Use of a symbolic math system to solve polarized light problems

James C. Wyant

The use of a symbolic math system, muMATH-79, to solve polarized light problems is described. The problems are setup using Jones calculus, and muMATH running on a Z-80 microprocessor multiplies out the Jones matrices and simplifies the final algebraic expressions. Only a minimal amount of operator interaction with the microcomputer is required. Several examples are given.

I. Introduction

Jones calculus is a commonly used tool for solving polarized light problems involving polarizers, retarders, and wave plates. While Jones calculus involves only 2 × 2 matrices and two row vectors, only a few polarization components need be present before the algebraic manipulations present are sufficiently involved that errors become common. If only numerical results are desired, it is fairly easy to use a computer to solve the problem and reduce chances of obtaining an incorrect result. It is more difficult, however, when the answer sought is an algebraic equation rather than just a numerical result. This paper describes use of a software package called muMATH-79,[1] created by the Soft Warehouse of Honolulu, Haw. to run on a Z-80 microprocessor, multiply the matrices involved in Jones calculus, and simplify the resulting equations, including simplifications of expressions containing trigonometric functions. While the simplification is not done completely automatically, and some operator interaction is required, the process is much less tedious than doing all the work by hand, and the chances of introducing an error into the process are greatly reduced. Plus it is a lot of fun—and it is not often that algebra is fun.

muMATH-79, which is essentially a symbolic math system, has recently been described in the literature.[2,3] For readers who are not already familiar with muMATH, the reading of Ref. 2 is encouraged. muMATH has capabilities far greater than the ones described in this paper for working with Jones calculus. muMATH-79 is a modular system consisting of packages for 611-digit arithmetic, algebra, trigonometric manipulation, matrix manipulation, equation solving, logarithmic manipulation, integration, and differentiation. The software gives exact results and works with integers and fractions rather than decimals. For example, adding $1/3$ and $1/5$ gives $8/15$ rather than a number such as 0.533333. If you find 50! (50 factorial) on your calculator you will get a number such as 3.041409318E64. muMATH gives you all 65 digits. The point being stressed is that if muMATH gives you an answer, which is limited to 611 digits, it is exact and not just a close approximation that we are used to obtaining from calculators and computers. muMATH treats everything as a string of symbols. All algebraic operations work on two strings of symbols to give a third string as the result. The strings can consist of numbers, letters, or functions.

To use muMATH with Jones calculus to solve polarized light problems, four muMATH packages are required: arithmetic, algebra, trigonometry, and matrix manipulation. About 48K of memory is required for these four packages and storing the Jones matrices described below plus memory space to perform the desired calculations.

II. Jones Calculus

It will be assumed that the reader is familiar with the use of Jones calculus. For those not familiar several excellent references exist.[4,5] Our preliminary discussion of Jones calculus will consist only of the listing of the Jones vectors and matrices required for muMATH calculations.

The author is with University of Arizona, Optical Sciences Center, Tucson, Arizona 85721.

Received 19 March 1981.
0003-6935/81/193321-06$00.50/0.

Table I. Jones Vectors for Polarized Light

State of polarization	Jones vector
Horizontal P-state (HOR)	$\begin{vmatrix} 1 \\ 0 \end{vmatrix}$
Vertical P-state (VER)	$\begin{vmatrix} 0 \\ 1 \end{vmatrix}$
P state at +45° ($P45$)	$\frac{1}{\sqrt{2}} \begin{vmatrix} 1 \\ 1 \end{vmatrix}$
P state at −45° ($PM45$)	$\frac{1}{\sqrt{2}} \begin{vmatrix} 1 \\ -1 \end{vmatrix}$
Right circularly polarized ($RCIR$)	$\frac{1}{\sqrt{2}} \begin{vmatrix} 1 \\ -i \end{vmatrix}$
Left circularly polarized ($LCIR$)	$\frac{1}{\sqrt{2}} \begin{vmatrix} 1 \\ i \end{vmatrix}$

Table I gives the basic Jones vectors for linearly polarized light oriented horizontally, vertically, at ±45°, and right- and left-handed circularly polarized light. Table II gives the Jones matrices for common polarizers and wave plates. The Jones matrices for other orientations can be obtained using the rotation matrices $ROTP$ and $ROTM$ where

$$ROTP = \begin{vmatrix} \cos(TH) & \sin(TH) \\ -\sin(TH) & \cos(TH) \end{vmatrix}, \quad ROTM = \begin{vmatrix} \cos(TH) & -\sin(TH) \\ \sin(TH) & \cos(TH) \end{vmatrix}, \tag{1}$$

where TH is the angle of rotation.

