2008 年度 特別研究

u-w 法を用いた樹木の成長解析ソフトウェア開発と検証

2008 年 1 月 23 日

龍谷大学理工学部
環境ソリューション工学科
T050592
青木　将晃
宮浦　富保　教授

目次

要旨・・・・・・・・・3P

1.はじめに・・・・・4P

2.u-w 法について・・4P

3.目的・・・・・・・6P

4.方法・・・・・・・9P

5.結果・・・・・・・8P

6.考察・・・・・・・12P

7.引用文献・・・・・14P

8.謝辞・・・・・・・14P

9.付録・・・・・・・15P

u-w 法を用いた樹木の成長解析ソフトウェア開発と検証

T050592　青木　将晃
宮浦　富保　教授

１．はじめに

樹木の成長はとてもゆっくりとしたものであるが、成長の記録を年輪として幹の内側に残しながら着実に大きくなっている。樹木の年輪を調べ、その成長の仕方を曲線で近似することにより、ある程度過去の森林の姿を再現できたり、森林の将来の状態を予測したりできると考えられる。現在までに数多くの成長曲線が提唱されてきたが、どれを選べばよいのかの判断基準は存在せず、それらの成長曲線では実際のデータをうまく近似出来ない場合が多かった。

Hozumi(1985)が提唱した u-w 法で得られる成長曲線は従来の成長曲線の多くを包含しており、実際の成長データのあてはめが従来の成長曲線式よりも柔軟に行える。u-w 法では以下のように定義される u と w の関係をプロットして近似曲線を求める。

w:幹の材積（㎥）　　$u = Aw^{-c}\left(1-\left(\frac{w}{W}\right)^{m}\right)$　ただし　$u = \frac{1}{w^2}\frac{dw}{dt} = -\frac{d}{dt}\left(\frac{1}{w}\right)$

しかし、u-w 法の適用には大変な手間を必要とするため、u-w 法による計算を効率よく正確に行い、成長曲線の解析に役立てることを目的として u-w 法の計算ソフトウェアを作成し、その検証を行った。

２．方法

Visual Basic 2008 というプログラム言語を使って、Windows アプリケーションとして成長解析ソフトウェアを作成した、また 2007 年に「龍谷の森」のコナラ二次林で伐採して得られた実際の年輪データを使用して成長解析ソフトウェアの有効性を検証した。

３．結果と考察

図 1 は、開発した成長解析ソフトウェアの実行例である。データから u と w の値を計算し、それを元に自動的に近似曲線を引くことができる。

今回開発したソフトウェアでは A, c, W それぞれを一定量ずつ変化させて u の推定値と実測値の残差平方和が最小になる値を探すという方法で近似曲線を探した。多くの樹木で近似できたが、一部うまく近似できないものもあった。

定数を解析した結果、c は 1～1.4 の範囲をとることが多かった。A と c には強い負の相関が見られた。

実際の樹木の成長は複数の成長曲線で近似されることが多く、この成長曲線が変わる現象を成長曲線の乗り換えと呼ぶ。成長曲線の乗り換えが起こるごとに W が大きくなる傾向がほとんどの樹木で見られた。また成長曲線の乗り換えの年代を調べた結果、樹種ごとに特徴があることが分かった。コナラでは 1970 年代に乗り換えが集中していた。

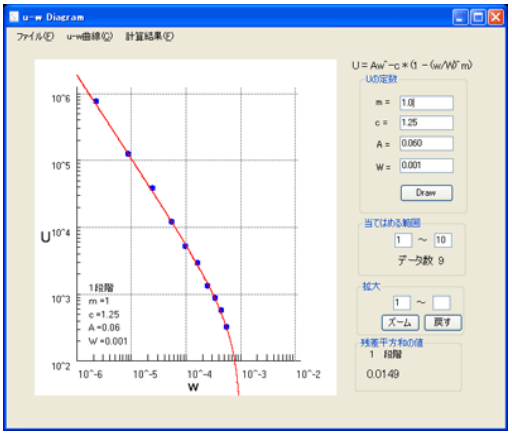


図 1,今回開発した u-w 図を描いて計算するソフトウェア

４．結論

ソフトウェア開発により、u-w 法による樹木の成長解析が効率的に行えるようになった。

u-w 曲線式は定数の変動幅が大きく、今回開発したソフトウェアで使ったアルゴリズムでは全ての曲線式に対応できなかったため今後の課題となった。

c は 1～1.4 の範囲をとることが多かったが、これは多くの場合ロジスティック曲線、指数関数曲線、ミッチャーリッヒ曲線などの有名な成長曲線式のいずれにも当てはまらないと言えるので、u-w 法を使った方がより正確な成長曲線を導き出せると考えられる。

u-w 図において複数の成長曲線で分けたほうがきれいに近似できるので、多くの樹木は成長曲線の乗り換えを何度も行いながら成長していると考えられる。

1.はじめに

樹木の成長はとてもゆっくりとしたものであるが、成長の記録を年輪として幹の内側に残しながら着実に大きくなっている。樹木の年輪を調べ、その成長の仕方を曲線で近似することにより、ある程度過去の森林の姿を再現できたり、森林の将来の状態を予測したりできると考えられる（宮浦,2002）。

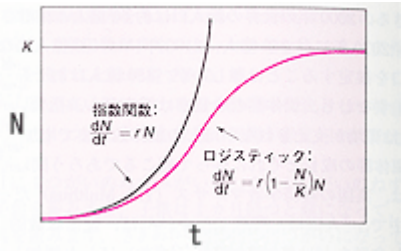


図１.成長曲線の例
日本生態学会（2004）より

個体サイズの変化の時間的経過を定量的にあらわす曲線を、一般的に成長曲線とよぶ。樹木の成長のしかたを記述する方法として、現在までに数多くの成長曲線が提唱されてきた。例えば実際によく使われるものとしては図１のように指数関数的に成長する指数曲線や、途中まで指数関数的に成長し上限値に近づくと成長が止まっていくロジスティック曲線などがある。図１で t は時間、N は個体数や個体サイズなど、K は N の上限値である。しかし数多くある成長曲線のうちどれを選べばよいのかという点についての合理的な判断基準は存在せず、それらの成長曲線では実際のデータをうまく近似出来ない場合が多かった（宮浦,1990）。

2.u-w 法について

u-w 法は一本の木の幹材積の成長解析方法として Hozumi(1985)によって提案された。
ここでいくつかの変数を定義する。

t：時間	[year]
w：幹材積(幹部分の体積)	[m^3]
W：w の上限値	[m^3]
v：成長速度、$\frac{dw}{dt}$	[$m^3 \cdot year^{-1}$]
s：相対成長速度、$\frac{1}{w}\frac{dw}{dt}$	[$year^{-1}$]
u：$\frac{s}{w} = \frac{v}{w^2} = -\frac{d}{dt}(\frac{1}{w})$	[$m^{-3} \cdot year^{-1}$]

u-w 法では、u をｗの関数として次式で表わす（Hozumi,1985）

$$u = Aw^{-c}\left(1 - \left(\frac{w}{W}\right)^m\right) \quad \cdot\cdot\cdot(1)$$

u と w の関係を表す u-w 曲線は従来の成長曲線の多くを包含するため、実際の成長データのあてはめが従来の成長曲線式よりも柔軟に行える。また生物的な根拠をもった方法であり、あてはめの結果得られる係数や定数はなんらかの生物的な意味を持っていることが期待される(宮浦,1990)。

この u-w 曲線式の A、c、m は定数である。また W は w の上限値である。(1)式の m,c,W の取り方によっては指数曲線、ロジスティック曲線、ミッチャーリッヒ曲線などの成長方程式を得ることができる。

実際の使い方の例としては以下の通りである。すなわち（1）式において例えば W=1 に固定しておき、m=1～２、c=-2～2 程度の範囲で w と c の値を何通りか変えて、両対数グラフ上に u-w 曲線を描く、この曲線をトレーシングペーパーに転写したものを定規として用いる（図 4)。次に表 1 のように年輪の樹齢・幹材積のデータから算出した u と w を両対数グラフ上にプロットし、作成した定規を重ねて最もよく適合する曲線を選ぶ。そして選んだ曲線から残りの定数である W と A を算出して成長曲線式を決定する。

図 2 は u-w 図をプロットした例である、実際にはこのように一本だけでなく複数の曲線で近似することが多いが、この曲線が変わる現象は成長曲線の乗り換えと呼ばれる。図 3 は幹材積の成長経過を片対数グラフにプロットしたものである。図 2 の方が成長曲線の乗り換えがはっきりとわかる。

u-w 法を用いて得られる係数や定数の値を、気象条件や林分構造の側面から吟味することにより、樹木の成長に関する内的な要因と外的要因の効果をある程度分離して理解できると期待されている（宮浦,1990)。

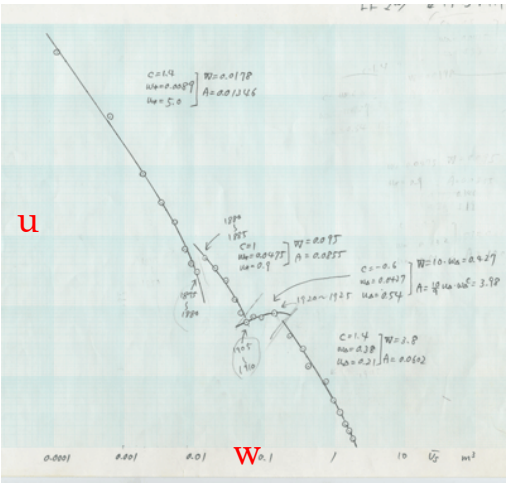


図 2.u-w 図

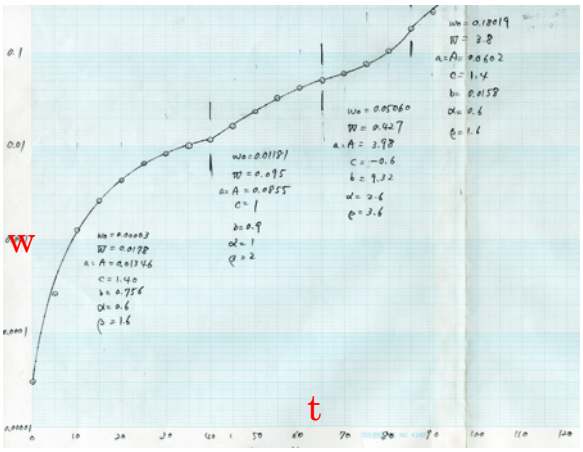


図 3.時間ごとの幹材積

表 1.u と w の計算例

樹齢	幹材積[m³]	u [m^{-3}・year^{-1}]	w [m³]
15	0.00092		
		139.3	0.00261
20	0.00567		
		21.9	0.01097
25	0.01885		
		6.42	0.0345
30	0.05706		

u と w の値は年齢を t とすると t_2-t_1 の間

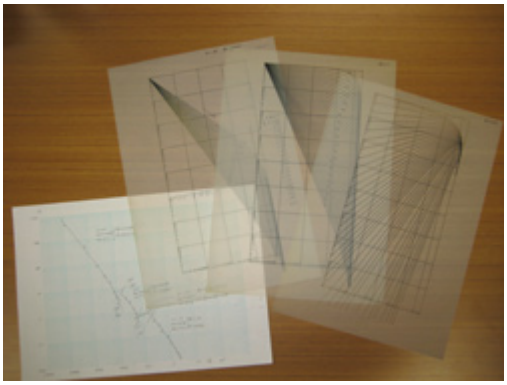


図 4.c=1.1、1.2、1.3…と変化させたグラフを描いた紙曲線の形を決めるときに使う。

の近似値であり、u≒u、w≒w である

3.目的

u-w 法を上記のように手作業で適用する方法は、データが大量にある場合、合うグラフを探すのにとても時間がかかってしまい、また目視でグラフを選ぶため良い判断基準がないという欠点がある。そのため u-w 図の作成を効率よく正確に行い、成長曲線の解析に役立てることを目的として u-w 法の計算ソフトウェアを作成する。次に実際にそれを使って年輪データの解析を行い、成長曲線を正しく計算できるか検証し、u-w 法について考察する。

4.方法

4-1 ソフトウェア開発

本ソフトウェア開発には Microsoft Visual Basic 2008 Express Edition を使った。またこれを使用して実際に成長解析を行った。Visual Basic は Windows 用のソフトウェア（GUI アプリケーション）を比較的簡単に作れるという特徴がある。　図 5 はソフトウェアのソースコードの一部である。

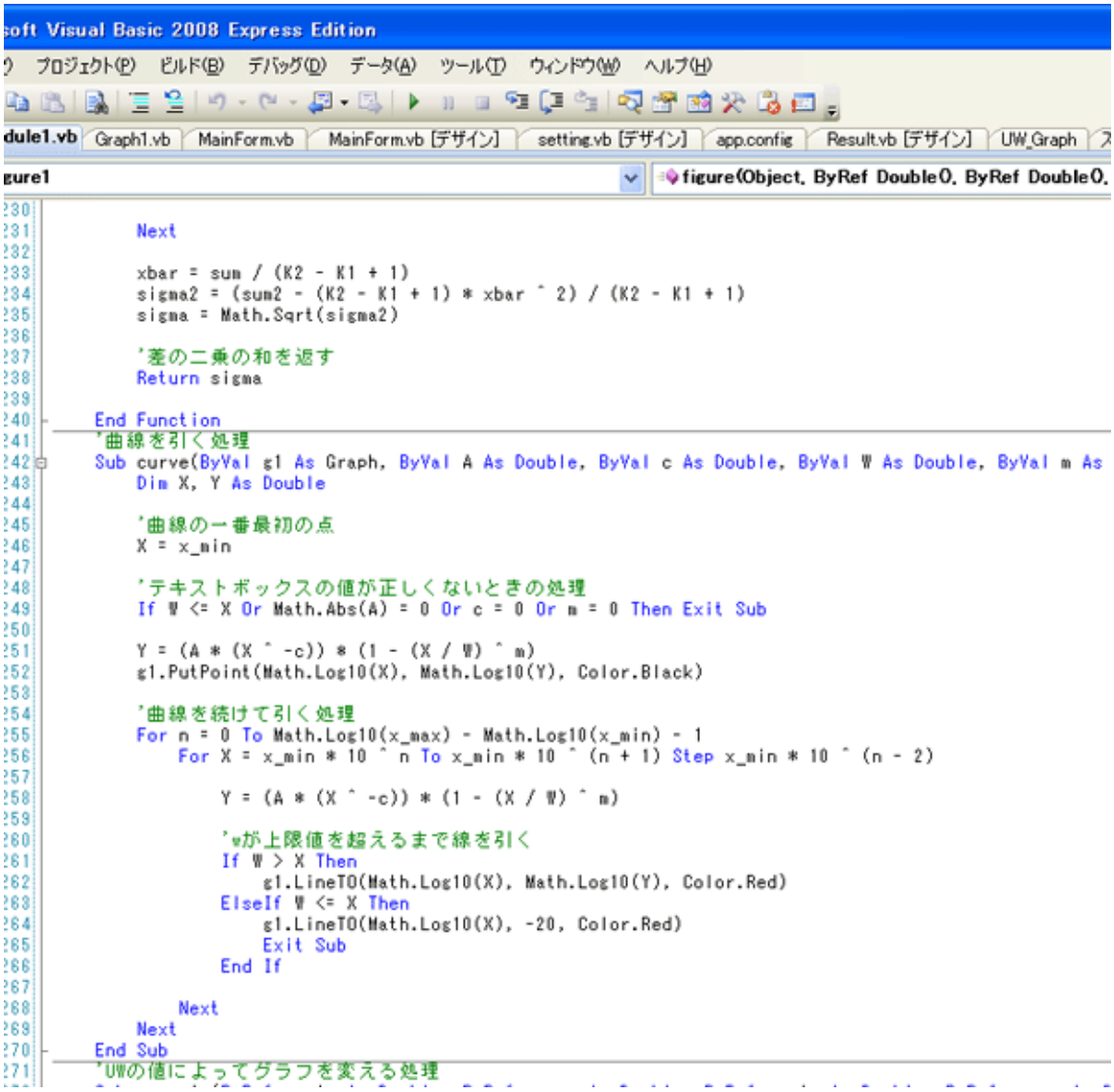


図 5.Microsoft Visual Basic 2008 による開発画面

4-2.プログラムの構造

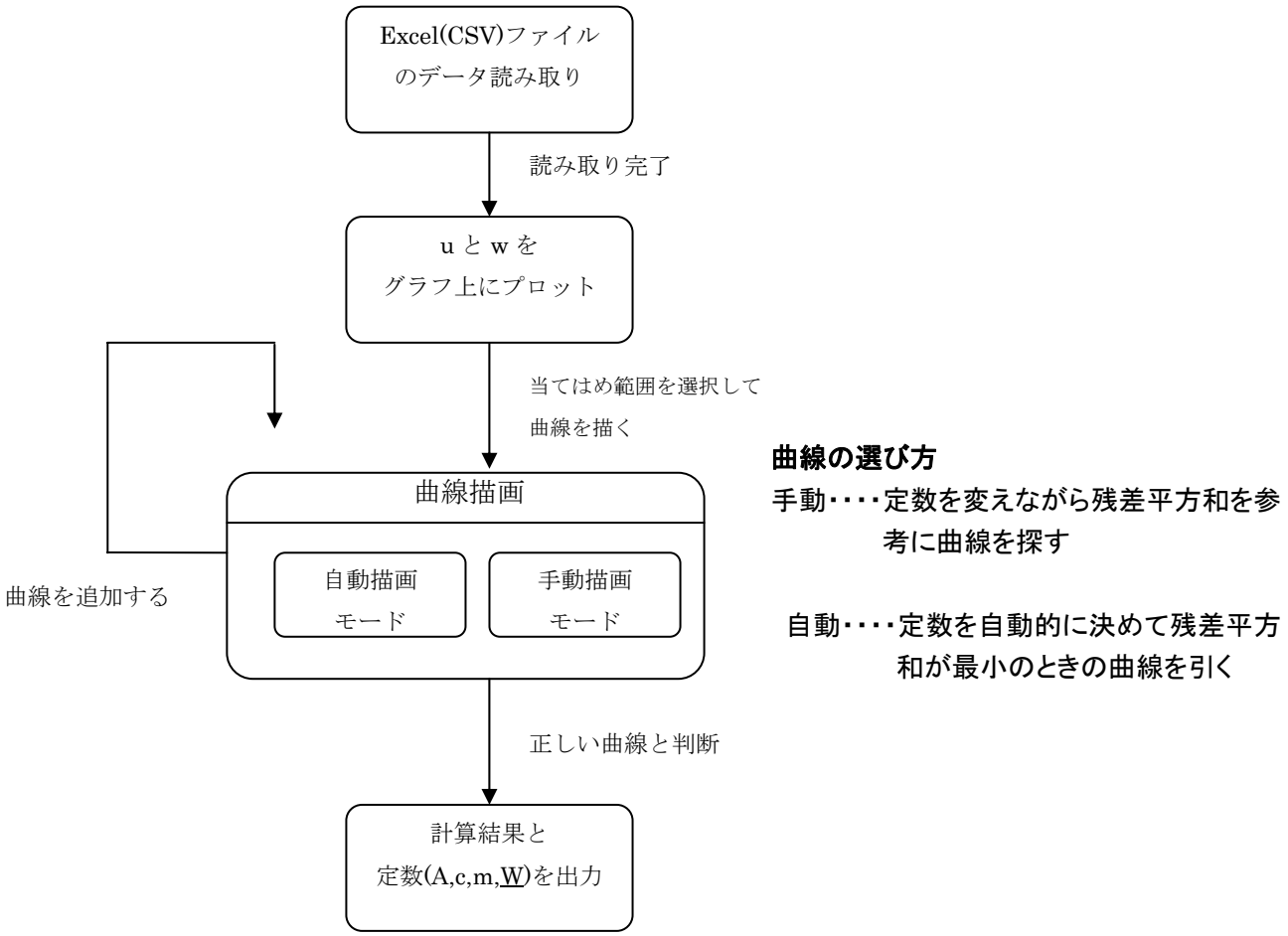


図 6.状態遷移図

図 6 はソフトウェアの動作を示した状態遷移図である。まず Excel ファイル（CSV ファイルでも可）から年輪データを読み取り、u と w を計算してプロットする、次に曲線を選び、最後に計算で得られた結果とグラフを保存するしくみとなっている。

Excel ファイルには 1 列目に樹齢、2 列目に樹齢に対応した w があるようにする。曲線を選ぶには、手動で選ぶ方法と自動で選ぶ方法がある。手動で選ぶ方法は定数の値をグラフの変化を見ながら入力し、よく当てはまるグラフを選ぶ。

自動的に曲線を引く方法は、A,c,w を一定量ずつ変化させ残差平方和が最小のときの A,c,w を返して曲線を引くというしくみとなっている。つまり S が最小のときが最も当てはまる曲線であるというようにした。キザミ幅は表 2 のようにした。ただし m はほとんどの場合 1 に近い値をとるので、ここでは常に 1 として計算している。S は u の実測値と推定値の残差平方和であり、以下の（2）式で表す。

$$S = \sum_{i=1}^{n} (u_n - f(w)_n)^2 \qquad (2)$$

u_n ：表 1 で計算した u

f(w)：(1)式にu_n のときの w と定数を適当にあてはめた u の推定値

表 2.曲線を自動で選ぶときの定数のキザミ幅

定数	初期値	加算値	終了値
A	0.01	0.01	3
c	-1	0.05	2
W	10^($\log_{10} w_{\min}$)		10^($\log_{10} w_{\min}$ + i×s)

$$s = \frac{\log_{10} w_{\max} - \log_{10} w_{\min}}{n}$$

i = 0、１、２ … n

表 2 の n は S×10 として、つまり 10^n から 10^n+1 までを等間隔で 10 分割するようにプログラムを作成した。n の値を変えれば W についてのキザミ幅を変えることができる。また u-w 図は図 2 のように複数の曲線で近似できることが多いので、複数本引けると判断した場合、別の曲線を追加できるようにもした。開発したソフトウェアのソースコードは付録に載せてある。

5.結果

5-1 開発したソフトウェア

図 7 はソフトウェアを起動した直後の状態である。[ファイル]ボタンを押すと使うファイルを選ぶ画面が表示される、選んだ Excel ファイルから樹齢と幹材積のデータを読み取り u と w が計算される。図 8 はデータファイルの例である。読み込んだデータは図 9 のようにグラフにプロットされる。

右の[Draw]ボタンを押すと曲線が引かれる、右上のテキストボックスに定数 A,c,m,W それぞれを入力するとそれに対応する曲線が引ける。[u-w 曲線]から[自動描画]を押すと自動的に近似曲線が引かれる、[曲線追加]を押すと曲線を追加できる。あてはめるデータの範囲は右のテキストボックスで選択できる、何も変えなければ全ての範囲が選択される。図 10 の右下の 0.0149 は残差平方和 S であり、この数値が小さくなるような曲線を選べばうまく近似することができる、自動描画では S の最小値を探して自動的に曲線を引くことができる。

グラフ画面の大きさはフォームの大きさを変えると自動で変わるようになっている。右下の[ズーム]ボタンを押すとグラフの一部を拡大できる、データが狭い範囲に集まっている場合はこれを使えば作業がしやすくなる。

[計算結果]を押すと図 11 のように現在計算したデータの u,w,A,c,m,W の値が Excel に出力される。また[ファイル]から[保存]を選ぶとグラフ画面も画像として保存することができる。

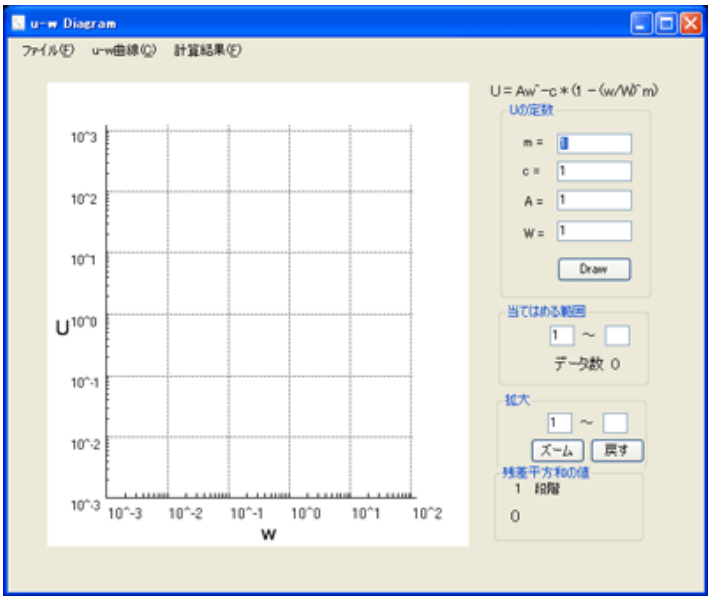


図 7.ソフトウェア起動時

	A	B	C
1	t(year)	w(m³)	
2	1988	1.82E-07	
3	1989	6.41E-07	
4	1990	4.67E-06	
5	1991	1.41E-05	
6	1992	2.09E-05	
7	1993	2.62E-05	
8	1994	3.58E-05	
9	1995	4.7E-05	
10	1996	5.68E-05	
11	1997	7.18E-05	
12	1998	8.42E-05	
13	1999	9.41E-05	
14	2000	0.000103	
15	2001	0.000116	

