CMBquick 2. Documentation

This is the doc file CMBquickDoc2.nb of version 0.0.3 of `CMBquick2`` `. Last update on 22 July 2011.

■ Preambule

Author

Cyril Pitrou

Institute of Cosmology and Gravitation, Portsmouth, UK

`cyril.pitrou@ens-lyon.org`

`http://icg.port.ac.uk/~pitrouc/cmbquick.htm`

Intro

`CMBquick2`` is a *Mathematica* Package which provides tools to compute non-linear perturbations solutions in cosmology.

It thus goes beyond the capabilities of CMBFAST or CAMB and extends the functions and procedures developed in `CMBquick1``

Since computations in *Mathematica* are slower than in low level compiled code like Fortran or C, it is unavoidably slow.

However, the purpose of this package is not the speed of computation, but rather to have a code easy to

understand, and thus to modify, and which provides patterns for second order integrations

Its functional design enables the user to compute non-linear perturbations interactovely.

This enables to check, plot, output and understand all the steps of the computation in a straightforward way, and is thus a

good tool to study and understamd the behaviour of non-linear equations in cosmology.

It is restricted to second order in cosmological perturbations but we use the abusive "non-linear" to distiguish the second order

functions from the first order (linear) functions.

The main features provided are

1.Second order transfer functions for all geometrical and physical variables.

2.Computations of the second order sources in the flat sky (lengthy).

3.Computation of the primordial bispectrum both in the flat sky and full sky methods.

4.Computation of the bispectrum generated by non-linear effects, once the second order sources have been stored.

Possible determination of the contamination to primordial non-Gaussianity (lenghy) that is the equivalent f_{NL}, and signal to noise ratios.

5.Integration of the distorsion field Y (of its multipoles). The computation of the spectrum of spectral distorsions is provided in a separate "example" notebook

6.Integration of the magnetic field generated by non-linear effects.. The computation of the spectrum of the magnetic field is provided in a separate "example" notebook

See `History2[]` and `ToBeDone2[]` for more information about versions and future developments in `` CMBquick2` ``

Note that this package is still in "alpha" version and is widely incomplete contrary to `` CMBquick1` `` and so is the documentation.

Loading the package

Loading the package is straightforward once it has been installed :

You just have to evaluate '**<< CMBquick/CMBquick2.m;**'

This loads all the definition, and also loads `CMBquick1` but does not start any computation.

We can also check how much memory space

it takes to store all these definitions, and the time it takes to load them (less than 2 second!):

```
In[1]:= MemBefore = MemoryInUse[];
        << CMBquick/CMBquick2.m; // Timing // First

           ------------------------------------------------------------------------------
             -------

           These packages come with ABSOLUTELY
             NO WARRANTY; for details type Disclaimer[].
           This is free software, and you are welcome to
             redistribute it under certain conditions.
           See the General Public License for details.

           ------------------------------------------------------------------------------
             -------

           Package CMBquick`CMBquick1`  version 0.0.3, {2011, 7, 22}

           CopyRight (C) 2009-2010, Cyril Pitrou, under the General Public License.

           ------------------------------------------------------------------------------
             -------

           Please send me your comments and
             don't hesitate to report the bugs and mistakes

           at cyril.pitrou@ens-lyon.org

           For information, browse also the
             Add-Ons in the Documentation Center of the Help.

           You can also find worked examples in the
             'Examples' subdirectory of your CMBquick installation.

           You can also download the pdf of the documentation on the  CMBquick  page.

           For information on versions and
             future development, type 'ToBeDone[]' and 'History[]'

           ------------------------------------------------------------------------------
             -------

           A file with interpolation
              functions for the Bessel functions has been found in
           /Users/cyrilp/Library/Mathematica/Applications/CMBquick/Data/jls.dat
           It will be used if you load it with 'LoadAndGenerateBesselsBinary[lmax]'
           unless you erase it and recreate it.
           ------------------------------------------------------------------------
              -------------

           Package CMBquick`CMBquick2`  version 0.0.3, {2011, 7, 22}

           CopyRight (C) 2009-2010, Cyril Pitrou, under the General Public License.

           ------------------------------------------------------------------------------
             -------

Out[2]= 1.3809

In[3]:= MemoryInUse[] - MemBefore

Out[3]= 12 306 040
```

GPL

```
In[4]:= Disclaimer[]
```

```
These are points 11 and 12 of the General Public License:

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
  FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT
  WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER
  PARTIES PROVIDE THE PROGRAM `AS IS´ WITHOUT WARRANTY OF ANY KIND,
  EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
  IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
  PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF
  THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU
  ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
  WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
  REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR
  DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL
  DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM
  (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED
  INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF
  THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER
  OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
```

Useful definitions for this notebook

A 2 quick overview

■ Second order transfer functions

The main feature of `CMBquick2`is to allow the numerical integration of the second order system of Einstein-Boltzmann hierarchy.

