Please refer to the latest version of the online notes. The analysis is more straightforward if the total binding rate is constant independent of epsilon: I have adjusted the rate matrix to achieve this.

First of all, epsilon quantifies the degree of irreversibility. This means that epsilon quantifies the amount of chemical work done during a loop. We can see this mathematically by substituting our rates into equation 21 of the slides - the result is dG_loop(1,2,3,4,1) = 4 kT ln (epsilon/(2-epsilon)). Clearly this is less than zero if 0 < epsilon < 1, and zero if epsilon =1.

The code below estimates the mean and variance of the time spent in the bound state (2,3,4) as a function of epsilon.

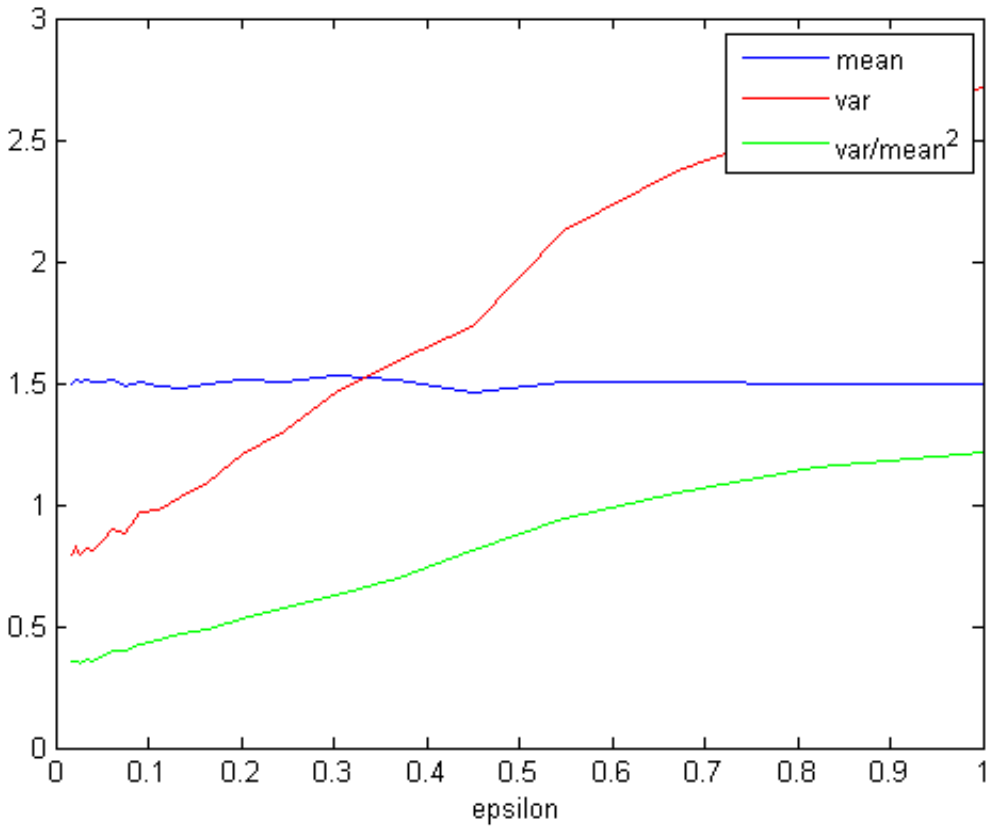
```
%Declare array for results.
output_data = zeros(21,4);
%Loop over several values of epsilon
for counter = 1:21
    %implement matrix with q=1 and epsilon determined by loop variable.
    rmatrix=zeros(4);
    epsilon = exp((1-counter)/5);
    rmatrix(1,2)=0.01*(2-epsilon);
    rmatrix(1,4)= 0.01*epsilon;
    rmatrix(2,1)= epsilon;
    rmatrix(2,3)=2-epsilon;
    rmatrix(3,2)=epsilon;
    rmatrix(3,4)=2-epsilon;
    rmatrix(4,3)=epsilon;
    rmatrix(4,1)=2-epsilon;

    %Declare variables to record average and variance of times
    tav=0;
    tsqav=0;
    for i=1:10000
        %start in state 1 and take a step
        init_state=1;
        output = gillespie(rmatrix,init_state);
        state=output(1);
        %Perform more steps until the system returns to state 1
        time =0;
        while state ~=1
            output = gillespie(rmatrix,state);
            state = output(1);
            dt = output(2);
            time = time + dt;
        end
        %Update average and variance according to new sample
        tav= (tav*(i-1) + time)/(i);
        tsqav = (tsqav*(i-1) + time*time)/(i);
    end
    outputdata(counter,1)= epsilon;
    outputdata(counter,2)= tav;
    outputdata(counter,3)= tsqav - tav*tav;
    outputdata(counter,4)= (tsqav - tav*tav)/(tav*tav);
end
```

The data shows that the mean time spent bound is epsilon independent, but the variance in binding time drops as epsilon

tends to 0. consequently, so does the variance /mean^2.

```
%Plot data
plot(outputdata(:,1),outputdata(:,2));
hold on
plot(outputdata(:,1),outputdata(:,3),'red');
plot(outputdata(:,1),outputdata(:,4),'green');
xlabel('epsilon')
legend('mean','var','var/mean^2');
hold off
```



