```
Imports System.Math

Public Class Masses
    Dim r0 As Double = 3.5            'kpc
    Dim v0 As Double = 0.256        '250 km/s -> kpc/Myr
    Dim rmax, tau, gamma, rscale, off As Double   'kpc,Myr,Myr

    Dim galaxy(,), sne(,) As Double

    'Dim sf As New SpecialFunctions
    Dim op As New Operations
    Dim mop As New MatrixOperations

    Dim x() As Double
    Dim y() As Double
    Dim y2() As Double
    Dim yout As Double
    Dim rand As New Random

    Function Omega_star(ByVal r As Double) As Double
        If r < 1 Then
            Return v0 / 1
        Else
            Return v0 / r
        End If
    End Function

    Function Omega(ByVal r As Double) As Double
        Return v0 / r0 * (1 - 1 / Sqrt(2)) - Omega_star(r)
    End Function

    Function phi(ByVal r As Double) As Double
        Return Exp(-r / r0)
    End Function

#Region "Spline"
    Private Sub spline(ByVal x() As Double, ByVal y() As Double, ByVal n As Integer, ByVal yp1 As Double, ByVal ypn As Double)
        Dim i, k As Integer
        Dim p, qn, sig, un, u() As Double

        ReDim y2(n)
        ReDim u(n)

        If yp1 > 9.9E+29 Then
            y2(0) = u(0) = 0.0
        Else
            y2(0) = -0.5
            u(0) = (3.0 / (x(1) - x(0))) * ((y(1) - y(0)) / (x(1) - x(0)) - yp1)
        End If

        For i = 1 To n - 1
            sig = (x(i) - x(i - 1)) / (x(i + 1) - x(i - 1))
            p = sig * y2(i - 1) + 2
            y2(i) = (sig - 1) / p
            u(i) = (y(i + 1) - y(i)) / (x(i + 1) - x(i)) - (y(i) - y(i - 1)) / (x(i) - x(i - 1))
            u(i) = (6 * u(i) / (x(i + 1) - x(i - 1)) - sig * u(i - 1)) / p
        Next i

        If ypn > 9.9E+29 Then
            qn = un = 0.0
        Else
            qn = 0.5
            un = (3.0 / (x(n) - x(n - 1))) * (ypn - (y(n) - y(n - 1)) / (x(n) - x(n - 1)))
        End If

        y2(n) = (un - qn * u(n - 1)) / (qn * y2(n - 1) + 1)
        For k = n - 1 To k >= 1 Step -1
            y2(k) = y2(k) * y2(k + 1) + u(k)

        Next
    End Sub

    Private Function splint(ByVal xa() As Double, ByVal ya() As Double, ByVal y2a() As Double, ByVal n As Integer, ByVal x As Double)
        Dim klo, khi, k As Integer
        Dim h, b, a As Double

        klo = 0
        khi = n
        Do While khi - klo > 1
            k = khi + klo >> 1

            If xa(k) > x Then
                khi = k
            Else
                klo = k
            End If
        Loop
        h = xa(khi) - xa(klo)
        a = (xa(khi) - x) / h
        b = (x - xa(klo)) / h
        yout = a * ya(klo) + b * ya(khi) + ((a ^ 3 - a) * y2a(klo) + (b ^ 3 - b) * y2a(khi)) * (h ^ 2) / 6

        Return yout
    End Function
#End Region

    Sub CreateGalaxy()
        ReDim galaxy(tbPixels.Text, tbPixels.Text)

        Dim csv As New CSVData
        csv.LoadCSV("c:\hosting\grb\io\ch" & tbMetal.Text & ".csv", True)
        Dim rows As Integer = csv.CSVDataSet.Tables(0).Rows.Count - 1
        ReDim x(rows)
        ReDim y(rows)
        For i As Integer = 0 To rows
            x(i) = csv.CSVDataSet.Tables(0).Rows(i).Item("Age(Myr)")
            y(i) = csv.CSVDataSet.Tables(0).Rows(i).Item(tbBand.Text)
        Next
        csv.Dispose()

        spline(x, y, rows - 1, 0, 0)

        Dim iMax As Integer = tbPixels.Text
        Dim jMax As Integer = iMax

        ToolStripProgressBar1.Maximum = iMax
        ToolStripProgressBar1.Minimum = 0
        ToolStripProgressBar1.ForeColor = Color.Green

        Dim ctr As Integer = tbPixels.Text / 2
        Dim r, theta, L As Double
        Dim k As Integer

        For i As Integer = 0 To iMax 'rows
            For j As Integer = 0 To jMax 'columns
                r = Sqrt((i - ctr) ^ 2 + (j - ctr) ^ 2) * rscale
                theta = Atan2(i - ctr, j - ctr)
                L = 0

                k = Round(Abs((1 / 3.1416) * (tau * Omega(1))), 0) + 1
