AnyBubble Manual

Manual version 2.01 corresponding to AnyBubble version 2.0

Introduction

The AnyBubble package finds a solution to the coupled differential equations

$$\frac{d^2\phi_i}{dr^2} + \frac{D-1}{r}\frac{d\phi_i}{dr} = \frac{\partial U}{\partial \phi_i} \ . \tag{1}$$

where $\{\phi_i\}$ is a set of fields with a potential $U(\phi_1, \phi_2, \ldots)$ and D is the number of dimensions of spacetime. The solution must start for $r = 0$ near a specified true vacuum in field space with $d\phi_i/dr = 0$ and asymptotically approach a specified false vacuum as $r \to \infty$. For more details see [1].

Loading the code

AnyBubble is a Mathematica package consisting of two files, `anybubble.m` and `powell.m`. You can load these files with the `<<` command. If you load just `anybubble.m` it will load `powell.m` if it has not been loaded. For this to work, `powell.m` must be on your Mathematica path.

FindBubble

The package has a single function, `FindBubble`, which has the following structure:

```
FindBubble[potential, field, tv, fv]
```

Here `field` is the name of the field (which should not have any definitions in Mathematica). The potential is a function of the individual components of the field, which are denoted as in function evaluation, `field[1]`, `field[2]`, `field[3]`, ... The parameters `tv` and `fv` are vectors giving the locations of the true and false vacuum, respectively. For example, you might write

```
FindBubble[f[1]^4 - 14 f[1]^2 + 24 f[1], f, {-3}, {2}]
```

Note that even if you have only one field, you must still use notation such as `f[1]` and the vacua must be specified as vectors.

The function `FindBubble` returns a vector `{action, profile}` where `action` is the Euclidean action,

$$S[\phi] = A_{D-1} \int_0^\infty dr\, r^{D-1} \left[\frac{1}{2} \partial_r \phi \partial_r \phi + U(\phi) - U(\mathrm{fv}) \right] , \tag{2}$$

with A_{D-1} the area of the unit $(D-1)$-dimensional sphere, and `profile` is a function that gives the value of the field as a function of r.

Suppose you want to find the bubble solution and action for the potential

$$U(\phi_1, \phi_2) = \sin(\phi_1 - \phi_2) + \frac{1}{2}\cos(\phi_1 + \phi_2) + \cos 3(\phi_1 + \phi_2) + 2\cos 3(\phi_1 - \frac{\phi_2}{2}) \tag{3}$$

This potential is shown in Fig.1. This potential has many minima including two (red and blue dots) at

$$\begin{aligned} \mathrm{min}_1 &= \{2.39338, 2.82768\} , \\ \mathrm{min}_2 &= \{4.56086, 2.81235\} . \end{aligned} \tag{4}$$

Here min_1 has a lower potential than min_2 and can be regarded as a "true" vacuum for transitions from min_2. The code for finding the bounce action and profile is

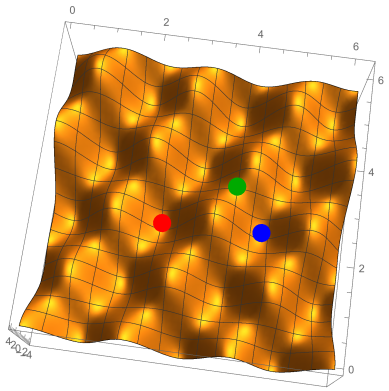

Figure 1: An example of a two-field potential.

```
result = FindBubble[Sin[phi[1] - phi[2]] + Cos[phi[1] + phi[2]]/2
             + Cos[3phi[1] + 3phi[2]]+ 2 Cos[3 phi[1] - (3/2) phi[2]],
           phi, {2.39338, 2.82768}, {4.56086,2.81235}]
```

The result is a vector such as

```
{30668.1, AnyBubble‘Private‘profile$10650}
```

where the second element is a function that can be evaluated at a point r. As an example, to see the profiles of the two fields, we write

```
Plot[result[[2]][r], {r, 0, 20}]
```

The profile function operates only on real-valued arguments. Thus, in the example above, `result[[2]][r]` returns `AnyBubble‘Private‘profile$10650[r]` without further evaluation. This means that `result[[2]][r][[1]]` returns the symbol `r`, because that is part 1 of `AnyBubble‘Private‘profile$10650[r]`. To define a function that returns a specific component of the profile, set it up to operate only on real-valued arguments, with syntax such as

```
profile1[r_/;Element[x,Reals]] := result[[2]][r][[1]]
```

Options

`FindBubble` has a number of options, which we explain here.

InitialProfilePoints

This option accepts a list of intermediate points $\{p_1, p_2, \ldots\}$ in order from the true toward the false vacuum. `FindBubble` will use in its initial guess for the field profile a smooth path going through these points. If there are several possible solutions, this gives you control over which one `FindBubble` finds, although it is not guaranteed to work: the solution-finding process may still find a solution differ from the one you intended. It also will generally improve the chance of success and decrease the runtime.

