A BASIC Program to Calculate the Evolution of Cartesian Product Operators

Stefano Mammi

Biopolymer Research Center, National Research Council
Via Marzolo 1, 35131 Padova, Italy

1 Introduction

The introduction of the product operator formalism [1] has greatly improved the description of multiple-pulse NMR experiments allowing the understanding of the fate of the magnetization in a direct manner. The use of this formalism has become widespread [2] because of its advantages over both the complete density matrix approach and the method of vector diagrams. It is much simpler than the former and it permits a clear description of the results while remaining rigorous. With respect to the latter, it allows one to visualize all the states of the magnetization and to follow their evolution over very complicated pulse sequences.

The rules that govern the evolution of product operators are very simple and lend themselves to automation by means of computer programs. Automation is especially desirable considering that the description of any two-dimensional experiment leads quickly to very long expressions which can be affected by trivial mistakes.

Recently, computer programs that perform such calculations have been reported in the literature. Among these are the program by Nakashima and McClung [3] and the more recent one by Shriver [4]. The first was written in FORTRAN 77 and describes the evolution of product operators in the spherical basis [3]. While this approach is extremely useful in following coherence pathways and thus deriving phase cycling schemes, many times the use of the cartesian basis is preferable as in the development of new pulse sequences.

The second program [4] was written in the new computer language *Mathematica* whose major advantage is the easy simplification of algebraic expressions. This program was written for MacIntosh systems and is not yet available for IBM personal computers. The input seems to be stepwise and rather cumbersome.

The program presented here runs within MS-DOS and describes the evolution of cartesian product operators. The program, named "EVOLVE", was written using the QuickBasic (C) 4.50 language. The input is from a file which contains all the information pertaining to the spin system and the pulse sequence, and the output is filed separately.

2 Features of the Program

A sample input file is presented in Fig. 1. In this example, an HMQC experiment [5] is applied to a system composed of an X nucleus and two protons, only one of which is coupled to the heteronucleus.

The first information in the input file is the spin system which can be made of up to four spins, denoted by capital letters. The initial magnetization is entered next.

```
### Spin System
HHX
### Initial Operators [Axis(Sp#)Axis(Sp#)....: coeff]
### Terms in sine or cosine must have exactly 6 characters in parenthesis
z(1): +1
z(2): +1
### Coupling Constants (Spin 1,Spin 2,J)
2,3,90
### Sequence (1 Line for Name, N Lines for Pulse Sequence)
HMQC - 2 spins + 1 spin
P(90)H PH1
D1
P(90)X PH2
D0
P(180)H PH3
D0
P(90)X PH4
D1
AQ PH5 DEC(X)
### Phase Cycles (as Bruker, separated by commas)
PH1,0
PH2,0,1,2,3
PH3,0
PH4,0
PH5,0,3,2,1
### Delays: D#=Num/Den*J(Sp#,Sp#) (#, Numerator, Denominator, Spin 1, Spin 2)
1,1,2,2,3
### Do you want to skip the Evolution under Chemical Shift?
N
### Print out only the Observable Operators?
Y
```

Figure 1. Sample input file for an HMQC sequence applied to a (H + HX) spin system.

In listing the operators, the spins are numbered from one to four to prevent ambiguities among like spins. Any number of operators can be listed with appropriate coefficients as required. It is not necessary to start with equilibrium magnetization: for example, the magnetization of any spin can be neglected. Moreover, it is possible to follow the evolution of a specific operator, *e.g.*, **2x(2)z(3)**, generated after the first **D1** in the sequence of Fig 1, by restricting the initial input to just that operator with its own coefficients, *e.g.*, **+cos(Ω 2*D1)**, utilizing an appropriately shortened pulse sequence. Next, the scalar coupling network is described in terms of the spins which are coupled and the relevant coupling constant.

The format for the pulse sequence is very similar to the Bruker one. This allows for simple transcription of sequences already in use and for easy modification of any part of the phase cycling scheme. Each pulse is written as a "**P**" followed by the flip angle in parenthesis, by the spin(s) to which it is applied and by the phase cycle to be used. Only 90° and 180° pulses are currently accepted. Each delay is written as a "**D**" followed by a single digit. The acquisition is referred to as "**AQ**" followed by its own phase cycle. Up to ten different phase cycles can be utilized each containing up to 128 steps. Each step is recorded as a number according to the usual notation: $0 \equiv +x$; $1 \equiv +y$; $2 \equiv -x$; $3 \equiv -y$.