Choice of equations

The Booleans value for the numerical integartions are (click for information)

```
In[18]:= ?$Integrate*
```

▼ CMBquick\`CMBquick1\`

$IntegrateTC

▼ CMBquick\`CMBquick2\`

$IntegrateM0
$IntegrateM1
$IntegrateM2
$IntegrateMagnetic
$IntegrateSources
$IntegrateSpectralDistorsions

We choose in this section to integrate only the brightness (scalar vectors and tensor), and not to integrate the spectral distorsions or the magnetic field. All these options are set to true by default when the package is loaded.

`$IntegrateM0` is the boolean for scalar second order equations (m=0)

`$IntegrateM1` is the boolean for vector second order equations (m=1)

`$IntegrateM2` is the boolean for tensor second order equations (m=2)

`$IntegrateMagnetic` is the boolean for magnetic field integration

`$IntegrateSpectralDistorsions` is the boolean for integrating the spectral distorsion Y

`$IntegrateSources` has to be set to `True` if we also want to memorize the expression of the quadratic sources.

```
In[19]:= $IntegrateSources = False;
         $IntegrateSpectralDistorsions = False;
         $IntegrateMagnetic = False;
```

We choose the configuration of k_1 k_2 and μ_{12} for which we want to compute the transfer functions.

```
In[22]:= k1t = 0.05 RescaleMp[CPL];
         k2t = k1t / 10;
         Print["k1 = ", k1t / RescaleMp[CPL],
          " Mpc^-1", "  and  k2 = ", k2t / RescaleMp[CPL], " Mpc^-1"]
         angle = 0;

           k1 = 0.05 Mpc^-1  and  k2 = 0.005 Mpc^-1
```

Energy densities

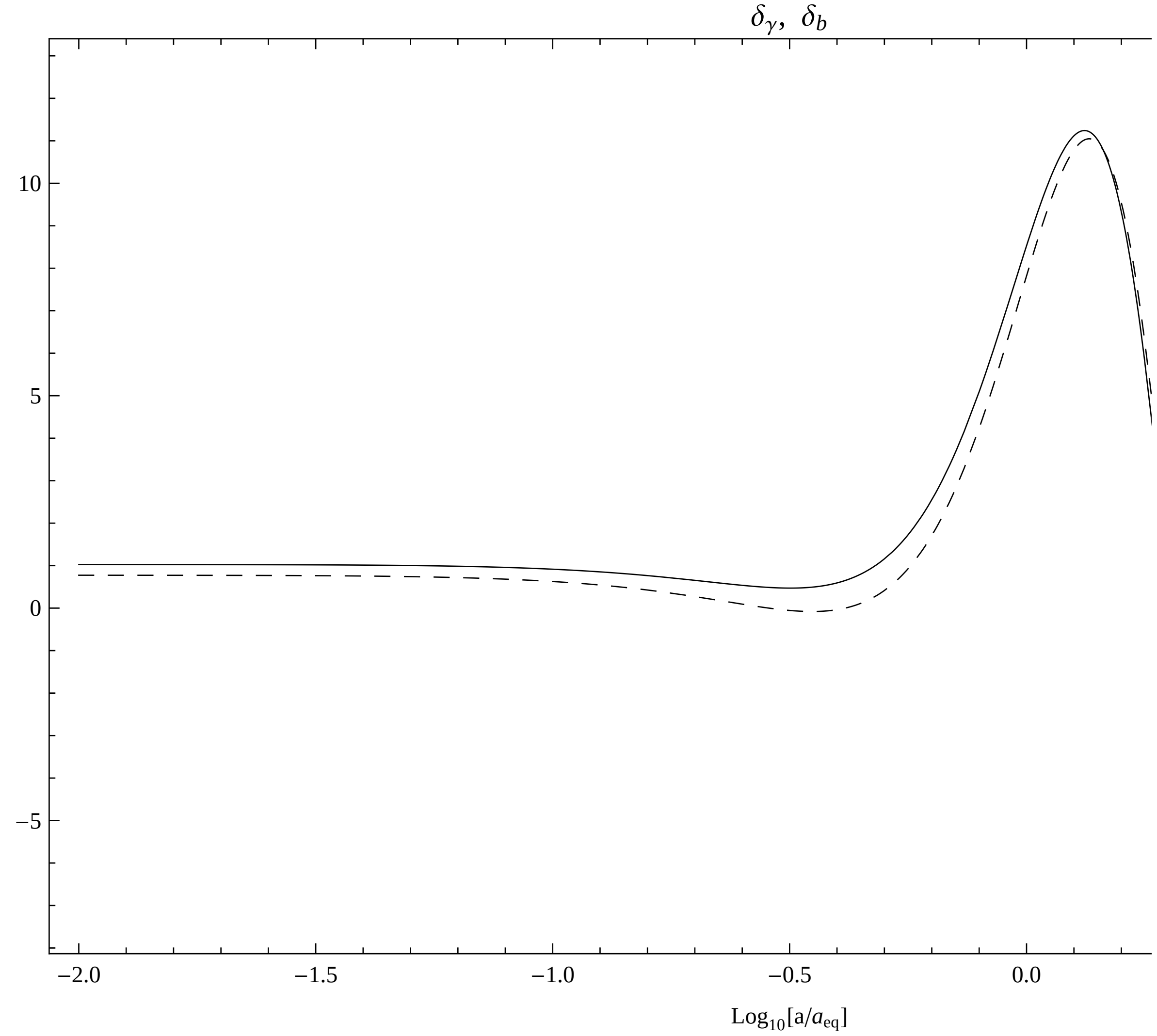
The moment l=m=0 is not exactly the energy density for radiation and neutrinos. We thus define the energy density of radiation:

```
In[26]:= dρr[cpl_][k1_, k2_, m_][y_] := Ro2[cpl][0, 0][k1, k2, m][y] / 4 +
           2 / 3 / 16 * Produp12[0, 0, 0, k1, k2, m] Ro1[cpl][1][k1][y] Ro1[cpl][1][k2][y]
```

