Documentation for rxn2Dparam.h and rxn2Dparam.c

Steve Andrews

History

2/6/17 Created this library by modifying rxnparam.h and rxnparam.c.

Functions for external use

`double rxn2Dactrate(double step,double sigmab);`

This returns an estimate of the effective activation-limited reaction rate for a 2D reaction, assuming that the reaction probability is set to 1. It is the reaction rate if the rdf is 1 for all $r > a$, where a = `sigmab`. The returned value needs to be divided by Δt. The correct equation is

$$k_{Na}\Delta t = \int_0^a 2\pi r \int_a^\infty 2\pi r' \,\mathrm{grn}(r,r')\,dr'\,dr$$

The inner integral is the rdf after one diffusion step, starting with an rdf that equals 0 from 0 to a, and 1 from a to ∞. The outer integral computes the flux into the region with $r < a$. The Green's function, grn(r,r'), is given below. Unfortunately, Mathematica cannot perform this integral with the correct Green's function, so I used an approximate Green's function instead (the one with r' in the denominator, shown below; it is not particularly good for small r or r' values). Using it,

$$\frac{k_{Na}\Delta t}{a^2} = \frac{1}{2}\left\{\pi + \left(2 - e^{-\frac{1}{2s'^2}}\right)s'\sqrt{2\pi} - \pi\left(s'^2+1\right)\mathrm{erf}\left(\frac{1}{s'\sqrt{2}}\right)\right\}$$

This is in the code.

Functions for investigating an absorbing disk

`double rdf2Dabsorbprob(double *r,double *rdf,int n,double prob);`

`rdf2Dabsorb` integrates the radial diffusion function (rdf) for $0 \le r \le 1$, multiplies those values by 1-`prob`, and returns the integral multiplied by `prob`. `r` is a vector of radii; `r[0]` may equal zero but that is not required; if not, then it is assumed that the rdf has zero slope at the origin.

Integration uses a cylindrical version of the trapezoid rule: at positions r_0 and r_1, the function f has values f_0 and f_1, leading to the linear interpolation

$$f = \frac{(r-r_0)f_1 + (r_1-r)f_0}{r_1-r_0}$$

$$A = \int_{r_0}^{r_1} 2\pi r f(r)\,dr$$

$$= \frac{\pi(r_1-r_0)}{3}\left[f_0(2r_0+r_1)+f_1(r_0+2r_1)\right]$$

The left end of the integral assumes zero slope for the rdf. The right end does not terminate exactly at 1, but includes the upper left triangle of the final trapezoid. That way, if there are two absorptions in a row, the second one will return an integral of 0, and area is properly conserved. The problem is that it does not terminate exactly at 1. Furthermore, the correct relative location of 1 between two `r[j]` points depends on the function. The best solution is to use an unevenly spaced `r[j]` vector, with a very narrow separation about 1 and no `r[j]` equal to 1.

```
void rdf2Ddiffuse(double *r,double *rdfa,double *rdfd,int n,double step);
```

`rdf2Ddiffuse` integrates the radial distribution function with the Green's function for radially symmetric diffusion to implement diffusion over a fixed time step. `r` is a vector of radii, `rdfa` is the input rdf, `rdfd` is the output rdf, `n` is the number of points, and `step` is the rms step length, equal to $(2Dt)^{1/2}$. `r[0]` may equal 0 but it is not required. It is assumed that `rdfa` has zero slope at $r = 0$. It is assumed that the rdf equals 1 for all radii beyond the range of `r`, leading to a constant source at large r values.

The numerical portion of the integral is carried out like the one in `rdfabsorb` but with a different integrand,

$$c(r) = \int_0^\infty 2\pi r' \,\mathrm{rdfa}(r')\,\mathrm{grn}(r,r')\,dr'$$

$\mathrm{grn}(r,r')$ is the Green's function, equal to

$$\mathrm{grn}(r,r') = G_s(r)G_s(r')I_0\left(\frac{rr'}{s^2}\right)$$

where $G_s(x)$ is a normalized Gaussian with mean 0 and standard deviation σ and $I_0(x)$ is an I_0 modified Bessel function. This equation is from Carslaw and Jaeger 14.8(1) (page 368) with the substitutions of his κ for my D and $(2Dt)^{1/2} = s$. Note that

$$\operatorname{grn}(0,r') = \frac{1}{s\sqrt{2\pi}} G_s(r') \qquad \text{and} \qquad \operatorname{grn}(r,0) = \frac{1}{s\sqrt{2\pi}} G_s(r)$$

Also, if $r' >> s$, then

$$\operatorname{grn}(r,r') \approx \frac{G_s(r'-r)}{2\pi\sqrt{rr'}} \approx \frac{G_s(r'-r)}{\pi(r+r')} \approx \frac{G_s(r'-r)}{2\pi r} \approx \frac{G_s(r'-r)}{2\pi r'}$$

These approximations were found by trying various functions in Mathematica and finding what ones qualitatively agreed well with the exact result. The first two functions are remarkably good, even when r and r' are as small as $2s$, and have the benefit of being symmetric with respect to r and r'. However, they don't integrate easily. The latter two functions are less good but can be integrated easily. All of these functions become more accurate with larger r and r' values. To create the assumption that the rdf is equal to 1 for all $r > r_{max}$, where r_{max} is the largest tabulated r value, the Green's function integral is computed from r_{max} to ∞ using

$$c(r) = \int_{r_{max}}^{\infty} 2\pi r' \frac{G_s(r-r')}{2\pi r'} dr' = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{r - r_{max}}{s\sqrt{2}}\right)\right]$$

This supplies a constant source of concentration that diffuses into the range of interest. Because it is assumed that the rdf equals 1 here, which is not the correct steady-state solution, this affects the shape of the computed rdf over the r values that are within a few rms step lengths of the last tabulated r value. However, it should have negligible effect for smaller r values.

`void rdf2Dreverserxn(double *r,double *rdf,int n,double step,double b,double flux);`

Analysis of the reverse reaction involves adding a delta function to the rdf and then convolving with the Green's function. However, this leads to numerical errors, although it is trivial analytically. This function is the analytic solution. It adds the diffusion Green's function to the rdf, based on a delta function at b, and after one diffusion step. `r` is a list of radii, `rdf` is the rdf, `step` is the rms step length, `b` is the delta function point (which does not have to be equal or unequal to a `r[j]` value), and `flux` is the area of the delta function.

`double rdfsteadystate(double *r,double *rdfa,double *rdfd,int n,double step,double b,double eps);`

`rdfsteadystate` calculates the radial distribution function (rdf) for alternating absorption and diffusion steps, for either irreversible or reversible reactions. `r` is a vector of radii, `rdfa` is input as a trial rdf and output as the result after absorption, `rdfd` is ignored on input but is output as the rdf after diffusion, `n` is the number of elements in the vectors, `step` is the rms step length, and `b` is either <0 if the reaction is irreversible or is the unbinding radius if the reaction is reversible. It executes until the fractional difference between successive steps is less than `eps`, but at least

30 times and no more than `maxit` times. It can also exit if it iterates more than `maxit` times before converging or if the flux exceeds `maxflux`; if either of these happens, the function returns -1.

`void rdfmaketable();`

`rdfmaketable` is used to create data tables of reaction rates, including those used above in `numrxnrate`. All input is requested from the user using the standard input and all output is sent to standard output. Runtime is about 1 minute with mode `i`, 200 pts, `eps`=1e-4 (on a Mac G4 laptop).