As an example, the potential in Fig.1 has a ridge at $\phi = \{3.91369, 3.50186\}$ (green dot). It makes sense that the profile gets close to this point. You can specify this by the option `InitialProfilePoints -> {{3.91369,3.50186}}`. If you do not specify `InitialProfilePoints`, `FindBubble` will use a straight line between the two vacua in the initial profile.

MaxIntervalGrowth

As explained in [1], the multi-shooting method avoids the instabilities typical to shooting problems. To do this, it attempts to make the intervals small enough so that the field will not grow in any one interval by more than `MaxIntervalGrowth`. (However, `FindBubble` only considers the rate of growth near the two vacua, so this limit is not strictly applied.)

If the code has trouble finding the solution, it may help to decrease the value of `MaxIntervalGrowth`, which will cause `FindBubble` to divide the interval into more pieces. The default value of `MaxIntervalGrowth` is 30.

StartAnalyticFraction

`FindBubble` uses analytic solutions for a quadratic approximation of the potential near the center of the bubble. This is used in a sphere of maximum radius `StartAnalyticFraction` times the distance between the false and true vacua in field space. Thus the the solution that you get is the solution to a slightly different problem than one you give, because the potential has been approximated. Reducing `StartAnalyticFraction` will increase the accuracy of the result by using approximations over a smaller region, but is likely to make the parameterization of the problem worse and so lead to slow convergence.

The default value is 0.01.

EndAnalyticFraction

This is similar to `StartAnalyticFraction` but applies to the region around the false vacuum, i.e., for large r.

The default value is 0.01.

MaxIterations

`FindBubble` uses Powell's hybrid method [2] to search for the bounce solution. If it takes more than `MaxIterations` steps, the Powell's method code will give up. The default is 500, but if you find that the code gets close to finding the solution, but needs more tries to find it, you can increase this option.

MaxReadjustments

Whenever the search for the solution of the differential equation reaches profiles which do not obey the constraints given by `StartAnalyticFraction` and `EndAnalyticFraction`, `FindBubble` readjusts the points that divide the intervals used for shooting. This option sets the maximum number of such readjustments. After each readjustment, the Powell's method procedure starts again, so the total number of steps taken could in principle be as large as `MaxIterations` times `MaxReadjustments`.

The default value is 20.

WorkingPrecision

The number of digits of precision to use for calculations in `FindBubble`. For this to be useful, your true and false vacua and your potential should be specified with at least as much precision. Even then, the quality of the result will be limited by `StartAnalyticFraction` and `EndAnalyticFraction`.

Verbose

If you set this option to True, `FindBubble` will print verbose output showing all of the steps in the shooting process.

PowellVerbosity

This is passed as the `Verbosity` option to the Powell's method code, causing it to produce a log of what it is doing. Possible values are

0 No output.
1 Print a dot for each step taken (the default).
2 One line for each step taken.
3 Everything but the Jacobian.
4 Everything.

SpaceTimeDimension

This is the number of spacetime dimensions D. It appears only in the coefficient of the first-order term in Eq. (1) and in the calculation of the action, Eq. (2). The default is 4.

AccuracyGoal

This is the number of digits of accuracy sought in the Powell's method calculation. If this is d, the calculation attempts to make the total of the squares of the differences between the field values and between their derivatives at the joining points of the shooting regions less than 10^{-d}.

The default value is 7. For simple cases with few fields, 8 is usually achievable. For a very large number of fields, you may have to decrease this further.

Parallel Usage

It should be possible to use AnyBubble with the with the parallelization facilities of Mathematica, such as `ParallelTable`. In most cases nothing special is required, but if you find that Mathematica does not pass on the AnyBubble code to the parallel kernels, you may need to supply the `DistributedContexts` option including `"AnyBubble‘"` and `"PowellHybrid‘"`.

Closing remarks

We hope that AnyBubble will meet your needs for finding tunneling instantons. If you discover problems, please feel free to email Ken Olum at kdo@cosmos.phy.tufts.edu. In particular, if you encounter potentials that this code cannot solve, please send us an example and perhaps we will be able to help.

The development of this code was funded in part by the National Science Foundation under grant 1213888 and 1518742.

References

[1] Ali Masoumi, Ken D. Olum, and Benjamin Shlaer. Efficient numerical solution to vacuum decay with many fields. Forthcoming.

[2] M. J. D. Powell. A hybrid method for nonlinear equations. In Philip Rabinowitz, editor, *Numerical methods for nonlinear algebraic equations*, chapter 6, pages 87–114. Gordon and Breach Science Publishers, New York, 1970.