

PHYS 3900 Numerical Project

Fred Hohman

NP01: 29 April 2013, **NP02:** 30 April 2013

Contents

1	NP01 Numerical Integration	2
1.1	Calculations, Error, and Tables	2
1.2	Figures	4
1.3	Error Theory	5
1.4	Results	5
1.5	Discussion	6
2	NP02 Numerical ODE Solvers	7
2.1	H ₂ Combustion Model A	7
2.2	ODE Solver Observations and Default Parameters	7
2.3	The Ignition Process: The Spark	7
2.4	Explosive Growth	8
2.5	Accuracy of E1 and E2 Algorithms	10

1 NP01 Numerical Integration

For Numerical Project NP01, we will be trying to evaluate

$$Y(t) := \int_{T_I}^{T_F} R(t) dt = \int_0^5 (t + \cos(\omega t)) dt, \text{ where } \omega = 1 \text{ and } \omega = 40,$$

using computer approximations. For this project I chose to perform my calculations and data analysis in Numbers from iWork '09, the Mac OS X equivalent of Microsoft Excel. This application shares nearly every single function as Excel needed for NP01, and operates analogously to other spreadsheet applications. I also used Mathematica 9 to check various algebra steps and sum approximations.

The four approximation methods we will be using are the Left Riemann Sum (LRS), the Right Riemann Sum (RRS), the Trapezoid Rule (TR), and the Simpson Rule (SR). We will carry out the calculations using a time stepwidth

$$h \equiv \frac{T_F - T_I}{K},$$

where K follows a doubling sequence of values: $K = 2, 4, 8, 16, 32, 64, 128, 256,$ and 512 .

1.1 Calculations, Error, and Tables

Once calculating the the various timewidths h for all K values, I organized my spreadsheets by making a column listing integer values of K , a column of t_k values, a column of $R(t_k)$ for $\omega=1$, and a column of $R(t_k)$ for $\omega=40$. I then wrote four functions to evaluate our integral by the four methods stated above.

LRS for both values of ω simply required summing the $R(t_k)$ values from $k=0$ to $k=K-1$ and multiplying by h . Similarly, RRS required summing the $R(t_k)$ values from $k=1$ to $k=K$ and multiplying by h .

For TR I summed the $R(t_k)$ values from $h=1$ to $k = K-1$, and multiplied by h . I then added the sum of the endpoints, $R(t_0)$ and $R(t_K)$ and multiplied their sum by a pre-factor of $\frac{h}{2}$.

Finally, for SR I had to add two other columns in order to calculate to the sum of even $R(t_k)$ and odd $R(t_k)$. Once those were found, I then multiplied by their respective pre-factors, $\frac{2h}{3}$ and $\frac{4h}{3}$. I then added the sum of the endpoints, $R(t_0)$ and $R(t_K)$ and multiplied their sum by a pre-factor of $\frac{h}{3}$.

I then calculated the numerical error

$$\Delta Y := \left| Y^{(K)} - Y \right|,$$

where $Y^{(K)}$ is the approximated value of the integral, and Y is the exact value. Note: when using $K = \infty$, the numerical integration error should be 0. I left in this calculation to verify that my formula for calculating the numerical integration error was correct.

