Exercise 2 (hard)

Toby Wiseman (Imperial College)

Mathematica Summer School, Porto 2014

AIM : The aim of this exercise is to solve for 4-d (Euclidean) static AdS-CFT bulk metrics where the boundary is S^1*S^2, but the S^2 is deformed preserving a U(1). For an undeformed round S^2 the bulk solution we are interested in is global AdS-Schwarzschild, but deforming the S^2 it no longer is.

[This exercise is uses the finite differencing from Excerise 1 as we shall see.]

WARNING : This is intended to be a harder exercise. Only attempt after completing and understanding Exercise 1.

The first part of this notebook is to compute the (Harmonic) Einstein equations. There are many packages freely available for Mathematica to do this. The one below is a simple one I wrote. Feel free to use it, but it certainly is not as complete as those that are publically available - for example; Ricci.m, diffgeo.m (by Matt Headrick), GRTensorM.m ....

You just run this to generate the Harmonic Einstein equations. Then the problem is to build code that numerically solves these using the methods from Exercise 1.

Harmonic Einstein equations - RUN THIS SUBSECTION (note that this takes a few moments to complete)

The following subsection derives the Einstein equations....

GRSetup2.nb

Trace, Outer Product and Covariant Derivative! - GREinstein.nb

Buildmetric

GRrun2.nb

Harmonic Einstein equations

```
T[x_, y_] = 1;

A[x_, y_] = 0;

B[x_, y_] = 1;

F[x_, y_] = 0;

S[x_, y_] = 1;
```

Create reference connection Γ0

```
Γ0 = Simplify[Γ];
```

```
Clear[T, A, B, F, S]
```

Difference the connections Γ and the reference connection

```
dΓ = {{d, d, u}, Γ[[2]] - Γ0[[2]]};
```

Create the ξ vector by contracting this difference

```
ξ = trT[dΓ, {{1, 2}}];
```

Take the covariant derivative $\nabla\xi$

```
dξ = covDT[ξ];
```

And symmetrize it

```
symdξ = {{d, d}, 1/2 (dξ[[2]] + Transpose[dξ[[2]]])};
```

Construct the harmonic Einstein equations

```
EinEq = R2[[2]] + 3 Gd - symdξ[[2]];
```

Construct the norm $\xi.\xi$

```
ξsqr = trT[outerT[ξ, ξ], {{1, 2}}];
```

Put the non-vanishing components of the Einstein equations into a list, EinList.

```
EinList =
  {EinEq[[1, 1]], EinEq[[2, 2]], EinEq[[3, 3]], EinEq[[2, 3]], EinEq[[4, 4]]};
```

PROBLEMS

The list ' EinList ' now contains the five harmonic Einstein equations for negative cosmological constant.

We have used the metric ansatz;

$$ds^2 = \frac{1}{x^2\ (2-x)^2}\left((1-x)^2\, P[x]\, T[x,y]\, dt^2 + \frac{4}{P[x]}\left(T[x,y] + A[x,y]\ (1-x)^2\right) dx^2 + B[x,y]\, dy^2 + 2\ (1-x)\ F[x,y]\, dx\, dy + Q[y]\, \mathrm{Sin}[y]^2\, S[x,y]\, d\phi^2\right)$$

where $P[x] = 4 - 5\ (1-x)^2 + 2\ (1-x)^4$

and where t is Euclidean time, and ϕ is an angle.

This ansatz involves the function Q[y] which will be fixed.

The reference metric is chosen to be that above for T = B = S = 1 and A = F = 0. We will also use this as an initial guess for a solution.

The computational domain is x ϵ (0, 1), y ϵ (0, 2 π).

For smooth T, A, B, F, S we want x = 0 to be the AdS conformal boundary, and we require T = B = S = 1 and A = F = 0 there to obtain the boundary metric;

ds^2 = dt^2 + dy^2 + Q[y] Sin[y]^2 dϕ^2

We indeed see this is (Euclidean time)*(round sphere) only for Q=1.