図 8.データファイルの例

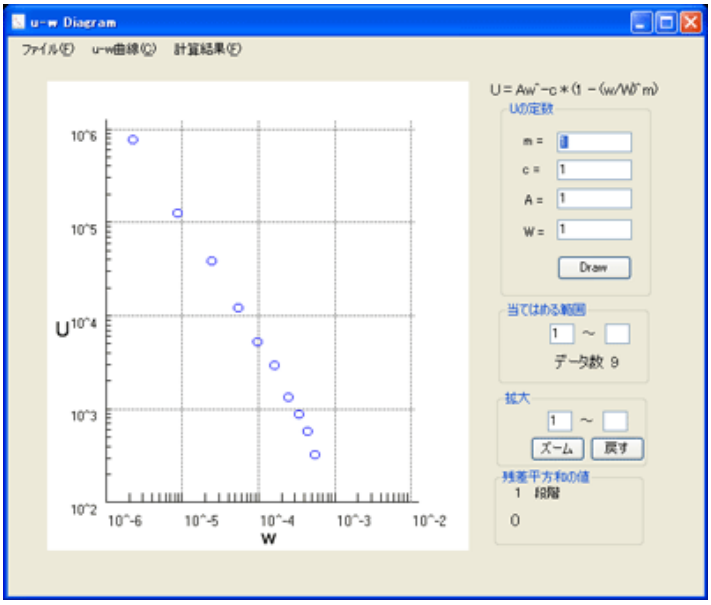


図 9.ファイルからデータを読み込んでプロット

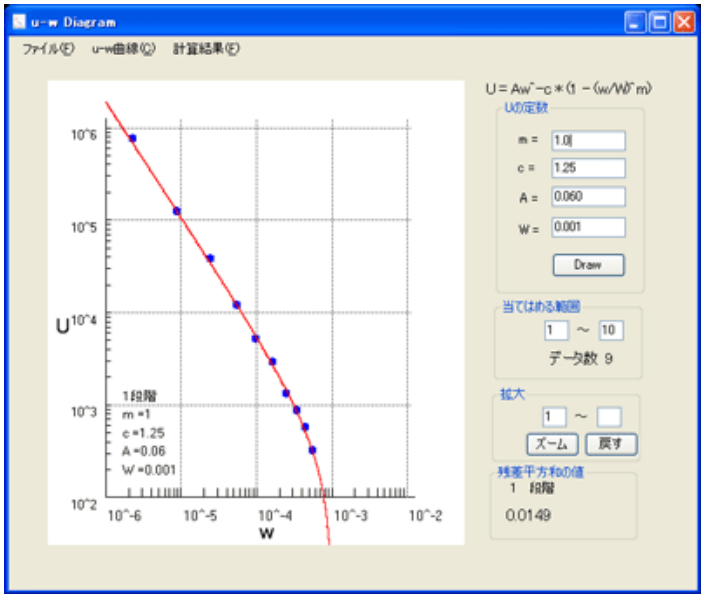


図 10.自動描画機能で曲線を引いたところ

	A	B	C	D	E	F
	w	u	A	c	W	m
1	2.23E-06	778365.2				
2	8.72E-06	129042.1				
3	2.45E-05	39540.87				
4	5.45E-05	12285.17				
5	9.82E-05	5310.362				
6	0.000163	3030.066				
7	0.000245	1361.589				
8	0.000337	904.7535				
9	0.000446	581.8031				
10	0.000557	330.5071	0.06	1.25	0.001	

図 11.計算結果を出力

5-2.成長解析

ソフトウェアの検証には、2007 年に龍谷の森で伐採して年輪解析した合計 45 本の年輪データを使用した。それぞれ u-w 曲線と定数を求めた後、w の成長曲線である w-t（樹木の幹材積と時間）のグラフを作成した。

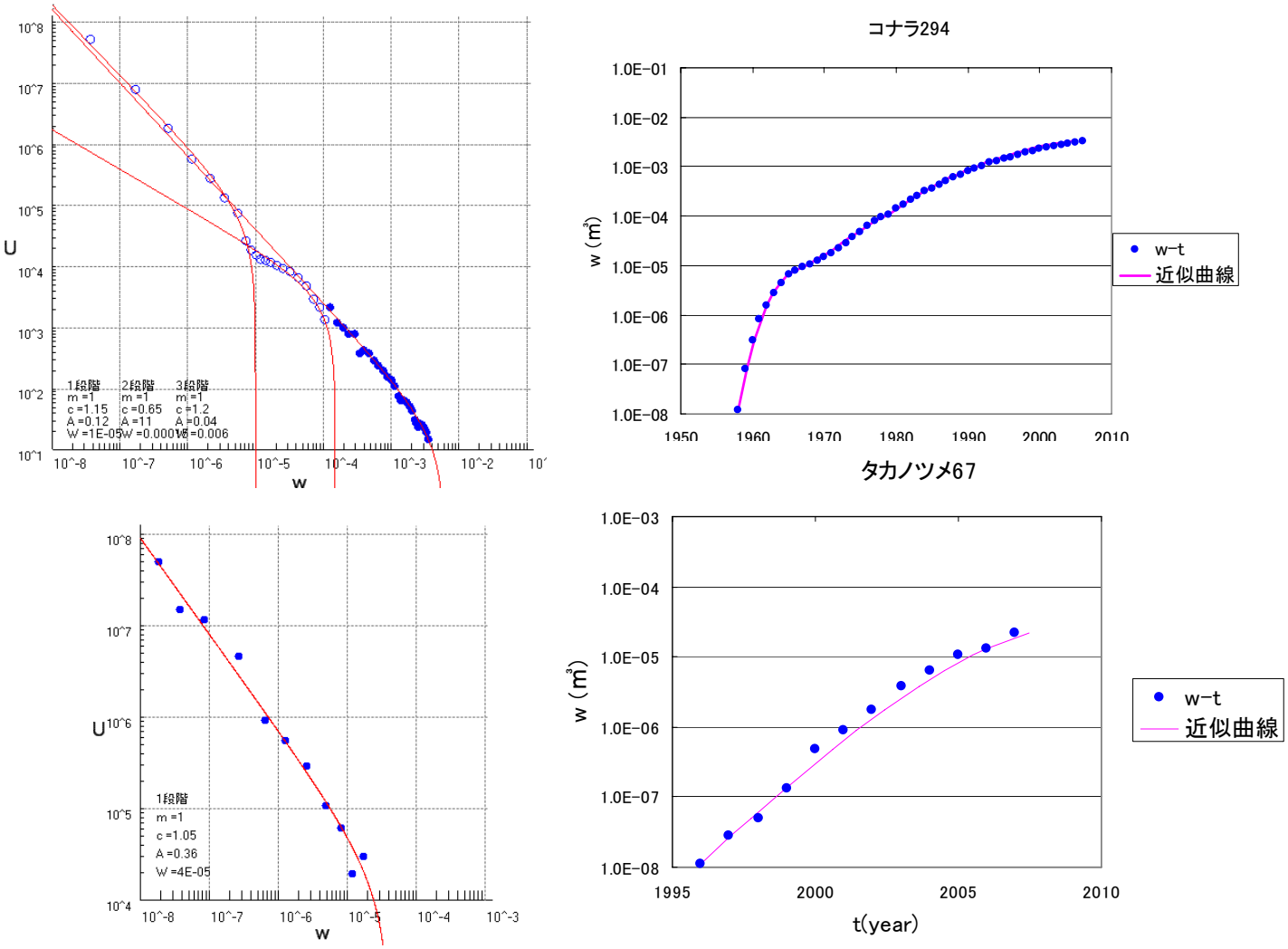


図 12.コナラとタカノツメを解析した結果
左が u-w 図で右は w の成長である、このコナラ 294 では 3 段階に分けて成長曲線をあてはめた。

図 12 のコナラの場合データは u-w 関係でも w-t 関係でもほぼ満足できる程度に曲線近似された。一方の場合,u-w 関係はおおむねよく近似されたが、w-t 関係につては、充分な近似とは言い難しい。

5-3.R による検証

R とは統計解析を行うためのオープンソースのソフトウェアである。R の非線形最小二乗法の機能を使って u と w から各定数を導き出した。初期値には今回開発したソフトウェアで計算した A,c,W の値を用いた。表 3 は開発したソフトウェア（初期値）と R で求めた推定値である（m の値はすべて 1 で統一してある）。

表 3,開発したソフトウェアと R による推定値の定数の比較

樹種 No.	年齢[yr]	開発したソフトウェア（初期値）			R		
		A	c	W	A	c	W
635	0～6	0.02	1.35	0.0000700	0.0330	1.320	0.0000531
391	0～10	0.01	1.35	0.0000800	0.000700	1.537	0.000320
660	0～23	0.01	1.4	0.00200	0.0201	1.322	0.00148
660	24～46	0.27	1	0.0900	0.711	0.847	0.0545
660	47～57	0.56	0.35	0.300	0.205	0.732	0.427
35	0～13	0.01	1.35	0.0000800	0.406	1.0600	0.0000473

今回開発したソフトウェアにより、u-w 曲線のデータのあてはめがかなり効率よく行えるようになった。このソフトウェアによって得られる係数値 A,c,W は第一近似的なものである。データへの近似の精度をあげるために、このソフトウェアで推定された A,c,W の値を初期値として、統計解析ソフト R の非線型最小二乗法を用いて、さらにあてはめを行った。なお、この場合も m=1 とした。45 本のデータのうち 41 本については、今回開発したソフトウェアで推定された係数値よりも適合度の高い推定値をえることができなかった。4 本については、R の非線形最小二乗法により表 3 に示すような推定値を得た。すなわち 45 本中 41 本については、今回開発したソフトウェアにより、十分な精度の A,c,W の推定値を得ることができたと言える。

6.考察

今回開発したソフトウェアでは、手動に頼ることなく全てのプロットデータを一度に自動で曲線近似できることが一番望ましかったが、そのようなアルゴリズムの関数は大変困難であったため実現できなかった。また定数のキザミを細かくすると計算時間が膨大な量になってしまうため、今回は表 2 で示したようにキザミ幅をある一定量変化させて定数を探す方法をとった。一部このキザミ幅で対応できない場合もあったが、手動で定数を選ぶ方法を使えば全ての数値に対応することは可能である。よって自動的に曲線を求め、それがうまくいかなかった場合は手動であてはめるという使い方が良いと思われる。成長曲線の乗り換えについてははっきりとした基準がなくプログラムで判断するのは困難なのでユーザーの判断に任せる形となった。

今回開発したソフトウェアを使用して 2007 年に龍谷の森で伐採して年輪解析した 42 本の年輪データを解析した結果、多くの曲線はうまく近似することができたが、一部のデータではこの方法でうまく近似できないものもあった。w-t 曲線に直した場合も同様に多くの場合はうまく近似できたが、一部プロットデータのずれたものがあった。しかし u-w 図で少ない曲線であてはめた場合と多くの曲線であてはめた場合では、後者の方が w-t に直したときにきれいな近似曲線を引けることが多かった。図 13 と図 14 は同じコナラを使ってあてはめ方を変えた時のグラフである。

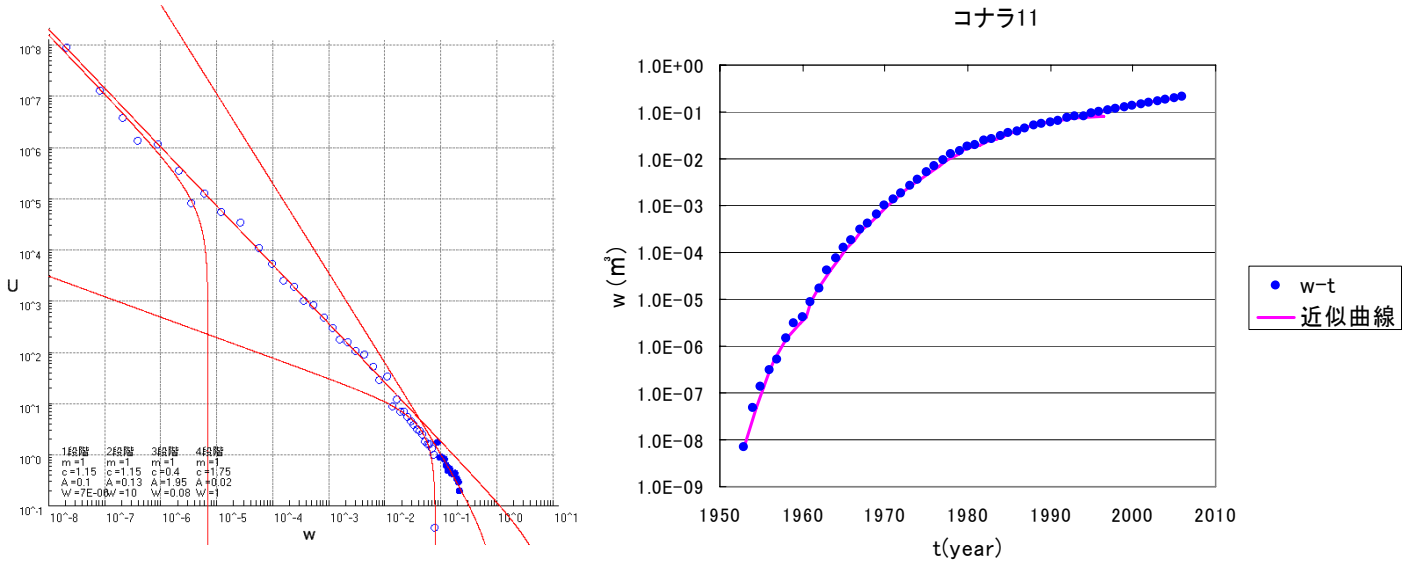


図 13.u-w 図で曲線を 4 段階に分けた場合

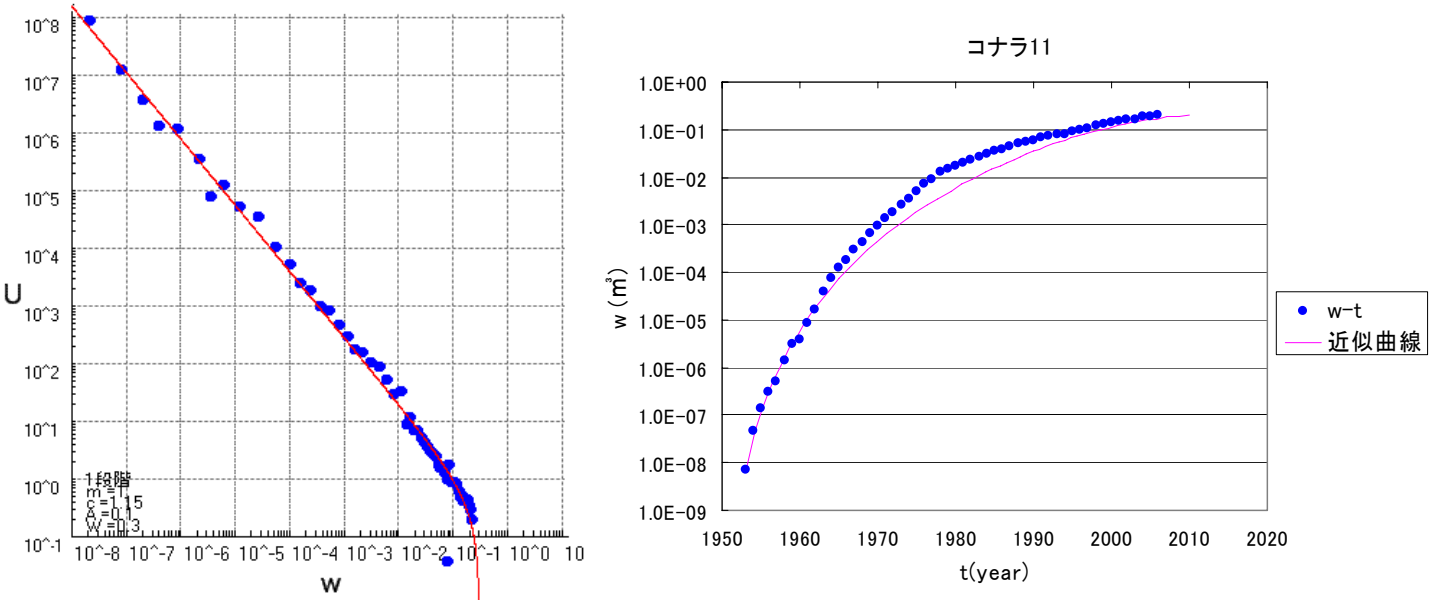


図 14.u-w 図で一本の曲線で近似した場合

定数 c は 1～1.4 の値をとることが多かったが、その範囲を大きく超えるものもまれにあった。特に A については変動幅が大きかったので上記の方法では課題が残る形となった。A と W については、A の値が大きくなると、u-w 曲線は全体が上方向に移動する。また W の値が大きくなると、u-w 曲線は右斜め下方向に移動する特徴がある（宮浦,1990）。これをうまく利用するアルゴリズムを組めば多くの数値に対応でき、より正確に成長解析ができるのではないかと考えられる。

また今回計算した定数では c は 1～1.4 の範囲をとることが多かったため、多くの場合ロジスティック曲線、指数関数曲線、ミッチャーリッヒ曲線などの有名な成長曲線式のいずれにも当てはまらないと言える。そのため u-w 法を使った方がより正確な成長曲線を導き出せると考えられる。成長曲線の乗り換え時期をまとめた結果、若い樹種は一本の曲線で近似できたものが多かったが、50 年生を超えるような大きなものはほとんどの樹木に乗り換えが発生しており、90 年生を超えるアカマツでは何度も乗り換えが発生していた。また成長曲線の乗り換えが起きるごとに W が大きくなる傾向がほとんどの樹木で見られたことと、u-w 図でも多くの曲線をあてはめたほうがうまく近似できたことから、多くの樹木は成長曲線の乗り換えを何度も行いながら、つまり成長曲線を何度も変えながら成長していくと考えられる。

図 15 はウワミズザクラとコナラの成長曲線の乗り換えが発生した年を一本ごとに示した図である。線の始まるところが樹木の発生した年である。乗り換えの発生した年をまとめた結果、全体ではどの年代でもまんべんなく乗り換えの発生があったが、コナラとウワミズザクラでは図 15 のように 1970 年代に乗り換えが多く発生しているという特徴が見られた。1978 年には松枯れ病の被害が全国ピークを示しており、滋賀県では 1970 年頃から被害が増加したという報告があった（引用）。そのため 70 年代に多く発生した乗り換えは松枯れ病によるアカマツの衰退によって、他の樹種の光条件等が改善されたためだと考えられる。今後 u-w 法を用いた今回開発したソフトウェアを使ってより多くの樹種を解析することにより、乗り換えの発生状況がさらに正確に分かってくるのではないかと期待できる。

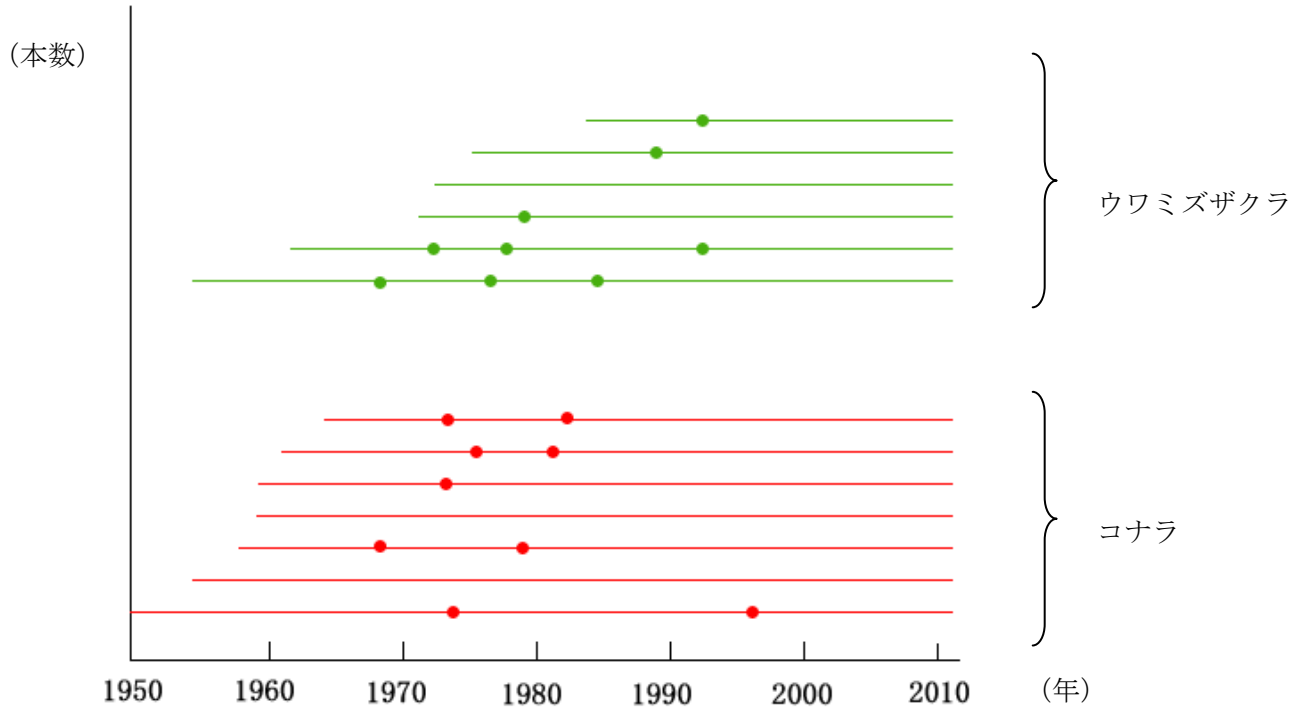


図 15.コナラとウワミズザクラの
年代ごとの乗り換え発生状況

７.引用文献

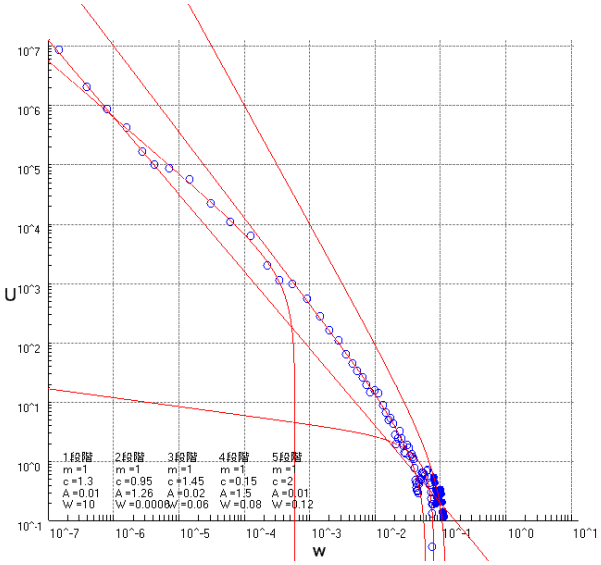
宮浦富保（1990）：新しい成長解析法「u-w 法」の紹介と応用例.北方林業 Vol.42 No5 129-135
宮浦富保（2002）：樹木の成長について.龍谷理工ジャーナル 14 巻 2 号 4-11
Kazuo Hozumi（1985） Phase Diagrammatic Approach to the Analysis of Growth Curve Using the u-w Diagram. Bot Mag Tokyo 98 239-250
日本生態学会（2004）：生態学入門、東京化学同人出版
松枯れ病（2009 年 1 月閲覧）：http://ajisime.web.fc2.com/Matugare/Matugare.htm

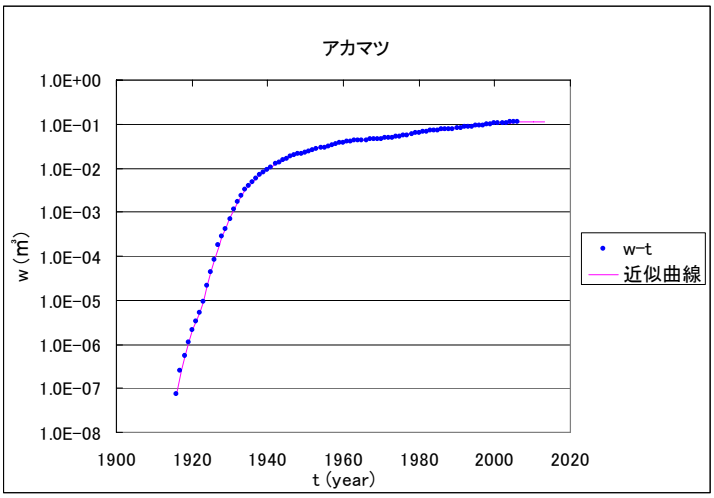
８.謝辞

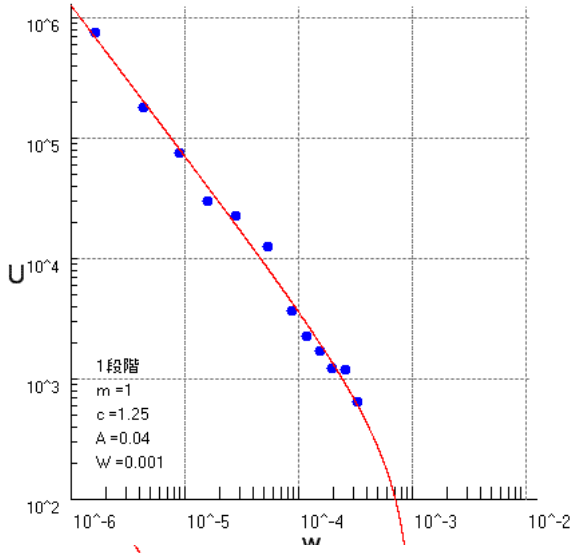
本論文の作成にあたり、方針と助言を与えてくださった龍谷大学理工学部環境ソリューション工学科の宮浦富保教授に心より感謝します。

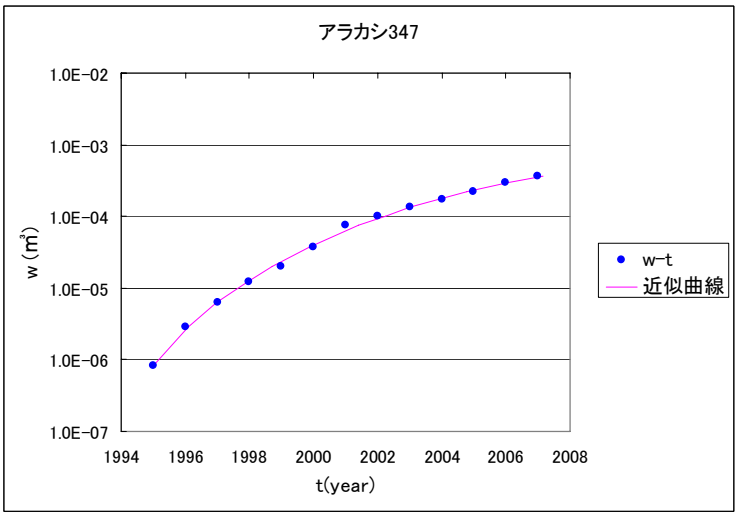
付録１

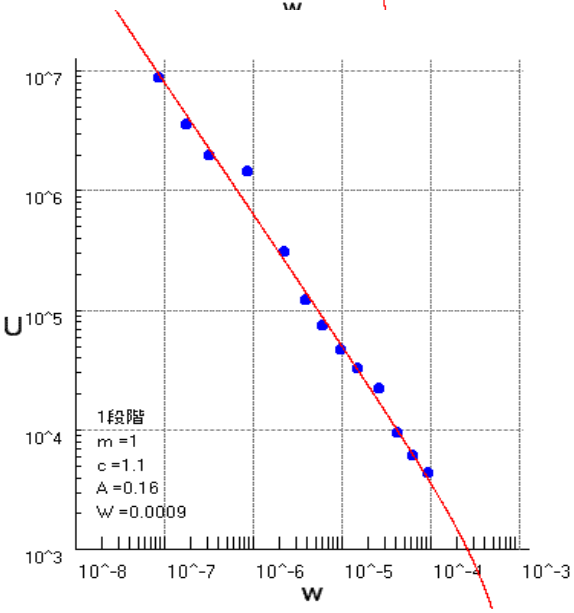
左が今回開発したソフトウェアで描画した u-w 図、右がそれを w と t の関係に直したグラフ。