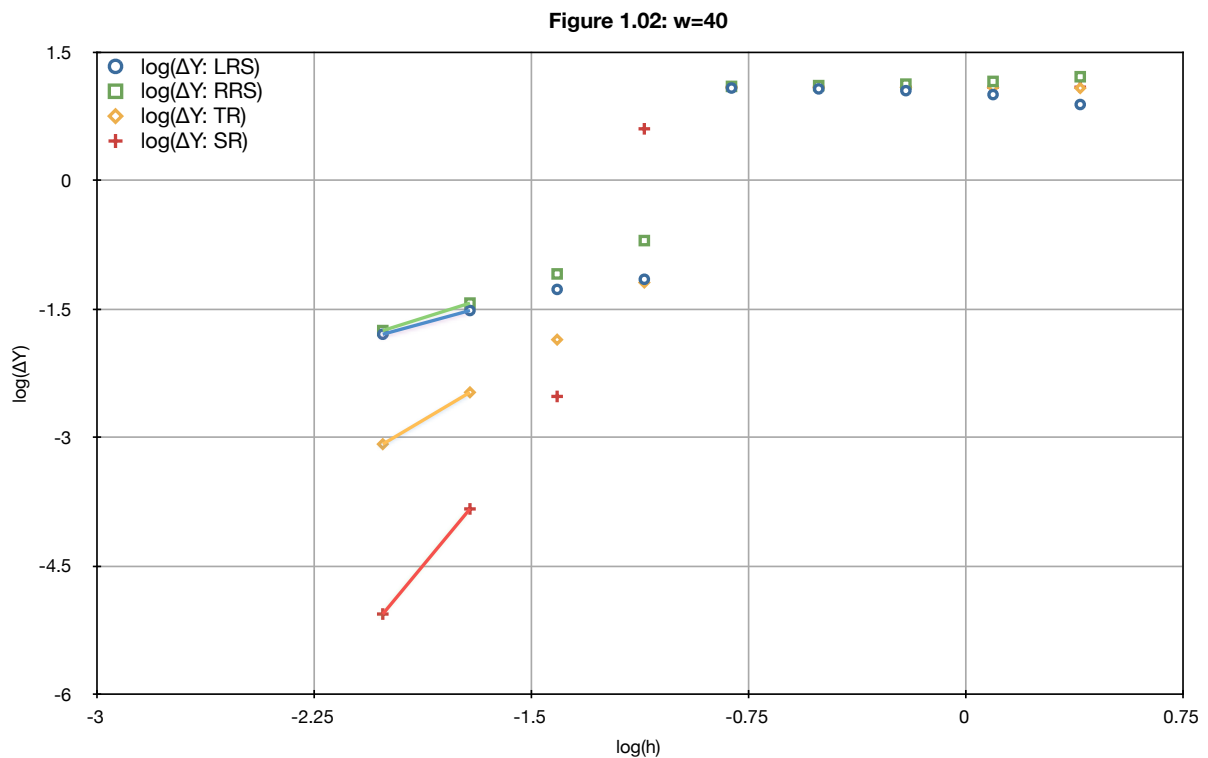
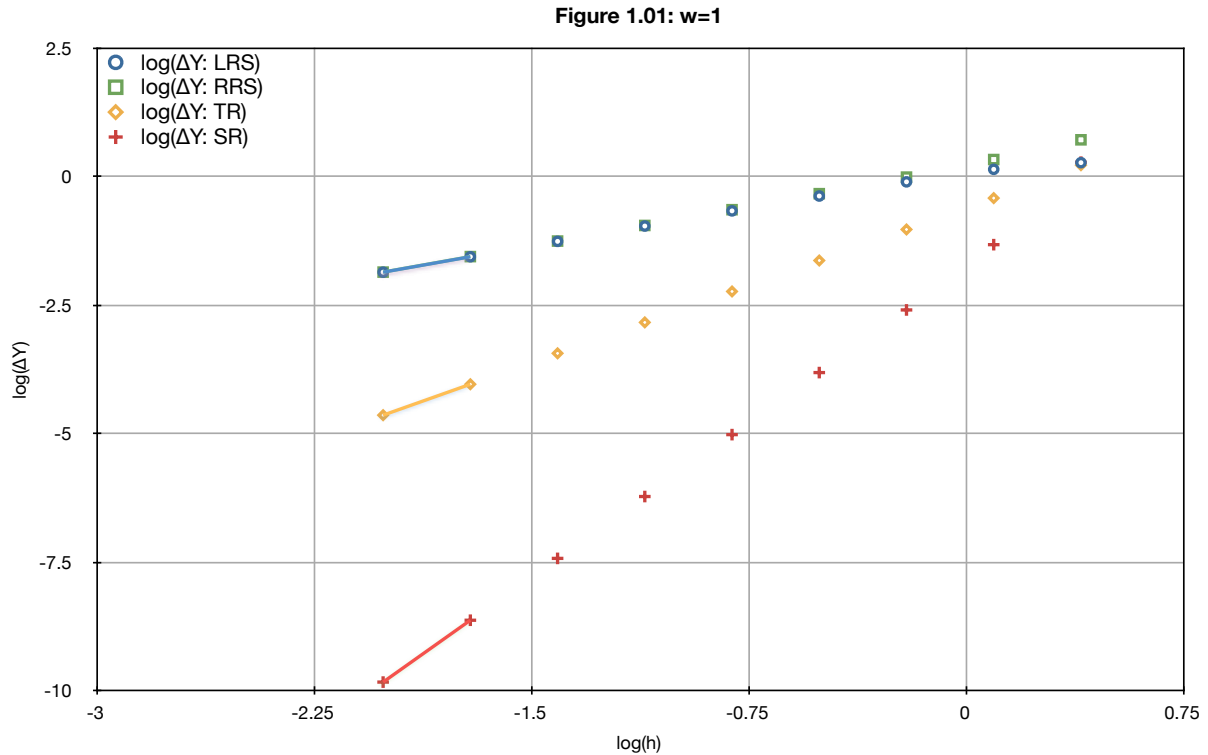
I then tabulated all of my data, including K values, stepwidth h , all four approximation method results $Y^{(K)}$, and the integration error ΔY for each method into two tables for $\omega=1$ and $\omega=40$, respectively titled Table 1.01 and Table 1.02.

Table 1.01: $w=1$									
K	h	LRS	RRS	TR	SR	ΔY : LRS	ΔY : RRS	ΔY : TR	ΔY : SR
2	2.5	7.7414228834	14.868889274	11.305156079	7.6977193082	1.8818042926	5.2456620984	1.6819289029	1.9255078678
4	1.25	8.2260727107	11.789805906	10.007939308	9.5755337182	1.3971544654	2.1665787301	0.3847121324	0.0476934578
8	0.625	8.8265542197	10.608420817	9.7174875185	9.6206702553	0.7966729563	0.9851936414	0.0942603425	0.0025569208
16	0.3125	9.2012099568	10.092143256	9.6466766062	9.6230729688	0.4220172192	0.4689160797	0.0234494302	0.0001542072
32	0.15625	9.4063490435	9.8518156929	9.6290823682	9.6232176222	0.2168781325	0.2285885169	0.0058551922	0.0000095538
64	0.078125	9.5133238648	9.7360571896	9.6246905272	9.6232265802	0.1099033112	0.1128300135	0.0014633512	0.0000005958
128	0.0390625	9.5679096547	9.6792763171	9.6235929859	9.6232271388	0.0553175213	0.0560491411	0.0003658099	0.0000000372
256	0.01953125	9.5954769611	9.6511602923	9.6233186267	9.6232271737	0.0277502149	0.0279331163	0.0000914507	0.0000000023
512	0.009765625	9.6093292058	9.6371708714	9.6232500386	9.6232271759	0.0138979702	0.0139436954	0.0000228626	0.0000000001
∞		9.623227176	9.623227176	9.623227176	9.623227176	0	0	0	0
K	log(h)	log(ΔY : LRS)	log(ΔY : RRS)	log(ΔY : TR)	log(ΔY : SR)				
2	0.3979400087	0.2745744549	0.7198003119	0.2258076337	0.2845452975				
4	0.096910013	0.1452444231	0.3357744752	-0.414864117	-1.32154119				
8	-0.204119983	-0.098719925	-0.0064784	-1.025670986	-2.592282731				
16	-0.505149978	-0.374669829	-0.328904875	-1.629867705	-3.811895288				
32	-0.806179974	-0.663784235	-0.64094559	-2.232458845	-5.019823181				
64	-1.10720997	-0.958989223	-0.94757536	-2.834651435	-6.224891167				
128	-1.408239965	-1.257137288	-1.251431038	-3.436744564	-7.429247909	Data for Small h for Liner Regression Lines			
256	-1.709269961	-1.55673365	-1.55388061	-4.038812839	-8.633426971	log(ΔY : LRS)	log(ΔY : RRS)	log(ΔY : TR)	log(ΔY : SR)
512	-2.010299957	-1.857048623	-1.855622114	-4.640874902	-9.837557236	-1.55673365	-1.55388061	-4.038812839	-8.633426971