Now we show that this reduction in variance of time spent bound can be translated into a lower sampling error. We calculate the mean and variance of r_T, the receptor occupancy during some fixed period T.

```
%Loop over several values of epsilon
for counter = 1:21
    %implement matrix with q=1 and epsilon determined by loop variable.
    rmatrix=zeros(4);
    epsilon = exp((1-counter)/5);
    rmatrix(1,2)=0.01*(2-epsilon);
    rmatrix(1,4)= 0.01*epsilon;
    rmatrix(2,1)= epsilon;
    rmatrix(2,3)=2-epsilon;
    rmatrix(3,2)=epsilon;
    rmatrix(3,4)=2-epsilon;
    rmatrix(4,3)=epsilon;
    rmatrix(4,1)=2-epsilon;
```

```
    %Declare variables to record average and variance of times
    rT_av=0;
    rTsq_av=0;
    for j=1:1200
        %calculate occupancy in some period T
        time = 0;
        state=1;
        T=2000;
        t_init=1000;
        %initialize to forget first state
        while time<t_init;
            output = gillespie(rmatrix,state);
            state=output(1);
            time = time+output(2);
        end
        %now sample for time T
        %initialize recording variables
        time =0;
        tbound=0;

        bound_flag=0;
        if state>1
            bound_flag=1;
        end
        while time<T
            output = gillespie(rmatrix,state);
            state=output(1);
            time = time+output(2);
            %increment time in bound state if we were bound at the end of
            %the previous step.
            if bound_flag
                tbound = tbound+output(2);
                if time >T
                    %Don't count times t>T.
                    tbound = tbound+T-time;
                end
            end
            %update bound_flag to indicate whether ligand is bound or not
            if state>1
                bound_flag=1;
            else
                bound_flag=0;
            end
        end
        %increment averages
        rT_av = (rT_av*j + tbound/T)/(j+1);
        rTsq_av = (rTsq_av*j + tbound*tbound/(T*T))/(j+1);
    end
    outputdata(counter,1)= epsilon;
    outputdata(counter,2)= rT_av;
    outputdata(counter,3)= sqrt(rTsq_av - rT_av*rT_av);
    outputdata(counter,4)= (rTsq_av - rT_av*rT_av)/ (rT_av*rT_av);
end
```

Berg-Purcell: variance/mean^2 = 2/<# of binding events>

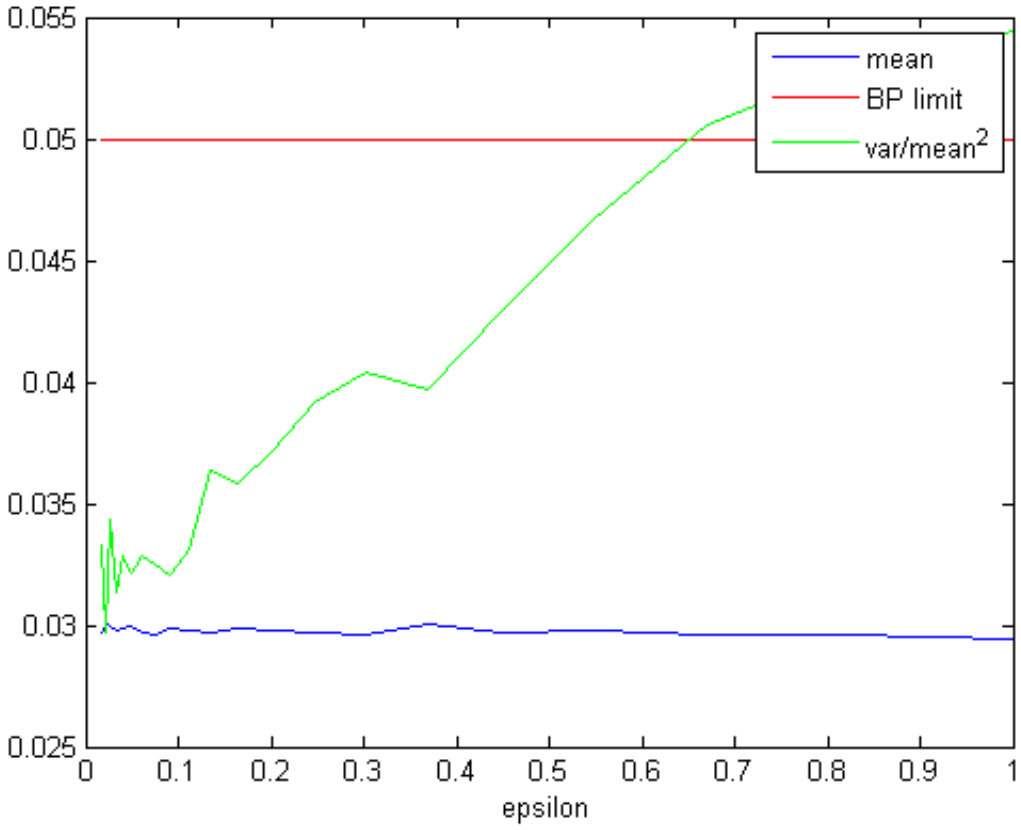
Time spent in bound state is only 3% of total.

=> number of binding events ~ T * total binding rate = 0.02*T = 40.

=> Burg-Purcell limit is approximately 2/40 = 0.05

Below I plot the mean of the time spent bound during T, the variance of this time / mean^2 and the approximation to the Berg-Purcell result for comparison. % Clearly, as epsilon -> 1, introducing the multi-state bound ensemble is unhelpful, but for small epsilon it is possible to sample more accurately than suggested by Berg and Purcell.

```
outputdata(:,3)=0.05;
plot(outputdata(:,1),outputdata(:,2));
hold on
plot(outputdata(:,1),outputdata(:,3),'red');
plot(outputdata(:,1),outputdata(:,4),'green');
xlabel('epsilon')
legend('mean','BP limit','var/mean^2');
hold off
```



Published with MATLAB® R2014a