

```

1 Imports System.Math
2
3 Public Class Masses
4     Dim r0 As Double = 3.5           'kpc
5     Dim v0 As Double = 0.256        '250 km/s -> kpc/Myr
6     Dim rmax, tau, gamma, rscale, off As Double 'kpc,Myr,Myr
7
8     Dim galaxy(,), sne(,) As Double
9
10    'Dim sf As New SpecialFunctions
11    Dim op As New Operations
12    Dim mop As New MatrixOperations
13
14    Dim x() As Double
15    Dim y() As Double
16    Dim y2() As Double
17    Dim yout As Double
18    Dim rand As New Random
19
20    Function Omega_star(ByVal r As Double) As Double
21        If r < 1 Then
22            Return v0 / 1
23        Else
24            Return v0 / r
25        End If
26    End Function
27
28    Function Omega(ByVal r As Double) As Double
29        Return v0 / r0 * (1 - 1 / Sqrt(2)) - Omega_star(r)
30    End Function
31
32    Function phi(ByVal r As Double) As Double
33        Return Exp(-r / r0)
34    End Function
35
36 #Region "Spline"
37 Private Sub spline(ByVal x() As Double, ByVal y() As Double, ByVal n As Integer, ByVal yp1 As
Double, ByVal ypn As Double)
38     Dim i, k As Integer
39     Dim p, qn, sig, un, u() As Double
40
41     ReDim y2(n)
42     ReDim u(n)
43
44     If yp1 > 9.9E+29 Then
45         y2(0) = u(0) = 0.0
46     Else
47         y2(0) = -0.5
48         u(0) = (3.0 / (x(1) - x(0))) * ((y(1) - y(0)) / (x(1) - x(0)) - yp1)
49     End If
50
51     For i = 1 To n - 1
52         sig = (x(i) - x(i - 1)) / (x(i + 1) - x(i - 1))
53         p = sig * y2(i - 1) + 2
54         y2(i) = (sig - 1) / p
55         u(i) = (y(i + 1) - y(i)) / (x(i + 1) - x(i)) - (y(i) - y(i - 1)) / (x(i) - x(i - 1))
56         u(i) = (6 * u(i) / (x(i + 1) - x(i - 1)) - sig * u(i - 1)) / p
57     Next i
58
59     If ypn > 9.9E+29 Then
60         qn = un = 0.0
61     Else
62         qn = 0.5
63         un = (3.0 / (x(n) - x(n - 1))) * (ypn - (y(n) - y(n - 1)) / (x(n) - x(n - 1)))
64     End If
65
66     y2(n) = (un - qn * u(n - 1)) / (qn * y2(n - 1) + 1)
67     For k = n - 1 To 1 Step -1
68         y2(k) = y2(k) * y2(k + 1) + u(k)
69
70     Next
71 End Sub
72
73 Private Function splint(ByVal xa() As Double, ByVal ya() As Double, ByVal y2a() As Double,
ByVal n As Integer, ByVal x As Double)
74     Dim klo, khi, k As Integer

```

```

75     Dim h, b, a As Double
76
77     klo = 0
78     khi = n
79     Do While khi - klo > 1
80         k = khi + klo >> 1
81
82         If xa(k) > x Then
83             khi = k
84         Else
85             klo = k
86         End If
87     Loop
88     h = xa(khi) - xa(klo)
89     a = (xa(khi) - x) / h
90     b = (x - xa(klo)) / h
91     yout = a * ya(klo) + b * ya(khi) + ((a ^ 3 - a) * y2a(klo) + (b ^ 3 - b) * y2a(khi)) * (h ^ 2) / 6
92
93     Return yout
94 End Function
95 #End Region
96
97 Sub CreateGalaxy()
98     ReDim galaxy(tbPixels.Text, tbPixels.Text)
99
100    Dim csv As New CSVData
101    csv.LoadCSV("c:\hosting\grb\io\ch" & tbMetal.Text & ".csv", True)
102    Dim rows As Integer = csv.CSVDataSet.Tables(0).Rows.Count - 1
103    ReDim x(rows)
104    ReDim y(rows)
105    For i As Integer = 0 To rows
106        x(i) = csv.CSVDataSet.Tables(0).Rows(i).Item("Age (Myr)")
107        y(i) = csv.CSVDataSet.Tables(0).Rows(i).Item(tbBand.Text)
108    Next
109    csv.Dispose()
110
111    spline(x, y, rows - 1, 0, 0)
112
113    Dim iMax As Integer = tbPixels.Text
114    Dim jMax As Integer = iMax
115
116    ToolStripProgressBar1.Maximum = iMax
117    ToolStripProgressBar1.Minimum = 0
118    ToolStripProgressBar1.ForeColor = Color.Green
119
120    Dim ctr As Integer = tbPixels.Text / 2
121    Dim r, theta, L As Double
122    Dim k As Integer
123
124    For i As Integer = 0 To iMax 'rows
125        For j As Integer = 0 To jMax 'columns
126            r = Sqrt((i - ctr) ^ 2 + (j - ctr) ^ 2) * rscale
127            theta = Atan2(i - ctr, j - ctr)
128            L = 0
129
130            k = Round(Abs((1 / 3.1416) * (tau * Omega(1))), 0) + 1
131            For n As Integer = -k To k
132                Dim arg As Double = (off - theta / Omega(r) - 2 * n / 2 * 3.1416 / Omega(r))
133                If arg < 0 Or arg > tau Then
134                    L += 0
135                Else
136                    L += 10 ^ ((4.8 - splint(x, y, y2, (rows - 1), arg)) / 2.5) * phi(r) / Abs
137                End If
138            Next
139            galaxy(i, j) = L
140        Next
141        ToolStripProgressBar1.Value = i
142    Next
143 End Sub
144
145 Sub CreateMassCutMap()
146     ReDim sne(tbPixels.Text, tbPixels.Text)
147
148     Dim Mass1 As Integer = tbMin.Text