And we can draw the evolutions of the energy densities of radiation and baryons.

```
In[27]:= MyPlotBandW[
           {dρr[CPL][k1t, k2t, angle][10^p], B0o2[CPL][k1t, k2t, angle][10^p] / 3},
           {p, Log[10, yi2], 1}, Frame → True, Axes → False,
           FrameLabel → "Log10[a/aeq]", PlotLabel → "δγ,   δb"]
```

`Out[27]=`



We thus see that the syntax is not very different from first order transfer functions. The mode `k` is replaced by the set of `k1,k2,`μ`.` And the syntax of the multipoles is slighlty modified since now we must specify L and M, instead of just L for the first order case.

```
In[28]:= ? Ro2
```

```
Moments of radiation at second
     order. Syntax is Ro2[cpl][L,M][k1,k2,m12][y].
```

```
In[29]:= ? B0o1
```

```
Baryons monopole (zeroth moment). Syntax is B0o1[cpl][k][y]
```

Velocities

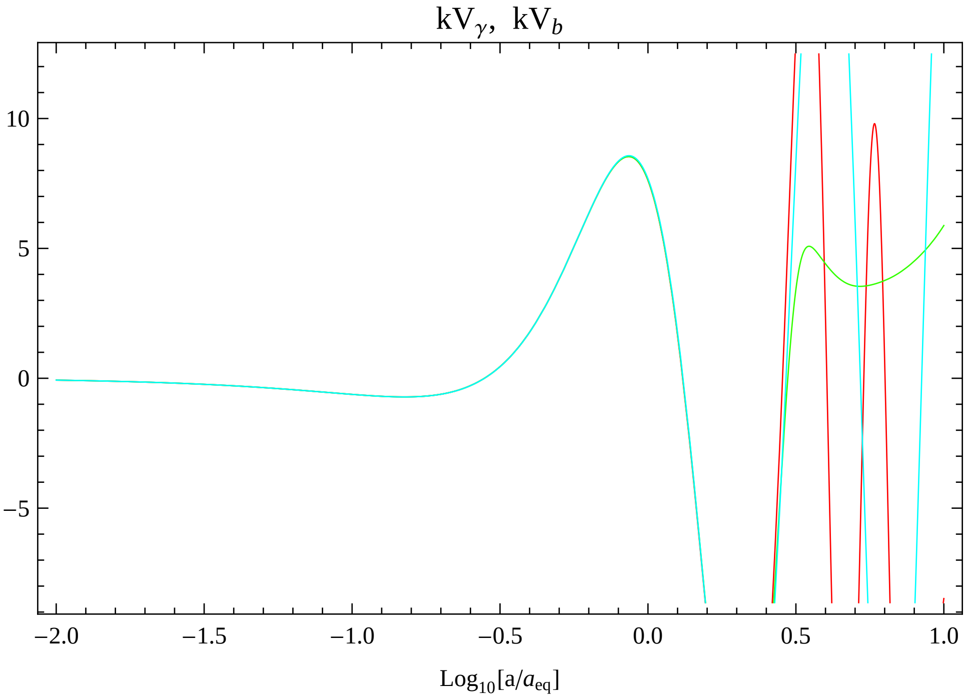
The first moment (l = 1 m = 0) is not exactly the velocity. We thus define the velocity of radiation, in order to be able to compare with the velocity of baryons and cold dark matter.

```
In[30]:= Vr[cpl_][k1_, k2_, m_][y_] := Ro2[cpl][1, 0][k1, k2, m][y] / 4 -
           (Ro1[cpl][0][k1][y] B1o1[cpl][k2][y] Rot2[1, 0, k2uz[k1, k2, m]] +
             Ro1[cpl][0][k2][y] B1o1[cpl][k1][y] Rot1[1, 0, k1uz[k1, k2, m]])
```