Table 1.02: $w=40$									
K	h	LRS	RRS	TR	SR	ΔY : LRS	ΔY : RRS	ΔY : TR	ΔY : SR
2	2.5	20.217391542	28.871299105	24.544345323	24.84115791	7.7828888394	16.436796402	12.109842621	12.406655208
4	1.25	22.599508902	26.926462683	24.762985793	24.835865949	10.1650062	14.491959981	12.32848309	12.401363247
8	0.625	23.735665975	25.899142866	24.81740442	24.835543963	11.301163272	13.464640163	12.382901718	12.40104126
16	0.3125	24.290124861	25.371863307	24.830994084	24.835523972	11.855622158	12.937360604	12.396491381	12.401021269
32	0.15625	24.563955953	25.104825176	24.834390564	24.835522724	12.12945325	12.670322473	12.399887862	12.401020022
64	0.078125	12.363933633	12.634368244	12.499150938	8.3874043964	0.07056907	0.1998655413	0.0646482357	4.0470983063
128	0.0390625	12.380795292	12.516012597	12.448403945	12.43148828	0.0537074109	0.0815098948	0.0139012419	0.0030144226
256	0.01953125	12.40406413	12.471672782	12.437868456	12.434356626	0.0304385731	0.0371700797	0.0033657533	0.0001460762
512	0.009765625	12.418435506	12.471672782	12.43533767	12.434494074	0.0160671963	0.0177371302	0.000834967	0.0000086285
∞		12.478167568	12.478167568	12.478167568	12.478167568	0	0	0	0
K	log(h)	log(ΔY : LRS)	log(ΔY : RRS)	log(ΔY : TR)	log(ΔY : SR)				
2	0.3979400087	0.8911408276	1.2158171757	1.0831384991	1.093654713				
4	0.096910013	1.0071076478	1.1611271261	1.0909096438	1.0934694285				
8	-0.204119983	1.0531231494	1.1291947516	1.092822426	1.0934581525				
16	-0.505149978	1.0739243498	1.1118456834	1.0932987828	1.0934574524				
32	-0.806179974	1.083841225	1.1027876683	1.0934177576	1.0934574087				
64	-1.10720997	-1.151385606	-0.699262076	-1.189443323	0.6071437538				
128	-1.408239965	-1.269965783	-1.088789668	-1.856946398	-2.520795856	Data for Small h for Liner Regression Lines			
256	-1.709269961	-1.51657571	-1.429806507	-2.472917718	-3.835420455	log(ΔY : LRS)	log(ΔY : RRS)	log(ΔY : TR)	log(ΔY : SR)
512	-2.010299957	-1.794059902	-1.751116647	-3.078330714	-5.064064637	-1.51657571	-1.429806507	-2.472917718	-3.835420455

1.2 Figures

From these tables, I then made two graphs, titled Figure 1.01 for $\omega=1$ and Figure 1.02 for $\omega=40$, that plot $\log(h)$ vs. $\log(Y^{(K)})$. The legend in corner details each approximation method's shape representations, along with corresponding colors. The linear regressions¹ are to be explained in the next section.



¹The "trendline" tool in Numbers '09 only draws a line through the endpoints of the selected data, and since we only used the smallest two h values to obtain the most accurate slope for each algorithm, Numbers '09 will draw a line only from these two points (one of the drawbacks Numbers has compared to Excel)!

1.3 Error Theory

According to mathematical theory,

$$\lim_{K \rightarrow \infty} h = \lim_{K \rightarrow \infty} \frac{T_F - T_I}{K} = 0$$

should imply that the error ΔY decreases asymptotically with the stepwidth h by a power law:

$$\Delta Y \cong Ch^p,$$

where p and C are constants. Based on these theoretical results, we can see why the data points for a given integrated and approximation method fall on a straight line, $y \cong ax + b$ for sufficiently small values of h . Notice that if we were to take the log of both sides of our ΔY approximation equation, we could arrive at an equation in the form of a line:

$$\log(\Delta Y) \cong \log(Ch^p) = \log(C) + \log(h^p) = p \log(h) + \log(C).$$

From here we can compare this equation to our linear equation to produce the relationships between slope a , intercept b , and constants p and C :

$$a = p, \quad b = \log(C) \quad \text{where} \quad x = \log(h), \quad y = \log(\Delta Y)$$

1.4 Results

Since all eight data sets from Figures 1.01 and 1.02 have several h values that are small enough to fall in the limit where the power law from above holds, we can see that $K=512$ is a sufficient approximation for these integrands; however, doubling K again would make the error continue to decrease, therefore generating another data point on our graphs that would, once again, fall on the linear regression line, since this new K value would have an even smaller h stepwidth than any h from Table 1.01 or 1.02.

I then applied a linear regression to the two smallest h values from the $\omega=1$ integrand, even though each point fell almost directly on the line (applying the regression to the two smallest h values will give the most accurate slope a and intercept b). For the $\omega=40$ integrand, I applied a linear regression with the same conditions as $\omega=1$, therefore I only used the h 's from $K=512$ and 256 to generate this line (Note: I only used the smallest two h values for each method instead of several, since the accuracy is highest for smallest h and biggest K). As stated before, greater values of K would make the linear regression more accurate; however, I believe $K=512$ is sufficient to determine the slope and intercept of the lines.

I then extracted the slope a and intercept b from the equations of these linear regressions. Once these values were obtained, I was then able to estimate p and C using the relationships as stated above in **1.3**. Once all this new data was collected, I tabulated the results and created Table 1.03 to compare all values of a , b , C , and p for LRS, RRS, TR, and SR, for both $\omega=1$ and $\omega=40$.

Table 1.03								
	w=1				w=40			
	Slope: a	Intercept: b	Pre-factor: C	Exponent: p	Slope: a	Intercept: b	Pre-factor: C	Exponent: p
log(ΔY : LRS)	0.9976	0.1485	1.4076672284	0.9976	0.9218	0.0590	1.1455129414	0.9218
log(ΔY : RRS)	1.0024	0.1594	1.4434442007	1.0024	1.0674	0.3946	2.4808471081	1.0674
log(ΔY : TR)	2.0000	-0.6203	0.2397176436	2.0000	2.0111	0.9647	9.2193435745	2.0111
log(ΔY : SR)	4.0000	-1.7963	0.0159845347	4.0000	4.0815	3.1409	1383.2478377	4.0815

1.5 Discussion

It's worth noting that some $Y^{(k)}$ values for LRS, RRS, TR, and SR might be slightly off for large h and small K ; this is an approximation after all. However, we can apply a simple method to check whether the algorithm approximations LRS, RRS, TR, and SR have decreasing error by calculating the factor by which the error goes down for each method. As stated in class, we can find these error decreasing factors by the ratio

$$\frac{Y^{(K)}}{Y^{(2K)}} \cong \begin{cases} 2, \text{ for LRS and RRS} \\ 4, \text{ for TR} \\ 16, \text{ for SR} \end{cases} \quad \text{for } K = 2, 4, 8, 16, 32, 64, 128 \text{ or } 256.$$

This implies that the method with the highest error decreasing factor will give a better approximation of our integral faster than others, and for bigger K . Also, these factors will be closest to the exact value of 2, 4, and 16 when h is smallest and K is largest. Applying this quick check for both $\omega=1$ and $\omega=40$ from Table 1.01 and 1.02 above, we see these factors present, providing further proof that our calculations are accurate.