```

```

149 Dim Mass2 As Integer = tbMax.Text
150
151 Dim csv As New CSVData
152 csv.LoadCSV("c:\hosting\grb\io\msto" & tbMetal.Text & ".csv", True)
153 Dim dtSN As DataTable = csv.CSVDataSet.Tables(0)
154 Dim rows As Integer = dtSN.Rows.Count - 1
155 ReDim x(rows)
156 ReDim y(rows)
157 For i As Integer = 0 To rows
158     x(i) = dtSN.Rows(i).Item("Log Age")
159     y(i) = dtSN.Rows(i).Item("Mass")
160 Next
161
162 Dim coef(5) As Single
163 For i As Integer = 0 To 4
164     coef(i) = dtSN.Rows(i).Item("Coeffs")
165 Next
166
167 csv.Dispose()
168 dtSN.Dispose()
169
170 Dim iMax As Integer = tbPixels.Text
171 Dim jMax As Integer = iMax
172
173 ToolStripProgressBar1.Maximum = iMax
174 ToolStripProgressBar1.Minimum = 0
175 ToolStripProgressBar1.ForeColor = Color.Red
176
177 Dim ctr As Integer = tbPixels.Text / 2
178 Dim r, theta, dth, L, tauprime As Double
179 Dim k As Integer
180
181 For i As Integer = 0 To iMax
182     For j As Integer = 0 To jMax
183         r = Sqrt((i - ctr) ^ 2 + (j - ctr) ^ 2) * rscale
184         theta = Atan2(i - ctr, j - ctr)
185         dth = rscale / r 'need to know angular separation between pixels
186         If r = 0 Then dth = 1
187         L = 0
188         tauprime = (10 ^ op.LInterpolate(y, x, Mass1)) / 1000000 'reverse interpolate to
get time from min Mass
189
190         k = Math.Round(Math.Abs((1 / 3.1416) * (tauprime * Omega(1))), 0) + 1
191         For n As Integer = -k To k
192             Dim arg As Double = (off - theta / Omega(r) - 2 * n / 2 * 3.1416 / Omega(r))
193             Dim arg1 As Double = (off - (theta - dth) / Omega(r) - 2 * n / 2 * 3.1416 /
Omega(r))
194             Dim arg2 As Double = (off - (theta + dth) / Omega(r) - 2 * n / 2 * 3.1416 /
Omega(r))
195             arg1 = Math.Log((arg1 / 1.1) * 1000000, 10)
196             arg2 = Math.Log((arg2 / 1.1) * 1000000, 10)
197             If arg < 3 Or arg > 15000 Then
198                 L += 0
199             Else
200                 Dim m1 As Double = 10 ^ (coef(0) + coef(1) * arg1 + coef(2) * arg1 ^ 2 +
coef(3) * arg1 ^ 3 + coef(4) * arg1 ^ 4)
201                 Dim m2 As Double = 10 ^ (coef(0) + coef(1) * arg2 + coef(2) * arg2 ^ 2 +
coef(3) * arg2 ^ 3 + coef(4) * arg2 ^ 4)
202
203                 If m1 > Mass2 Then m1 = Mass2
204                 If m2 > Mass2 Then m2 = Mass2
205                 If m1 < Mass1 Then m1 = Mass1
206                 If m2 < Mass1 Then m2 = Mass1
207
208                 L += (1.1 / 1.3 * (m2 ^ -1.3 - m1 ^ -1.3) * phi(r) / dth)
209             End If
210         Next
211         If L > 0 Then
212             sne(i, j) = L
213         Else
214             sne(i, j) = 0
215         End If
216     Next
217     ToolStripProgressBar1.Value = i
218 Next
219

