psi1   = 135*Pi/180,
    omega1 = 0,
    omega2 = 0
];

params_TAMA20 := [ h=0, l=.1982222222 pi, lambda=.7752222222 pi,
  psi2=1/4 pi, psi1=3/4 pi, omega1=0, omega2=0 ]
> DV_TAMA20:=evalf(subs(params_TAMA20,Det_vecs),12);

DV_TAMA20 := [
  [-.394641550942 107, .336579525359 107, .369940864286 107 ],
  [.772651341534, .270378770836, .574373767425 ],
  [-.145055918411, -.805638601252, .574373767424 ],
  [.287974438054 .0460229363990 .263553488145]
  [.0460229363990 -.287974438055 .309018075911]
  [.263553488145 .309018075911 0 ] ]

```

### **Glasgow-10 (1995) (G1095)**

Reference:

B. Allen, "Gravitational Wave Detector Sites," gr-qc/9607075 (1996).

```

> params_G1095:= [
  h      = 0,
  l      = 55.87*Pi/180,
  lambda = -4.28*Pi/180,
  psi1   = 152*Pi/180,
  psi2   = 242*Pi/180,
  omega1 = 0,
  omega2 = 0
];

params_G1095 := [ h=0, l=.3103888889 pi, lambda=-.02377777778 pi,
  psi1=38/45 pi, psi2=121/90 pi, omega1=0, omega2=0 ]
> DV_G1095:=evalf(subs(params_G1095,Det_vecs),12);

DV_G1095 := [

```

```
[ .357683021395 107, -267687.639613, .525633471276 107 ],
[ -.453424109740, -.851482838592, .263407578267 ],
[ .693799397147, -.522707959431, -.495397603614 ],
[ -.137882090094    .374368657593    .112135606031
  .374368657593    .225899706782    -.241617651471
  .112135606031    -.241617651471    -.0880176166887 ] ]
```

## Response Functions and a Simple Example

We now have the information necessary in to calculate the beam pattern functions for a gravitational wave that arrives from a source at known declination and right ascension at a given GMST and with a given polarization angle  $\psi$ . We begin by defining a procedure that will return the required results as a list.

### Definition of the "Response" procedure

MAPLE Response() procedure. '#' denotes the beginning of a comment. Comments end at carriage returns.

```
> Response:= proc(
    dec,          # declination
    ra,          # right ascension
    time,        # Greenwich mean sidereal time
    pol_angle,   # polarization angle
    lat,         # North latitude
    long,        # East longitude
    height,      # height above reference ellipsoid
    psi_arm1,    # arm 1 polarization angle
    psi_arm2,    # arm 2 polarization angle
    omega_arm1, # arm 1 angle to horizontal
    omega_arm2) # arm 2 angle to horizontal

    # specify the local and global variables. Note that delta, alpha
    # etc. MUST be global.
    local parms,arm_vec1,arm_vec2,strain,F_p,F_c;
    global delta,alpha,GMST,psi,l,lambda,h,

    WGS84_params,Rdef,sky_to_celestial,Omega,h_tensor,R,F_plus,F_cross
;

    # create a set of parameter values to with which to evaluate.
    parms:={
        delta=dec,
        alpha=ra,
        GMST=time,
        psi=pol_angle,
        l=lat,
        lambda=long,
        h=height,
```

```

        psi1=psi_arm1,
        psi2=psi_arm2,
        omegal=omega_arm1,
        omega2=omega_arm2
    };

# create the two arm vectors
    arm_vec1:=create([1], subs(WGS84_params, subs(Rdef, subs(parms,
op(Omega_1)))));
    arm_vec2:=create([1], subs(WGS84_params, subs(Rdef, subs(parms,
op(Omega_2)))));

# dot the strain vectors into h_tensor to get the strain in the
detector
    strain:= get_compts( prod(h_tensor,

lin_com(1,prod(arm_vec1,arm_vec1),-1,prod(arm_vec2,arm_vec2)),
[1,1], [2,2]));

# extract response functions, i.e. the coefficients of the plus
and cross polarizations
    F_p:=subs(parms,subs(sky_to_celestial,coeff(strain,h_plus)))/2;

F_c:=subs(parms,subs(sky_to_celestial,coeff(strain,h_cross)))/2;

# make sure the labels we are using for the response functions are
not assigned.
    F_plus:= 'F_plus';
    F_cross:='F_cross';