When comparing p values for $\omega=1$ and $\omega=40$, notice that the p values for each approximation algorithm are very close (almost identical!); the various p 's do not vary greatly since the slope of each $\log(\Delta Y)$ vs. h line should be equal, since the same approximation algorithm methods, LRS, RRS, TR, SR, are being used for both values of ω . In contrast, when comparing C values for both $\omega=1$ and $\omega=40$, each corresponding C is related by $\log(C)=b$, *i.e.*, the bigger b gets then the bigger C gets. So whereas p was dependent upon the approximation algorithm used, C is dependent upon the integrand, specifically the period of our $\cos(\omega t)$ terms. Notice how the $\omega=40$ graph is a "shifted" graph of the $\omega=1$ to the left. Therefore, C values for $\omega=40$ will be bigger than those of $\omega=1$, since in the limit as $h \rightarrow 0$ and $K \rightarrow \infty$, lines of the same slope shifted left will have a higher intercept. Furthermore, we can see that the exact shift of the $\omega=40$ graph compared to the $\omega=1$ graph will be $\log(40)$, *i.e.*, in general: $\log(\omega)$.

Finally, comparing Figures 1.01 and 1.02, we can see that the $\omega=40$ integrand will require smaller h values to obtain an accurate approximation by simply looking at each graph, Figures 1.01 and 1.02, and noticing that for an accurate approximation, h must fall behind the vertical line defined at $\log(\frac{2\pi}{\omega})$. Notice all points on Figure 1.01 fall to the left of this vertical line $\log(2\pi)$, but only h values where $K=64$ or greater fall to the left of $\log(\frac{2\pi}{40})$. Qualitatively speaking, h values must be smaller for $\omega=40$ since the graph of the $\omega=40$ integrand varies much more than the $\omega=1$ integrand, so the h values need to be small enough to accurately model the gaps in between each local max and min of the $\omega=40$ integrand.

2 NP02 Numerical ODE Solvers

For Numerical Project NP02, we will be trying to solve the coupled rate equations (ODE system) for H_2 Combustion Model A, posted on one of the UGA KinSolver websites. For this project I also chose to perform my calculations and data analysis in Numbers from iWork '09, the Mac OS X equivalent of Microsoft Excel.

2.1 H_2 Combustion Model A

I have downloaded all appropriate files from the class website and have read through **np_H20.pdf** to familiarize myself with the H_2 Combustion Model A.

2.2 ODE Solver Observations and Default Parameters

Similar to **2.1**, I have experimented with **KS PHYS 3900_H20** website, including changing the values of the model parameters (initial conditions of the 6 given molecules, forward and backward reaction rate coefficients $k_1, k_2, k_3, \bar{k}_1, \bar{k}_2,$ and \bar{k}_3 , initial and final times T_I and T_F , time step number K , and stepwidth Δk), selecting different integration options (E1, E2, and others), and familiarized myself with the “plot” function, as well as obtaining and importing data into my spreadsheet software.

We will use this section to state the default parameters for the various solutions to the ODE system that will be used in the upcoming sections. So let forward and backward coefficients

$$k_1 = k_2 = k_3 = 1; \quad \bar{k}_1 = \bar{k}_2 = \bar{k}_3 = 0.0001,$$

initial time and final times

$$T_I = 0; \quad T_F = 10,$$

number of time steps and stepwidth

$$K = 200000; \quad \Delta k = 1000,$$

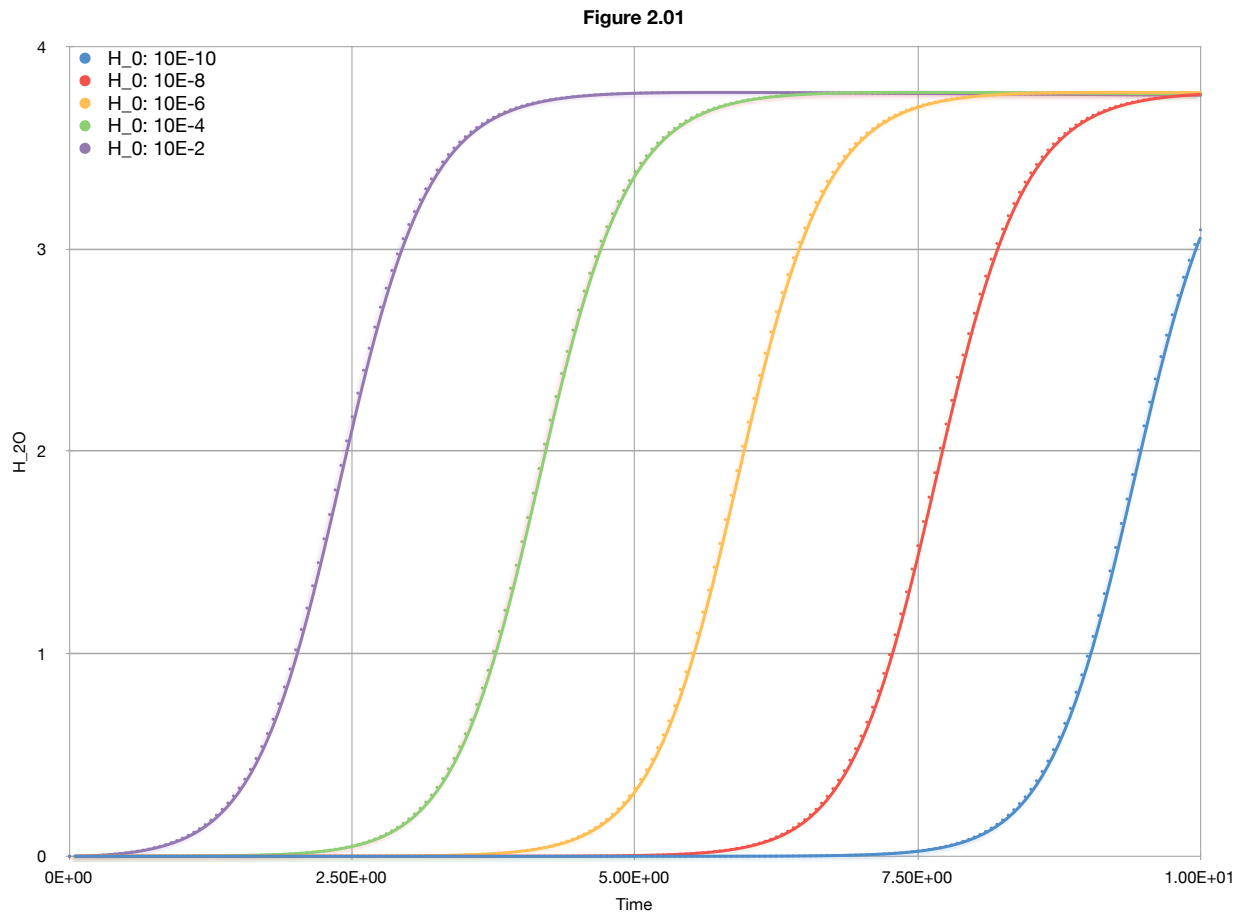
and integration method to be E1.