If $R(0)$ is the Jones matrix for an element of orientation 0° and $R(\theta)$ is the Jones matrix for the same element rotated an angle θ,

$$R(\theta) = ROTM \cdot R(0) \cdot ROTP. \tag{2}$$

By using Eq. (1) and Table II almost any desired Jones matrix can be found.

III. Use of muMATH

The first step in using muMATH to solve polarized light problems after a software package consisting of the arithmetic, algebra, trigonometry, and matrix packages are put together as described in the muMATH-79 manual[1] is to type in the vectors and matrices given in Tables I and II. While this is a tedious process, it needs to be done only once, since once this package is created, it can be saved on disk, and it can be the starting point for future problem solving.

muMATH prompts user input with a question mark and a space. All user commands must end with a ; or a $. When a $ is used the answer is not typed out. If **HOR** is the Jones vector for horizontally polarized light, we could define **HOR** with the following statement:

```
? HOR:{1,0};
@ {1,                (3)
   0}.
```

The first line is the input line (? was typed by the computer), and the second line is the computer's response showing that it accepted our definition. { and } are used at the beginning and ends of the matrix or vector. As shown in the next example, [and] are used at the beginning and ends of rows of a matrix. If $HPOL$ is the Jones matrix for a horizontal linear polarizer, $HPOL$ can be defined with the following statement:

```
? HPOL:{[1,0],[0,0]};
@ {[1, 0],           (4)
   [0, 0]}.
```

Again the second (and third) lines are the computer's response. In the same manner all the vectors and matrices given in Tables I and II and $ROTP$ and $ROTM$ can be stored in the program. It should be remembered that ½ is written as ½ and not as 0.5. $\sqrt{2}$ is written as 2^(½) [or perhaps 2↑(½), depending upon the computer display]. π is written as #PI, e is written as #E, and i is #I. Hence expn(iP) is written as #E^ #I*P. muMATH knows that #I is the square root of −1 and that #E represents the base of the natural logarithms.

While almost any other Jones matrix that is wanted can be obtained using muMATH and the matrices in Table II, plus the rotation matrices, there are a few that are used often enough that it makes sense to calculate them once and store them for future use. For example, the matrix of a retarder of retardation phase P having a fast axis at angle TH from the horizontal is frequently used. From Eq. (2) it follows that to obtain this matrix we need to multiply matrix $ROTM$ times matrix $RETH$

Table II. Jones Matrices for Common Polarizers and Waveplates

Optical element	Jones matrix
Horizontal linear polarizer ($HPOL$)	$\begin{vmatrix} 1 & 0 \\ 0 & 0 \end{vmatrix}$
Vertical linear polarizer ($VPOL$)	$\begin{vmatrix} 0 & 0 \\ 0 & 1 \end{vmatrix}$
Linear polarizer at +45° ($POL45$)	$\frac{1}{2}\begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix}$
Linear polarizer at −45° ($POLM45$)	$\frac{1}{2}\begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix}$
λ/4 plate, fast axis vertical (QWV)	$\mathrm{expn}(i\pi/4)\begin{vmatrix} 1 & 0 \\ 0 & -i \end{vmatrix}$
λ/4 plate, fast axis horizontal (QWH)	$\mathrm{expn}(-i\pi/4)\begin{vmatrix} 1 & 0 \\ 0 & i \end{vmatrix}$
λ/2 plate, fast axis vertical (HWV)	$\mathrm{expn}(+i\pi/2)\begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix}$
λ/2 plate, fast axis horizontal (HWH)	$\mathrm{expn}(-i\pi/2)\begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix}$
Retarder, fast axis vertical (retardation phase = P) ($RETV$)	$\mathrm{expn}(iP/2)\begin{vmatrix} 1 & 0 \\ 0 & \mathrm{expn}(-iP) \end{vmatrix}$
Retarder, fast axis horizontal (retardation phase = P) ($RETV$)	$\mathrm{expn}(-iP/2)\begin{vmatrix} 1 & 0 \\ 0 & \mathrm{expn}(iP) \end{vmatrix}$