The other x boundary at x = 1 is the (Euclidean) ' horizon' in the bulk where the S^1 of (Euclidean) time smoothly caps off. For our (carefully designed!) ansatz, at the horizon we require for regularity that the metric funcions T, A, B, F, S are all simply smooth functions of (1 - x)^2. Hence regularity requires that these functions are reflection symmetric about x = 1.

Since the y coordinate is an azimuthal coordinate for the (not necessarily round) S^2 we then require that all functions are periodic in y.

Q[y]=1 should give a bulk solution which is AdS-Schwarzschild with T=B=S=1 and A=F=0. For Q[y]≠1 we must solve for T,A,B,F,S with the above boundary conditions to find the solution.

Note that Q[y] must also be periodic in y, and for regularity at y=0, it must be even in y there. Note also that while ϕ is an angular coordinate, it need not have period 2π - if Q[0] ≠ 0 then it won't.

And example non-trivial Q[y] we take later is Q[y] = 1 + 0.5 Cos[y]

IMPORTANT : We now see that the function space is precisely the same as that we constructed in Exercise 1. Hence we can use all that code to discretize these Einstein equations.

In addition the code above computes ξsqr = $\xi.\xi$ which should be zero for a solution to the Einstein equations.

Problem 1 - the ansatz

Check that for Q = 1, then T = B = S = 1 and A = F = 0 is indeed a solution of both the Einstein equations;

(R2[[2]] + 3 Gd)

and also the harmonic Einstein equations

R2[[2]] + 3 Gd - symdξ[[2]]

You can see the metric using the command; Dt[ind].Gd.Dt[ind]

Show that this metric can be written;

$$ds^2 = h[r]\, dt^2 + \frac{dr^2}{h[r]} + r^2 \left(d\theta^2 + \mathrm{Sin}[\theta]^2\, d\phi^2 \right)$$

for $h[r] = r^2 + 1 - \frac{2}{r}$. This is the usual form for global AdS - Schwarzschild. In particular you should find the relation between (r, θ) and (x, y).

Problem 2 - discretizing the equations and a check that Q=1 is solved by AdS-Schwarschild

Fix Q[y] = 1 and then use the discretization scheme in Exercise 1 to build a discretized harmonic Einstein equation list for the interior points of the ' grid' list. The main difference is that in Exercise 1 there was just one equation and one function f. Now at each point we have 5 equations (the elements of EinList) and these depend on the 5 functions T, A, B, F, S.

For example, instead of just fval, you will need Tval, Aval, Bval, Fval and Sval to store the function values at the grid points.

Likewise instead of just dxfval, you will now need lists dxTval, dxAval, dxBval, dxFval, dxFval etc ... and similarly for the other derivatives.

Start with reasonable resolution, **NX = 8, NY = 8**, and use **InterpolationOrder -> 4**

Once the list is built show that then setting the values for T, A, B, F, S so that T = B = S = 1 and B = F = 0 the equations are indeed satisfied (to machine precision) at all interior grid points.

Problem 3 - solve for a non-trivial Q

Now choose a non - trivial function **Q = 1 + 0.5 Cos[y]**

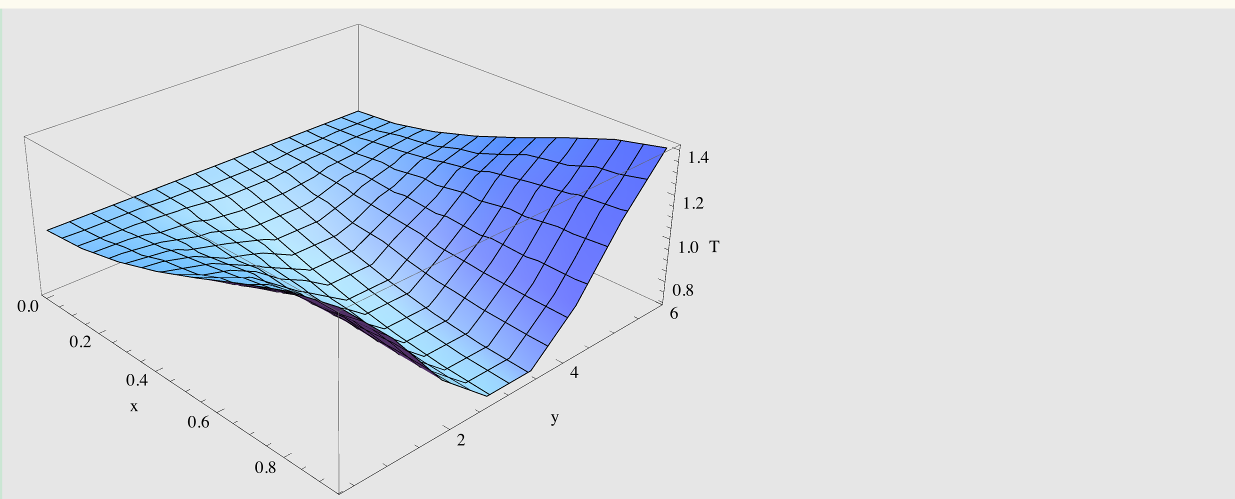
The bulk solution is now not simply AdS - Schwarzschild.