# return a list with the response functions
    RETURN([F_plus=simplify(F_p),F_cross=simplify(F_c)]);
end:

```

## Response for reference detector

As a check, calculate the known response functions for a reference detector and source. The detector is at the equator  $l = 0$  and prime meridian  $\lambda = 0$ , with arms running due North  $\psi_1 = 0$  and East  $\psi_2 = \frac{\pi}{2}$ , on the reference ellipsoid  $h = 0$ , and in the local horizontal

$[\omega_1 = 0, \omega_2 = 0]$ . The source is at declination  $\delta = 0$  and right ascension  $\alpha = 0$ , and the wave arrives at the reference detector at  $GMST = 0$  and with polarization angle  $\psi = 0$ . The response functions should be  $[F_{plus} = 1, F_{cross} = 0]$ .

```

> Response_REF:=eval(subs(params_REF,Response(0,0,0,0,1,lambda,h,psi
1,psi2,omegal,omega2)));
Response_REF := [ F_plus=1, F_cross=0 ]

```

## Response Functions for LIGO Hanford (LHO) and LIGO

# Livingston (LLO) Detectors

Response functions for Hanford and Livingston detectors for an generic source.

## Response for LHO

```
> Response_LHO:=simplify(subs(params_LHO,Response(delta,alpha,GMST,psi,l,lambda,h,psil,psi2,omega1,omega2)));
Response_LHO := [ F_plus=.3104536563 cos(ψ) sin(ψ) sin(δ)
-.1552268269 sin(-1. α+GMST) cos(-1. α+GMST) cos(δ)2
-.6209073106 cos(ψ)2 sin(-1. α+GMST) cos(-1. α+GMST)
+.4947780908 cos(δ) sin(δ) cos(-1. α+GMST)
-.9895561806 cos(δ) sin(δ) cos(-1. α+GMST) cos(ψ)2
+.4559956647 cos(δ) sin(-1. α+GMST) sin(δ)
-1.424276333 cos(ψ)2 cos(-1. α+GMST)2 cos(δ)2
-1.424276335 cos(ψ)2+.4928680670 cos(ψ)2 cos(δ)2
+.9895561807 sin(ψ) cos(δ) cos(ψ) sin(-1. α+GMST)
-.6209073102 sin(ψ) sin(δ) cos(-1. α+GMST)2 cos(ψ)
+.3104536568 sin(-1. α+GMST) cos(-1. α+GMST)
+2.848552664 cos(ψ)2 cos(-1. α+GMST)2-2.848552664
cos(ψ) sin(-1. α+GMST) sin(ψ) sin(δ) cos(-1. α+GMST)
+.7121381649 cos(-1. α+GMST)2 cos(δ)2+.3104536565
cos(ψ)2 sin(-1. α+GMST) cos(-1. α+GMST) cos(δ)2+.7121381662
-.9119913309 sin(ψ) cos(δ) cos(ψ) cos(-1. α+GMST)
-1.424276333 cos(-1. α+GMST)2-.2464340348 cos(δ)2
-.9119913309 cos(δ) sin(-1. α+GMST) sin(δ) cos(ψ)2, F_cross
=-.4947780908 cos(δ) sin(-1. α+GMST)-.3104536569
cos(ψ) sin(-1. α+GMST) sin(ψ) cos(-1. α+GMST) cos(δ)2
+1.424276334 cos(ψ) sin(ψ)
+.4559956650 cos(δ) cos(-1. α+GMST)
-.4928680678 cos(ψ) sin(ψ) cos(δ)2-.1552268269 sin(δ)
+.3104536564 cos(ψ)2 sin(δ)
-2.848552664 cos(ψ)2 sin(-1. α+GMST) sin(δ) cos(-1. α+GMST)
```

```

+ .6209073092 cos( ψ ) sin( -1. α+GMST ) sin( ψ ) cos( -1. α+GMST )
+1.424276334 cos( ψ ) sin( ψ ) cos( -1. α+GMST )2 cos( δ )2
+1.424276334 sin( δ ) cos( -1. α+GMST ) sin( -1. α+GMST )
+ .9895561806 cos( ψ ) cos( δ ) sin( ψ ) sin( δ ) cos( -1. α+GMST )
- .6209073097 cos( ψ )2 sin( δ ) cos( -1. α+GMST )2
+ .9119913300 cos( ψ ) cos( δ ) sin( ψ ) sin( -1. α+GMST ) sin( δ )
+ .3104536564 sin( δ ) cos( -1. α+GMST )2
- 2.848552664 cos( ψ ) sin( ψ ) cos( -1. α+GMST )2
+ .9895561807 cos( ψ )2 cos( δ ) sin( -1. α+GMST )
- .9119913310 cos( ψ )2 cos( δ ) cos( -1. α+GMST ) ]

```

## Response for LLO