And we can draw on the same plot the velocity of baryons and radiation. We also draw the fluid velocity obtained from the tight coupling approximation.

```
In[31]:= MyPlotColors[{-Vr[CPL][k1t, k2t, angle][10^p], -B1o2[CPL][k1t, k2t, angle][10^p],
           -F1o2[CPL][k1t, k2t, angle][10^p]}, {p, Log[10, yi2], 1},
          Axes -> False, FrameLabel → "Log10[a/aeq]", PlotLabel → "kVγ,   kVb"]
```

Out[31]=



```
In[32]:= ? B1o1
         ? F1o1
```

Baryons first moment. Syntax is B1o1[cpl][k][y]

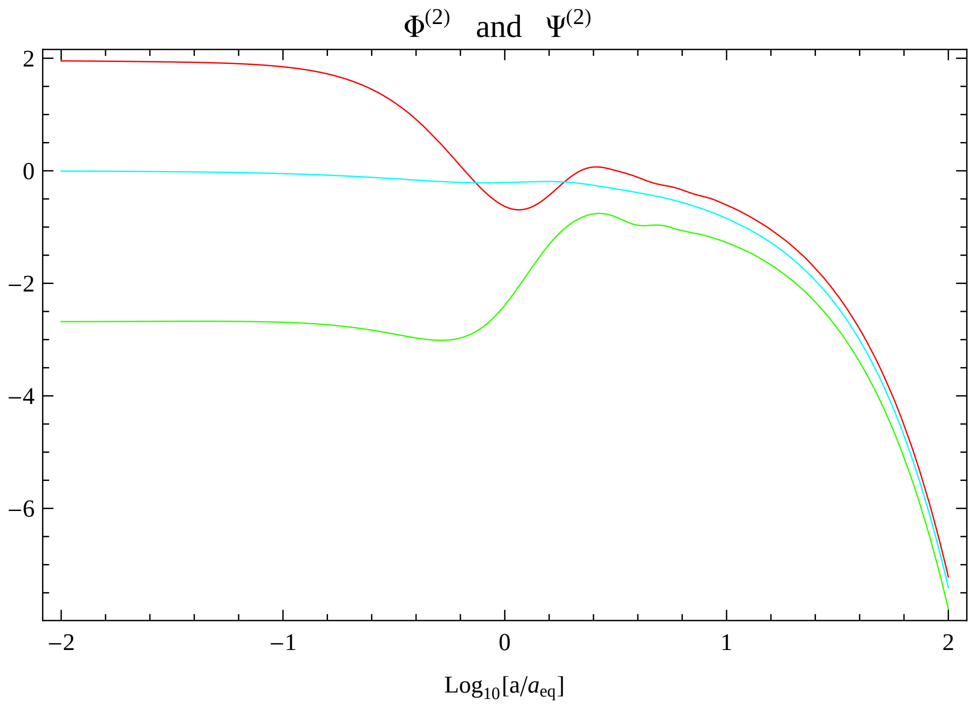
Tight coupled fluid first moment. Syntax is F1o1[cpl][k][y]

Potentials

We plot the two gravitational potentials Φ and Ψ and their asymptotic form which is to be reached in the matter domination

```
In[34]:= MyPlotColors[{Φ2[CPL][k1t, k2t, angle][10^p], Ψ2[CPL][k1t, k2t, angle][10^p],
           Φ2Asymptotic[CPL][k1t, k2t, angle][10^p]}, {p, -2, 2}, Frame → True,
          Axes → False, FrameLabel → "Log10[a/aeq]", PlotLabel -> "Φ(2)    and    Ψ(2)   "]
```

