

```

1 pro result_rk5, t, h, xx0, vv0, NN, delta, output, regions
2
3 compile_opt idl2, hidden
4
5 ;;;;;;;;;;;;;;
6 ;;
7 ;; This does a 5th order RK step and computes the error estimate
8 ;; See Numerical Recipes, ch 17.2
9 ;;
10 ;;
11 ;;;;;;;;;;;;;;
12
13 hh = h # [1., 1., 1.]
14 tt = t # [1., 1., 1.]
15
16 ;; RK coefficients
17 c = [0., 0.2d, 0.3d, 0.8d, 8./9.d, 1d]
18 b = [35d/384d, 0d, 500d/1113d, 125d/192d, -2187d/6784d, 11d/84d]
19 bs = [5179d/57600d, 0d, 7571d/16695d, 393d/640d, -92097d/339200d, 187d/2100d]
20 bdiff = b-bs
21
22 ;;;;;;;;;;;;;;
23 ;; Figure out which points to look at
24
25 x0 = xx0 + (tt+c[0]*hh)*vv0
26 x1 = xx0 + (tt+c[1]*hh)*vv0
27 x2 = xx0 + (tt+c[2]*hh)*vv0
28 x3 = xx0 + (tt+c[3]*hh)*vv0
29 x4 = xx0 + (tt+c[4]*hh)*vv0
30 x5 = xx0 + (tt+c[5]*hh)*vv0
31
32 ;; Compute density at each point
33 k0 = hh * results_density(x0, output, regions)
34 k1 = hh * results_density(x1, output, regions)
35 k2 = hh * results_density(x2, output, regions)
36 k3 = hh * results_density(x3, output, regions)
37 k4 = hh * results_density(x4, output, regions)
38 k5 = hh * results_density(x5, output, regions)
39
40 ;; Compute N at t+h
41 t += h
42
43 temp = k0*b[0] + k1*b[1] + k2*b[2] + k3*b[3] + k4*b[4] + k5*b[5]
44 if (min(temp) LT 0) then stop
45 NN += temp
46
47 ;; Estimate the error
48 delta = abs(k0*bdiff[0] + k1*bdiff[1] + k2*bdiff[2] + k3*bdiff[3] + k4*bdiff[4] +
49 k5*bdiff[5])
50
51 q1 = where(nn LT 0, nq) & if (nq NE 0) then stop

```

52  
53 end