                For n As Integer = -k To k
                    Dim arg As Double = (off - theta / Omega(r) - 2 * n / 2 * 3.1416 / Omega(r))
                    If arg < 0 Or arg > tau Then
                        L += 0
                    Else
                        L += 10 ^ ((4.8 - splint(x, y, y2, (rows - 1), arg)) / 2.5) * phi(r) / Abs(Omega(r))
                    End If
                Next
                galaxy(i, j) = L
            Next
            ToolStripProgressBar1.Value = i
        Next
    End Sub

    Sub CreateMassCutMap()
        ReDim sne(tbPixels.Text, tbPixels.Text)

        Dim Mass1 As Integer = tbMin.Text
        Dim Mass2 As Integer = tbMax.Text

        Dim csv As New CSVData
        csv.LoadCSV("c:\hosting\grb\io\msto" & tbMetal.Text & ".csv", True)
        Dim dtSN As DataTable = csv.CSVDataSet.Tables(0)
        Dim rows As Integer = dtSN.Rows.Count - 1
        ReDim x(rows)
        ReDim y(rows)
        For i As Integer = 0 To rows
            x(i) = dtSN.Rows(i).Item("Log Age")
            y(i) = dtSN.Rows(i).Item("Mass")
        Next

        Dim coef(5) As Single
        For i As Integer = 0 To 4
            coef(i) = dtSN.Rows(i).Item("Coeffs")
        Next

        csv.Dispose()
        dtSN.Dispose()

        Dim iMax As Integer = tbPixels.Text
        Dim jMax As Integer = iMax

        ToolStripProgressBar1.Maximum = iMax
        ToolStripProgressBar1.Minimum = 0
        ToolStripProgressBar1.ForeColor = Color.Red

        Dim ctr As Integer = tbPixels.Text / 2
        Dim r, theta, dth, L, tauprime As Double
        Dim k As Integer

        For i As Integer = 0 To iMax
            For j As Integer = 0 To jMax
                r = Sqrt((i - ctr) ^ 2 + (j - ctr) ^ 2) * rscale
                theta = Atan2(i - ctr, j - ctr)
                dth = rscale / r  'need to know angular separation between pixels
                If r = 0 Then dth = 1
                L = 0
                tauprime = (10 ^ op.LInterpolate(y, x, Mass1)) / 1000000 'reverse interpolate to get time from min Mass

                k = Math.Round(Math.Abs((1 / 3.1416) * (tauprime * Omega(1))), 0) + 1
                For n As Integer = -k To k
                    Dim arg As Double = (off - theta / Omega(r) - 2 * n / 2 * 3.1416 / Omega(r))
                    Dim arg1 As Double = (off - (theta - dth) / Omega(r) - 2 * n / 2 * 3.1416 / Omega(r))
                    Dim arg2 As Double = (off - (theta + dth) / Omega(r) - 2 * n / 2 * 3.1416 / Omega(r))
                    arg1 = Math.Log((arg1 / 1.1) * 1000000, 10)
                    arg2 = Math.Log((arg2 / 1.1) * 1000000, 10)
                    If arg < 3 Or arg > 15000 Then
                        L += 0
                    Else
                        Dim m1 As Double = 10 ^ (coef(0) + coef(1) * arg1 + coef(2) * arg1 ^ 2 + coef(3) * arg1 ^ 3 + coef(4) * arg1 ^ 4)
                        Dim m2 As Double = 10 ^ (coef(0) + coef(1) * arg2 + coef(2) * arg2 ^ 2 + coef(3) * arg2 ^ 3 + coef(4) * arg2 ^ 4)

                        If m1 > Mass2 Then m1 = Mass2
                        If m2 > Mass2 Then m2 = Mass2
                        If m1 < Mass1 Then m1 = Mass1
                        If m2 < Mass1 Then m2 = Mass1

                        L += (1.1 / 1.3 * (m2 ^ -1.3 - m1 ^ -1.3) * phi(r) / dth)
                    End If
                Next
                If L > 0 Then
                    sne(i, j) = L
                Else
                    sne(i, j) = 0
                End If

            Next
            ToolStripProgressBar1.Value = i
        Next
    End Sub

    Private Sub btSimulate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btSimulate.Click
        rscale = tbRadius.Text / (tbPixels.Text / 2)
        tau = tbTau.Text
        rmax = tbRadius.Text
        off = tbOff.Text

        CreateGalaxy()
        CreateMassCutMap()

