Software Spec – NumerovRadialIntegrator Class

# Background

Numerov’s method is useful for solving DEs of the form:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Schrodinger’s equation is of the form:

|  |  |  |
| --- | --- | --- |
|  |  |  |

The numerical solution of this equation can be obtained using the following formulas:

## Uniform Mesh

Formulas for use on a uniform mesh:

|  |  |  |
| --- | --- | --- |
| Forward |  |  |
| Backward |  |  |

## Nonuniform Mesh

If the interpoint spacing is non-uniform such that

|  |  |  |
| --- | --- | --- |
| OLD VERSION, Forward |  |  |
| OLD VERSION, Backward |  |  |
| NEW VERSION, Forward |  |  |
| NEW VERSION, Backward |  |  |

Need to unit test the old vs. the new versions against a known (analytic) solution and determine which one is correct.

## Starting the Integration

### Forward integration

Given the initial conditions y(0) and dy/dr(0), y1 can be obtained as follows:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Since, for integration of the radial Schrodinger equation, y0 is defined to be 0, this expression simplifies to:

|  |  |  |
| --- | --- | --- |
|  |  |  |

### Backward integration

Given the initial conditions y(rmax) and dy/dr(rmax), y-1 can be obtained as follows:

|  |  |  |
| --- | --- | --- |
|  |  |  |

# Implementation

Class NumerovIntegrator

Public Shared Function Integrate(mesh as IRadialMesh, v as double(), istart as integer, iend as integer, ystart as double, yprimestart as double) as Double()

* Exceptions:
* Mesh is Nothing 🡪 Throw ArgumentNullException
* Mesh has fewer than 3 points 🡪 Throw ArgumentException
* Dimensions of V is different than Mesh 🡪 Throw ArgumentException
* If iStart or iEnd out of range of Mesh dimensions, 🡪 Throw InvalidArgumentException
* Ystart or yprimestart = NaN, MAXVALUE or MINVALUE 🡪 Throw InvalidArgumentException
* If iStart > iEnd, integration is in the backward direction.

Public Shared Function StepForward(a as double, b as double, y0 as double, yminus as double, vminus as double, v0 as double, vplus as double) as Double

* Branches internally depending on the value of a and b.

Public Shared Function StepBackward(a as double, b as double, y0 as double, yplus as double, vminus as double, v0 as double, vplus as double) as Double

* Branches internally depending on the value of a and b.

NOT NEEDED

Structure NumerovIntegrationResult

Sub New(y as Double(), yPrime as Double(), istart as integer, iend as integer)

Public Readonly Property y as Double()

Public Readonly Property yPrime as Double()

Public Readonly Property iStart as Integer

Public Readonly Property iEnd as Integer