2.3 The Ignition Process: The Spark

To study the ignition process, let's first set all initial parameters to default as given in **2.2**, with the exceptions of $[H_2]_o=6.0$, $[O_2]_o=3.0$, and all other species to 0.0. Notice how the plot of the ODE solution shows each species as constants for all defined time. This is because for H_2 and O_2 to be able to react, each species needs an appropriate reaction partner. We can see these two specific species are not compatible by inspecting the Chemical Circuit/Reaction Network Model for Combustion Model A as shown in class. For H_2 to react, it needs some concentration of O ; likewise, for O_2 to react it needs some concentration of H . So if no reaction partner is present for either H_2 or O_2 , the two species will simply mix together but not react, *i.e.*, the initial concentrations will not change with time since no reaction is taking place between the two species.

Let's now add a "spark" to ignite the combustible H_2-O_2 mixture. We will run five simulations using default parameters from **2.2**, except where $[H_2]_o=6.0$, $[O_2]_o=3.0$, and our spark concentration can be chosen as $[H]_o$ or $[O]_o$. I chose to use $[H]_o$ for the spark concentration, so for each of the five simulations, let $[H]_o=1.0 \times 10^{-10}, 1.0 \times 10^{-8}, 1.0 \times 10^{-6}, 1.0 \times 10^{-4},$ and 1.0×10^{-2} .

After running each simulation, I imported the kin.o01.txt file into Numbers '09 and plotted each $[H_2O]$ concentration vs. time t , as seen in Figure 2.01 below.



When the spark is added, notice the chemical reaction takes place. All species concentrations change with time, and eventually the H_2O species (as graphed above for each spark concentration) reaches a maximum concentration. Geometrically, our graphs are nearly identical except for a shift along the time axis. Also, when the spark concentration changes by increasing factors of 10, notice that the H_2O concentration maximum stays the same for every solution; however, the time it takes for the H_2O reaction to occur decreases. So the greater the spark concentration that is used, the quicker the H_2O species will start to combust, but H_2O will always level out at the same maximum concentration over the same amount of time.

2.4 Explosive Growth

To study the explosive growth of reaction products during ignition of the chemical combustion taking place, let's once again set all parameters to default as given in **2.2**, with the exceptions of $[H_2]_o=6.0$, $[O_2]_o=3.0$, a spark of $[O]_o=10^{-9}$, and all other concentrations set to 0.

After running the simulation, I imported the kin.o01.txt file into Numbers '09 and plotted the $[H_2O]$ concentration vs. time t and the $[O]$ concentration vs. time t , as seen in Figure 2.02 and 2.03 below.