Out[34]=

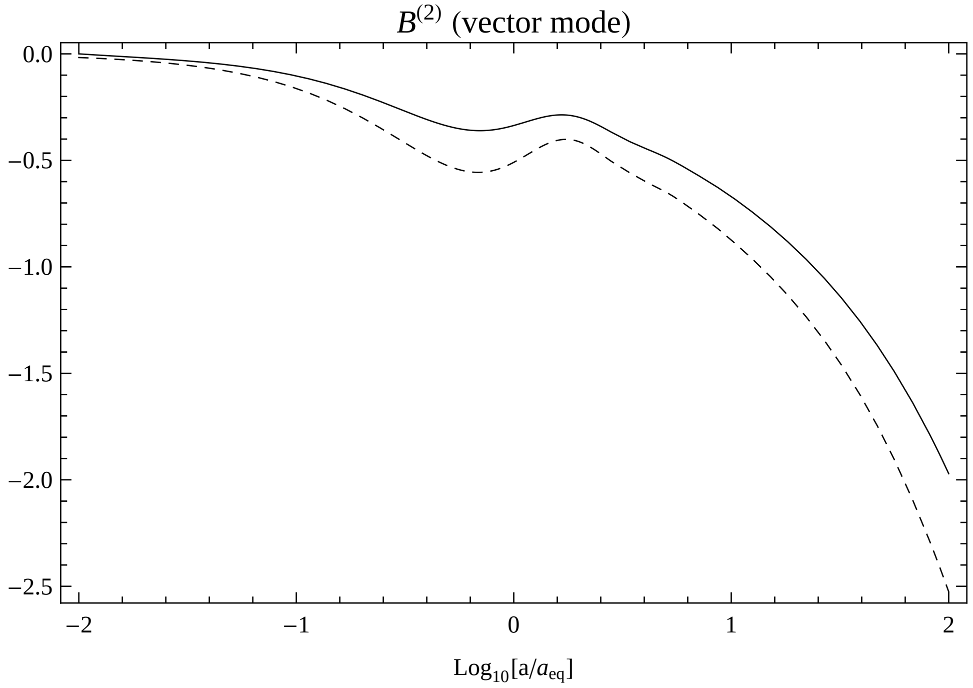


Vector modes

We plot the second order vector perturbations, together with its asymptotic value in matter dominated universe

```
In[35]:= MyPlotBandW[{VD2[CPL][k1t, k2t, angle][10^p], ConvV2[CPL][k1t, k2t, angle][10^p]},
          {p, -2, 2}, Frame → True, Axes → False(*,PlotRange→{0,0.00001}*),
          FrameLabel → "Log10[a/aeq]", PlotLabel -> "B(2)  (vector mode)"]
```

Out[35]=



```
In[36]:= ? V2
         ? ConvV2
```

```
Information::notfound : Symbol V2 not found. ≫
```

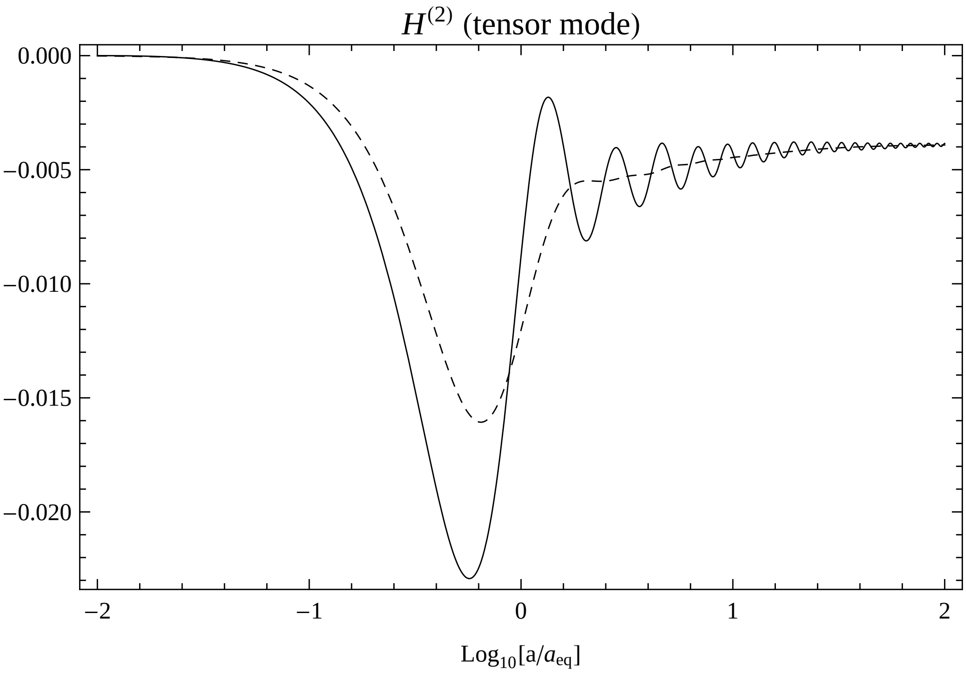
```
Exact solution of vector modes at
    second order in a matter dominated universe.
```

Tensor modes

We plot the second order tensor perturbations, together with its asymptotic value in matter dominated universe

```
In[38]:= MyPlotBandW[{T2[CPL][k1t, k2t, angle][10^p], ConvT2[CPL][k1t, k2t, angle][10^p]},
  {p, -2, 2}, Frame → True, Axes → False,
  FrameLabel → "Log10[a/aeq]", PlotLabel -> "H(2) (tensor mode)"]
```

Out[38]=



```
In[39]:= ? T2
         ? ConvT2
```

Second order Tensor modes $H^{(2)}$

```
Exact solution of Tensor modes at
    second order in a matter dominated universe.
```

Convergence tests

We know the form of the gravitational potential in the matter dominated era, for sub-Hubble modes.