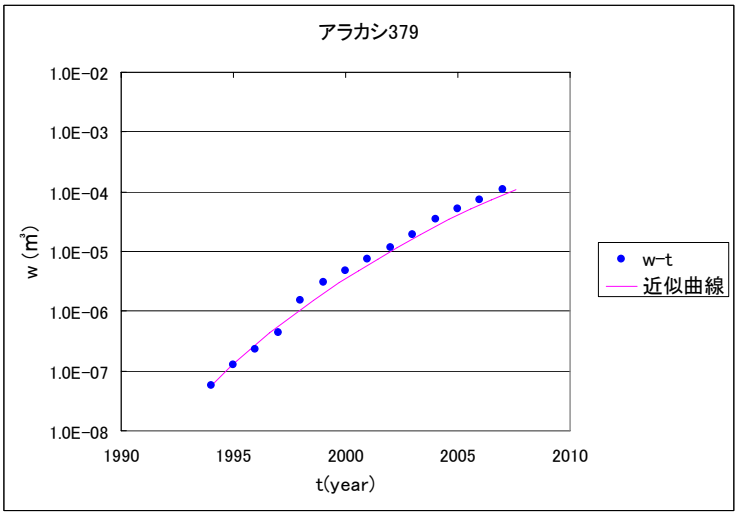


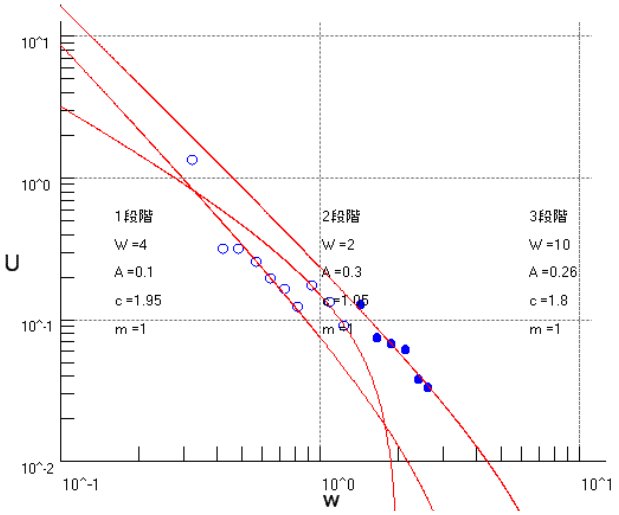


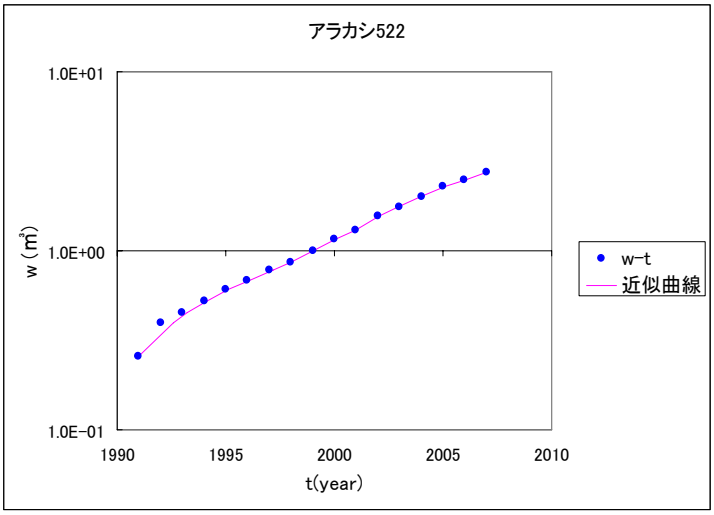


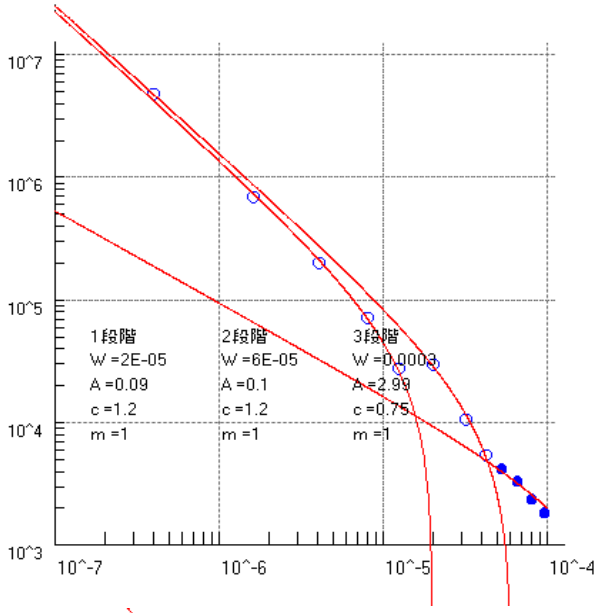


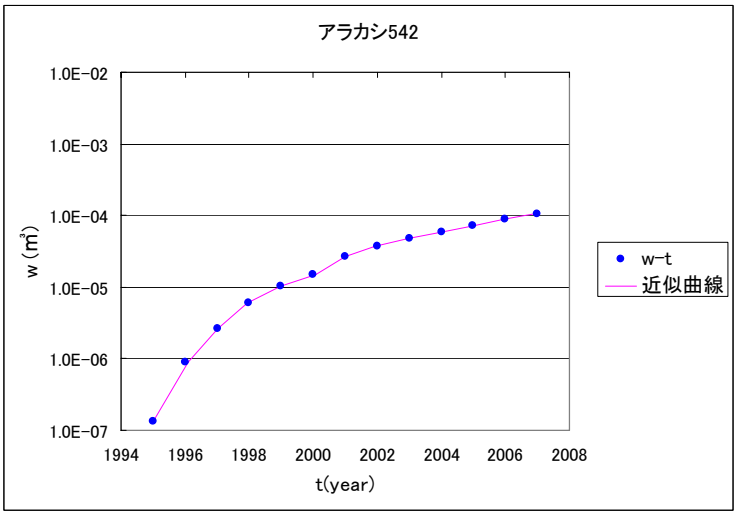


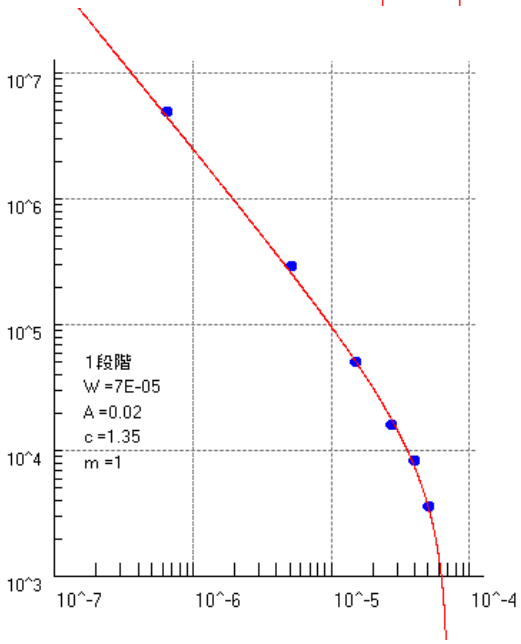


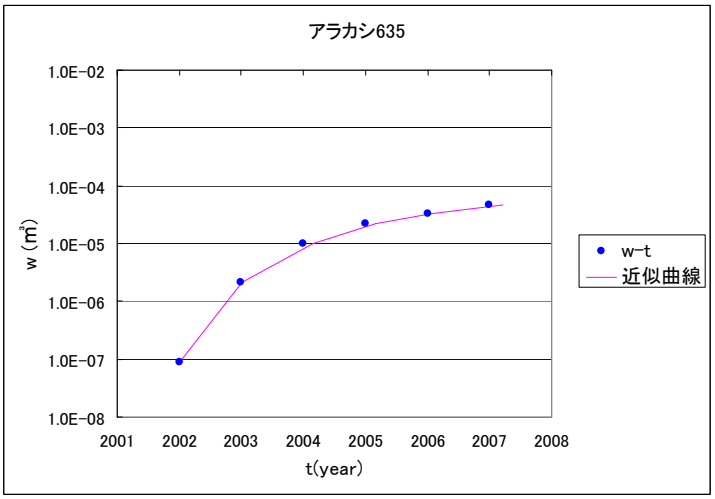

U
W
1段階
W =4
A =0.1
c =1.95
m =1
2段階
W =2
A =0.3
c =1.05
m =1
3段階
W =10
A =0.26
c =1.8
m =1

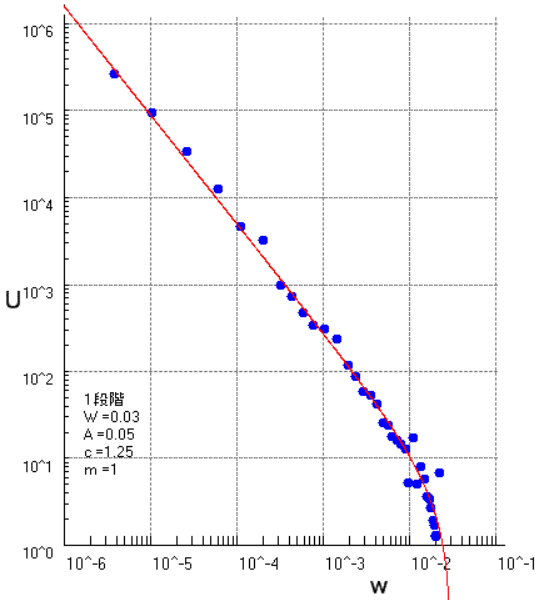

アラカシ522
w (m³)
t(year)
w-t
近似曲線

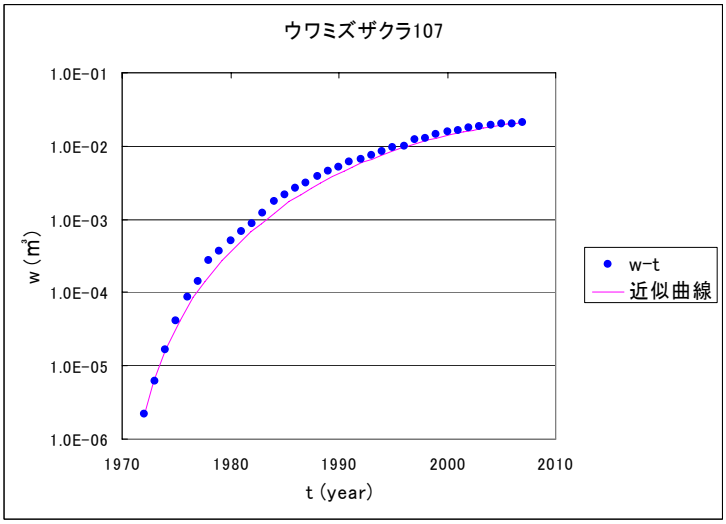

1段階
W =2E-05
A =0.09
c =1.2
m =1
2段階
W =6E-05
A =0.1
c =1.2
m =1
3段階
W =0.0003
A =2.99
c =0.75
m =1

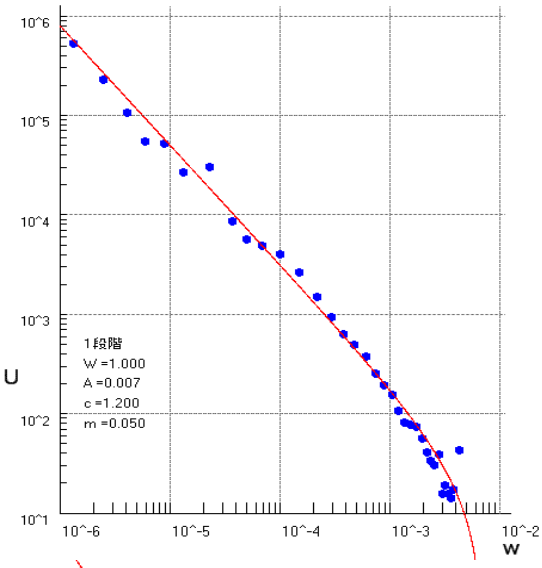

アラカシ542
w (m³)
t(year)
w-t
近似曲線

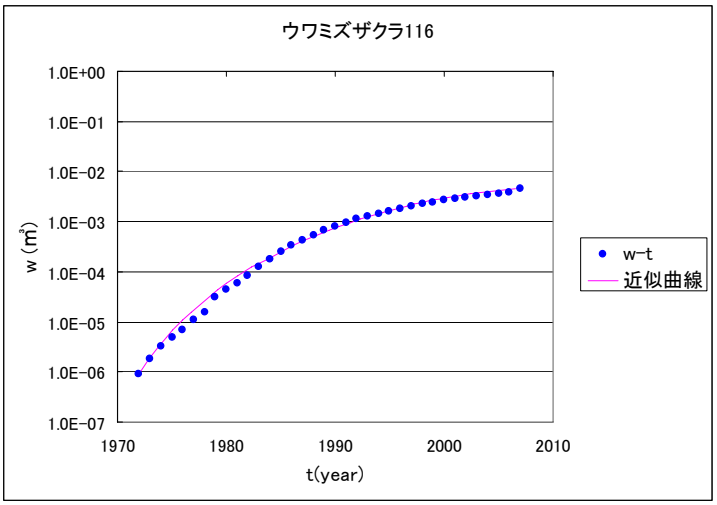

1段階
W =7E-05
A =0.02
c =1.35
m =1

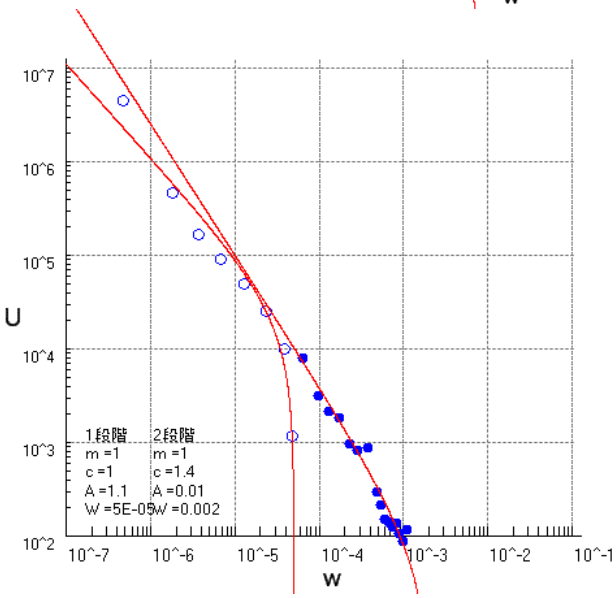

アラカシ635
w (m³)
t(year)
w-t
近似曲線

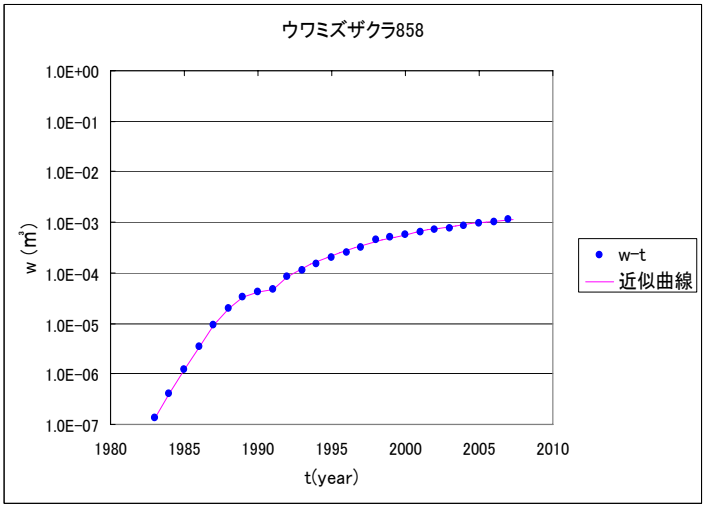

10^6
10^5
10^4
10^3
10^2
10^1
10^0
U
1段階
W =0.03
A =0.05
c =1.25
m =1
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
w

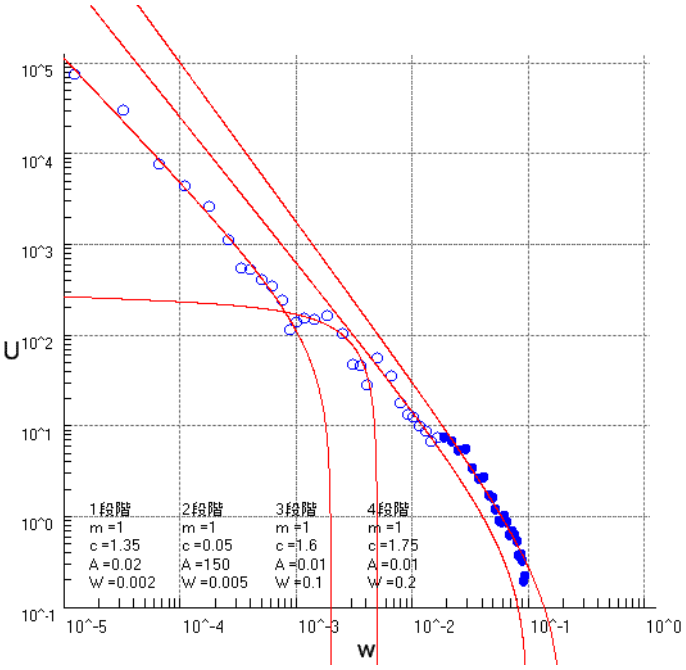

ウワミズザクラ107
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
w (m³)
1970
1980
1990
2000
2010
t (year)
w-t
近似曲線

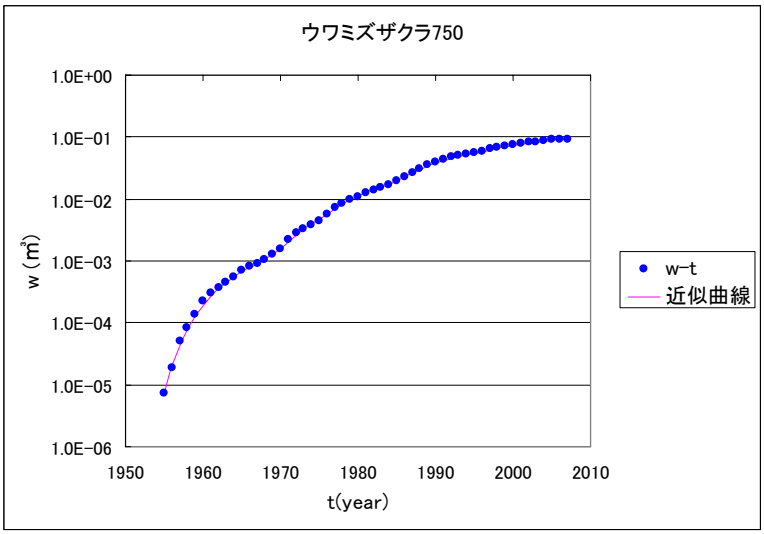

10^6
10^5
10^4
10^3
10^2
10^1
U
1段階
W =1.000
A =0.007
c =1.200
m =0.050
10^-6
10^-5
10^-4
10^-3
10^-2
w

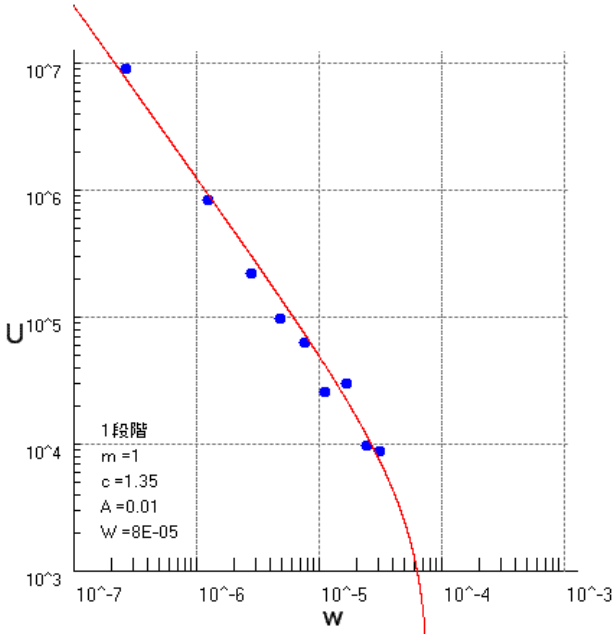

ウワミズザクラ116
1.0E+00
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
w (m³)
1970
1980
1990
2000
2010
t(year)
w-t
近似曲線

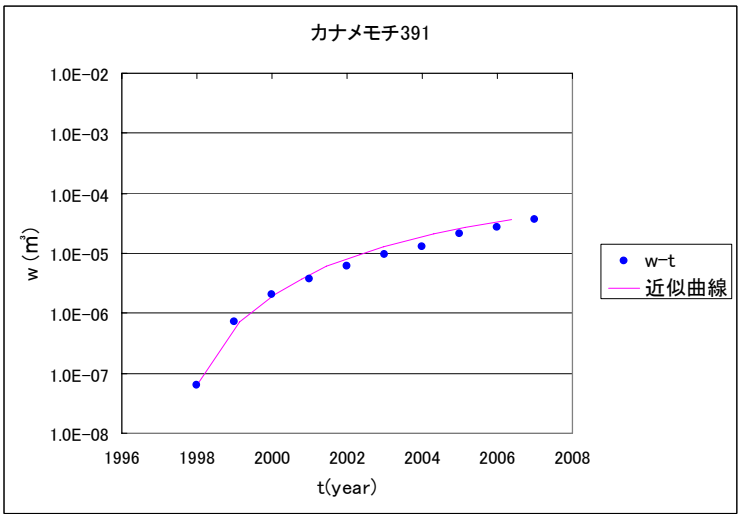

10^7
10^6
10^5
10^4
10^3
10^2
U
1段階
m =1
c =1
A =1.1
W =5E-05
2段階
m =1
c =1.4
A =0.01
W =0.002
10^-7
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
w

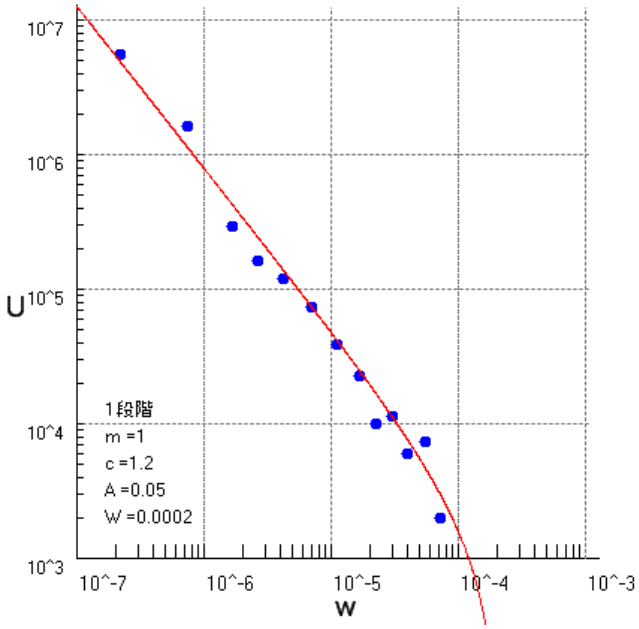

ウワミズザクラ858
1.0E+00
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
w (m³)
1980
1985
1990
1995
2000
2005
2010
t(year)
w-t
近似曲線

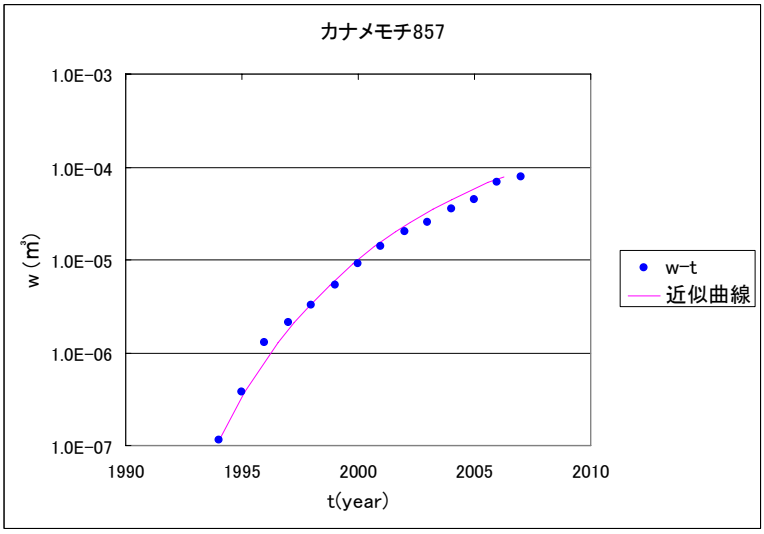

1段階
m =1
c =1.35
A =0.02
W =0.002
2段階
m =1
c =0.05
A =150
W =0.005
3段階
m =1
c =1.6
A =0.01
W =0.1
4段階
m =1
c =1.75
A =0.01
W =0.2
U
W