times matrix *ROTP*. The command used to obtain this matrix, which will be called *RETTH* (retarder at angle *TH*), and the computer's response is given in Fig. 1.

Figure 1 illustrates the main drawback of using muMATH; it is difficult to read computer written equations. Fortunately, with experience the equations become easier to read. For the reader's convenience, the computer's response given in Fig. 1 can be rewritten as

$$RETTH = \begin{vmatrix} \cos^2(TH)\,\mathrm{expn}(-iP/2) + \mathrm{expn}(iP/2)\sin^2(TH) & \cos(TH)\sin(TH)\,\mathrm{expn}(-iP/2) - \mathrm{expn}(iP/2)\cos(TH)\sin(TH) \\ \cos(TH)\sin(TH)\,\mathrm{expn}(-iP/2) - \mathrm{expn}(iP/2)\cos(TH)\sin(TH) & \sin^2(TH)\,\mathrm{expn}(-iP/2) + \mathrm{expn}(iP/2)\cos^2(TH) \end{vmatrix}. \quad (5)$$

```
? RETTH:ROTM.RETH.ROTP;

@ {[COS(TH)^2/#E^(#I*P/2)+#E^(#I*P/2)*SIN(TH)^2, COS(TH)*SIN(
TH)/#E^(#I*P/2)-#E^(#I*P/2)*COS(TH)*SIN(TH)],
   [COS(TH)*SIN(TH)/#E^(#I*P/2)-#E^(#I*P/2)*COS(TH)*SIN(TH),
SIN(TH)^2/#E^(#I*P/2)+#E^(#I*P/2)*COS(TH)^2]}
```

Fig. 1. Calculation of matrix of a retarder of retardation *P* having a fast axis at an angle *TH* from the horizontal.

Table III. Effect of Control Variables in Algebra Packaqe on Algebraic Expressions[1]

Control variable	Transformation produced with positive value	Transformation produced with negative value
NUMNUM	$A(B+C) \to AB + AC$	$AB + AC \to A(B+C)$
DENDEN	$\frac{1}{A}\left(\frac{1}{B+C}\right) \to \frac{1}{AB+AC}$	$\frac{1}{AB+AC} \to \frac{1}{A}\left(\frac{1}{B+C}\right)$
DENNUM	$\frac{B+C}{A} \to \frac{B}{A} + \frac{C}{A}$	$\frac{B}{A} + \frac{C}{A} \to \frac{B+C}{A}$
NUMDEN	$\frac{A}{B+C} \to \frac{1}{\frac{B}{A}+\frac{C}{A}}$	$\frac{1}{\frac{B}{A}+\frac{C}{A}} \to \frac{A}{B+C}$
BASEXP	$A^{(B+C)} \to A^B A^C$	$A^B A^C \to A^{B+C}$
EXPBAS	$(AB)^C \to A^C B^C$	$A^C B^C \to (AB)^C$
PWREXPD	$(A+B)^2 \to A^2 + 2AB + B^2$	$(A+B)^{-2} \to \frac{1}{(A^2 + 2AB + B^2)}$

Table IV. Effect of Control Variable NUMNUM on Algebraic Expressions[2]

Value of NUMNUM	Example
0	$3A(B+C)(D+E) \to 3A(B+C)(D+E)$
2 and its multiples	$\to A(3B+3C)(D+E)$
3 and its multiples	$\to 3(AB+AC)(D+E)$
5 and its multiples	$\to 3A[D(B+C)+E(B+C)]$
6 (= 2 · 3)	$\to (3AB+3AC)(D+E)$
10 (= 2 · 5)	$\to A[D(3B+3C)+E(3B+3C)]$
15 (= 3 · 5)	$\to 3(ABD+ABE+ACD+ACE)$
30 (= 2 · 3 · 5)	$\to 3ABD+3ABE+3ACD+3ACE$
−2, −3, −6	Same as 2, 3, 6 except factor out instead of distribute