Figure 2.02

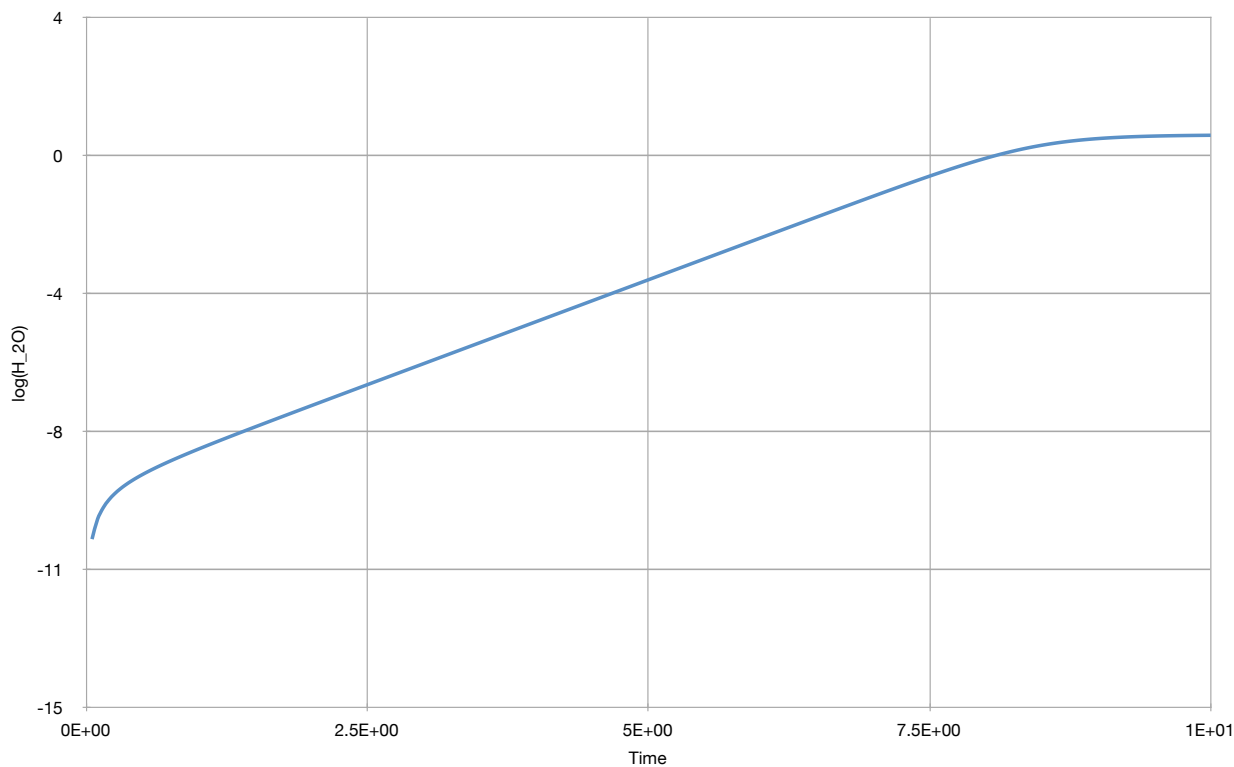
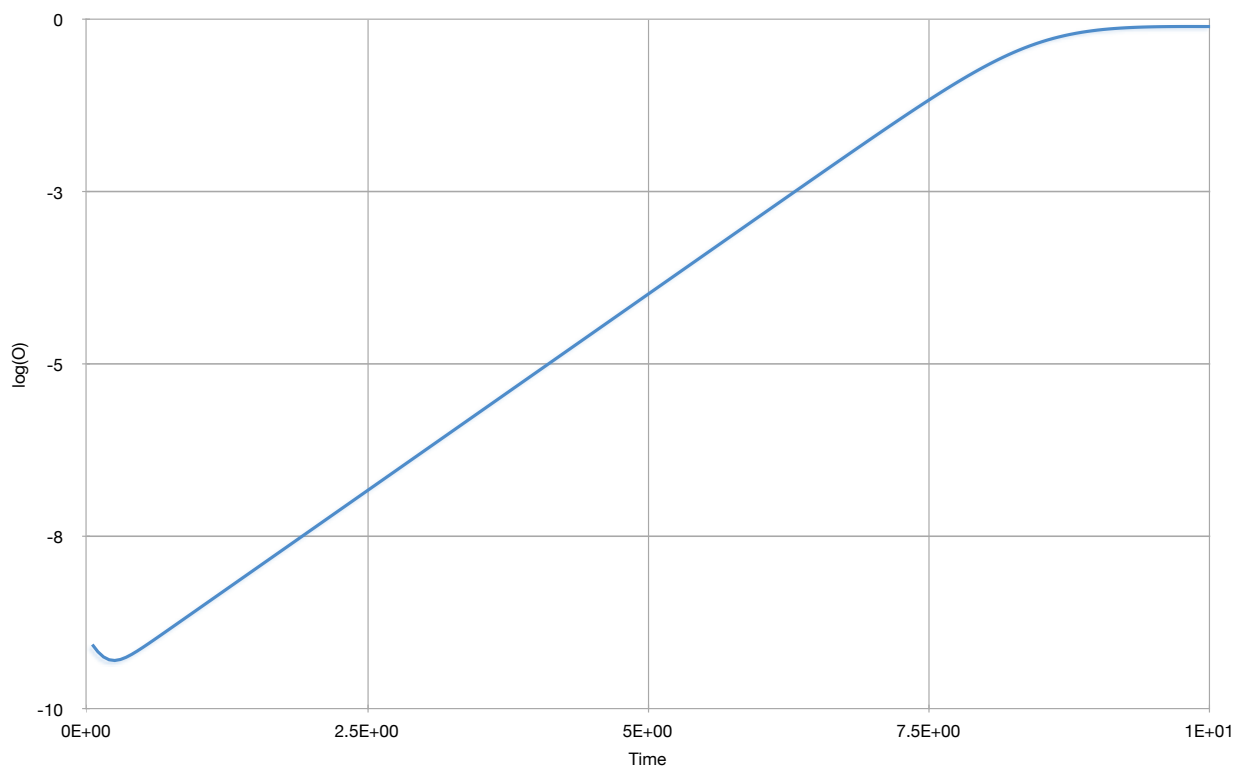


Figure 2.03



The H_2O vs. time graph starts out with a large positive slope, and becomes linear rather quickly. The graph stays linear until leveling off at a maximum concentration. In contrast, the O vs. time graph starts out with a negative slope, but changes sign and becomes linear just like H_2O . Similarly, the O concentration graph also levels out to a maximum concentration.

The approximate linear dependence of the $\log([H_2O])$ and $\log([O])$ plotted vs. time indicates

that the rate of H_2O and the rate of O are exponentially related to time in the differential equations. We can find a functional dependence of each concentration that represents the numerical data plotted using a line of best of fit for the “log-linear” phase: $y \cong mx + b$. After finding the slope and intercept of a linear regression line to just the ignition phase of both Figure 2.02 and 2.03, we can see that the intercept is simply $b = \log([H_2O]_o)$ or $\log([O]_o)$ during the ignition phase. The slope of these linear regressions represents the rate at which the concentration of H_2O and O increases on a logarithmic scale *i.e.*, $m = \frac{\Delta \log(\text{concentration})}{\Delta t}$, where “concentration” can be $[H_2O]$ or $[O]$.

From the reaction network, we can see that some product concentrations will grow exponentially because of the mathematical structure of the 6 differential equations. If a certain species is used in a reaction more than that same certain species is produce from a different reaction, then that species will eventually die out to a 0 concentration. For H_2O and O , this is not the case. From the equations listed in class, we can see that H_2O and O (among other spices) are produced more than they are used as reactants. For example, our O vs. time graph first starts out as a dip before increasing. If you were to look at 6 differential equations in this system, you would notice that O is used as a reactant for before it is created as a product; however, multiple reactions produce O once the combustion system starts, *i.e.*, O is first used to start the combustion but then created many more times from the other reactions. The concentration of O will decrease, then increase until a certain species is longer longer present to continue the combustion.