We can thus check the convergence. We have several functions to assess the convegence. Click on the table for exact definitions

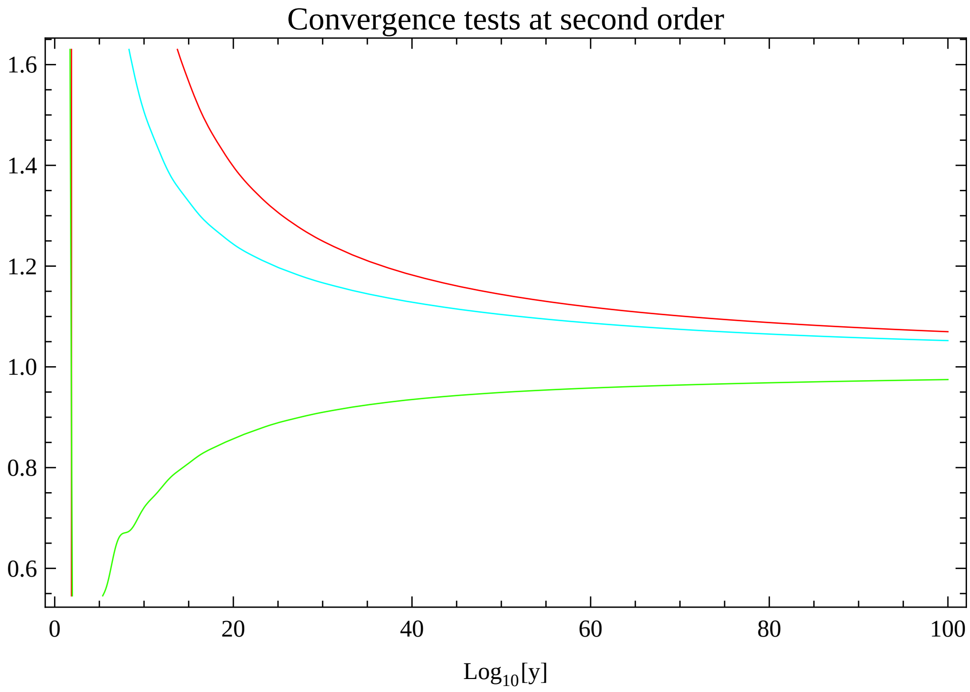
```
In[41]:= ?Test*
```

```
▼ CMBquick`CMBquick2`

TestConvDm    TestConvPhi    TestConvPsi    TestPoisson1    TestPoisson2
```

```
In[42]:= MyPlotColors[{TestPoisson2[CPL][k1t, k2t, angle][u],
           TestConvPhi[CPL][k1t, k2t, angle][u], TestConvPsi[CPL][k1t, k2t, angle][u]},
          {u, 1, 100}, Frame → True, Axes → False, FrameLabel → "Log10[y]",
          PlotLabel -> "Convergence tests at second order"]
```

Out[42]=



Primordial bispectrum

(Full Sky method implemented in CMBquick1)

We first load the previously stored Bessel functions

```
In[43]:= LoadAndGenerateBesselsBinary[2000];
```

```
Previously, Bessel functions were computed up to lmax=2500

No need to compute other Bessel functions
```

We choose the range of l-values

```
In[44]:= lmin = 4;
         lmax = 1000;
         Listls = Select[ListlUsedinBesselFunction, # <= lmax &];
```

We choose to multiply the resulting bispectra by this function in ordet to obtain a result of order unity

```
In[47]:= FactorMultiplicatif[l_] := l^2 * (l + 1)^2 / (2 Pi)^2 * 10^16;
```

The function in the case of equilateral bispectum is just bLLL and is defined in CMBquick1. Otherwise the function is bL1L2L3. We can check that and obtain information:

```
In[48]:=  ?bL*L*
```

```
▼ CMBquick`CMBquick1`
    bL1L2L3                              bLLL
```