While the above expression for *RETTH* is correct, it can be simplified somewhat. While some of this simplification could be done by hand, we will let the computer do all the work. To do this simplification, the control variable must be changed. Table III gives a list of the control variables contained in the algebra package and illustrates the effect of assigning positive or negative values to the control variables. Table IV illustrates in detail how the value of a control variable *NUMNUM* changes the result. To change the value of a control variable, such as *NUMNUM*, to say 3, we can use the command

$$?\ NUMNUM{:}3;. \quad (6)$$

If we want to find the expression for *RETTH* now that *NUMNUM* is 6, we can use the command

$$?\ EVAL(RETTH);. \quad (7)$$

The control variables are described in sufficient detail in the muMATH-79 manual. It should be obvious that proper selection of the control variables can greatly simplify complicated expressions, while at the same time poor choice of the control variables can make the expression more complicated. This author can testify that little thought is required to change a simple one-line expression into an extremely complicated, but correct, eight-line expression. While initially the optimum use of control variables is difficult, after a little experience it becomes much easier. However, an easier way to select the proper control variables is described next.

The default values of the algebraic control variables are

$$NUMNUM = 6 \quad DENNUM = 6 \quad DENDEN = 2$$
$$NUMDEN = 0 \quad EXPBAS = 30 \quad BASEXP = -30.$$
$$PWREXPD = 0$$

As expected, for many operations, these values for the control variables are acceptable. The command *FLAGS*( ); will print out the current values of the control variables. Three additional commands give the control variables temporary assignments. These three temporary control variables assignments appear to be almost always sufficient, and additional variation of the algebraic control variables is rarely needed. The first command, *EXPAND* (*expr*), evaluates *expr* to yield a fully expanded denominator distributed over the terms

```
? FCTR(RETTH);

@ {[COS(TH)^2/#E^(#I*P/2)+#E^(#I*P/2)*SIN(TH)^2, (1/#E^(#I*P/
2)-#E^(#I*P/2))*COS(TH)*SIN(TH)],
   [(1/#E^(#I*P/2)-#E^(#I*P/2))*COS(TH)*SIN(TH), SIN(TH)^2/#E
^(#I*P/2)+#E^(#I*P/2)*COS(TH)^2]}


? TRGEXPD(#ANS,5);

@ {[COS(TH)^2/#E^(#I*P/2)+#E^(#I*P/2)*SIN(TH)^2, SIN(2*TH)/(2
*#E^(#I*P/2))-#E^(#I*P/2)*SIN(2*TH)/2],
   [SIN(2*TH)/(2*#E^(#I*P/2))-#E^(#I*P/2)*SIN(2*TH)/2, SIN(TH
)^2/#E^(#I*P/2)+#E^(#I*P/2)*COS(TH)^2]}

? TRGEXPD(#ANS,-7);

@ {[-#I*COS(TH)^2*SIN(P/2)+#I*SIN(TH)^2*SIN(P/2)+COS(TH)^2*COS
(P/2)+COS(P/2)*SIN(TH)^2, -#I*SIN(2*TH)*SIN(P/2)],
   [-#I*SIN(2*TH)*SIN(P/2), #I*COS(TH)^2*SIN(P/2)-#I*SIN(TH)^
2*SIN(P/2)+COS(TH)^2*COS(P/2)+COS(P/2)*SIN(TH)^2]}


? TRGEXPD(#ANS,3);

@ {[-#I*COS(2*TH)*SIN(P/2)+COS(P/2), -#I*SIN(2*TH)*SIN(P/2)],
   [-#I*SIN(2*TH)*SIN(P/2), #I*COS(2*TH)*SIN(P/2)+COS(P/2)]}


? RETTH:#ANS$
```