```

```

220 End Sub
221
222 Private Sub btSimulate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btSimulate.Click
223     rscale = tbRadius.Text / (tbPixels.Text / 2)
224     tau = tbTau.Text
225     rmax = tbRadius.Text
226     off = tbOff.Text
227
228     CreateGalaxy()
229     CreateMassCutMap()
230
231     mop.CreateColorImage(galaxy, "C:\users\cody\documents\uv project\Sims\galaxy.bmp")
232     mop.CreateColorImage(sne, "C:\users\cody\documents\uv project\Sims\sne.bmp")
233
234     PictureBox1.ImageLocation = "C:\users\cody\documents\uv project\Sims\galaxy.bmp"
235     PictureBox2.ImageLocation = "C:\users\cody\documents\uv project\Sims\sne.bmp"
236
237     Dim frdt As DataTable = Fruchter(galaxy, sne)
238
239     Plot(frdt)
240     op.ExportToCSV(frdt, "C:\users\cody\documents\uv project\sims\fruchter" & tbMin.Text & "-"
& tbMax.Text & ".csv")
241 End Sub
242
243 Sub Plot(ByVal dt As DataTable)
244     Dim fp As NPlot.PointPlot = New NPlot.PointPlot
245     Dim pp As NPlot.PointPlot = New NPlot.PointPlot
246     Dim lp As NPlot.LinePlot = New NPlot.LinePlot
247
248     Dim XDAT As ArrayList = New ArrayList
249     Dim YDAT As ArrayList = New ArrayList
250     Dim XDAT2 As ArrayList = New ArrayList
251     Dim YDAT2 As ArrayList = New ArrayList
252
253     XDAT.Clear()
254     YDAT.Clear()
255     XDAT2.Clear()
256     YDAT2.Clear()
257     PlotSurface2D1.Clear()
258     PlotSurface2D1.Title = "Fruchter Plot"
259     PlotSurface2D1.BackColor = Color.Empty
260
261     For Each row As DataRow In dt.Rows
262         XDAT.Add(row.Item("FracL"))
263         YDAT.Add(row.Item("FracS"))
264     Next
265
266     'For Each row As DataRow In dtp.Rows
267     '    XDAT2.Add(row.Item("FracL"))
268     'YDAT2.Add(row.Item("FracS"))
269     'Next
270
271     pp.AbscissaData = XDAT
272     pp.DataSource = YDAT
273     pp.Marker.Type = NPlot.Marker.MarkerType.FilledCircle
274     pp.Marker.Size = 1
275     pp.Marker.Color = Color.Red
276     'fp.AbscissaData = XDAT2
277     'fp.DataSource = YDAT2
278     'fp.Marker.Color = Color.Blue
279     'fp.Marker.Type = NPlot.Marker.MarkerType.Circle
280     'fp.Marker.Size = 8
281     lp.AbscissaData = XDAT
282     lp.DataSource = XDAT
283     lp.Color = Color.Green
284     PlotSurface2D1.Add(pp)
285     'PlotSurface2D1.Add(fp)
286     PlotSurface2D1.Add(lp)
287     PlotSurface2D1.XAxis1.Label = "Fraction of Light"
288     PlotSurface2D1.YAxis1.Label = "Fraction of SNe"
289     PlotSurface2D1.XAxis1.WorldMin = 0
290     PlotSurface2D1.XAxis1.WorldMax = 1
291     'PlotSurface2D1.YAxis1.WorldMin = 0
292     PlotSurface2D1.YAxis1.WorldMax = 1
293

```

```

294         PlotSurface2D1.Refresh()
295     End Sub
296
297     Function Fruchter(ByVal gal(,) As Double, ByVal trans(,) As Double)
298         Dim n As Integer = gal.GetLength(0)
299         Dim m As Integer = gal.GetLength(1)
300
301         Dim totL As Double = 0
302         Dim totS As Double = 0
303         Dim fracL As Double = 0
304         Dim fracS As Double = 0
305
306         Dim dtL As New DataTable
307         dtL.Columns.Add("i", System.Type.GetType("System.Int32"))
308         dtL.Columns.Add("j", System.Type.GetType("System.Int32"))
309         dtL.Columns.Add("L", System.Type.GetType("System.Double"))
310         dtL.Columns.Add("S", System.Type.GetType("System.Double"))
311         dtL.Columns.Add("FracL", System.Type.GetType("System.Double"))
312         dtL.Columns.Add("FracS", System.Type.GetType("System.Double"))
313
314         For i As Integer = 0 To n - 1
315             'random flattening goes here
316
317             For j As Integer = 0 To m - 1
318                 Dim newRow As DataRow = dtL.NewRow
319                 newRow.Item("i") = i
320                 newRow.Item("j") = j
321                 newRow.Item("L") = gal(i, j)
322                 newRow.Item("S") = trans(i, j)
323                 dtL.Rows.Add(newRow)
324
325                 totL += gal(i, j)
326                 totS += trans(i, j)
327             Next
328         Next
329
330         Dim dv As New DataView(dtL)
331         dv.Sort = "L"
332
333         For i As Integer = 0 To dv.Count - 1
334             dv(i).Item("FracL") = fracL / totL
335             dv(i).Item("FracS") = fracS / totS
336             fracL += dv(i).Item("L")
337             fracS += dv(i).Item("S")
338         Next
339         Return dtL
340     End Function
341 End Class

```