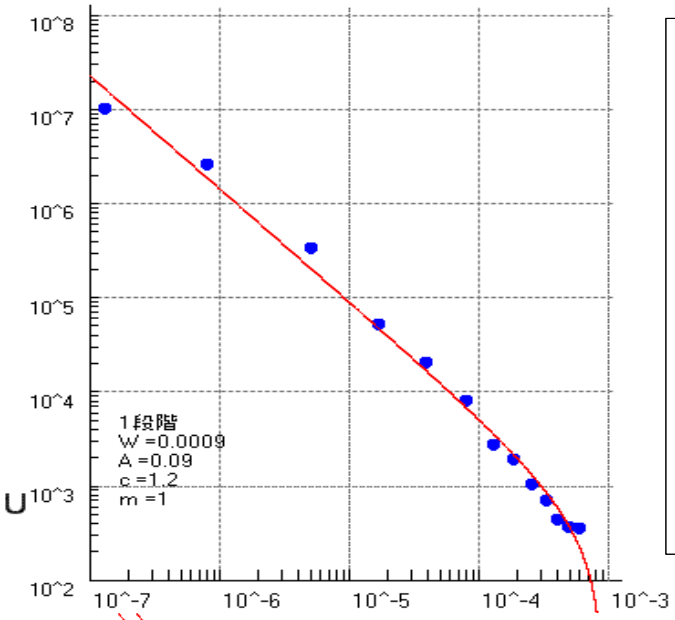

ウワミズザクラ750
w (m³)
t(year)
w-t
近似曲線

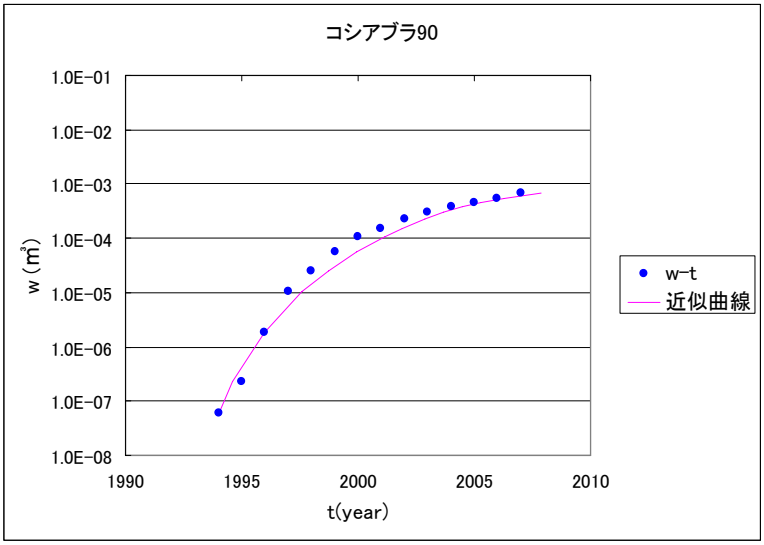

1段階
m =1
c =1.35
A =0.01
W =8E-05
U
W

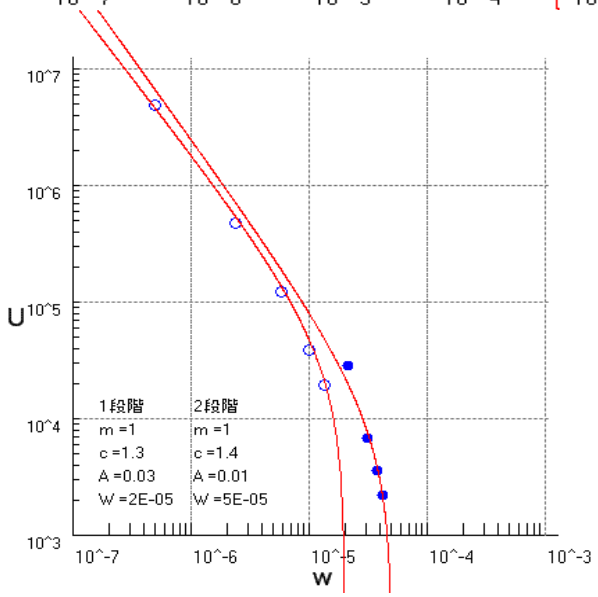

カナメモチ391
w (m³)
t(year)
w-t
近似曲線

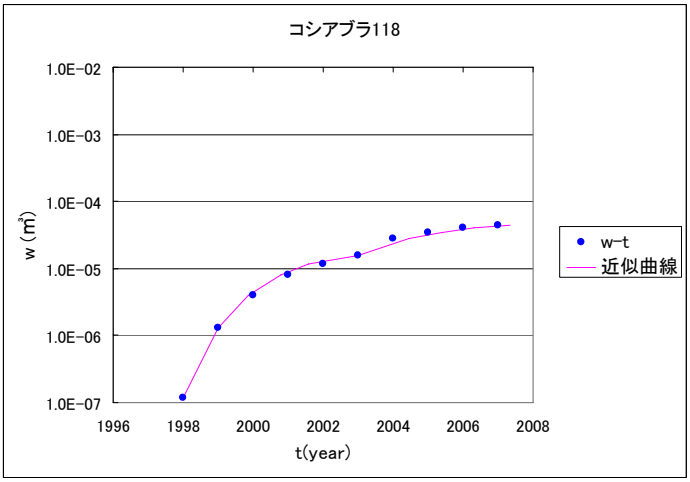

1段階
m =1
c =1.2
A =0.05
W =0.0002
U
W

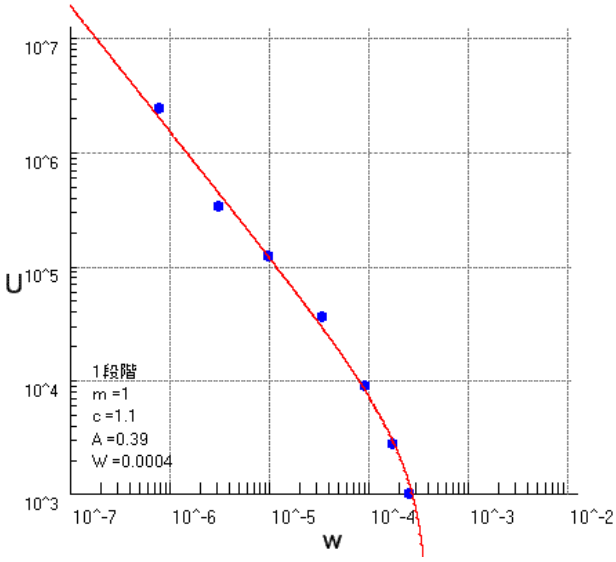

カナメモチ857
w (m³)
t(year)
w-t
近似曲線

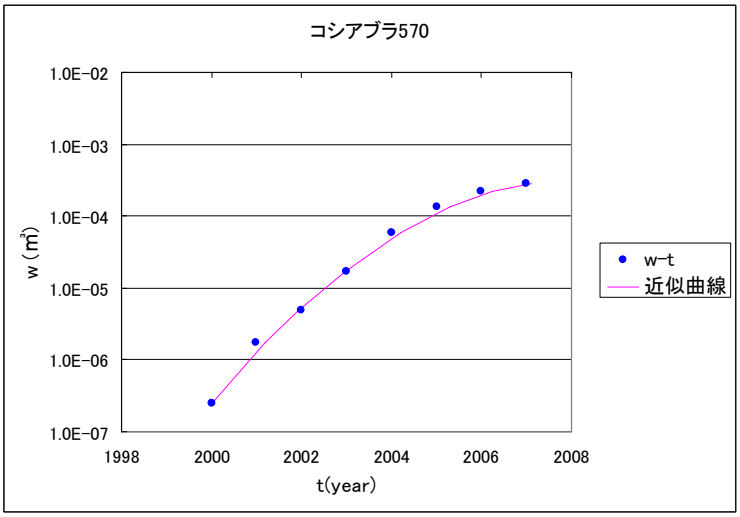

10^8
10^7
10^6
10^5
10^4
10^3
10^2
U
1段階
W =0.0009
A =0.09
c =1.2
m =1
10^-7
10^-6
10^-5
10^-4
10^-3

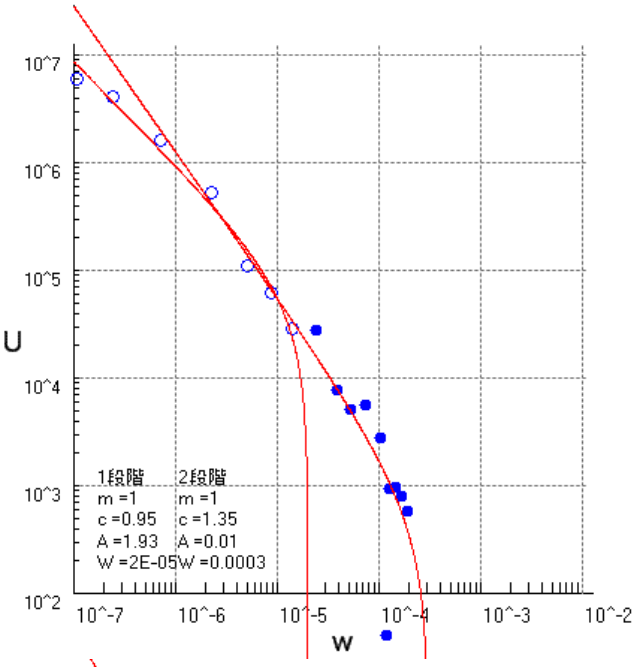

コシアブラ90
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
1.0E-08
w (m³)
1990
1995
2000
2005
2010
t(year)
w-t
近似曲線

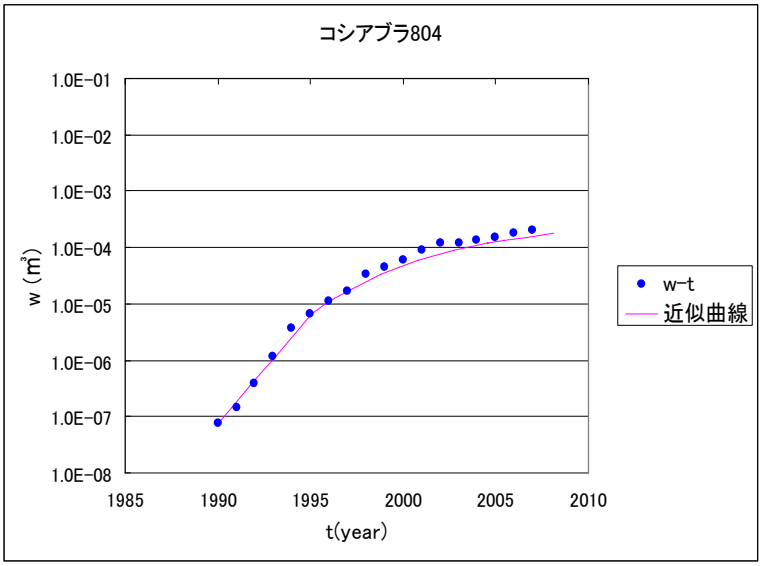

10^7
10^6
10^5
10^4
10^3
U
1段階
m =1
c =1.3
A =0.03
W =2E-05
2段階
m =1
c =1.4
A =0.01
W =5E-05
10^-7
10^-6
10^-5
10^-4
10^-3
w

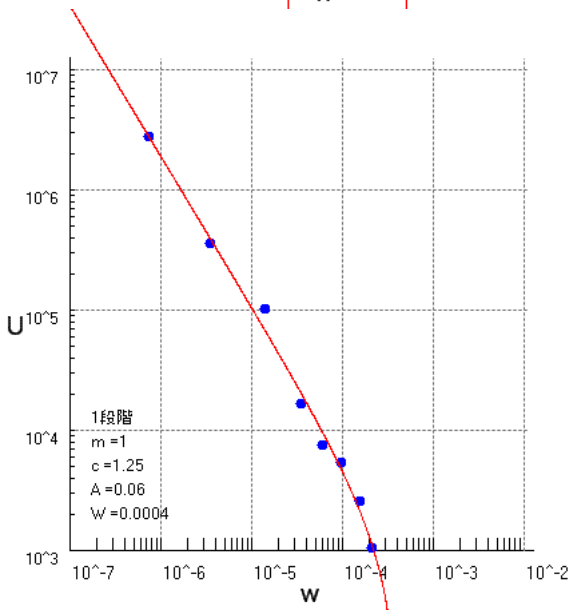

コシアブラ118
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
w (m³)
1996
1998
2000
2002
2004
2006
2008
t(year)
w-t
近似曲線

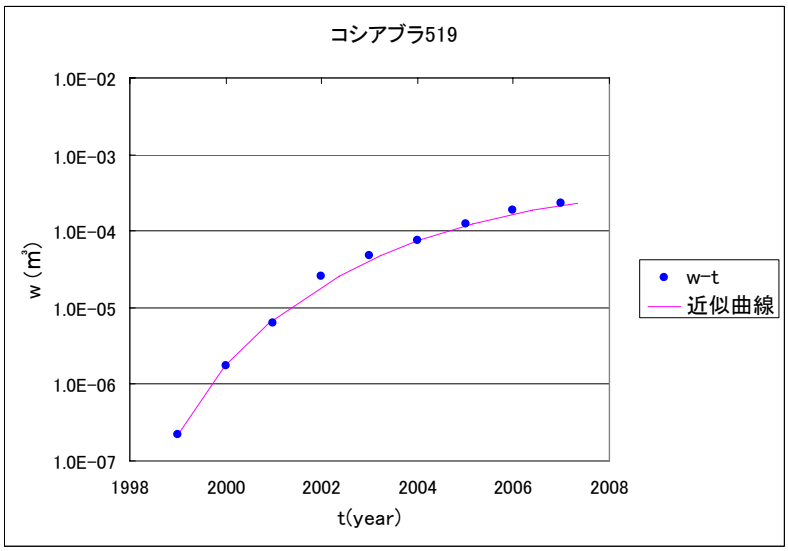

10^7
10^6
10^5
10^4
10^3
U
1段階
m =1
c =1.1
A =0.39
W =0.0004
10^-7
10^-6
10^-5
10^-4
10^-3
10^-2
w

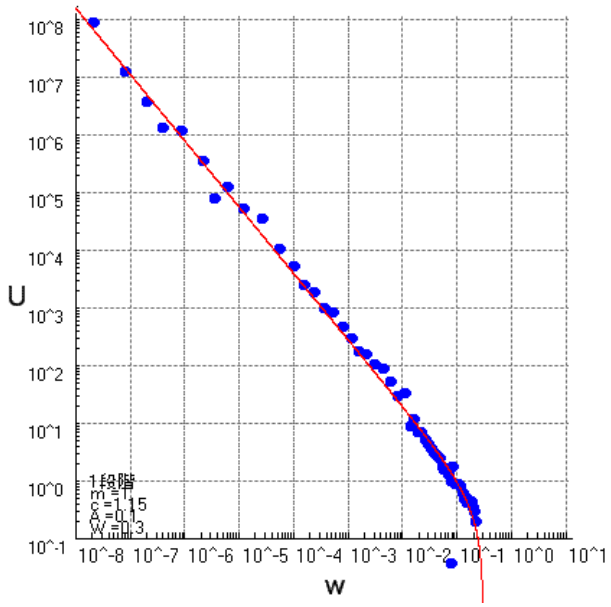

コシアブラ570
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
w (m³)
1998
2000
2002
2004
2006
2008
t(year)
w-t
近似曲線

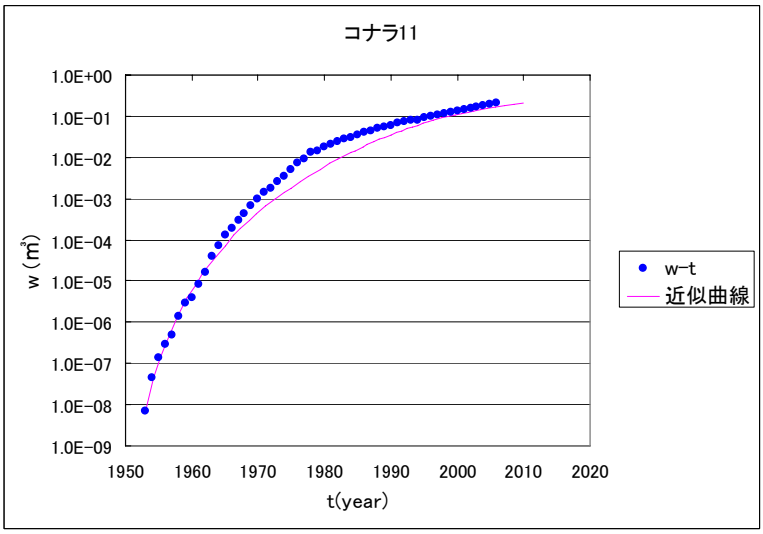

U
w
1段階
m =1
c =0.95
A =1.93
W =2E-05
2段階
m =1
c =1.35
A =0.01
W =0.0003

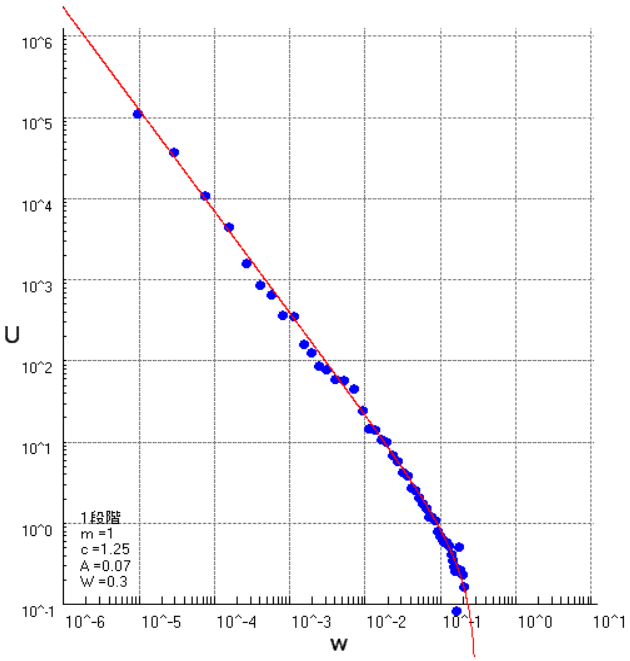

コシアブラ804
w (m³)
t(year)
w-t
近似曲線

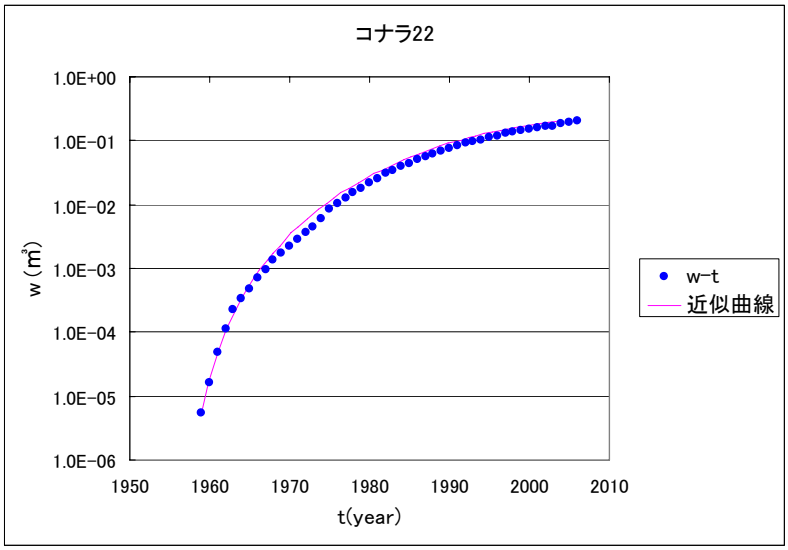

U
w
1段階
m =1
c =1.25
A =0.06
W =0.0004

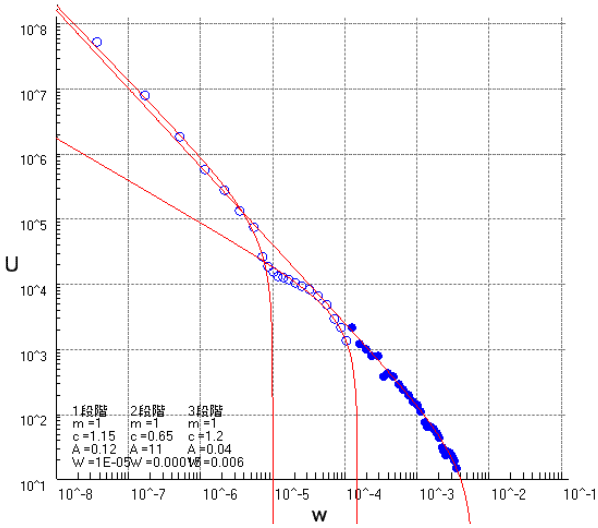

コシアブラ519
w (m³)
t(year)
w-t
近似曲線

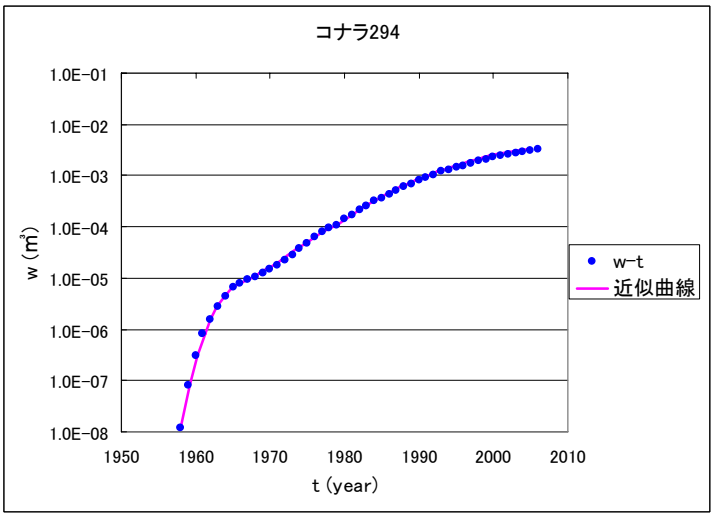

U
w
1段階
m =1
c =1.15
A =0.1
W =0.3

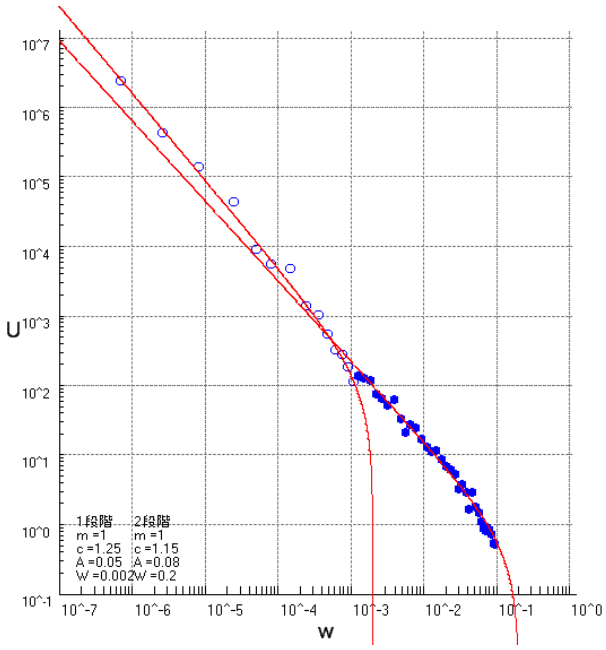

コナラ11
w (m³)
t(year)
w-t
近似曲線

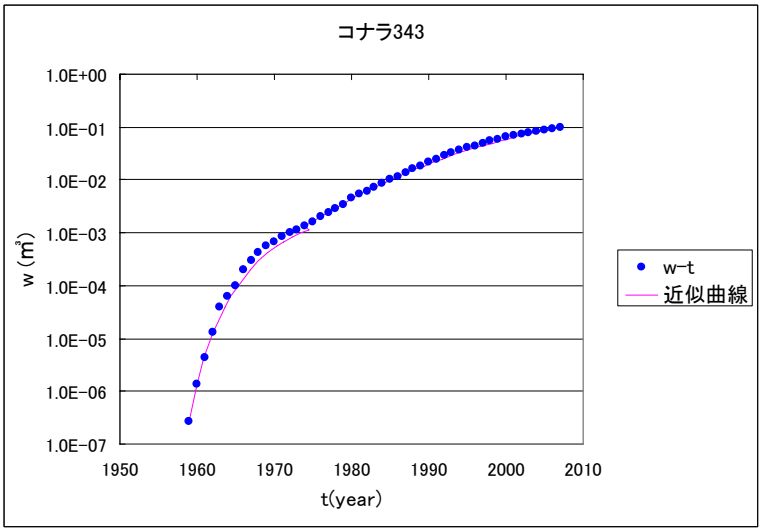

1段階
m =1
c =1.25
A =0.07
W =0.3
U
w

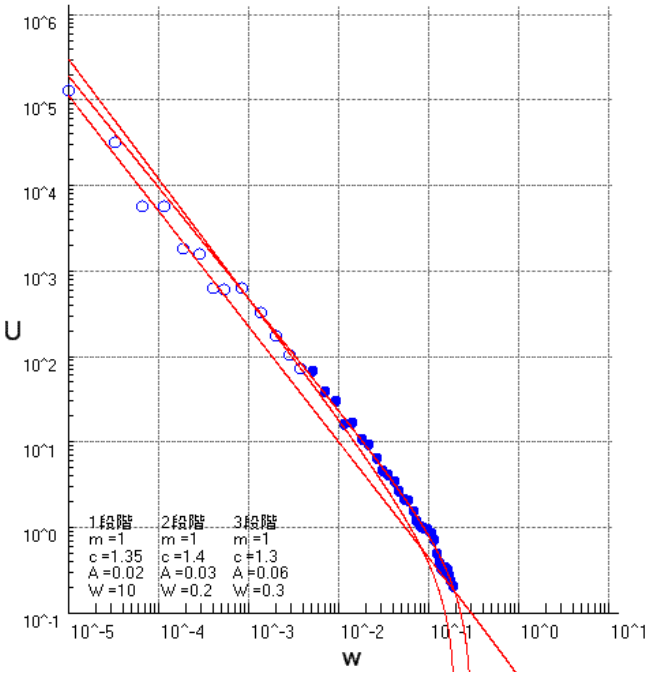

コナラ22
w (m³)
t(year)
w-t
近似曲線

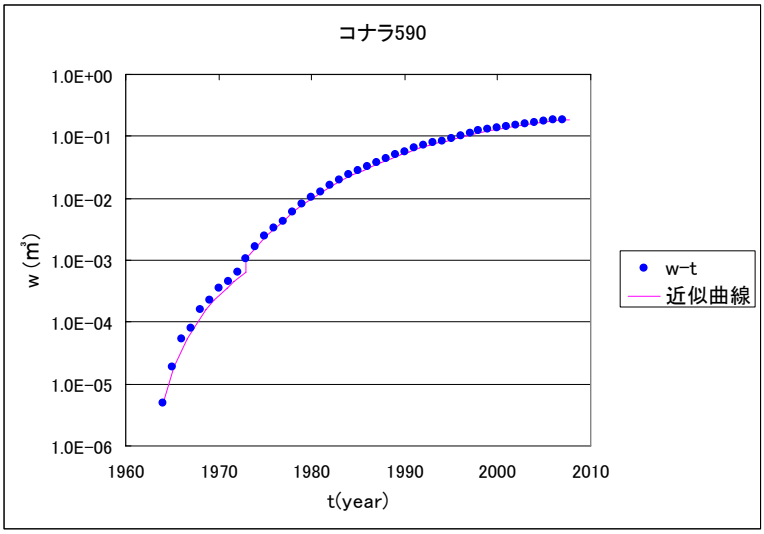

1段階
m =1
c =1.15
A =0.12
W =1E-05
2段階
m =1
c =0.65
A =11
W =0.0001
3段階
m =1
c =1.2
A =0.04
W =0.006
U
w