The overall combustion eventually stops because of the lack of O_2 and H_2 . The O_2 and H_2 species are used as reactants in the system but never created as a byproduct of any other reaction, therefore the O_2 and H_2 concentrations will slowly diminish until there is no more O_2 and H_2 in the mixture.

2.5 Accuracy of E1 and E2 Algorithms

To study the accuracy and convergence behavior of the E1 and E2 ODE solvers, let the forward rate coefficients $k_1 = 100.0$, $k_2 = 1.0$, and $k_3 = 1.0$; backward rate coefficients $\bar{k}_1 = 2.0$, $\bar{k}_2 = 0.02$, and $\bar{k}_3 = 0.02$; initial concentrations $[H_2]_o = 12$, $[O_2]_o = 3$, $[OH]_o = 0.001$; and initial and final times $T_I = 0$ and $T_F = 4$.

I first calculated a high-accuracy solution with the E2 solver with $K = 800000$ and $\Delta k = 200000 = \frac{K}{4}$. This kin.o01.txt file will contain the values $Y_n(\tau)$ for $\tau = 0, 1, 2, 3$, and 4. These will be the five “checkpoints” for calculation of an average error and corresponding nearly exact solution values as our exact reference solution in order to evaluate the error at each τ -point.

Now let’s study the accuracy of both E1 and E2 for a doubling sequence of time step number K , similar to NP01, where $K = K_o, 2K_o, 4K_o, 8K_o, \dots, 512K_o$, starting from the smallest possible K_o that gives a numerically stable solution. This numerical stable solution will have a small enough K_o to not produce an overflow in the ODE solver, representing a solution that barely resembles an approximation solution of the ODE system. So let $K_o = 500$; for every doubling of K_o , I also made sure to change $\Delta k = \frac{K}{4}$.

With each K value of the doubling sequence, I ran the E1 ODE solver and downloaded the kin.o01.txt file generated with $K = \frac{K}{4}$, then imported each file into a spreadsheet to analyze the upcoming calculations. I also ran this same sequence of K values for E2. Once the data was imported, I created Tables 2.01 and 2.02 to list results for the K values, timewidths h , error of each τ point $\left| Y_n^{(K)}(\tau) - Y_n(\tau) \right|$, and total error of E:

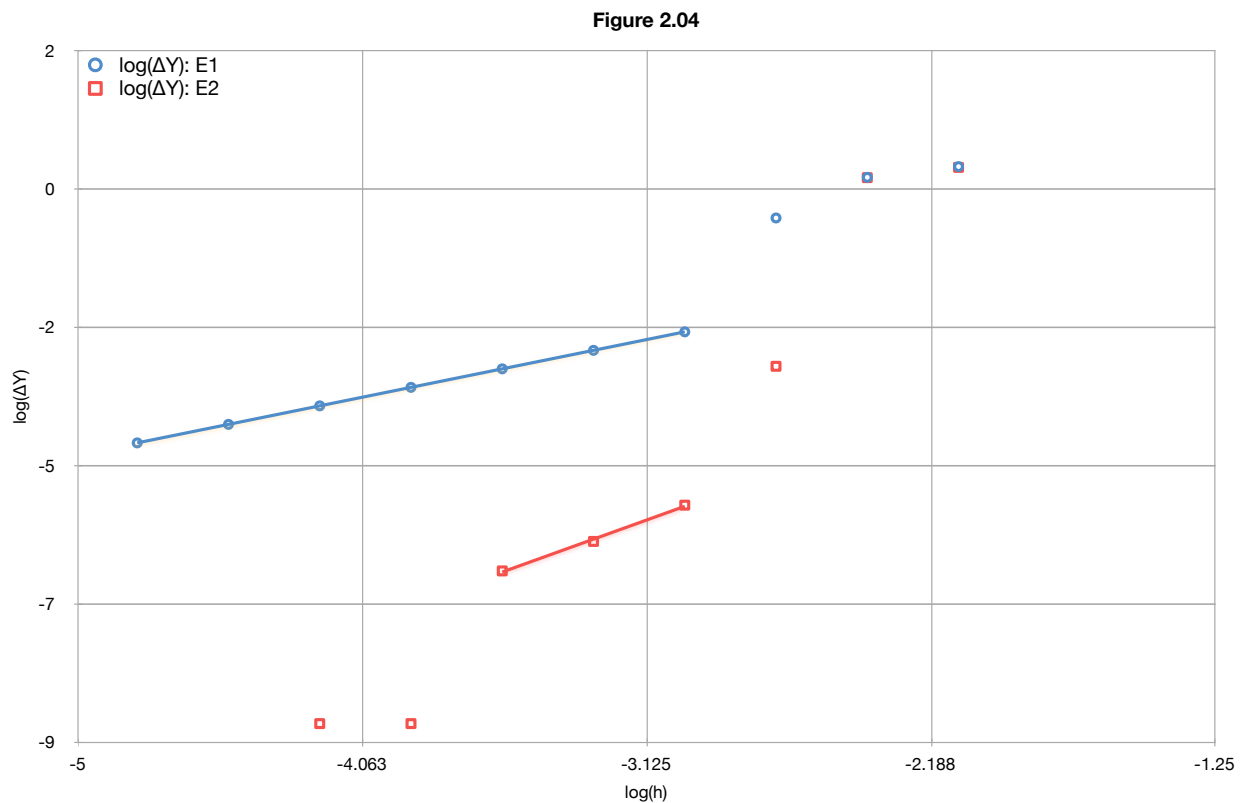
$$\Delta Y := \frac{1}{5} \sum_{\tau} \left| Y_n^{(K)}(\tau) - Y_n(\tau) \right|,$$

where $Y_n^{(K)}(\tau)$ is the approximate H_2O concentration obtained with E1 or E2, and $Y_n(\tau)$ is the high-accuracy solution state above (Note: Table 2.01 and 2.02 and the columns describing the error at various τ points is obtained from only the H_2O concentration).