Fig. 2. Simplification of matrix *RETTH* given in Fig. 1.

of a fully expanded numerator. The second command, *EXPD* (*expr*), evaluates *expr* to yield a fully expanded numerator over a fully expanded denominator. The third command, *FCTR* (*expr*), evaluates *expr* to yield a semifactored numerator over a semifactored denominator. *EXPD* followed by *FCTR* is often useful in simplifying expressions. Sometimes it is useful to work on only a numerator or a denominator of an expression. The commands *NUM* (*expr*) and *DEN* (*expr*) return only the numerator or the denominator of an expression, so the numerator or denominator can be worked on separately.

In addition to the algebraic control variables there is a trigonometric control variable, *TRGEXPD*, that is extremely useful. *TRGEXPD* controls the use of multiple angle and angle sum expressions and replacement of trig functions by complex exponentials. The default value of *TRGEXPD* is 0. When *TRGEXPD* is a positive multiple of 2, tangents, cotangents, secants, and cosecants are replaced by corresponding expressions involving sines and cosines. Negative multiples of 2 have the opposite effect. If *TRGEXPD* is a positive multiple of 3, integer powers of signs and cosines are expanded in terms of sines and cosines of multiple angles. For example, when *TRGEXPD* = 30, $\cos^2(x) \rightarrow [1 + \cos(2x)]/2$. A negative multiple of 3 has the opposite effect. *TRGEXPD* = 30 is useful in proving most trig identities and appears to be extremely useful in simplifying algebraic expressions containing trig functions.

If *TRGEXPD* is a negative multiple of 5, sines and cosines of angle sums and differences are expanded in terms of sines and cosines of nonsums and nondifferences. For example, if *TRGEXPD* = −15, $\cos(x + y) \rightarrow \cos(x)\cos(y) - \sin(x)\sin(y)$. If *TRGEXPD* is a positive multiple of 7, sines and cosines are converted to complex exponentials. For example, when *TRGEXPD* = 14, then $\cos(x) \rightarrow [\#E\hat{}(\#I{*}x) + 1/\#E\hat{}(\#I{*}x)]/2$. If *TRGEXPD* is a negative multiple of 7 the opposite transformation is obtained.

Changing the value of an option variable does not affect the values of the expressions which have already been evaluated. After changing the value of *TRGEXPD*, or other variables, it may be necessary to use *EVAL* to get the desired effect. For the control variable *TRGEXPD*, a temporary change of its value can be obtained with the command *TRGEXPD* (*expr*, *N*), where now *expr* is evaluated with *TRGEXPD* set equal to *N*.

We now have enough background material to simplify our expressions for *RETTH*. There is of course no unique answer. The preferred final form and approach depend somewhat upon the person doing the simplification. One simplification procedure is illustrated in Fig. 2.

Looking at Fig. 1, or Eq. (5), it appears as though the first step in the simplification process should be to factor *RETTH*. Since the expression is already completely expanded, expanding before factoring will not change the results. The answer obtained using *FCTR*(*RETTH*); is stored as #*ANS*. After the computer does the expansion, it appears as though it would be good to use a trig identity to change a product of a sine times a cosine into a double-angle trig function. This is accomplished by using *TRGEXPD* (#*ANS*,5). Next it appears as though it would be advantageous to

```
? A:HWTH.RCIR;

@ {-#I*COS(2*TH)/2^(1/2)-SIN(2*TH)/2^(1/2),
   -#I*SIN(2*TH)/2^(1/2)+COS(2*TH)/2^(1/2)}

? FCTR(A);

@ {-(#I*COS(2*TH)+SIN(2*TH))/2^(1/2),
   (-#I*SIN(2*TH)+COS(2*TH))/2^(1/2)}

? A:#ANS$

? TRGEXPD(A,7);

@ {-#I/(2^(1/2)*#E^(2*#I*TH)),
   1/(2^(1/2)*#E^(2*#I*TH))}
```

Fig. 3. Example showing that a rotating halfwave plate frequency shifts circularly polarized light.

convert complex exponentials into sines and cosines, which is accomplished with the command *TRGEXPD*(#*ANS*,−7);. It appears as though the new #*ANS* can be simplified if integer powers of sines and cosines are expanded in terms of sines and cosines of multiple angles. The new #*ANS* obtained is acceptable as the final value of *RETTH*, so *RETTH* is set equal to #*ANS*.