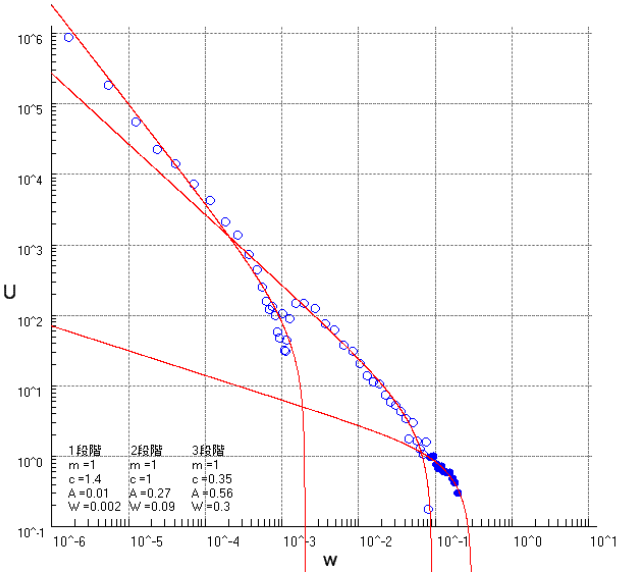

コナラ294
w (m³)
t (year)
w-t
近似曲線

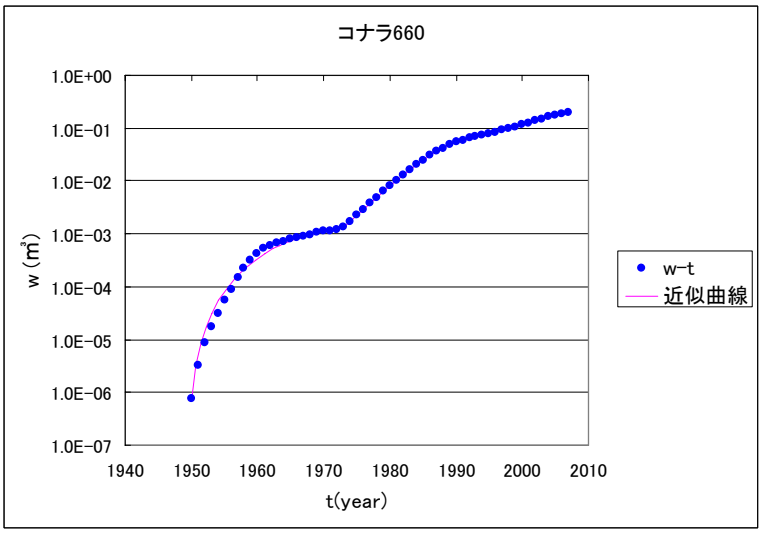

1段階
m =1
c =1.25
A =0.05
W =0.002
2段階
m =1
c =1.15
A =0.08
W =0.2
U
w

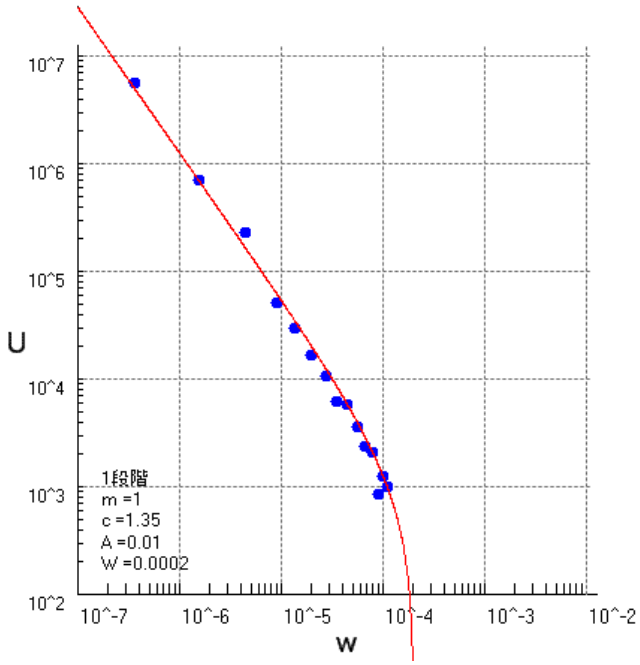

コナラ343
w (m³)
t(year)
w-t
近似曲線

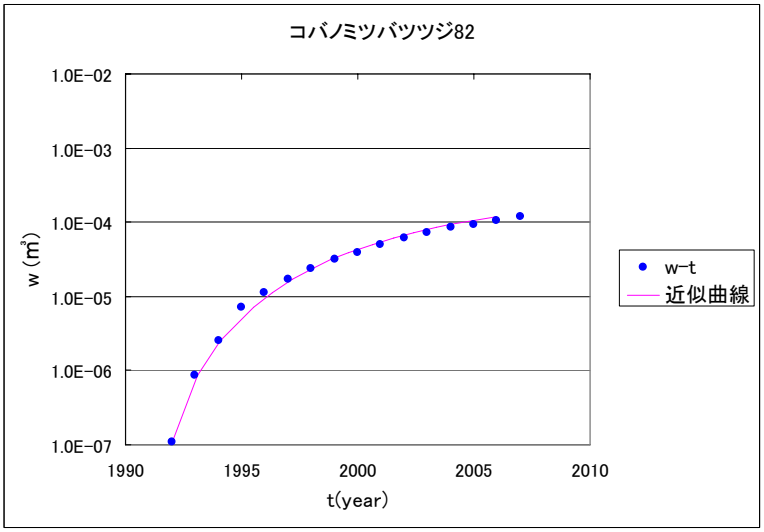

U
w
1段階
m =1
c =1.35
A =0.02
W =10
2段階
m =1
c =1.4
A =0.03
W =0.2
3段階
m =1
c =1.3
A =0.06
W =0.3

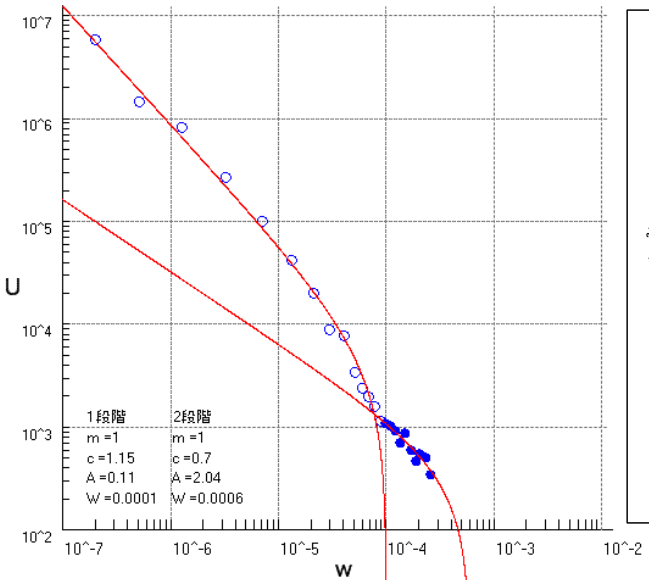

コナラ590
w (m³)
t(year)
w-t
近似曲線

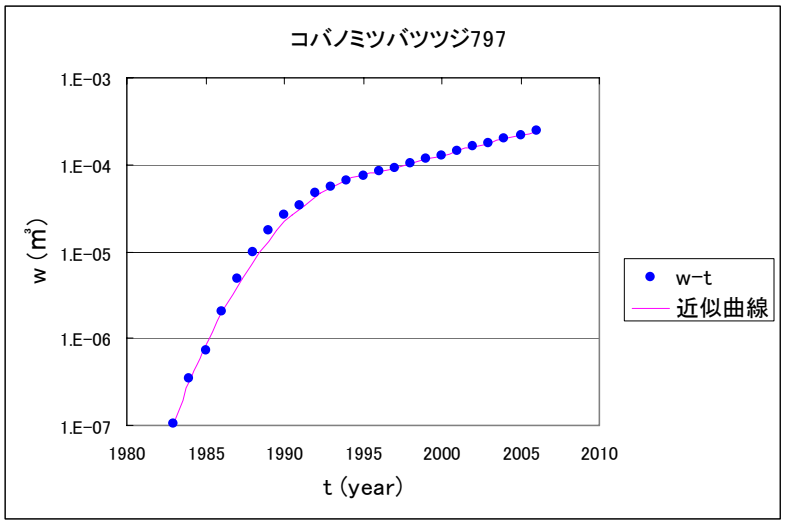

U
w
1段階
m =1
c =1.4
A =0.01
W =0.002
2段階
m =1
c =1
A =0.27
W =0.09
3段階
m =1
c =0.35
A =0.56
W =0.3

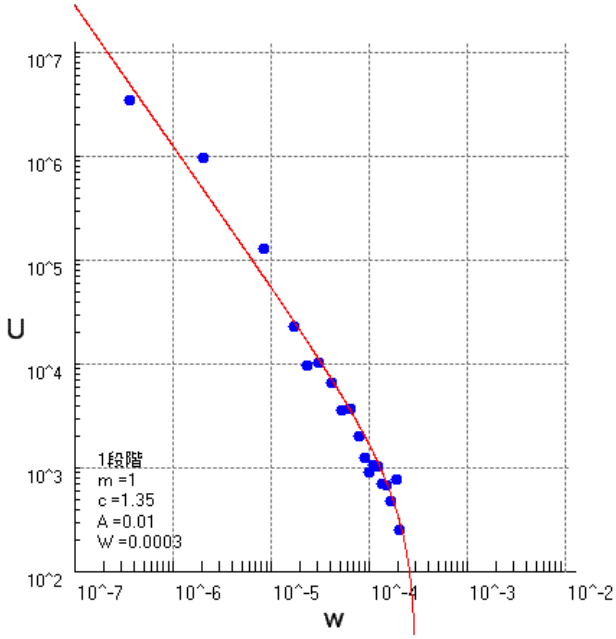

コナラ660
w (m³)
t(year)
w-t
近似曲線

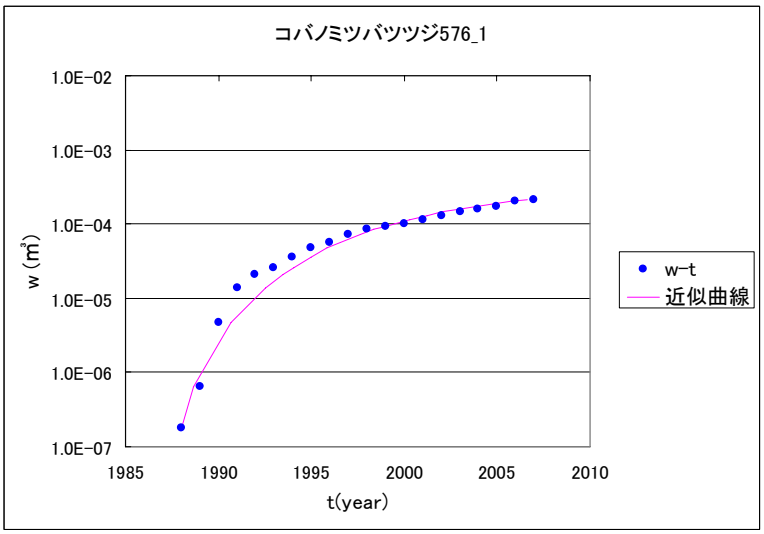

U
w
1段階
m =1
c =1.35
A =0.01
W =0.0002

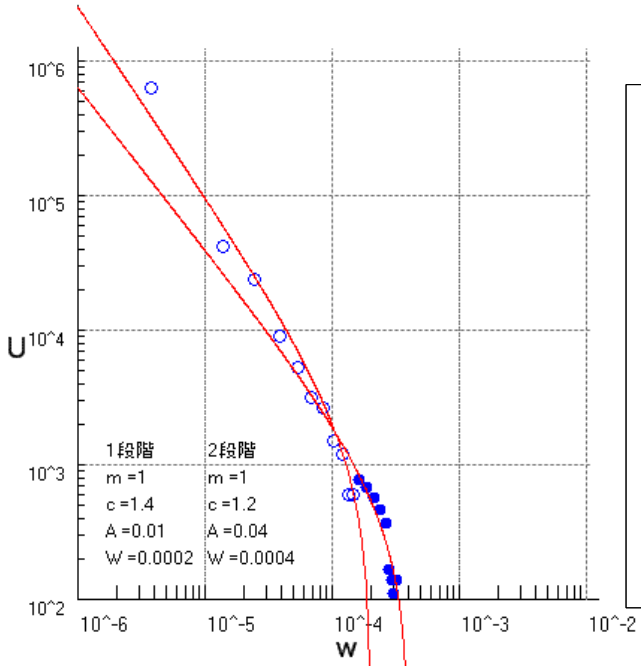

コバノミツバツツジ82
w (m³)
t(year)
w-t
近似曲線

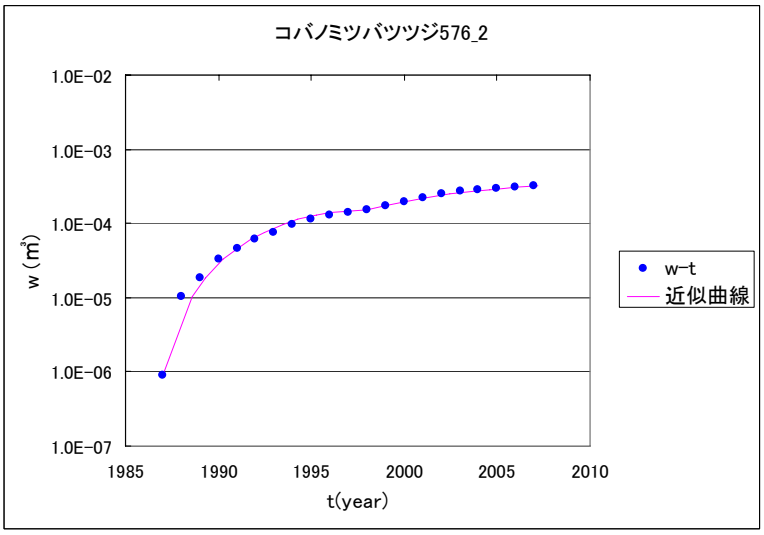

U
w
1段階
m =1
c =1.15
A =0.11
W =0.0001
2段階
m =1
c =0.7
A =2.04
W =0.0006

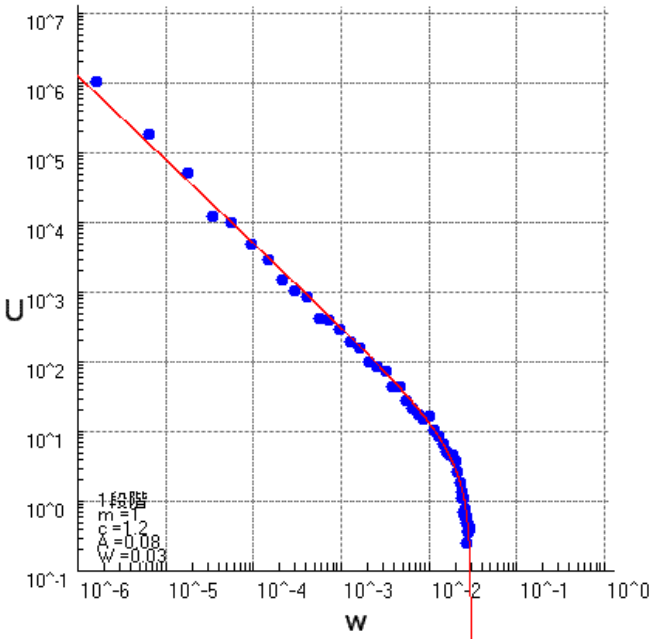

コバノミツバツツジ797
w (m³)
t (year)
w-t
近似曲線

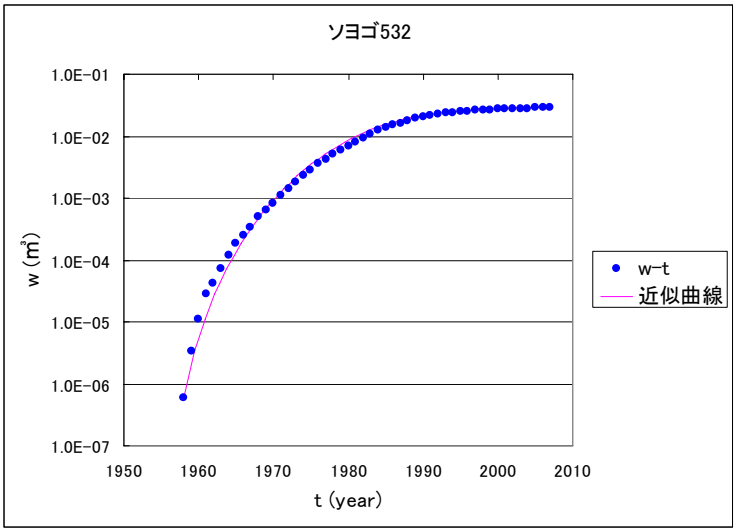

U
w
1段階
m =1
c =1.35
A =0.01
W =0.0003

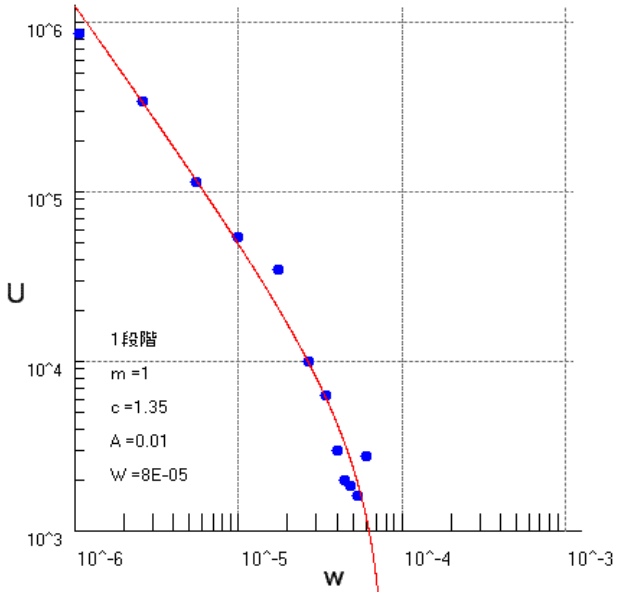

コバノミツバツツジ576_1
w (m³)
t(year)
w-t
近似曲線

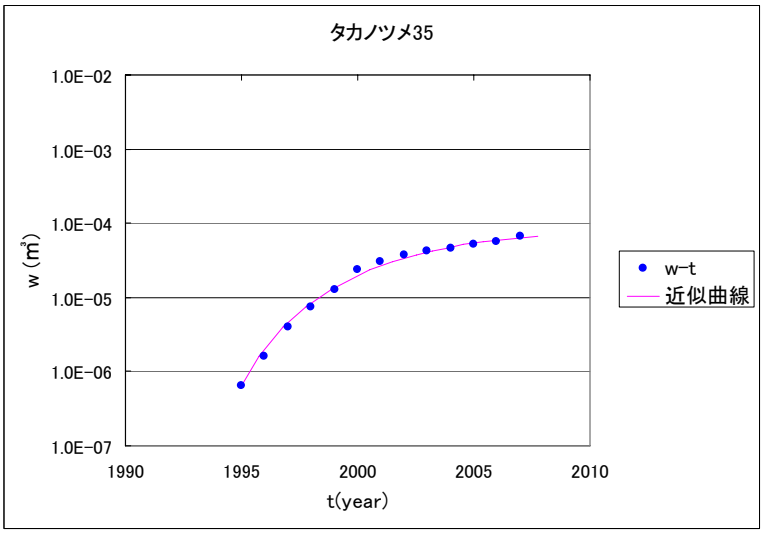

U
w
1段階
m =1
c =1.4
A =0.01
W =0.0002
2段階
m =1
c =1.2
A =0.04
W =0.0004

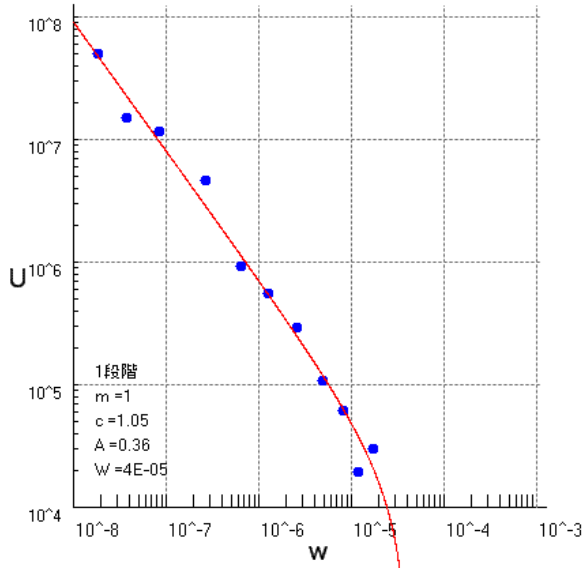

コバノミツバツツジ576_2
w (m³)
t(year)
w-t
近似曲線

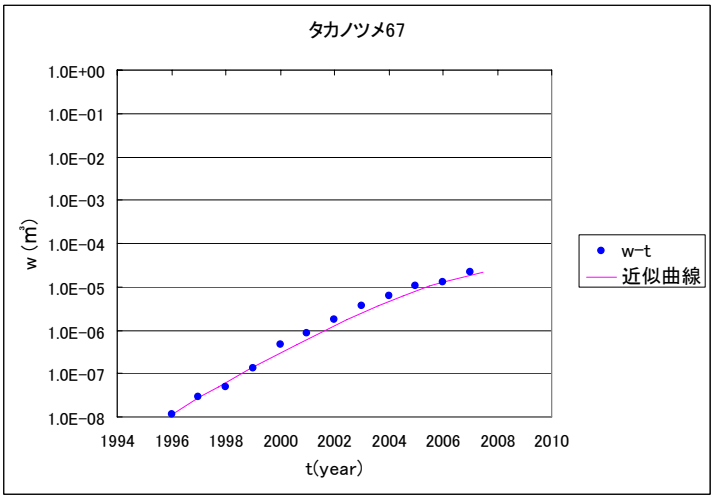

10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
10^-1
U
1段階
m =1
c =1.2
A =0.08
W =0.03
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
10^0
w

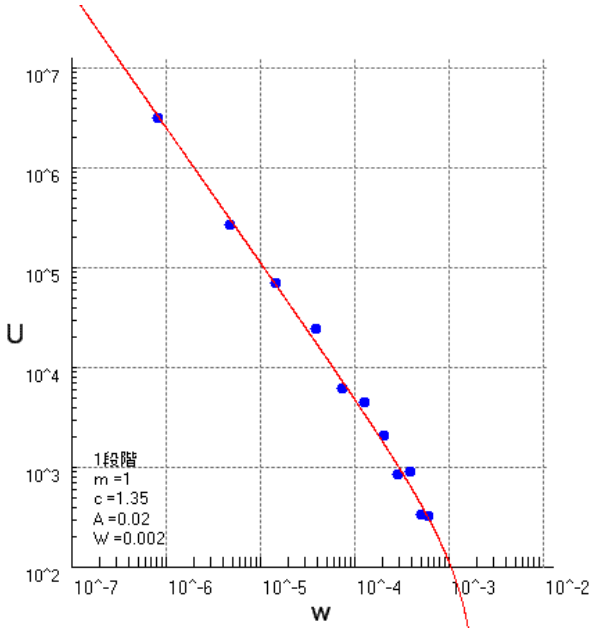

ソヨゴ532
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
w (m³)
1950
1960
1970
1980
1990
2000
2010
t (year)
w-t
近似曲線

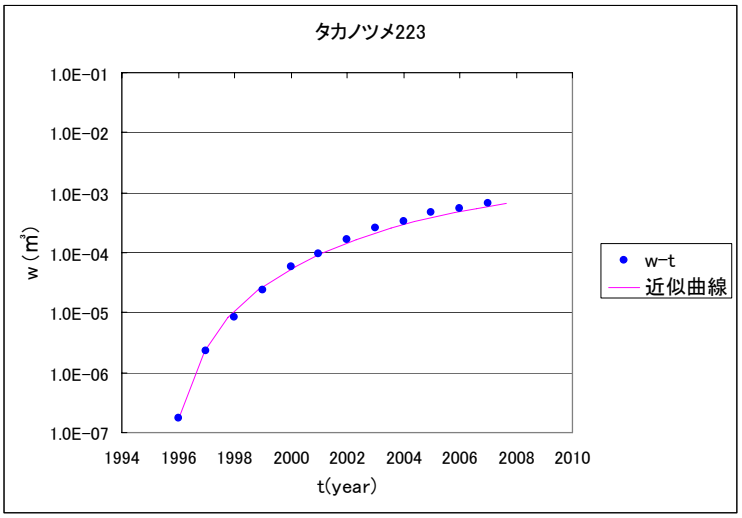

10^6
10^5
10^4
10^3
U
1段階
m =1
c =1.35
A =0.01
W =8E-05
10^-6
10^-5
10^-4
10^-3
w