Table 2.01: E1								
K-Value	Numerical K	h	Error of T=0	Error of T=1	Error of T=2	Error of T=3	Error of T=4	ΔY
1*K_0	500	0.008	0.00E+00	4.11E+00	4.28E+00	1.54E+00	1.46E+00	2.28E+00
2*K_0	1000	0.004	0.00E+00	2.08E+00	4.07E+00	8.46E-01	6.20E-01	1.52E+00
4*K_0	2000	0.002	0.00E+00	3.61E-02	1.36E+00	2.00E-01	6.91E-02	3.33E-01
8*K_0	4000	0.001	0.00E+00	4.28E-04	2.09E-02	2.10E-03	1.80E-05	4.69E-03
16*K_0	8000	0.0005	0.00E+00	2.15E-04	1.05E-02	1.05E-03	9.00E-06	2.34E-03
32*K_0	16000	0.00025	0.00E+00	1.08E-04	5.23E-03	5.24E-04	5.00E-06	1.17E-03
64*K_0	32000	0.000125	0.00E+00	5.38E-05	2.62E-03	2.62E-04	2.00E-06	5.87E-04
128*K_0	64000	0.0000625	0.00E+00	2.69E-05	1.31E-03	1.31E-04	1.00E-06	2.93E-04
256*K_0	128000	0.00003125	0.00E+00	1.35E-05	6.54E-04	6.60E-05	0.00E+00	1.47E-04
512*K_0	256000	0.000015625	0.00E+00	6.74E-06	3.27E-04	3.30E-05	0.00E+00	7.33E-05
K	K-Value	log(h)	log(ΔY)	Data for Small h for Liner Regression Lines				
1*K_0	500	-2.09691001	0.3576					
2*K_0	1000	-2.39794001	0.1829					
4*K_0	2000	-2.69897	-0.4774					
8*K_0	4000	-3	-2.3293	-3	-2.3293			
16*K_0	8000	-3.30103	-2.6299	-3.30103	-2.6299			
32*K_0	16000	-3.60205999	-2.9306	-3.60205999	-2.9306			
64*K_0	32000	-3.90308999	-3.2317	-3.90308999	-3.2317			
128*K_0	64000	-4.204119983	-3.5326	-4.204119983	-3.5326			
256*K_0	128000	-4.50514998	-3.8336	-4.50515	-3.8336			
512*K_0	256000	-4.80617997	-4.1346	-4.80618	-4.1346			

Table 2.02: E2								
K-Value	Numerical K	h	Error of T=0	Error of T=1	Error of T=2	Error of T=3	Error of T=4	ΔY
1*K_0	500	0.008	0.00E+00	4.18E+00	4.17E+00	1.38E+00	1.32E+00	2.21E+00
2*K_0	1000	0.004	0.00E+00	2.00E+00	4.07E+00	8.49E-01	6.24E-01	1.51E+00
4*K_0	2000	0.002	0.00E+00	1.30E-04	5.49E-03	8.08E-04	2.80E-05	1.29E-03
8*K_0	4000	0.001	0.00E+00	6.30E-07	2.90E-05	6.00E-06	0.00E+00	7.13E-06
16*K_0	8000	0.0005	0.00E+00	1.60E-07	7.00E-06	2.00E-06	0.00E+00	1.83E-06
32*K_0	16000	0.00025	0.00E+00	4.00E-08	2.00E-06	1.00E-06	0.00E+00	6.08E-07
64*K_0	32000	0.000125	0.00E+00	1.00E-08	0.00E+00	0.00E+00	0.00E+00	2.00E-09
128*K_0	64000	0.0000625	0.00E+00	1.00E-08	0.00E+00	0.00E+00	0.00E+00	2.00E-09
256*K_0	128000	0.00003125	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
512*K_0	256000	0.000015625	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
K	K-Value	log(h)	log(ΔY)	Data for Small h for Liner Regression Lines				
1*K_0	500	-2.09691001	0.3449					
2*K_0	1000	-2.39794001	0.1785					
4*K_0	2000	-2.69897	-2.8893					
8*K_0	4000	-3	-5.1472					
16*K_0	8000	-3.30103	-5.7371					
32*K_0	16000	-3.60205999	-6.2161					
64*K_0	32000	-3.90308999	-8.6990					
128*K_0	64000	-4.204119983	-8.6990	-3	-5.1472			
256*K_0	128000	-4.50514998		-3.30103	-5.7371			
512*K_0	256000	-4.80617997		-3.60206	-6.2161			

Once I made these tables, I generated Figure 2.04 by plotting $\log(\Delta Y)$ vs. time t for both E1 and E2, using different symbols and colors to distinguish E1 from E2.



Analogous from NP01, the error ΔY is again predicted by mathematical theory to follow a power law as $h \rightarrow 0$. We can use the same derivation as in 1.3 to arrive at the same relationship between $\Delta Y \cong Ch^p$ and our linear regressions², given by:

$$a = p, \quad b = \log(C) \quad \text{where} \quad x = \log(h), \quad y = \log(\Delta Y)$$

Similar to NP01, I calculated each p and C value and created Table 2.03 to compare the values between each algorithm E1 and E2.

Table 2.03				
	Slope: a	Intercept: b	Pre-factor: C	Exponent: p
E1	0.9996	0.6698	4.6751979088	0.9996
E2	1.7755	0.1608	1.4481048231	1.7755

Some Observations: First notice that the slope a of E2 is greater than the slope of E1. Similar to NP01 where the largest positive slope represented the most accurate approximation algorithm used, we can see that E2 is a more accurate integration method used when compared to E1, since $1.7755 > 0.9996$. This slope comes from comparing ΔY values from Table 2.01 and Table 2.02. We can see that error approaches 0 much faster in the E2 method rather than the E1 method, which is why the slope is steeper in Figure 2.04, since the same time width h is being used for each approximation.

Also notice that the error ΔY in the E2 method for $K = 256K_o$ (and larger K) is 0. So if the error is 0, then the log of the error $\log(\Delta Y) = \log(0) = -\infty$, which is undefined on Figure

²The "trendline" tool in Numbers '09 only draws a line through the endpoints of the selected data, and since we only used the smallest two h values to obtain the most accurate slope for each algorithm, Numbers '09 will draw a line only from these two points (one of the few drawbacks Numbers has compared to Excel)!

2.04 (hence absence of the last two data points for the E2 linear regression). This is most likely caused by the rounding of the concentrations of various species given from the kin.o01.txt files.

Under further inspection, I noticed that these numbers given in the kin.o01.txt files only carried decimals out 6 places. If the equation solver were to carry decimals farther out instead of rounding at 6, then our errors for each method E1 and E2 would have more nonzero answers, resulting in a better defined linear regression.