There are six more Jones matrices that are used frequently enough that it is convenient to store them in the program. (Storing too many items in memory reduces the memory available for calculations, which results in additional computation time.) The six matrices are quarterwave and halfwave plates at ±45° (*QW*45, *QWM*45, *HW*45, and *HWM*45) and quarterwave and halfwave plates at angle *TH* (*QWTH* and *HWTH*). While these six matrices can be found in Ref. 4 as well as several other references, they can be easily calculated from the matrices already stored. As an example illustrating the use of muMATH, we will calculate the matrix representing a halfwave plate with the fast axis oriented at an angle of 45° with respect to the horizontal direction. As mentioned above, we will call this matrix *HW*45.

We have two convenient ways to calculate *HW*45 using matrices already stored in the program. One way is to operate on *QWH* with the rotation matrices *ROTM* and *ROTP*, where *TH* is set equal to #*PI*/4. The second technique is to evaluate *RETTH* with *TH* equal to #*PI*/4 and *P* equal to #*PI*. The procedure for using this second method for finding *QW*45 is illustrated below:

```
? TH:#PI/4$

? P:#PI$

? HW45:EVAL(RETTH);

@{[0, -#I],
  [-#I, 0]}

? TH:'TH$

? P:'P$.
```

In the last two steps *TH* and *P* are set back to their unbound (undefined) condition. It is important that these last two steps are included, since otherwise for future calculations *TH* is set equal to #*PI*/4, and *P* is set equal to #*PI*.

The other five matrices mentioned above can be found using the same technique. Once these matrices are stored in memory, our program should be saved for later use so we do not have to go through the setup procedure again. (The muMATH-79 manual describes the procedure for storing the program for future use.)

Now that we have our Jones calculus muMATH program, let us use it to solve an example polarized light problem. The example to be illustrated is that a rotating halfwave plate will frequency shift circularly polarized light. Figure 3 shows the solution. In memory, we have stored the expression for a halfwave plate at angle *TH*, *HWTH*. In the first command line we operate on right-handed circularly polarized light with *HWTH*. The result can be simplified somewhat by factoring out 2^(½) from each expression. To achieve this, we use the command *FCTR*( ). We obtained the desired result, so let us replace *A* with *FCTR*(*A*). The answer of *FCTR*(*A*) has been stored as #*ANS*, so *A*:#*ANS*$ replaces *A* with *FCTR*(*A*). It appears as though the new expressions for *A* can be simplified by expressing cos and sin as complex exponentials. To do so, we need to reevaluate with *TRGEXPD* set equal to 7. In the last command line we try this, and we get the desired result. Right-handed circularly polarized has been converted to left-handed circularly polarized light, and the phase of both x and y components have been multiplied by expn($-i2TH$). That is, a rotating halfwave plate where *TH* is linearly proportional to time, say $TH = \omega t$, will frequency shift the transmitted light.

IV. Conclusions

It is this author's experience that muMATH-79[6] is extremely useful for solving problems using Jones calculus. Problems considerably more complicated than the ones given in the examples have been solved using the same procedures illustrated in the examples. While at first the process appears mysterious, after a little experience the proper values of the control variables to be used, in particular the value of *TRGEXPD* and when to use *FCTR* or *EXPD*, become readily apparent. Sometimes an incorrect choice is made, however; if the algebra is done by hand we sometimes choose the incorrect trig identity, expansions, or factoring. When this is done, we simply go back a step or two and begin again. The thought process used is similar to what we would do if we were to do the algebra by hand. However, the advantage is that now we just have to decide what to do, and the computer does the work. Probably the biggest advantage is that the computer does not make a mistake in algebra as we easily can. While the expression may not be as simple as we might like, it is at least correct. The time required to perform each command is from a few seconds to perhaps a minute, or more, if an especially complicated expression is being evaluated or the computer memory is nearly full. The biggest problem with using muMATH is that the equations are not as easy to read as they would be if they were handwritten. While the expressions do become much easier to read with experience, the awkward way the equations are written is definitely the weak point of the entire process.