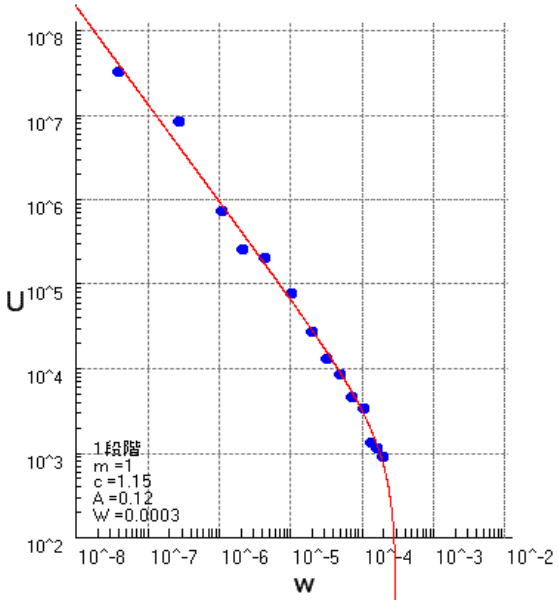

タカノツメ35
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
w (m³)
1990
1995
2000
2005
2010
t(year)
w-t
近似曲線

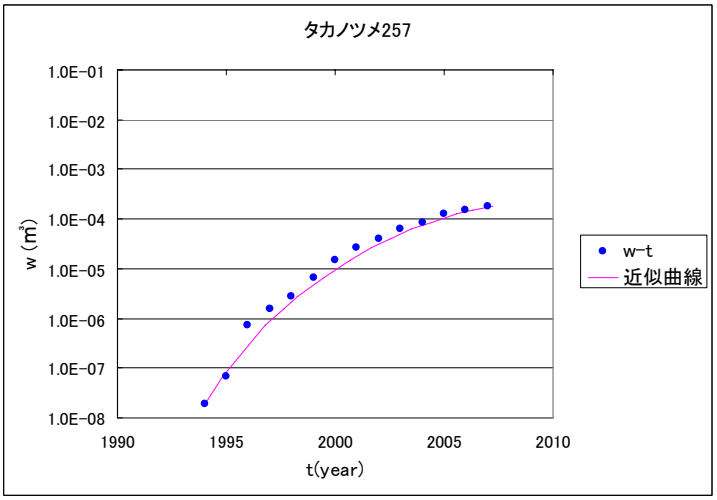

10^8
10^7
10^6
10^5
10^4
U
1段階
m =1
c =1.05
A =0.36
W =4E-05
10^-8
10^-7
10^-6
10^-5
10^-4
10^-3
w

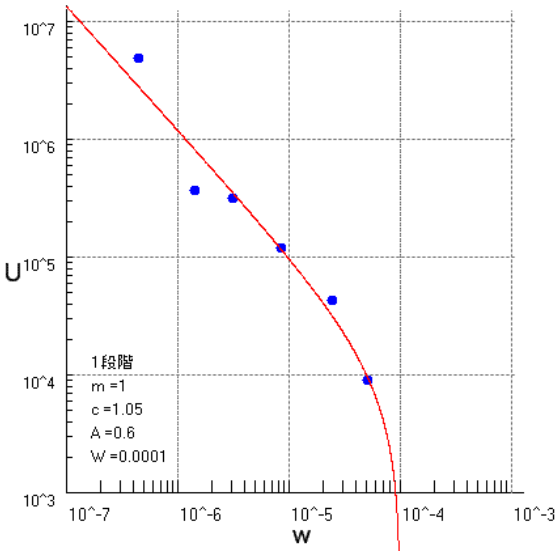

タカノツメ67
1.0E+00
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
1.0E-08
w (m³)
1994
1996
1998
2000
2002
2004
2006
2008
2010
t(year)
w-t
近似曲線

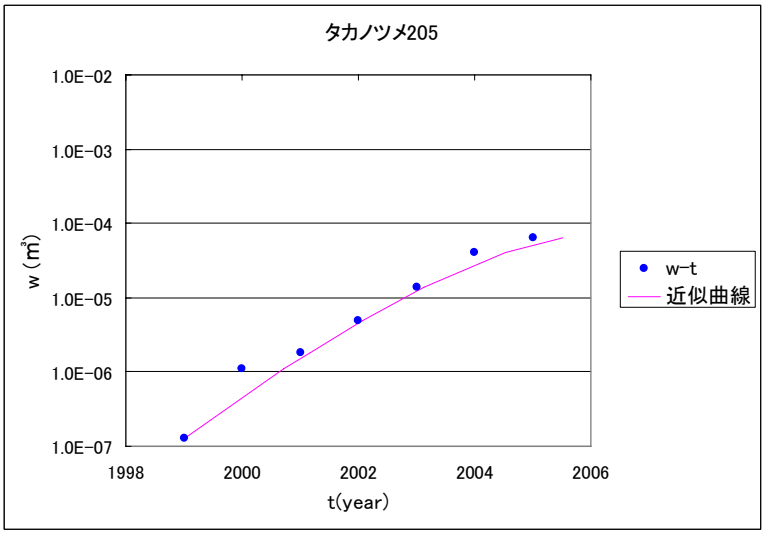

10^7
10^6
10^5
10^4
10^3
10^2
U
1段階
m =1
c =1.35
A =0.02
W =0.002
10^-7
10^-6
10^-5
10^-4
10^-3
10^-2
w

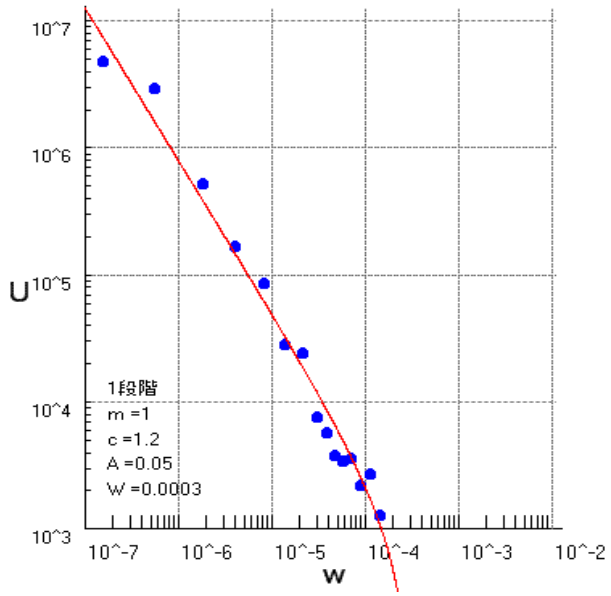

タカノツメ223
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
w (m³)
1994
1996
1998
2000
2002
2004
2006
2008
2010
t(year)
w-t
近似曲線

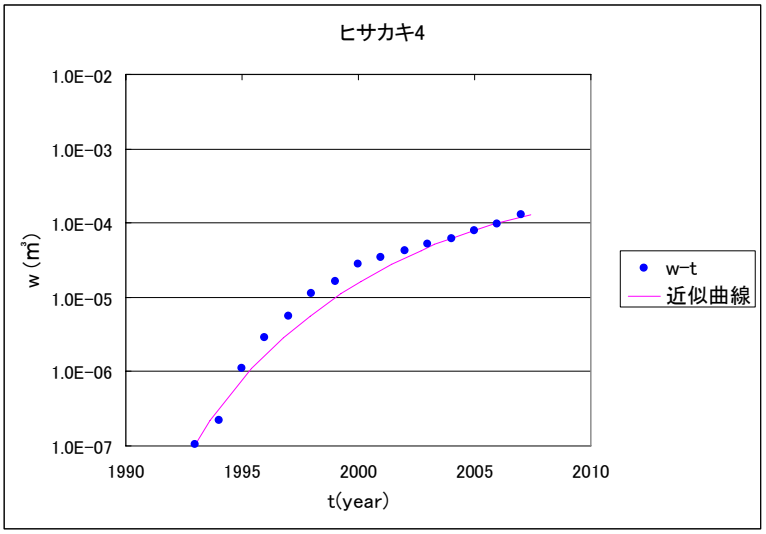

10^8
10^7
10^6
10^5
10^4
10^3
10^2
U
1段階
m =1
c =1.15
A =0.12
W =0.0003
10^-8
10^-7
10^-6
10^-5
10^-4
10^-3
10^-2
w

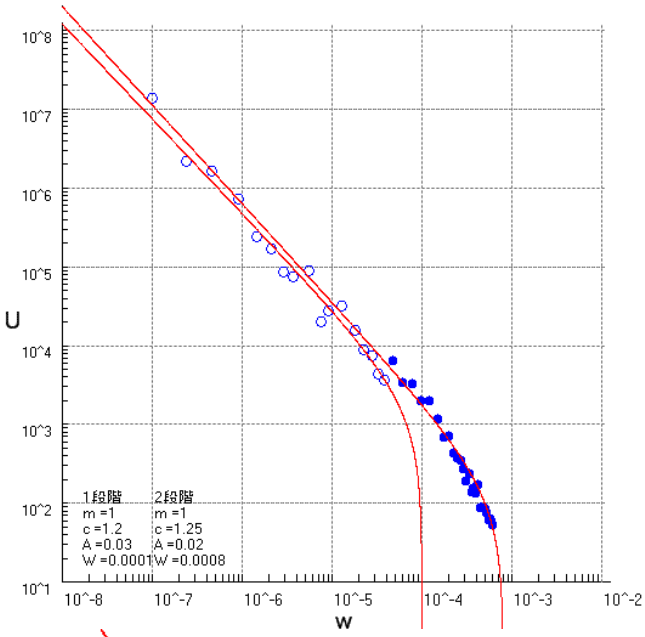

タカノツメ257
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
1.0E-08
w (m³)
1990
1995
2000
2005
2010
t(year)
w-t
近似曲線

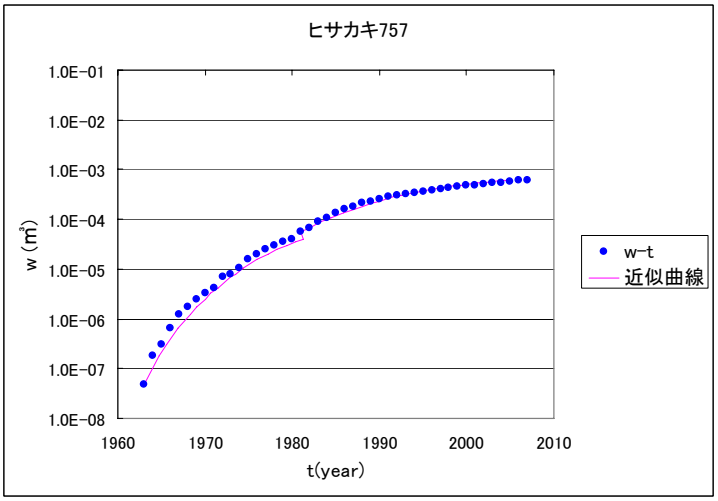

10^7
10^6
10^5
10^4
10^3
U
1段階
m =1
c =1.05
A =0.6
W =0.0001
10^-7
10^-6
10^-5
10^-4
10^-3
w

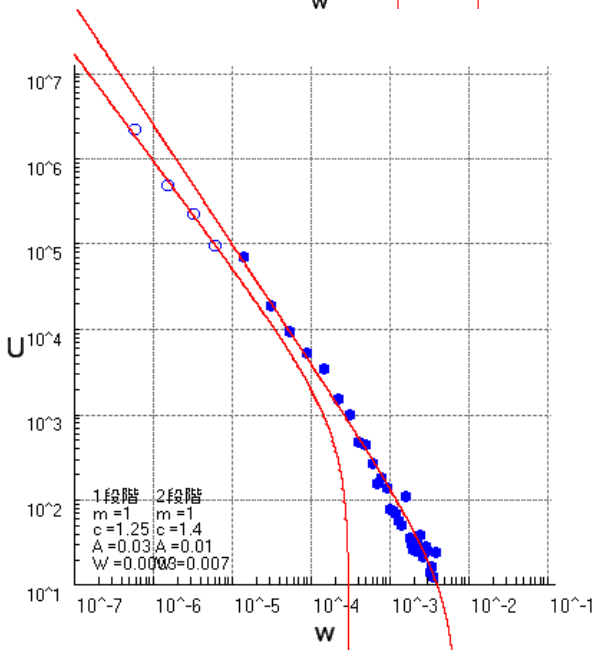

タカノツメ205
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
w (m³)
1998
2000
2002
2004
2006
t(year)
w-t
近似曲線

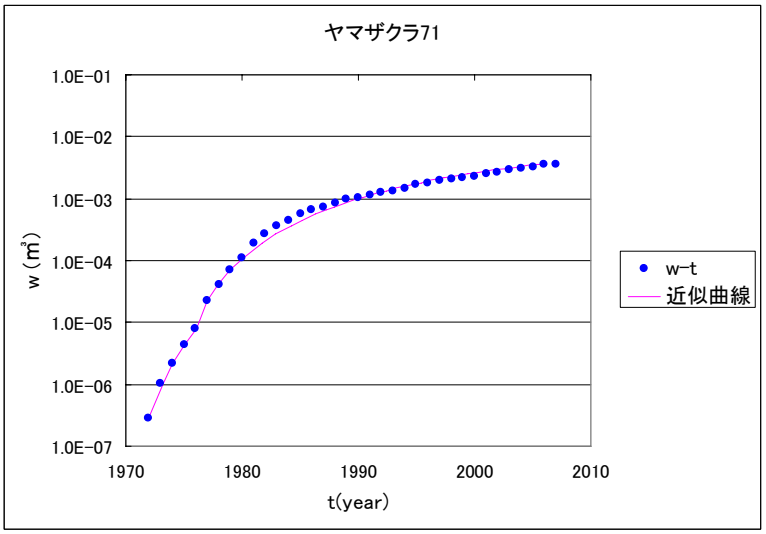

10^7
10^6
10^5
10^4
10^3
U
1段階
m =1
c =1.2
A =0.05
W =0.0003
10^-7
10^-6
10^-5
10^-4
10^-3
10^-2
w

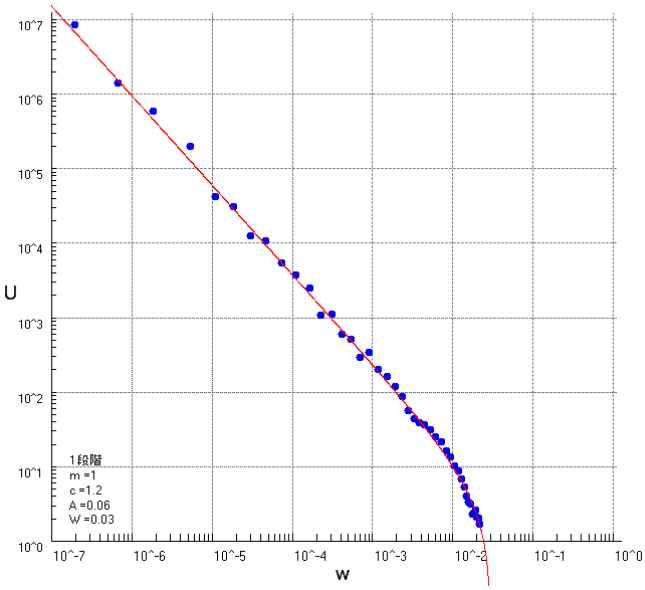

ヒサカキ4
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
w (m³)
1990
1995
2000
2005
2010
t(year)
w-t
近似曲線

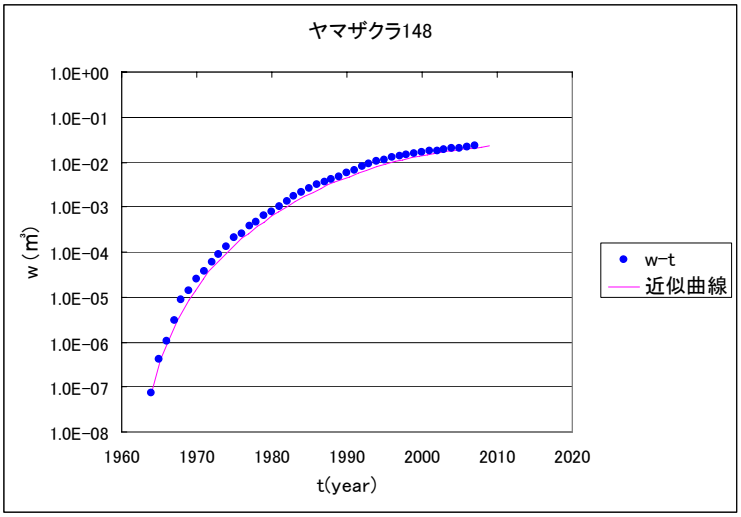

10^8
10^7
10^6
10^5
10^4
10^3
10^2
10^1
U
1段階
m =1
c =1.2
A =0.03
W =0.0001
2段階
m =1
c =1.25
A =0.02
W =0.0008
10^-8
10^-7
10^-6
10^-5
10^-4
10^-3
10^-2
w

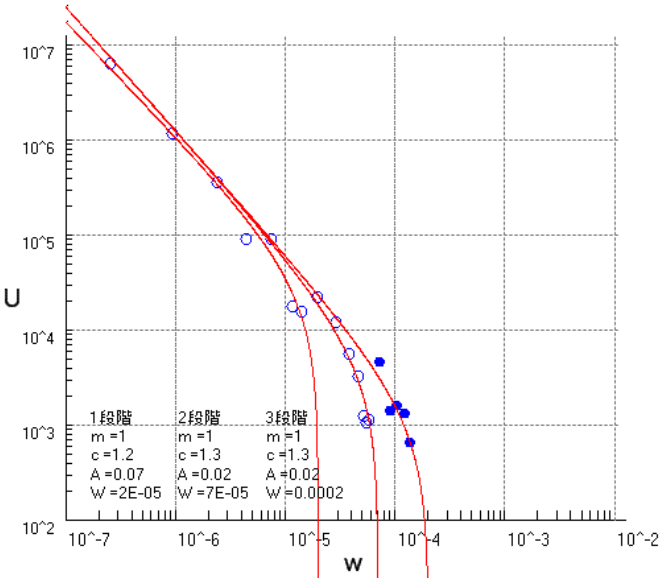

ヒサカキ757
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
1.0E-08
w (m³)
1960
1970
1980
1990
2000
2010
t(year)
w-t
近似曲線

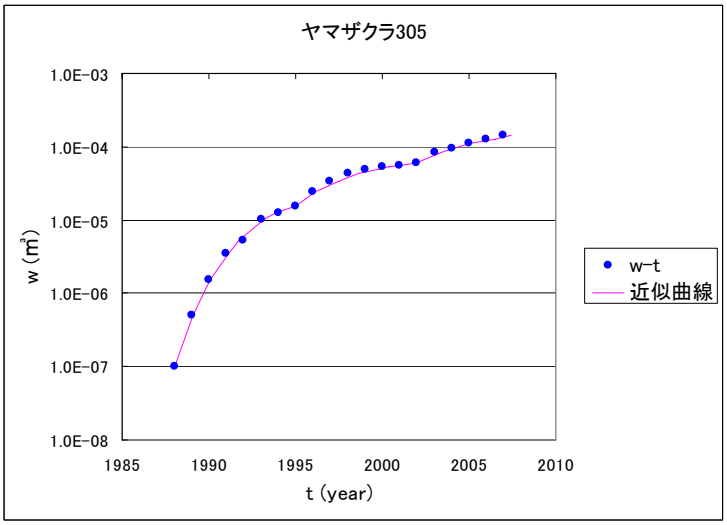

10^7
10^6
10^5
10^4
10^3
10^2
10^1
U
1段階
m =1
c =1.25
A =0.03
W =0.0008
2段階
m =1
c =1.4
A =0.01
W =0.007
10^-7
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
w

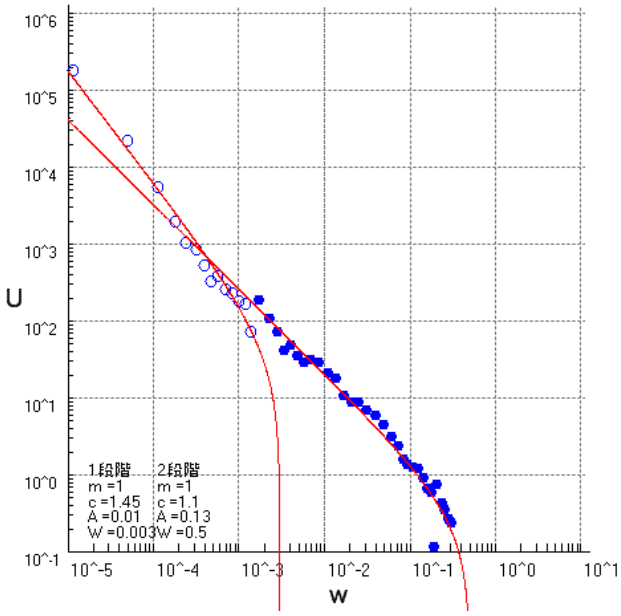

ヤマザクラ71
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
w (m³)
1970
1980
1990
2000
2010
t(year)
w-t
近似曲線

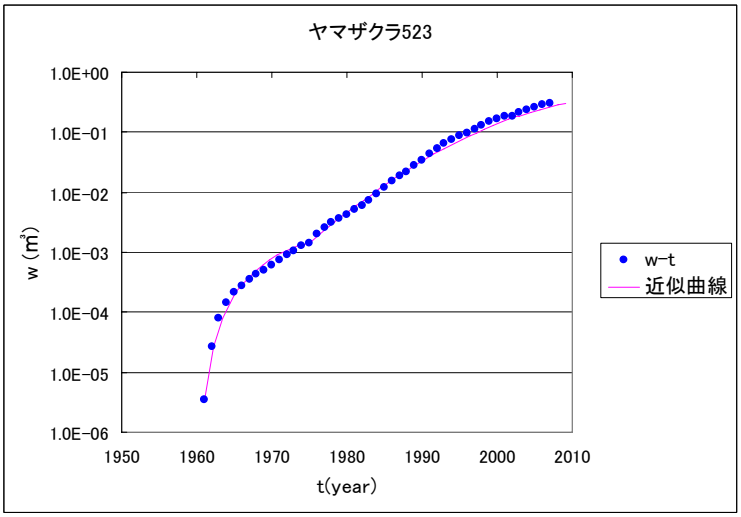

U
w
1段階
m =1
c =1.2
A =0.06
W =0.03

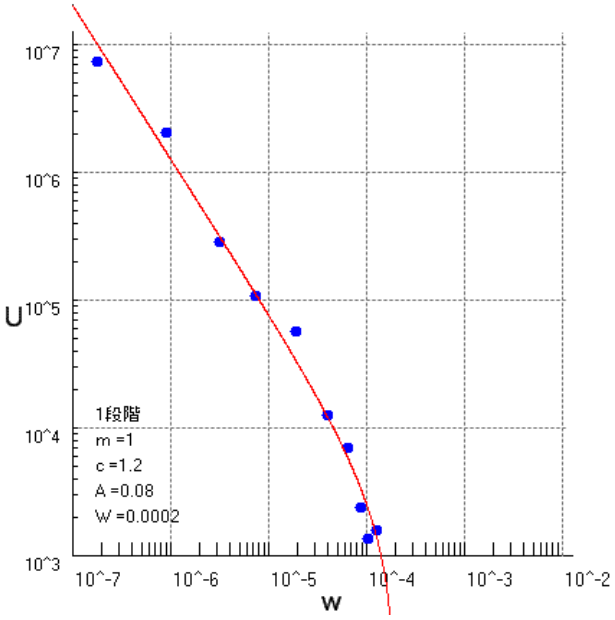

ヤマザクラ148
w (m³)
t(year)
w-t
近似曲線

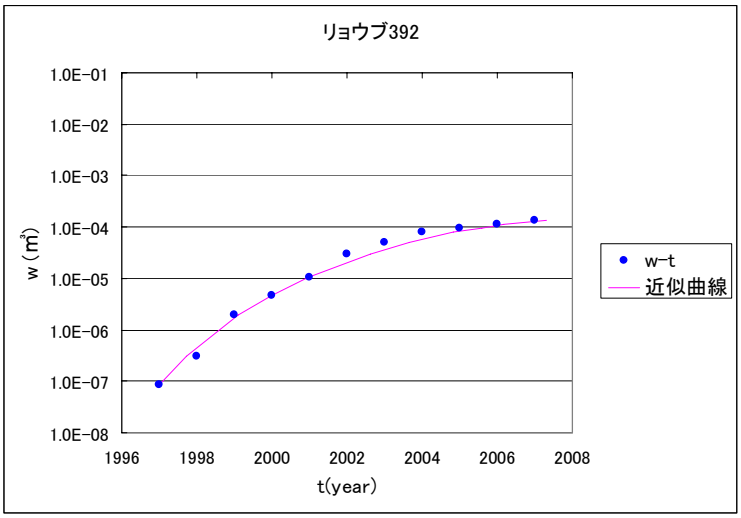

U
w
1段階
m =1
c =1.2
A =0.07
W =2E-05
2段階
m =1
c =1.3
A =0.02
W =7E-05
3段階
m =1
c =1.3
A =0.02
W =0.0002