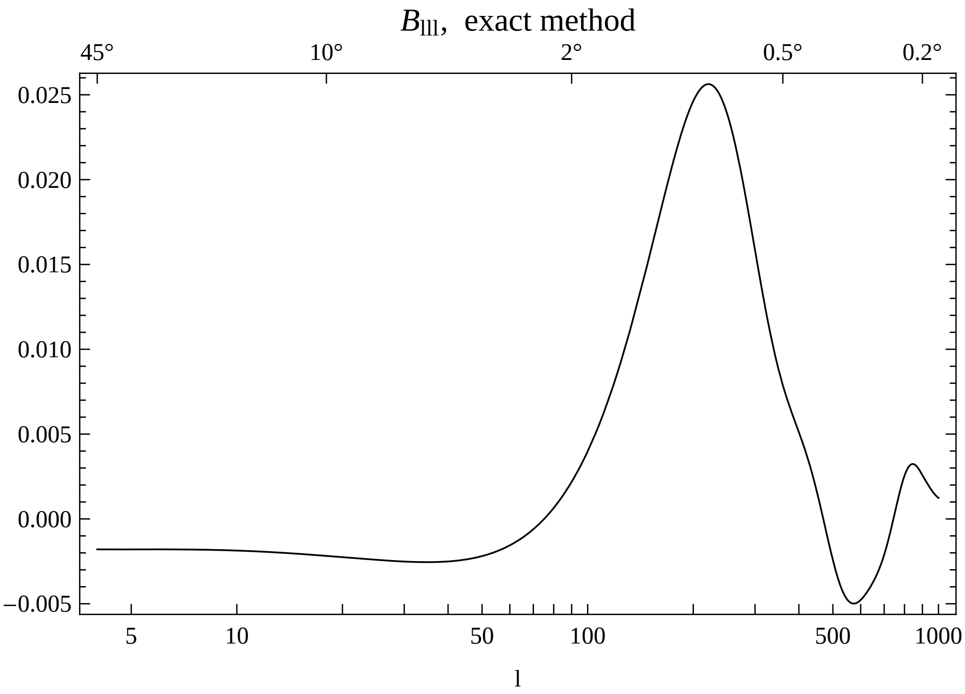
And we store the bispectra in a list by defining the following rules:

```
In[49]:=  Clear[bLLLlistI]
          bLLLlist[cpl_, el_] :=
            Table[{l, FactorMultiplicatif[l] bLLL[cpl, el][l]}, {l, Listls}];
          bLLLlistI[cpl_, el_] := bLLLlistI[cpl, el] = Interpolation[bLLLlist[cpl, el]]
```

We then plot the result

```
In[52]:=  MyLogLinearPlotBandW[{-bLLLlistI[CPL, EL][l]},
           {l, lmin, lmax}, PlotStyle → {{GrayLevel[0], Thickness[0.002]}},
           FrameLabel → {"l"}, PlotLabel -> "Blll,  exact method",
           Frame → True, Axes -> False, FrameTicks → MyTicks]
```

Out[52]=



Flat Sky Method

We select the directory where stored files are located (You should create it if it does not exits or choose something else, or just ignore and work in your current directory).

```
In[53]:=  SetDirectory["~/CMB"];
```

We now compute the bispectrum on the list of l' s in the flat sky method

```
In[54]:= IncludeEarly = True;

         bLLLFSlist[cpl_] := Table[
            {l, FactorMultiplicatif[l] BispectrumfNLone[cpl, FlatSky2, Local][l, l, -0.5]},
            {l, Listls}];
         bLLLFSlistI[cpl_] := bLLLFSlistI[cpl] = Interpolation[bLLLFSlist[cpl]]
```

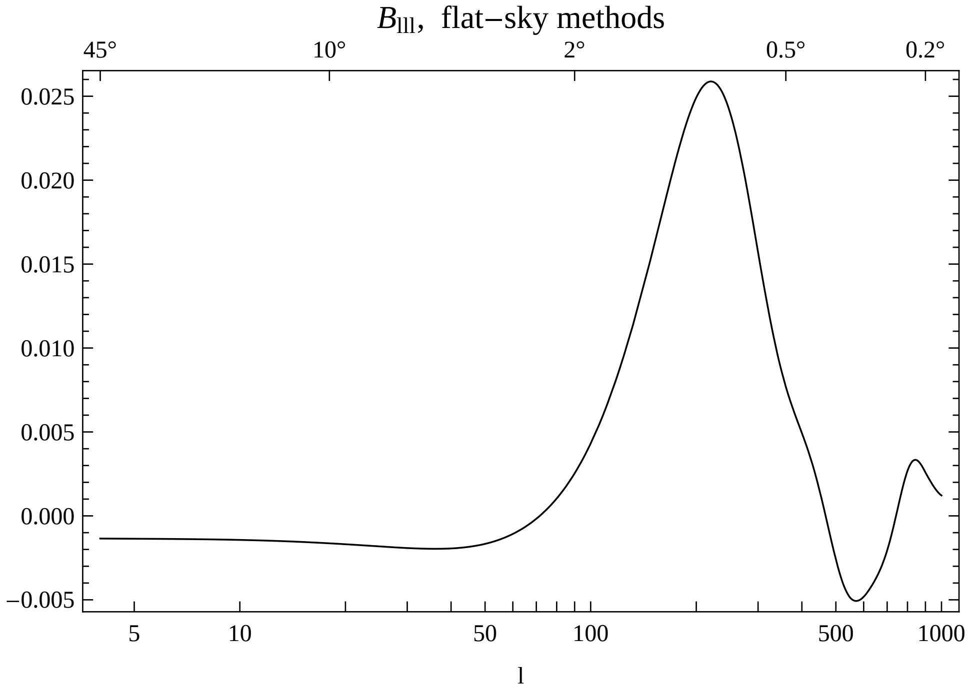
```
In[57]:= ?BispectrumfNLone
```

fNL primordial in the Flat Sky approximation. Syntax is BispectrumfNLone[cpl,FlatSkymethod,NGtype][l1,l2,m12], where m12 is the angle between the multipoles l1 and l2m and NGtype can be either 'Local' or 'Equilateral'. FlatSKyMethod has to be 'FlatSkyX' with X=1,2,3.