References

1. *The muSIMP/muMATH-79 Reference Manual* (The Soft Warehouse, Honolulu, Haw., Mar. 1980).
2. G. Williams, The muSIMP/muMATH-79 Symbolic Math System, Byte **5**, 328 (1980).
3. M. Schindler, *Math Package for μCs does Algebra, Calculus*, Electronic Design, 34 (7 June 1980).
4. W. A. Shurcliff, *Polarized Light* (Harvard U. P., Cambridge, 1966), p. 109.
5. E. Hecht and A. Zajac, *Optics* (Addison-Wesley, New York, 1974), p. 268.
6. muMATH-79 for CP/M operating systems as well as the TRS-80 can be purchased from Microsoft, 400 108th Ave., N.E., Bellevue, Wash. 98004. (CP/M is a trade mark of Digital Research, and TRS-80 is a trade mark of Radio Shack, a Tandy Corp.)

UNITED STATES CONGRESS · OFFICE OF TECHNOLOGY ASSESSMENT

PUBLICATION BRIEF

Patent-Term Extension

The Office of Technology Assessment (OTA) is an advisory arm of the U.S. Congress whose basic function is to help legislators anticipate and plan for the positive and negative impacts of technological changes. Address: OTA, U.S. Congress, Washington, D.C. 20510. Phone: 202/224-0885. (OTA offices are located at 600 Pennsylvania Ave., S.E.) John H. Gibbons, *Director*.

Proposals to extend patent terms for products subject to premarketing regulations would, if implemented, provide additional incentives for conducting pharmaceutical research and development. But evidence is insufficient to determine whether these incentives by themselves would appreciably increase pharmaceutical innovation.

Patents were intended to promote innovation by providing inventors with the right to exclude others from making, using, or selling a patented invention. Because drug developers usually obtain patents before their drugs have been approved by the Food and Drug Administration, the length of the approval process can directly affect the length of time during the patent term that a new pharmaceutical is marketed (the effective patent term).

Drug developers believe that pharmaceutical research is becoming less profitable as a result of shorter effective patent terms, governmental actions encouraging competition from drugs generically equivalent to drugs with expired patents, and higher costs of research.

To date, the profits of the pharmaceutical industry have remained high, revenues have increased steadily, and R&D expenditures have increased to levels which more than compensate for the inflation in biomedical research costs. However, the effects of the decline in effective patent terms and the increased competition resulting from Government actions may not have been fully felt.

Patent-term extension has numerous implications for society, industry, and innovation. The extension would increase the attractiveness of research on drugs for large markets; it would not increase the economic attractiveness of research on drugs for small markets.

Drugs with extended patent terms would generate additional revenues when the majority of the proposed extensions are to begin in the 1990's. The long-term stability of the relationship between R&D expenditures and revenues suggests that increases in research activities would not occur until that time and that 8 or 9 percent of the additional revenues generated would be spent on R&D activities. Industry spokesmen maintain that increased R&D expenditures could be expected sooner because firms would make their research decisions on the basis of anticipated increases in revenues.

As a result of patent-term extension, the prices of drugs whose patents are extended would be higher during the extended period than they would have been without the extension. Consumers would, however, benefit if more and better pharmaceuticals were developed. It is expected that both the benefits and the additional costs would affect the elderly and the chronically ill more than other segments of society.

Patent-term extension would delay and in some cases prevent the entry of firms primarily selling drugs that are generically equivalent to drugs with expired patents. The revenues of these firms are determined by the remaining market value of drugs with expired patents—and because of reduced marketing time, the remaining market values would be reduced.

Copies of the OTA report, "Patent-Term Extension and the Pharmaceutical Industry" are available from the U.S. Government Printing Office. The GPO stock number is 052-003-00842-4; the price is $4.25. Copies of the full report **for congressional use** are available by calling 4-8996.