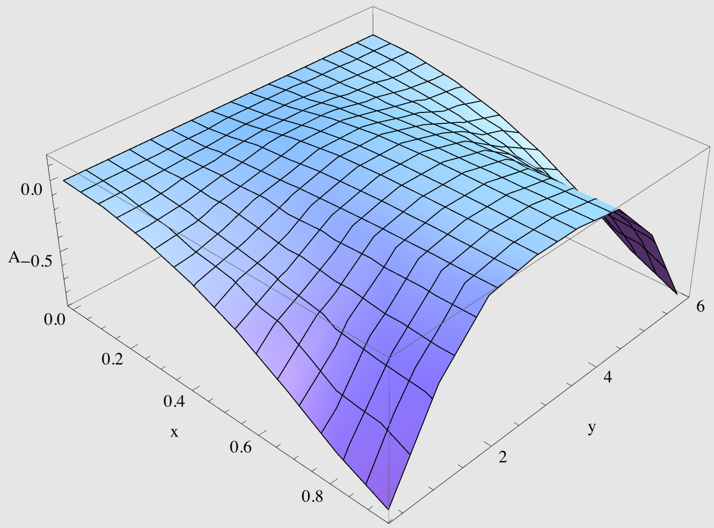
Do the same as in the previous problem to discretize the equations.

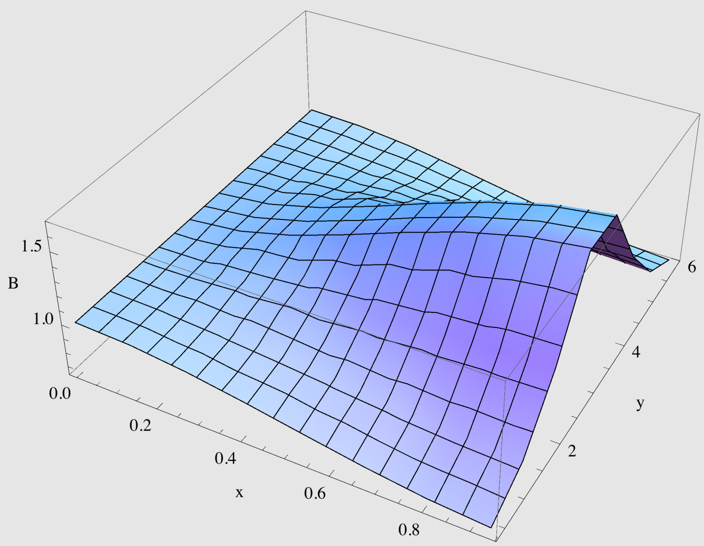
Now we will take T = B = S = 1 and B = F = 0 as an initial guess for our solution - obviously it obeys all the boundary conditions.