```

> Response_LLO:=simplify(subs(params_LLO,Response(delta,alpha,GMST,psi,l,lambda,h,psil,psi2,omega1,omega2)));
Response_LLO := [ F_plus=
1.040573095 cos( ψ )2 cos( -1. α+GMST )2 cos( δ )2
- .5608410611 cos( ψ ) sin( ψ ) sin( δ )
- .3632312613 cos( δ ) sin( -1. α+GMST ) sin( δ )
- .4945891833 cos( δ ) sin( δ ) cos( -1. α+GMST )
+1.121682122 cos( ψ )2 sin( -1. α+GMST ) cos( -1. α+GMST )
+ .2804205304 sin( -1. α+GMST ) cos( -1. α+GMST ) cos( δ )2
+1.040573095 cos( ψ )2 + .3865389495 cos( ψ )2 cos( δ )2
- .5608410613 sin( -1. α+GMST ) cos( -1. α+GMST ) - .5202865474
+1.121682123 sin( ψ ) sin( δ ) cos( -1. α+GMST )2 cos( ψ )
+1.040573094 cos( -1. α+GMST )2
- 2.081146184 cos( ψ )2 cos( -1. α+GMST )2
- .5202865471 cos( -1. α+GMST )2 cos( δ )2 - .1932694746 cos( δ )2 -
.5608410611 cos( ψ )2 sin( -1. α+GMST ) cos( -1. α+GMST ) cos( δ )2
+ 2.081146185
cos( ψ ) sin( -1. α+GMST ) sin( ψ ) sin( δ ) cos( -1. α+GMST )
+ .7264625220 sin( ψ ) cos( δ ) cos( ψ ) cos( -1. α+GMST )

```

$$\begin{aligned}
& + .9891783665 \cos(\delta) \sin(\delta) \cos(-1. \alpha + GMST) \cos(\psi)^2 \\
& + .7264625219 \cos(\delta) \sin(-1. \alpha + GMST) \sin(\delta) \cos(\psi)^2 \\
& - .9891783675 \sin(\psi) \cos(\delta) \cos(\psi) \sin(-1. \alpha + GMST), F_{cross} = \\
& 2.081146184 \cos(\psi)^2 \sin(-1. \alpha + GMST) \sin(\delta) \cos(-1. \alpha + GMST) \\
& + 1.121682122 \cos(\psi)^2 \sin(\delta) \cos(-1. \alpha + GMST)^2 \\
& - .9891783659 \cos(\psi)^2 \cos(\delta) \sin(-1. \alpha + GMST) \\
& - .7264625222 \cos(\psi) \cos(\delta) \sin(\psi) \sin(-1. \alpha + GMST) \sin(\delta) \\
& + .4945891833 \cos(\delta) \sin(-1. \alpha + GMST) \\
& + .7264625219 \cos(\psi)^2 \cos(\delta) \cos(-1. \alpha + GMST) \\
& - .9891783664 \cos(\psi) \cos(\delta) \sin(\psi) \sin(\delta) \cos(-1. \alpha + GMST) \\
& - .3632312613 \cos(\delta) \cos(-1. \alpha + GMST) \\
& - 1.040573095 \cos(\psi) \sin(\psi) \\
& - .3865389495 \cos(\psi) \sin(\psi) \cos(\delta)^2 + .2804205304 \sin(\delta) \\
& - .5608410613 \cos(\psi)^2 \sin(\delta) \\
& - 1.040573096 \sin(\delta) \cos(-1. \alpha + GMST) \sin(-1. \alpha + GMST) \\
& + 2.081146185 \cos(\psi) \sin(\psi) \cos(-1. \alpha + GMST)^2 \\
& - 1.121682123 \cos(\psi) \sin(-1. \alpha + GMST) \sin(\psi) \cos(-1. \alpha + GMST) \\
& - .5608410612 \sin(\delta) \cos(-1. \alpha + GMST)^2 + .5608410612 \\
& \cos(\psi) \sin(-1. \alpha + GMST) \sin(\psi) \cos(-1. \alpha + GMST) \cos(\delta)^2 \\
& - 1.040573097 \cos(\psi) \sin(\psi) \cos(-1. \alpha + GMST)^2 \cos(\delta)^2 ]
\end{aligned}$$

## Times of Arrival

The time of arrival of a gravitational wave at a detector can be calculated if one knows the [sky angles](#) of the source, the detectors [location vector](#) of the detector, and the GMST at which the wave arrives at the origin of the [earth fixed coordinates](#). The method is straightforward. One uses the sky angles, which are the azimuthal and polar angle for a unit vector pointing to the source, to construct just such a vector. The difference between the distance the wave travels to the coordinate origin and to the detector is thus given by the projection of the detectors location vector onto this unit vector. Finally, the difference in time of arrival is just the difference in distance divided by the speed of light.

### The source position unit vector

Because the sky angles  $\{\phi, \theta\}$  are just the azimuthal and polar angles for this vector, the unit vector is