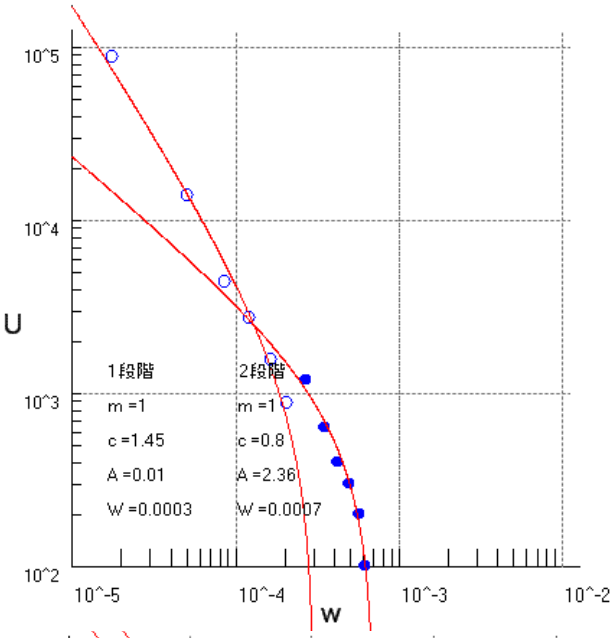

ヤマザクラ305
w (m³)
t (year)
w-t
近似曲線

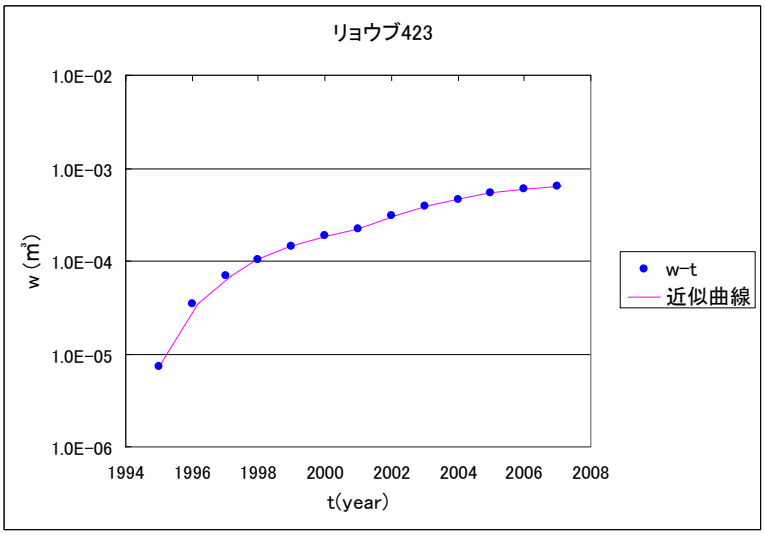

U
w
1段階
m =1
c =1.45
A =0.01
W =0.003
2段階
m =1
c =1.1
A =0.13
W =0.5

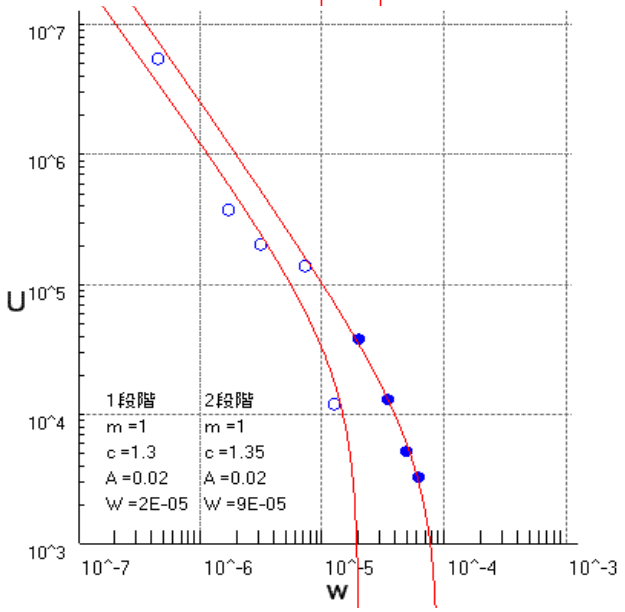

ヤマザクラ523
w (m³)
t(year)
w-t
近似曲線

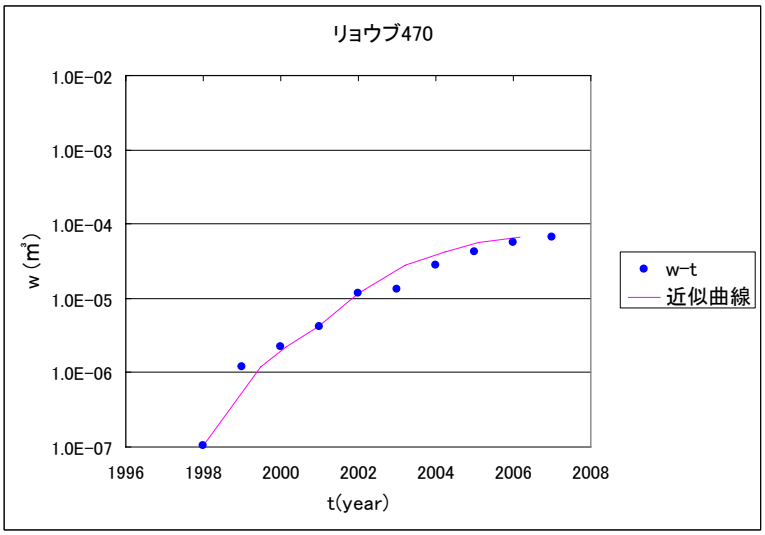

10^7
10^6
10^5
10^4
10^3
U
1段階
m =1
c =1.2
A =0.08
W =0.0002
10^-7
10^-6
10^-5
10^-4
10^-3
10^-2
w

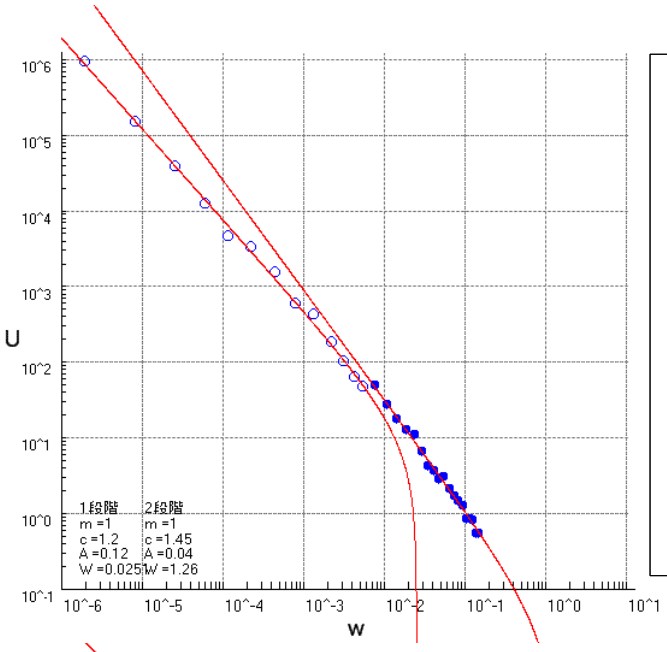

リョウブ392
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
1.0E-08
w (m³)
1996
1998
2000
2002
2004
2006
2008
t(year)
w-t
近似曲線

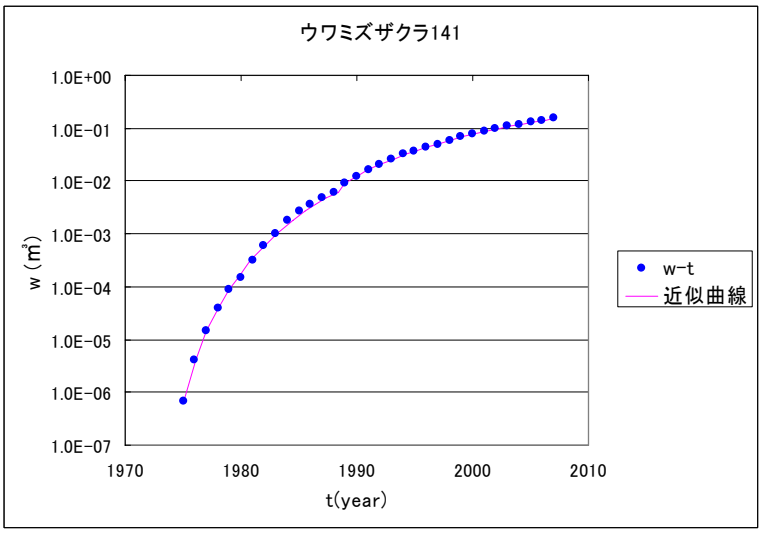

10^5
10^4
10^3
10^2
U
1段階
m =1
c =1.45
A =0.01
W =0.0003
2段階
m =1
c =0.8
A =2.36
W =0.0007
10^-5
10^-4
10^-3
10^-2
w

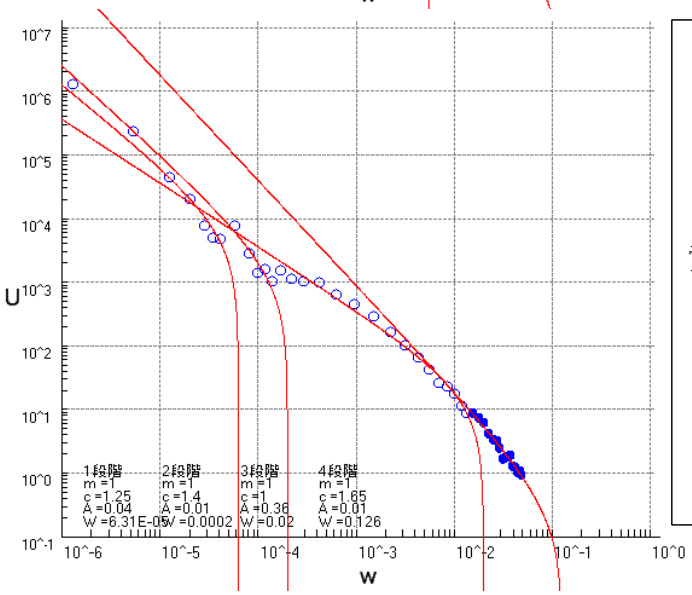

リョウブ423
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
w (m³)
1994
1996
1998
2000
2002
2004
2006
2008
t(year)
w-t
近似曲線

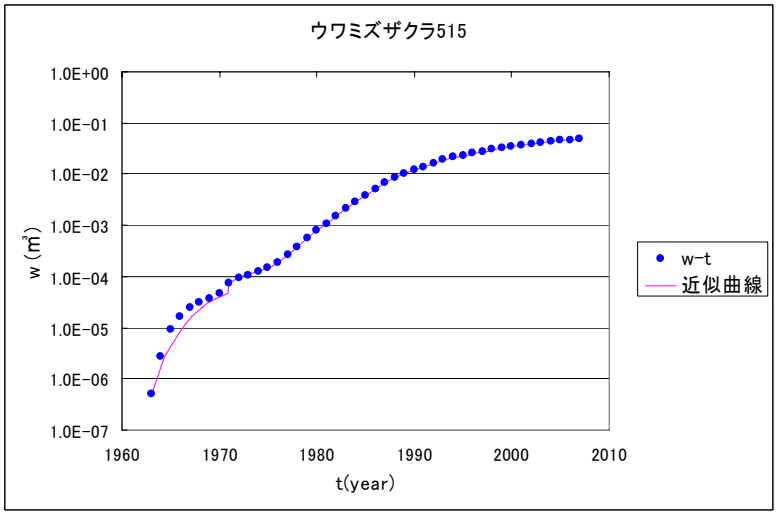

10^7
10^6
10^5
10^4
10^3
U
1段階
m =1
c =1.3
A =0.02
W =2E-05
2段階
m =1
c =1.35
A =0.02
W =9E-05
10^-7
10^-6
10^-5
10^-4
10^-3
w

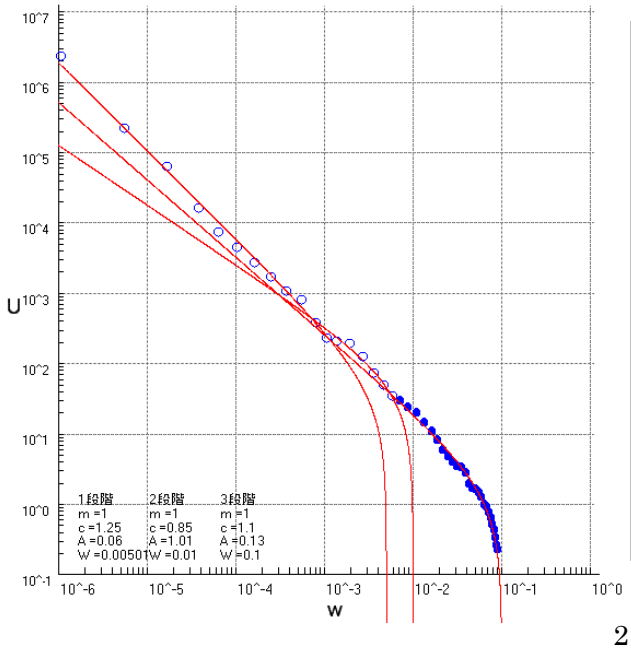

リョウブ470
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
w (m³)
1996
1998
2000
2002
2004
2006
2008
t(year)
w-t
近似曲線

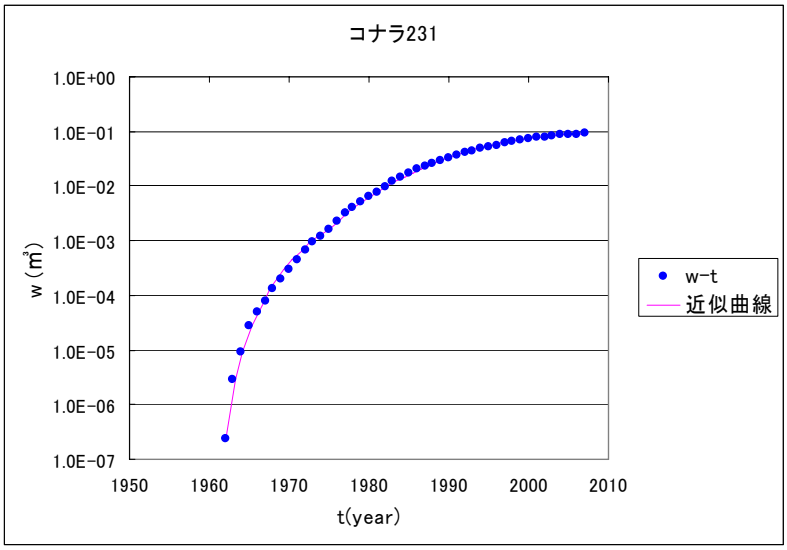

U
w
1段階
m =1
c =1.2
A =0.12
W =0.025
2段階
m =1
c =1.45
A =0.04
W =1.26


ウワミズザクラ141
w (m³)
t(year)
w-t
近似曲線


U
w
1段階
m =1
c =1.25
A =0.04
W =6.31E-05
2段階
m =1
c =1.4
A =0.01
W =0.0002
3段階
m =1
c =1
A =0.36
W =0.02
4段階
m =1
c =1.65
A =0.01
W =0.126


ウワミズザクラ515
w (m³)
t(year)
w-t
近似曲線


U
w
1段階
m =1
c =1.25
A =0.06
W =0.00501
2段階
m =1
c =0.85
A =1.01
W =0.01
3段階
m =1
c =1.1
A =0.13
W =0.1