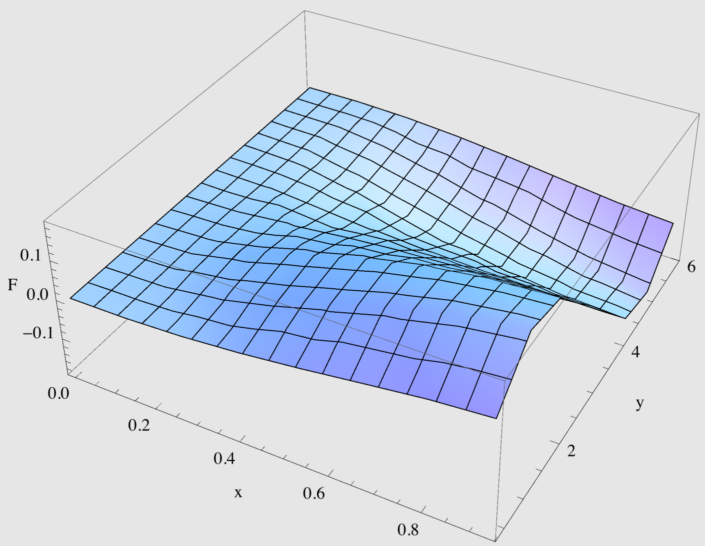
Then use the same method as in Exercise 1 to use FindRoot to solve the harmonic Einstein equations.

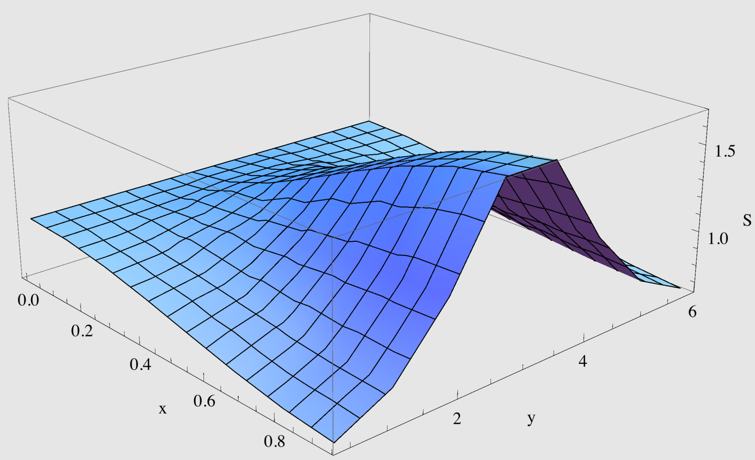
Plot the various metric funtions. For NX = NY = 8 you should obtain a solution looking like;




0.0
A
−0.5
0.0
0.2
0.4
x
0.6
0.8
2
y
4
6

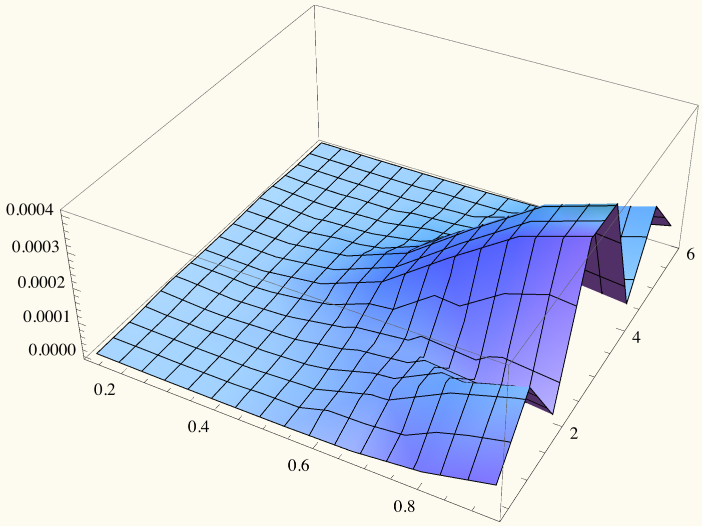

1.5
B
1.0
0.0
0.2
0.4
x
0.6
0.8
2
y
4
6





Problem 4 - examine $\xi.\xi$

Modify your FindRoot routine to plot ξsqr at each step. By construction it is zero for the initial guess (since this is the reference metric), then increases, then should decrease to be small in the final solution. For NX = NY = 8 and InterpolationOrder -> 4 and the above Q you should find;



You can experiment with changing resolution/interpolation order to see how the quality of the solution changes.