And we plot the result

```
In[63]:= MyLogLinearPlotBandW[{-bLLLFSlistI[CPL][l]},
          {l, lmin, lmax}, PlotStyle → {{GrayLevel[0], Thickness[0.002]}},
          FrameLabel → {"l"}, PlotLabel -> "Blll,  flat-sky methods",
          Frame → True, Axes -> False, FrameTicks → MyTicks]
```

Out[63]=



Bispectrum generated by non-linear effects, and other non-linear effects

We will not detail this part in the documentationm but we instead refer to the example notebook (located in CMBquick-/Examples in your installation directory).

Indeed this is not very user friendly, it is slow and not very general or robust. This is work and research in progress! (please contact me if you need more on that topic)

Obtaining information

You can obtain an interactive list of all functions publicly defined in CMBquick2 with the command `?"CMBquick`CMBquick2`*"`

You have to click on the name within the table below to obtain information on each variable. Documentation is not complete yet.

```
In[64]:= ?"CMBquick`CMBquick2`*"

▼ CMBquick`CMBquick2`

a2MagneticM1                   ELNoLate       Proddown21      SRLMIPP                  δNo2
a2MagneticM1ChgFrame           Eo2            Produp12        StoreSecondOrderResults  δRo2
a2MagneticM1dV                 Equilateral    Produp21        Sys2                     ζ2
a2MagneticM1Pi                 F0o2           PUB             T2                       Φ2
a2MagneticM1pure2              F1o2           Pure2Early      T2p                      Φ2Asymptotic
AllEarly                       fNL            Pure2LSS        Temperature              Φp2
AllLSS                         FNLshape       QuadraticEarly  Tensor                   Ψ2
B0o2                           IncludeEarly   QuadraticLSS    TestConvDm               Ψp2
B1M1o2                         k              Ro2             TestConvPhi              $Accuracy2
B1o2                           k1uz           Rot1            TestConvPsi              $AdaptivePrecision2
BispectreCorrectifTemperature  k2uz           Rot2            TestPoisson1             $CMBquick1VersionExpected
BispectreRSISW                 krmaxl1        RSISW           TestPoisson2             $Doppler2
BispectreRSISWl1l2l3           krmaxl2        S1Keitaro       TotalBispectrum          $dxeInSources
BispectrumEvolutionAllYTT      krsko          S1OverH2        TransferB2               $early2
BispectrumEvolutionM           k$             S2Keitaro       TransferB2ChgFrame       $IntegrateM0
```

BispectrumEvolutionTot
BispectrumEvolutionTotM
BispectrumfNLone
BispectrumPartialfNLone
BispectrumPartielEvolution
BispectrumPure2
BispectrumPure2M
BispectrumQuadratic
BispectrumQuadraticM
BispectrumType
Bo2
C0o2
C1M1o2
C1o2
Cll2M
ClM
CMBquick2
ConvT2
ConvV2
CorrectionTemperature
Distorsions
ELall
l3ofl1l2
LastLIntrinsic
LastLSources
Listmu
LoadedQlBool
LoadSecondOrderResults
Local
MagneticField
mdek1k2k3
mdek1k2k3true
mu3del1l2l3
mu3del1l2l3exact
mumin
Ncoskr
No2
Nth
PointsIntegrationNumerique2
PointsIntkrBispectrum
PointsIntkrBispectrumEvol
Precisionk2
PrimordialBispectrum
Proddown12
S2OverH2
S3
S4OverH
SB0
SB1M
SBLM
SC0
SC1M
Scalar
SElecdV
SElecforpure2
SElecPi
SELM
SF0
SF1M
SlmY
SNLM
SRI
SRIF
SRIY
SRLM
SRLMBetter
TransferB2dV
TransferB2Pi
TransferB2pure2
TransferT2
TransferT2Early
TransferT2inFS
TransferT2Quad
TransferT2QuadEarly
TransferY2
TransferY2Quad
TransferY2QuadRei
TransferY2QuadReiIlglg
TransferY2Rei
VarianceT
VD2
Vector
VND2
VNo2
VRo2
Y
yi2
YTT
$IntegrateM1
$IntegrateM2
$IntegrateMagnetic
$IntegrateSources
$IntegrateSpectralDistorsions
$IPPSources
$ISW2
$ISWInSources
$krCappedBool
$late2
$LensingInSources
$LoadDistorsionsFinished
$LoadMagneticFinished
$LoadTemperatureFinished
$lss2
$MaxNLSSWidthBool
$OrderInterpolation
$Quadrupole2
$SignQuadraticTerms
$StiffnessSwitching2
$SW2
$Version