コナラ231
w (m³)
t(year)
w-t
近似曲線

付録 2
以下は開発したソフトウェアのソースコードである。

```
'グラフスケールの基礎設定
Public Class Graph

    Protected xOLd, yOLd As Single
    Protected kx, ky As Single
    Protected MyPen As New Pen(Color.Black, 1)
    Protected MyBrush As New SolidBrush(Color.Black)
    Protected g As Graphics
    Protected bmp As Bitmap

    'コンストラクタ
    Public Sub New(ByRef pic As System.Windows.Forms.PictureBox, ByVal xmin As Single,
ByVal xmax As Single, ByVal ymin As Single, ByVal ymax As Single, ByVal dx As Single,
ByVal dy As Single)

        pic.Image = New Bitmap(pic.Width, pic.Height)

        g = Graphics.FromImage(pic.Image)
        g.FillRectangle(Brushes.White, 0, 0, pic.Width, pic.Height) '背景色を白に

        kx = (pic.Width - 2 * dx) / (xmax - xmin)
        ky = (pic.Height - 2 * dy) / (ymax - ymin)

        g.TranslateTransform(dx - xmin * kx, pic.Height - dy + ymin * ky)
        g.ScaleTransform(1, -1)

    End Sub
    'グラフの主軸
    Public Sub Axis(ByVal x1 As Double, ByVal x2 As Double, ByVal y1 As Double, ByVal y2
As Double, ByVal cL As Color, ByVal index As Integer)
        Select Case index
            Case 1
                MyPen.Color = cL
                g.DrawLine(MyPen, CInt(kx * x1), CInt(ky * 0), CInt(kx * x2), CInt(ky *
y1))
                g.DrawLine(MyPen, CInt(kx * x1), CInt(ky * y1), CInt(kx * x1), CInt(ky *
y2))

            Case 2
                MyPen.Color = cL
                g.DrawLine(MyPen, CInt(kx * x1), CInt(ky * y1), CInt(kx * x2), CInt(ky *
```

```
y1))
                g.DrawLine(MyPen, CInt(kx * x1), CInt(ky * y2), CInt(kx * x1), CInt(ky *
y1))
            Case 3
                MyPen.Color = cL
                g.DrawLine(MyPen, CInt(kx * x1), CInt(ky * y2), CInt(kx * x2), CInt(ky *
y2))
                g.DrawLine(MyPen, CInt(kx * x1), CInt(ky * y2), CInt(kx * x1), CInt(ky *
y1))
            Case 4
                MyPen.Color = cL
                g.DrawLine(MyPen, CInt(kx * x1), CInt(ky * 0), CInt(kx * x2), CInt(ky * 0))
                g.DrawLine(MyPen, CInt(kx * 0), CInt(ky * y1), CInt(kx * 0), CInt(ky * y2))
        End Select

    End Sub
    '線を引く
    Public Sub Line(ByVal x1 As Double, ByVal y1 As Double, ByVal x2 As Double, ByVal y2
As Double, ByVal cL As Color, ByVal index As Integer)
        MyPen.Color = cL
        Select Case index

            Case 1
                MyPen.DashStyle = Drawing2D.DashStyle.Solid
                g.DrawLine(MyPen, CInt(kx * x1), CInt(ky * y1), CInt(kx * x2), CInt(ky *
y2))

            Case 2
                MyPen.DashStyle = Drawing.Drawing2D.DashStyle.Dash
                g.DrawLine(MyPen, CInt(kx * x1), CInt(ky * y1), CInt(kx * x2), CInt(ky *
y2))
        End Select
        xOLd = x2：yOLd = x2
    End Sub
    '続けて線を引く
    Public Sub LineTO(ByVal x As Double, ByRef y As Double, ByVal cL As Color)
        MyPen.Color = cL
        g.DrawLine(MyPen, CInt(kx * xOLd), CInt(ky * yOLd), CInt(kx * x), CInt(ky * y))
        xOLd = x：yOLd = y

    End Sub
    Public Sub PutPoint(ByVal x As Double, ByVal y As Double, ByVal cL As Color)
        MyBrush.Color = cL
        g.FillRectangle(MyBrush, CInt(kx * x), CInt(ky * y), 1, 1)
```

```
        xOLd = x : yOLd = y
    End Sub

    'テキストの初期設定
    Public Sub Drawtext(ByVal s As String, ByVal x As Double, ByVal y As Double, ByVal cL
As Color, ByVal fsize As Integer)
        Dim myFont As New Font("MS 明朝", fsize, FontStyle.Regular)
        MyBrush.Color = cL
        g.ScaleTransform(1, -1)
        g.DrawString(s, myFont, MyBrush, kx * x, -ky * y)
        g.ScaleTransform(1, -1)

    End Sub

    Public Sub Drawtext_y_Axis1(ByVal s As String, ByVal x As Double, ByVal y As Double,
ByVal cL As Color, ByVal fsize As Integer)
        Dim myFont As New Font("MS 明朝", fsize, FontStyle.Regular)
        MyBrush.Color = cL
        g.ScaleTransform(1, -1)
        g.DrawString(s, myFont, MyBrush, (kx * x) - 35, -ky * y)
        g.ScaleTransform(1, -1)
    End Sub
    Public Sub Drawtext_y_Axis2(ByVal s As String, ByVal x As Double, ByVal y As Double,
ByVal cL As Color, ByVal fsize As Integer)
        Dim myFont As New Font("MS 明朝", fsize, FontStyle.Regular)
        MyBrush.Color = cL
        g.ScaleTransform(1, -1)
        g.DrawString(s, myFont, MyBrush, (kx * x) - 50, -ky * y)
        g.ScaleTransform(1, -1)
    End Sub
    Public Sub Drawtext_x_Axis1(ByVal s As String, ByVal x As Double, ByVal y As Double,
ByVal cL As Color, ByVal fsize As Integer)
        Dim myFont As New Font("MS 明朝", fsize, FontStyle.Regular)
        MyBrush.Color = cL
        g.ScaleTransform(1, -1)
        g.DrawString(s, myFont, MyBrush, kx * x, (-ky * y) + 20)
        g.ScaleTransform(1, -1)
    End Sub
    'プロットのマーカー設定のプロージャー
    Public Sub Plot(ByVal x As Double, ByVal y As Double, ByVal cL As Color, ByVal size As
Integer, ByVal index As Integer)
        Try
            MyBrush.Color = cL
            MyPen.Color = cL
```

```
            Select Case index
                Case 1 '×記号

                    g.DrawLine(MyPen, CInt(kx * x) - size, CInt(ky * y) - size, CInt(kx * x)
+ size, CInt(ky * y) + size)
                    g.DrawLine(MyPen, CInt(kx * x) - size, CInt(ky * y) + size, CInt(kx * x)
+ size, CInt(ky * y) - size)

                Case 2 '○記号
                    g.DrawEllipse(MyPen, CInt(kx * x) - size, CInt(ky * y) - size, 2 * size, 2
* size)


                Case 3 '●記号
                    g.FillEllipse(MyBrush, CInt(kx * x) - size, CInt(ky * y) - size, 2 * size, 2
* size)
            End Select

        Catch
            MsgBox("計算が正しくありません。")
        End Try

    End Sub
End Class

'フォーム
Public Class MainForm

    Private g1 As Graph

    Const dx As Integer = 55 '横の余白
    Const dy As Integer = 45 '縦の余白

    Public y_min As Double = 0.001 ' x の最小値
    Public y_max As Double = 1000 ' x の最大値
    Public x_min As Double = 0.001 ' y の最小値
    Public x_max As Double = 100 ' y の最大値

    Private log_y_min As Double = Math.Log10(y_min)
    Private log_y_max As Double = Math.Log10(y_max)
    Private log_x_min As Double = Math.Log10(x_min)
    Private log_x_max As Double = Math.Log10(x_max)
```

```
Public W_index(200) As Double 'W の値を格納する配列
Public U_index(200) As Double 'U の値を格納する配列

Public Plot As Integer 'プロットの数
Public k1, k2 As Integer 'プロットの範囲

Public line As Integer = 0 '曲線の数をカウント
Public consts(20, 3) As Double '曲線ごとの定数を格納(曲線の数の上限、4 つの定数)

Private k(10) As Integer 'プロットの範囲を記憶

Private ret As DialogResult 'ダイアログの結果を格納する変数

'初期化
Private Sub MainForm_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

    log_y_min = Math.Log10(y_min)
    log_y_max = Math.Log10(y_max)
    log_x_min = Math.Log10(x_min)
    log_x_max = Math.Log10(x_max)

    g1 = New Graph(pcbox1, log_x_min, log_x_max, log_y_min, log_y_max, dx, dy)

    GraphBase(g1, x_min, x_max, y_min, y_max)

End Sub
'Excel ファイルの場合
<System.Runtime.InteropServices.ComVisible(True)> Private Sub Excel ファイル EToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Excel ファイル EToolStripMenuItem.Click
    'Excel 関連の変数

    Dim xlsSheet As Object
    Dim xls As Object
    Dim xlsBook As Object

    Dim x_max2, x_min2, y_max2, y_min2 As Double

    'ダイアログの設定
    OpenFileDialog1.FileName = ""
    OpenFileDialog1.Filter = "Excel ファイル(*.xls) | *.xls"
    OpenFileDialog1.InitialDirectory = "C:¥temp"
```

```
        ret = OpenFileDialog1.ShowDialog()

        Try
            'ダイアログを開いたら以下の処理
            If ret = Windows.Forms.DialogResult.OK Then

                xls = CreateObject("Excel.Application")
                xlsBook = xls.Workbooks.Open(OpenFileDialog1.FileName)
                xlsSheet = xlsBook.sheets("Sheet1")

                'UW の計算
                figure(xlsSheet, W_index, U_index, Plot)

                x_max2 = W_index(Plot - 1)
                x_min2 = W_index(0)
                y_max2 = U_index(0)
                y_min2 = U_index(Plot - 1)

                'グラフメモリの再計算
                max_min(x_min, x_max, y_min, y_max, x_min2, x_max2, y_min2, y_max2)

                'グラフの書き込み()
                MainForm_Load(sender, e)

                'U と W をプロット
                PlotPoint(g1, W_index, U_index, Plot)

                Label12.Text = Plot

                'Excel の終了
                xls.Quit()
                xlsSheet = Nothing
                xls = Nothing
                xlsBook = Nothing
            End If
        Catch
            MsgBox("Sheet1 に正しい内容を入力してください")
        End Try
    End Sub

    'CSV ファイルの場合
    Private Sub CSV ファイル CToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CSV ファイル CToolStripMenuItem.Click
        'CSV 読み込み
```

```
        Dim sr As System.IO.StreamReader
        Dim x_max2, x_min2, y_max2, y_min2 As Double

        OpenFileDialog1.FileName = ""
        OpenFileDialog1.Filter = "CSV ファイル(*.csv) | *.csv"
        OpenFileDialog1.InitialDirectory = "C:¥temp"

        ret = OpenFileDialog1.ShowDialog()

        Try
            'ダイアログを開いたら以下の処理
            If ret = Windows.Forms.DialogResult.OK Then

                'CSV ファイルの読み込み
                sr = New System.IO.StreamReader(OpenFileDialog1.FileName,
System.Text.Encoding.Default)

                '計算
                figure(sr, W_index, U_index, Plot)

                x_max2 = W_index(Plot - 1)
                x_min2 = W_index(0)
                y_max2 = U_index(0)
                y_min2 = U_index(Plot - 1)

                'グラフメモリの再計算
                max_min(x_min, x_max, y_min, y_max, x_min2, x_max2, y_min2, y_max2)

                'グラフの書き込み
                MainForm_Load(sender, e)

                'U と W をプロット
                PlotPoint(g1, W_index, U_index, Plot)

                Label12.Text = Plot

                sr.Close()
            End If
        Catch ex As Exception
            MsgBox("このファイルの内容では計算できません。")
        End Try
    End Sub
    'フォームのサイズを変更したとき
    Private Sub frmGraphsample1_SizeChanged(ByVal sender As Object, ByVal e As
```

```
System.EventArgs) Handles Me.SizeChanged
        Dim A, c, W, m As Double
        g1 = New Graph(pcbox1, log_x_min, log_x_max, log_y_min, log_y_max, dx, dy)

        'メモリの呼び出し
        GraphBase(g1, x_min, x_max, y_min, y_max)

        'ダイアログを開いていない場合ここで終了
        If ret = Windows.Forms.DialogResult.OK Then

            'U と W をプロット
            PlotPoint(g1, W_index, U_index, Plot)

        End If
        For i = 0 To line

            A = consts(i, 0)
            c = consts(i, 1)
            W = consts(i, 2)
            m = consts(i, 3)

            curve(g1, A, c, W, m, x_min, x_max)

        Next
    End Sub
    '曲線の描画
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim A, c, W, m As Double

        consts(line, 0) = Val(TextBox3.Text)
        consts(line, 1) = Val(TextBox2.Text)
        consts(line, 2) = Val(TextBox4.Text)
        consts(line, 3) = Val(TextBox1.Text)

        A = consts(line, 0)
        c = consts(line, 1)
        W = consts(line, 2)
        m = consts(line, 3)


        g1 = New Graph(pcbox1, log_x_min, log_x_max, log_y_min, log_y_max, dx, dy)
        GraphBase(g1, x_min, x_max, y_min, y_max)
```

```
'ファイルを開いているときだけプロットと誤差の計算-------------------------
If ret = Windows.Forms.DialogResult.OK Then

    k1 = Val(TextBox5.Text)
    k2 = Val(TextBox6.Text)

    '誤差を計算する範囲を設定する処理
    If k1 = 0 Or k2 < k1 Or k1 > Plot Then
        k1 = 1
    End If
    If k2 > Plot Or k2 = 0 Or k2 > Plot Then
        k2 = Plot
    End If

    TextBox5.Text = k1
    TextBox6.Text = k2

    'U と W をプロット
    PlotPoint(g1, W_index, U_index, Plot, k1, k2)

    '誤差の表示
    Label7.Text = Format(LSM(A, c, W, m, W_index, U_index, k2, k1), "0.0000")

    k(line) = k2
End If '-----------------------------------------------------------------

'line の数だけ曲線を描く
For i = 0 To line

    A = consts(i, 0)
    c = consts(i, 1)
    W = consts(i, 2)
    m = consts(i, 3)

    curve(g1, A, c, W, m, x_min, x_max)

    g1.Drawtext(i + 1 & "段階", Math.Log10(x_min) + (i * 0.8) + 0.2,
Math.Log10(y_min) + 1.2, Color.Black, 9)
    g1.Drawtext("m =" & m, Math.Log10(x_min) + (i * 0.8) + 0.2,
Math.Log10(y_min) + 1.0, Color.Black, 9)
    g1.Drawtext("c =" & c, Math.Log10(x_min) + (i * 0.8) + 0.2, Math.Log10(y_min)
+ 0.8, Color.Black, 9)
    g1.Drawtext("A =" & A, Math.Log10(x_min) + (i * 0.8) + 0.2, Math.Log10(y_min)
+ 0.6, Color.Black, 9)
```

```
            g1.Drawtext("W =" & W, Math.Log10(x_min) + (i * 0.8) + 0.2,
Math.Log10(y_min) + 0.4, Color.Black, 9)

        Next
    End Sub

    '曲線を自動的に引く、計算速度遅
    Private Sub 自動描写ToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles 自動描写ToolStripMenuItem.Click
        Dim A, c, W, m, s1, s2 As Double
        Dim min As Double = 1000
        Dim A_sub, c_sub, W_sub As Double
        Dim U As Double = 0
        Dim U2 As Double = 0

        m = 1
        s1 = Math.Log10(x_max) - Math.Log10(x_min)
        s2 = s1 / (s1 * 10)

        consts(line, 0) = Val(TextBox3.Text)
        consts(line, 1) = Val(TextBox2.Text)
        consts(line, 2) = Val(TextBox4.Text)
        consts(line, 3) = Val(TextBox1.Text)

        g1 = New Graph(pcbox1, log_x_min, log_x_max, log_y_min, log_y_max, dx, dy)
        GraphBase(g1, x_min, x_max, y_min, y_max)

        'Excel ファイルを開いているときだけプロットの計算
        If ret = Windows.Forms.DialogResult.OK Then

            k1 = Val(TextBox5.Text)
            k2 = Val(TextBox6.Text)

            If k1 = 0 Or k2 < k1 Or k1 > Plot Then
                k1 = 1
            End If
            If k2 > Plot Or k2 = 0 Or k2 > Plot Then
                k2 = Plot
            End If

            TextBox5.Text = k1
            TextBox6.Text = k2

            'U と W をプロット
```

```
        PlotPoint(g1, W_index, U_index, Plot, k1, k2)

        '一番誤差が低いときの定数を返す処理
        '定数を変化させるループ
        For N As Integer = 0 To s1 * 10
            W_sub = 10 ^ (Math.Log10(x_min) + (N * s2))

            For A_sub = 0.01 To 4 Step 0.01
                For c_sub = -1 To 2 Step 0.05
                    '最小二乗法の計算()
                    U = LSM(A_sub, c_sub, W_sub, m, W_index, U_index, k2, k1)

                    ' 最小値を返す()
                    If U < min Then
                        min = U
                        A = A_sub
                        c = c_sub
                        W = W_sub

                    End If

                Next
            Next
        Next

        W = keta(W, 3, 0)

        consts(line, 0) = keta(A, 3, 0)
        consts(line, 1) = keta(c, 3, 0)
        consts(line, 2) = keta(W, 3, 0)
        consts(line, 3) = keta(m, 3, 0)

        TextBox3.Text = consts(line, 0)
        TextBox2.Text = consts(line, 1)
        TextBox4.Text = consts(line, 2)
        TextBox1.Text = consts(line, 3)

        '誤差の表示
        Label7.Text = Format(min, "0.0000")
        k(line) = k2
    End If

                    'line の数だけ曲線を描く
                    For i = 0 To line
```

```
                    A = consts(i, 0)
                    c = consts(i, 1)
                    W = consts(i, 2)
                    m = consts(i, 3)

                    curve(g1, A, c, W, m, x_min, x_max)

                    g1.Drawtext(i + 1 & "段階", Math.Log10(x_min) + (i * 0.8) + 0.2,
Math.Log10(y_min) + 1.2, Color.Black, 9)
                    g1.Drawtext("m =" & m, Math.Log10(x_min) + (i * 0.8) + 0.2,
Math.Log10(y_min) + 1.0, Color.Black, 9)
                    g1.Drawtext("c =" & c, Math.Log10(x_min) + (i * 0.8) + 0.2,
Math.Log10(y_min) + 0.8, Color.Black, 9)
                    g1.Drawtext("A =" & A, Math.Log10(x_min) + (i * 0.8) + 0.2,
Math.Log10(y_min) + 0.6, Color.Black, 9)
                    g1.Drawtext("W =" & W, Math.Log10(x_min) + (i * 0.8) + 0.2,
Math.Log10(y_min) + 0.4, Color.Black, 9)

                Next
    End Sub
    '終了
    Private Sub 終了ToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles 終了ToolStripMenuItem.Click

        Me.Close()
        End
    End Sub
    '曲線のやり直し
    Private Sub 消去CToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles 消去CToolStripMenuItem.Click
        pcbox1.Refresh()
        g1 = New Graph(pcbox1, log_x_min, log_x_max, log_y_min, log_y_max, dx, dy)
        GraphBase(g1, x_min, x_max, y_min, y_max)

        line = 0
        Label10.Text = 1

        Erase k
        ReDim k(10)

        If ret = Windows.Forms.DialogResult.OK Then

            'U と W をプロット
```

```
            PlotPoint(g1, W_index, U_index, Plot)
        End If

    End Sub

    '曲線を追加
    Private Sub 追加描写AToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles 追加描写AToolStripMenuItem.Click
        Dim A, c, W, m As Double


        g1 = New Graph(pcbox1, log_x_min, log_x_max, log_y_min, log_y_max, dx, dy)
        GraphBase(g1, x_min, x_max, y_min, y_max)

        'Excel ファイルを開いているときだけプロットの計算
        If ret = Windows.Forms.DialogResult.OK Then

            k1 = Val(TextBox5.Text)
            k2 = Val(TextBox6.Text)

            If k1 = 0 Or k2 < k1 Or k2 > Plot Then
                k1 = 1
            End If
            If k2 > Plot Or k2 = 0 Or k2 > Plot Then
                k2 = Plot
            End If

            TextBox5.Text = k1
            TextBox6.Text = k2

            'U と W をプロット
            PlotPoint(g1, W_index, U_index, Plot, k1, k2)

            '誤差の表示
            Label7.Text = Format(LSM(A, c, W, m, W_index, U_index, k2, k1), "0.0000")
            k(line) = k2
        End If

        '曲線を一本ずつ増やす処理----------------------------------------------------------

        consts(line, 0) = Val(TextBox3.Text)
        consts(line, 1) = Val(TextBox2.Text)
        consts(line, 2) = Val(TextBox4.Text) '上限値,これより増えるとエラー
        consts(line, 3) = Val(TextBox1.Text)
```

```
        'line の数だけ曲線を描く
        For i = 0 To line

            A = consts(i, 0)
            c = consts(i, 1)
            W = consts(i, 2)
            m = consts(i, 3)

            curve(g1, A, c, W, m, x_min, x_max)

            g1.Drawtext(i + 1 & "段階", Math.Log10(x_min) + (i * 0.8) + 0.2,
Math.Log10(y_min) + 1.2, Color.Black, 9)
            g1.Drawtext("m =" & m, Math.Log10(x_min) + (i * 0.8) + 0.2,
Math.Log10(y_min) + 1.0, Color.Black, 9)
            g1.Drawtext("c =" & c, Math.Log10(x_min) + (i * 0.8) + 0.2, Math.Log10(y_min)
+ 0.8, Color.Black, 9)
            g1.Drawtext("A =" & A, Math.Log10(x_min) + (i * 0.8) + 0.2, Math.Log10(y_min)
+ 0.6, Color.Black, 9)
            g1.Drawtext("W =" & W, Math.Log10(x_min) + (i * 0.8) + 0.2,
Math.Log10(y_min) + 0.4, Color.Black, 9)
        Next

        '曲線の数を増やす
        line += 1

        Label10.Text = line + 1
    End Sub
    '計算結果
    Private Sub 計算結果FToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles 計算結果FToolStripMenuItem.Click

        Dim xlsSheet As Object
        Dim xls As Object
        Dim xlsBook As Object

        xls = CreateObject("Excel.Application")
        xlsBook = xls.workbooks.add
        xlsSheet = xlsBook.sheets("Sheet1")
        xlsBook.Application.visible = True

        xlsSheet.cells(1, 2) = "w"
        xlsSheet.cells(1, 3) = "u"
        xlsSheet.cells(1, 4) = "A"
```

```
        xlsSheet.cells(1, 5) = "c"
        xlsSheet.cells(1, 6) = "W"
        xlsSheet.cells(1, 7) = "m"

        'U と W の計算結果
        For i = 1 To Plot
            xlsSheet.cells(i + 1, 1) = i
            xlsSheet.cells(i + 1, 2) = W_index(i - 1)
            xlsSheet.cells(i + 1, 3) = U_index(i - 1)
        Next

        '定数の計算結果
        For j = 0 To line
            xlsSheet.cells(k(j) + 1, 4) = consts(j, 0)
            xlsSheet.cells(k(j) + 1, 5) = consts(j, 1)
            xlsSheet.cells(k(j) + 1, 6) = consts(j, 2)
            xlsSheet.cells(k(j) + 1, 7) = consts(j, 3)
        Next

        xls.Quit()
        xlsSheet = Nothing
        xls = Nothing
        xlsBook = Nothing
    End Sub
    'グラフを画像として保存
    Private Sub グラフの保存 SToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles グラフの保存 SToolStripMenuItem.Click
        Dim ext As String

        SaveFileDialog1.FileName = "無題.gif"
        SaveFileDialog1.Filter = "GIF ファイル(*.gif) | *.gif"
        If SaveFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then

            ext = System.IO.Path.GetExtension(SaveFileDialog1.FileName).ToUpper()

            Select Case ext
                Case ".BMP", ".bmp"
                    pcbox1.Image.Save(SaveFileDialog1.FileName,
System.Drawing.Imaging.ImageFormat.Bmp)

                Case ".jpg", ".JPG"
                    pcbox1.Image.Save(SaveFileDialog1.FileName,
System.Drawing.Imaging.ImageFormat.Jpeg)
```

```
                Case ".gif", ".GIF"
                    pcbox1.Image.Save(SaveFileDialog1.FileName,
System.Drawing.Imaging.ImageFormat.Gif)

            End Select
        End If
    End Sub
    '拡大
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Dim x_max2, x_min2, y_max2, y_min2 As Double
        Dim z1, z2 As Integer

        If ret = Windows.Forms.DialogResult.OK Then

            'UW の計算
            z1 = Val(TextBox7.Text)
            z2 = Val(TextBox8.Text)

            x_max2 = W_index(z2 - 1)
            x_min2 = W_index(z1)
            y_max2 = U_index(z1)
            y_min2 = U_index(z2 - 1)

            'グラフメモリの再計算
            max_min(x_min, x_max, y_min, y_max, x_min2, x_max2, y_min2, y_max2)

            'グラフの書き込み()
            MainForm_Load(sender, e)

            'U と W をプロット
            ZoomPoint(g1, W_index, U_index, Plot, z1, z2)

        End If
    End Sub
    '縮小
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        Dim x_max2, x_min2, y_max2, y_min2 As Double

        If ret = Windows.Forms.DialogResult.OK Then
            x_max2 = W_index(Plot - 1)
            x_min2 = W_index(0)
            y_max2 = U_index(0)
```

```
            y_min2 = U_index(Plot - 1)

            'グラフメモリの再計算
            max_min(x_min, x_max, y_min, y_max, x_min2, x_max2, y_min2, y_max2)

            'グラフの書き込み()
            MainForm_Load(sender, e)

            'U と W をプロット
            PlotPoint(g1, W_index, U_index, Plot)
        End If
    End Sub
End Class

'MainForm で使う関数
Module figure1
    'グラフメモリ
    Public Sub GraphBase(ByVal g1 As Graph, ByVal x_min As Double, ByVal x_max As
Double, ByVal y_min As Double, ByVal y_max As Double)
        Try
            'グラフ軸の設定
            g1.Axis(Math.Log10(x_min), Math.Log10(x_max) + 0.1, Math.Log10(y_min),
Math.Log10(y_max) + 0.1, Color.Black, 2)

            'X 軸のメモリ(対数)
            '主軸
            For i = Math.Log10(x_min) + 1 To Math.Log10(x_max)
                g1.Line(i, Math.Log10(y_min), i, Math.Log10(y_max) + 0.1, Color.Gray, 2)
            Next

            For j = 0 To Math.Log10(x_max) - Math.Log10(x_min) - 1
                For i As Double = x_min * 10 ^ j To x_min * 10 ^ (j + 1) Step x_min * 10 ^ j
                    g1.Line(Math.Log10(i), Math.Log10(y_min), Math.Log10(i),
Math.Log10(y_min) + 0.1, Color.Black, 1)
                Next
            Next

            'Y 軸のメモリ(対数)
            '主軸
            For i = Math.Log10(y_min) + 1 To Math.Log10(y_max)
                g1.Line(Math.Log10(x_min), i, Math.Log10(x_max) + 0.05, i, Color.Gray, 2)
            Next

            For j = 0 To Math.Log10(y_max) - Math.Log10(y_min) - 1
```

```
                For i As Double = y_min * 10 ^ j To y_min * 10 ^ (j + 1) Step y_min * 10 ^ j
                    g1.Line(Math.Log10(x_min), Math.Log10(i), Math.Log10(x_min) + 0.05,
Math.Log10(i), Color.Black, 1)
                Next
            Next

            'X 軸メモリの数値
            For i = Math.Log10(x_min) To Math.Log10(x_max) Step 1
                g1.Drawtext("10^" & i, Math.Log10(10 ^ i), Math.Log10(y_min) - 0.1,
Color.Black, 10)
            Next

            'Y 軸メモリの数値
            For i = Math.Log10(y_min) To Math.Log10(y_max) Step 1
                g1.Drawtext_y_Axis1("10^" & i, Math.Log10(x_min), Math.Log10(10 ^ i),
Color.Black, 8)
            Next

            g1.Drawtext_x_Axis1("w", (Math.Log10(x_max) + Math.Log10(x_min)) / 2,
Math.Log10(y_min), Color.Black, 14)
            g1.Drawtext_y_Axis2("U", Math.Log10(x_min), (Math.Log10(y_max) +
Math.Log10(y_min)) / 2, Color.Black, 14)

        Catch ex As Exception
            MsgBox(ex.Message)

            Exit Sub
        End Try
    End Sub
    'Excel のセルの値から W と U の値を計算
    Public Sub figure(ByVal Sheet As Object, ByRef W_index() As Double, ByRef U_index()
As Double, ByRef i As Integer)
        Dim a, b, c, X As Double
        Dim s, t, r As Integer
        Dim N As Integer


        Try
            '数値がある行を探索
            For i = 1 To 300
                If IsNumeric(Sheet.Cells(i, 1).value) = True Then
                    N = i
                    Exit For
                End If
```

```
            Next



            '空セルまで繰り返す
            For i = N To 300

                If Sheet.Cells(i + 1, 1).value = Nothing Then Exit For

                '時間間隔
                s = Sheet.Cells(i, 1).value 't1
                t = Sheet.Cells(i + 1, 1).value 't2
                r = t - s 't2 - t1

                '幹樹材積の差
                a = Sheet.Cells(i, 2).value 'w1
                b = Sheet.Cells(i + 1, 2).value 'w2
                c = b - a 'w2 -w1
                X = Math.Log(b) - Math.Log(a) 'lnw2-lnw1

                'U-W の値
                W_index(i - N) = c / X
                U_index(i - N) = (X ^ 2) / (c * r)
            Next

            i = i - N
        Catch
            MsgBox("セルの値、または値の場所が正しくありません。")
        End Try
    End Sub
    'CSV ファイルの値から W と U の値を計算
    Public Sub figure(ByVal sr As System.IO.StreamReader, ByRef W_index() As Double,
ByRef U_index() As Double, ByRef i As Integer)
        Dim dat As String
        Dim sbuf(1) As String
        Dim delim() As Char = {","c}
        Dim data(30, 1) As Double
        Dim a, b, c, X As Double
        Dim s, t, r As Integer

        Try
            i = 0

            '空セルまで繰り返す
```

```
            Do While sr.Peek() >= 0

                '1 行分
                dat = sr.ReadLine()
                sbuf = dat.Split(delim)

                data(i, 0) = CDbl(sbuf(0))
                data(i, 1) = CDbl(sbuf(1))

                i += 1
            Loop

            For j = 0 To i - 2

                If data(j + 1, 0) = False Then Exit For

                '時間間隔
                s = data(j, 0) 't1
                t = data(j + 1, 0) 't2
                r = t - s 't2 - t1

                '幹樹材積の差
                a = data(j, 1) 'w1
                b = data(j + 1, 1) 'w2
                c = b - a 'w2 -w1
                X = Math.Log(b) - Math.Log(a) 'lnw2-lnw1

                'U-W の値
                W_index(j) = c / X
                U_index(j) = (X ^ 2) / (c * r)
            Next

            i = i - 1
        Catch
            MsgBox("セルの値、または値の場所が正しくありません。")
        End Try
    End Sub
    '計算結果をプロット
    Sub PlotPoint(ByVal g1 As Graph, ByVal W_index() As Double, ByVal U_index() As
Double, ByRef j As Integer, ByVal k1 As Integer, ByVal k2 As Integer)
        Try
            For i = 0 To j - 1
                If k1 - 1 <= i And k2 - 1 >= i Then
                    g1.Plot(Math.Log10(W_index(i)), Math.Log10(U_index(i)), Color.Blue, 4,
```

```
3)
                Else
                    g1.Plot(Math.Log10(W_index(i)), Math.Log10(U_index(i)), Color.Blue, 4,
2)
                End If
            Next

        Catch ex As Exception
            MsgBox(ex.Message)

        End Try
    End Sub
    'k1 と k2 を返さない PlotPoint
    Sub PlotPoint(ByVal g1 As Graph, ByVal W_index() As Double, ByVal U_index() As
Double, ByRef j As Integer)
        Try
            For i = 0 To j - 1

                g1.Plot(Math.Log10(W_index(i)), Math.Log10(U_index(i)), Color.Blue, 4, 2)

            Next
        Catch ex As Exception
            MsgBox(ex.Message)

        End Try
    End Sub
    '拡大図用
    Sub ZoomPoint(ByVal g1 As Graph, ByVal W_index() As Double, ByVal U_index() As
Double, ByRef j As Integer, ByVal z1 As Integer, ByVal z2 As Integer)
        Try
            For i = z1 To z2 - 1

                If i + 1 > i And z2 <= j Then
                    g1.Plot(Math.Log10(W_index(i)), Math.Log10(U_index(i)), Color.Blue, 4,
2)
                End If
            Next

        Catch ex As Exception
            MsgBox(ex.Message)

        End Try
    End Sub
    '残差平方和
```

```
Function LSM(ByVal A As Double, ByVal c As Double, ByVal W As Double, ByVal m As Double, ByRef W_index() As Double, ByRef U_index() As Double, ByVal K2 As Integer, ByVal K1 As Double) As Double

    Dim X, fx, u As Double
    Dim sum As Double = 0

    '誤差を計算する範囲を設定
    For i = K1 - 1 To K2 - 1 Step 1

        X = W_index(i)
        fx = (A * (X ^ -c)) * (1 - (X / W) ^ m)
        u = U_index(i)

        sum += ((Math.Log10(fx) - Math.Log10(u)) ^ 2)
    Next

    '差の二乗の和を返す
    Return sum

End Function
'未使用
Function LSM_CS(ByVal A As Double, ByVal c As Double, ByVal W As Double, ByVal m As Double, ByRef W_index() As Double, ByRef U_index() As Double, ByVal K2 As Integer, ByVal K1 As Double) As Double

    Dim X, fx, u, xx As Double
    Dim sum As Double = 0
    Dim sum2 As Double = 0
    Dim xbar, sigma2, sigma As Double

    '誤差を計算する範囲を設定
    For i = K1 - 1 To K2 - 1 Step 1

        X = W_index(i)
        fx = (A * (X ^ -c)) * (1 - (X / W) ^ m)
        u = U_index(i)

        'xx = Math.Abs((Math.Log10(fx) - Math.Log10(u)))
        xx = (Math.Log10(fx) - Math.Log10(u)) ^ 2

        sum += xx
        sum2 += xx ^ 2
```

```
        Next

        xbar = sum / (K2 - K1 + 1)
        sigma2 = (sum2 - (K2 - K1 + 1) * xbar ^ 2) / (K2 - K1 + 1)
        sigma = Math.Sqrt(sigma2)

        '差の二乗の和を返す
        Return sigma

    End Function
    '曲線を引く処理
    Sub curve(ByVal g1 As Graph, ByVal A As Double, ByVal c As Double, ByVal W As Double, ByVal m As Double, ByVal x_min As Double, ByVal x_max As Double)
        Dim X, Y As Double

        '曲線の一番最初の点
        X = x_min

        'テキストボックスの値が正しくないときの処理
        If W <= X Or Math.Abs(A) = 0 Or c = 0 Or m = 0 Then Exit Sub

        Y = (A * (X ^ -c)) * (1 - (X / W) ^ m)
        g1.PutPoint(Math.Log10(X), Math.Log10(Y), Color.Black)

        '曲線を続けて引く処理
        For n = 0 To Math.Log10(x_max) - Math.Log10(x_min) - 1
            For X = x_min * 10 ^ n To x_min * 10 ^ (n + 1) Step x_min * 10 ^ (n - 2)

                Y = (A * (X ^ -c)) * (1 - (X / W) ^ m)

                'w が上限値を超えるまで線を引く
                If W > X Then
                    g1.LineTO(Math.Log10(X), Math.Log10(Y), Color.Red)
                ElseIf W <= X Then
                    g1.LineTO(Math.Log10(X), -20, Color.Red)
                    Exit Sub
                End If

            Next
        Next
    End Sub
    'UW の値によってグラフを変える処理
    Sub max_min(ByRef x_min As Double, ByRef x_max As Double, ByRef y_min As Double, ByRef y_max As Double, ByVal x_min2 As Double, ByVal x_max2 As Double, ByVal y_min2
```

```
As Double, ByVal y_max2 As Double)
        x_min = xmin(x_min, x_min2)
        x_max = xmax(x_max, x_max2)
        y_min = ymin(y_min, y_min2)
        y_max = ymax(y_max, y_max2)
    End Sub
    Function keta(ByVal x As Double, ByVal N As Integer, ByVal c As Integer)
        Dim t, m As Double
        ' x を n 桁かつ約（ c ＝ 1 ：切りあげ， c ＝ 0 ：四捨五入： c ＝－ 1 ：切り捨て）
        t = 0.5 * c + 0.5
        m = Int(Math.Log(x) / Math.Log(10))
        keta = Int(10 ^ (N - 1 - m) * x + t) * 10 ^ (m + 1 - N)
    End Function

    '以下 4 つ初期値を変える処理
    Function xmin(ByVal x_min As Double, ByVal min As Double) As Double
        For n = -10 To 10
            For i = 10 ^ n To 10 ^ (n + 1) Step 10 ^ n

                If min < i Then
                    x_min = 10 ^ n

                    Return x_min
                End If
            Next
        Next

    End Function
    Function xmax(ByVal x_max As Double, ByVal max As Double) As Double
        For n = -10 To 10
            For i = 10 ^ n To 10 ^ (n + 1) Step 10 ^ n

                If max < i Then
                    x_max = 10 ^ (n + 2)

                    Return x_max
                End If
            Next
        Next
    End Function
    Function ymin(ByVal y_min As Double, ByVal min As Double) As Double
        For n = -10 To 10
            For i = 10 ^ n To 10 ^ (n + 1) Step 10 ^ n
                If min < i Then
```

```
                y_min = 10 ^ n

                Return y_min
            End If
        Next
    Next
End Function
Function ymax(ByVal y_max As Double, ByVal max As Double) As Double

    For n = -10 To 10
        For i = 10 ^ n To 10 ^ (n + 1) Step 10 ^ n

            If max < i Then
                y_max = 10 ^ (n + 1)

                Return y_max
            End If
        Next
    Next
End Function
End Module
```