```
> e_sky:=vector([cos(phi)*sin(theta),sin(phi)*sin(theta),cos(theta)]
);
      e_sky := [ sin(θ) cos(φ), sin(θ) sin(φ), cos(θ) ]
```

## The time delay from the origin to the detector

First, calculate the required inner product

```
> det_orig_dist:=simplify(subs(sky_to_celestial,subs(WGS84_params,su
bs(Rdef,dotprod(e_sky,convert(Lambda,array),orthogonal)))));
det_orig_dist := (
  .4068063159 1012 cos(δ) cos(-1. α+GMST) cos(l) cos(λ) +
  cos(δ) cos(-1. α+GMST) cos(l) cos(λ) h
  √ .27233161 108 cos(l)2 + .4040829998 1010
  - .4068063159 1012 cos(δ) sin(-1. α+GMST) cos(l) sin(λ) - 1.
  cos(δ) sin(-1. α+GMST) cos(l) sin(λ) h
  √ .27233161 108 cos(l)2 + .4040829998 1010
  + .4040829998 1012 sin(δ) sin(l)
  + sin(δ) sin(l) h √ .27233161 108 cos(l)2 + .4040829998 1010 )
  / √ .27233161 108 cos(l)2 + .4040829998 1010
```

The GMST appearing in the above formula is actually the GMST at which the wave arrives at the detector. However, we lose only higher order corrections by taking it to be the GMST at which the wave arrives at the origin of the earth fixed coordinates. We therefore have that the time difference between the arrival of a wave at the origin and at the detector is

```
> Delta_GMST:= -simplify(evalf(det_orig_dist/(3*108)));
Delta_GMST := -.1000000000 10-17 (
  .1356021053 1022 cos(δ) cos(-1. α+GMST) cos(l) cos(λ) +
  .3333333333 1010 cos(δ) cos(-1. α+GMST) cos(l) cos(λ) h
  √ .27233161 108 cos(l)2 + .4040829998 1010
  - .1356021053 1022 cos(δ) sin(-1. α+GMST) cos(l) sin(λ) -
  .3333333333 1010 cos(δ) sin(-1. α+GMST) cos(l) sin(λ) h
  √ .27233161 108 cos(l)2 + .4040829998 1010
  + .1346943333 1022 sin(δ) sin(l) + .3333333333 1010
  sin(δ) sin(l) h √ .27233161 108 cos(l)2 + .4040829998 1010 ) /
```

$$\sqrt{.27233161 \cdot 10^8 \cos(\delta)^2 + .4040829998 \cdot 10^{10}}$$

## An example: Time difference between LIGO Hanford (LHO) and LIGO Livingston (LLO) detectors

Using this result, we find that the time difference between a wave arriving at the Hanford and Livingston observatories for a generic source is

```
> GMST[LHO-LLO] := evalf(subs(params_LHO, Delta_GMST) - subs(params_LLO, Delta_GMST));
GMST_LHO - LLO := .006957129568 cos(δ) cos(-1. α + GMST)
+ .00553862846 cos(δ) sin(-1. α + GMST) - .00458697738 sin(δ)
```

## References

Clifford M. Will and Alan G. Wiseman, *Physical Review D* 54, page 4813 (1996).

William Althouse, Larry Jones, Albert Lazzarini, *Determination of Global and Local Coordinate Axes for the LIGO Sites*, LIGO-T980044-08-E (1999).

Warren G. Anderson, Patrick R. Brady, Jolien D. E. Creighton, and Éanna É. Flanagan, *An excess power statistic for detection of burst sources of gravitational radiation*, gr-qc/0008066