        mop.CreateColorImage(galaxy, "C:\users\cody\documents\uv project\Sims\galaxy.bmp")
        mop.CreateColorImage(sne, "C:\users\cody\documents\uv project\Sims\sne.bmp")

        PictureBox1.ImageLocation = "C:\users\cody\documents\uv project\Sims\galaxy.bmp"
        PictureBox2.ImageLocation = "C:\users\cody\documents\uv project\Sims\sne.bmp"

        Dim frdt As DataTable = Fruchter(galaxy, sne)

        Plot(frdt)
        op.ExportToCSV(frdt, "C:\users\cody\documents\uv project\sims\fruchter" & tbMin.Text & "-" & tbMax.Text & ".csv")
    End Sub

    Sub Plot(ByVal dt As DataTable)
        Dim fp As NPlot.PointPlot = New NPlot.PointPlot
        Dim pp As NPlot.PointPlot = New NPlot.PointPlot
        Dim lp As NPlot.LinePlot = New NPlot.LinePlot

        Dim XDAT As ArrayList = New ArrayList
        Dim YDAT As ArrayList = New ArrayList
        Dim XDAT2 As ArrayList = New ArrayList
        Dim YDAT2 As ArrayList = New ArrayList

        XDAT.Clear()
        YDAT.Clear()
        XDAT2.Clear()
        YDAT2.Clear()
        PlotSurface2D1.Clear()
        PlotSurface2D1.Title = "Fruchter Plot"
        PlotSurface2D1.BackColor = Color.Empty

        For Each row As DataRow In dt.Rows
            XDAT.Add(row.Item("FracL"))
            YDAT.Add(row.Item("FracS"))
        Next

        'For Each row As DataRow In dtp.Rows
        '    XDAT2.Add(row.Item("FracL"))
        'YDAT2.Add(row.Item("FracS"))
        'Next

        pp.AbscissaData = XDAT
        pp.DataSource = YDAT
        pp.Marker.Type = NPlot.Marker.MarkerType.FilledCircle
        pp.Marker.Size = 1
        pp.Marker.Color = Color.Red
        'fp.AbscissaData = XDAT2
        'fp.DataSource = YDAT2
        'fp.Marker.Color = Color.Blue
        'fp.Marker.Type = NPlot.Marker.MarkerType.Circle
        'fp.Marker.Size = 8
        lp.AbscissaData = XDAT
        lp.DataSource = XDAT
        lp.Color = Color.Green
        PlotSurface2D1.Add(pp)
        'PlotSurface2D1.Add(fp)
        PlotSurface2D1.Add(lp)
        PlotSurface2D1.XAxis1.Label = "Fraction of Light"
        PlotSurface2D1.YAxis1.Label = "Fraction of SNe"
        PlotSurface2D1.XAxis1.WorldMin = 0
        PlotSurface2D1.XAxis1.WorldMax = 1
        'PlotSurface2D1.YAxis1.WorldMin = 0
        PlotSurface2D1.YAxis1.WorldMax = 1

        PlotSurface2D1.Refresh()
    End Sub

    Function Fruchter(ByVal gal(,) As Double, ByVal trans(,) As Double)
        Dim n As Integer = gal.GetLength(0)
        Dim m As Integer = gal.GetLength(1)

        Dim totL As Double = 0
        Dim totS As Double = 0
        Dim fracL As Double = 0
        Dim fracS As Double = 0

        Dim dtL As New DataTable
        dtL.Columns.Add("i", System.Type.GetType("System.Int32"))
        dtL.Columns.Add("j", System.Type.GetType("System.Int32"))
        dtL.Columns.Add("L", System.Type.GetType("System.Double"))
        dtL.Columns.Add("S", System.Type.GetType("System.Double"))
        dtL.Columns.Add("FracL", System.Type.GetType("System.Double"))
        dtL.Columns.Add("FracS", System.Type.GetType("System.Double"))

        For i As Integer = 0 To n - 1
            'random flattening goes here

            For j As Integer = 0 To m - 1
                Dim newRow As DataRow = dtL.NewRow
                newRow.Item("i") = i
                newRow.Item("j") = j
                newRow.Item("L") = gal(i, j)
                newRow.Item("S") = trans(i, j)
                dtL.Rows.Add(newRow)

                totL += gal(i, j)
                totS += trans(i, j)
            Next
        Next

        Dim dv As New DataView(dtL)
        dv.Sort = "L"

        For i As Integer = 0 To dv.Count - 1
            dv(i).Item("FracL") = fracL / totL
            dv(i).Item("FracS") = fracS / totS
            fracL += dv(i).Item("L")
            fracS += dv(i).Item("S")
        Next
        Return dtL
    End Function
End Class
```