Many sequences require decoupling during specific delays, including the acquisition.

```
z(2): +1
z(1): +1
   -- 90 (H)+x -->
y(2): -1
y(1): -1
   -- D1 -->
2x(2)z(3): +cos(Ω 2*D1)
2y(2)z(3): +sin(Ω 2*D1)
x(1): +sin(Ω 1*D1)
y(1): -cos(Ω 1*D1)
   -- 90 (X)+x -->
2x(2)y(3): -cos(Ω 2*D1)
2y(2)y(3): -sin(Ω 2*D1)
x(1): +sin(Ω 1*D1)
y(1): -cos(Ω 1*D1)
   -- D0 -->
2x(2)x(3): +cos(Ω 2*D0)cos(Ω 2*D1)sin(Ω 3*D0)-sin(Ω 2*D0)sin(Ω 2*D1)sin(Ω 3*D0)
2x(2)y(3): +sin(Ω 2*D0)sin(Ω 2*D1)cos(Ω 3*D0)-cos(Ω 2*D0)cos(Ω 2*D1)cos(Ω 3*D0)
2y(2)x(3): +sin(Ω 2*D0)cos(Ω 2*D1)sin(Ω 3*D0)+cos(Ω 2*D0)sin(Ω 2*D1)sin(Ω 3*D0)
2y(2)y(3): -sin(Ω 2*D0)cos(Ω 2*D1)cos(Ω 3*D0)-cos(Ω 2*D0)sin(Ω 2*D1)cos(Ω 3*D0)
x(1): +cos(Ω 1*D0)sin(Ω 1*D1)+sin(Ω 1*D0)cos(Ω 1*D1)
y(1): +sin(Ω 1*D0)sin(Ω 1*D1)-cos(Ω 1*D0)cos(Ω 1*D1)
   -- 180 (H)+x -->
2x(2)x(3): +cos(Ω 2*D0)cos(Ω 2*D1)sin(Ω 3*D0)-sin(Ω 2*D0)sin(Ω 2*D1)sin(Ω 3*D0)
2x(2)y(3): +sin(Ω 2*D0)sin(Ω 2*D1)cos(Ω 3*D0)-cos(Ω 2*D0)cos(Ω 2*D1)cos(Ω 3*D0)
2y(2)x(3): -sin(Ω 2*D0)cos(Ω 2*D1)sin(Ω 3*D0)-cos(Ω 2*D0)sin(Ω 2*D1)sin(Ω 3*D0)
2y(2)y(3): +sin(Ω 2*D0)cos(Ω 2*D1)cos(Ω 3*D0)+cos(Ω 2*D0)sin(Ω 2*D1)cos(Ω 3*D0)
x(1): +cos(Ω 1*D0)sin(Ω 1*D1)+sin(Ω 1*D0)cos(Ω 1*D1)
y(1): -sin(Ω 1*D0)sin(Ω 1*D1)+cos(Ω 1*D0)cos(Ω 1*D1)
   -- D0 -->
2x(2)x(3): +cos(Ω 2*D1)cos(Ω 3*D0)sin(Ω 3*D0)+cos(Ω 2*D1)cos(Ω 3*D0)sin(Ω 3*D0)
2x(2)y(3): +cos(Ω 2*D1)sin(Ω 3*D0)sin(Ω 3*D0)-cos(Ω 2*D1)cos(Ω 3*D0)cos(Ω 3*D0)
2y(2)x(3): -sin(Ω 2*D1)cos(Ω 3*D0)sin(Ω 3*D0)-sin(Ω 2*D1)cos(Ω 3*D0)sin(Ω 3*D0)
2y(2)y(3): +sin(Ω 2*D1)cos(Ω 3*D0)cos(Ω 3*D0)-sin(Ω 2*D1)sin(Ω 3*D0)sin(Ω 3*D0)
x(1): +sin(Ω 1*D1)
y(1): +cos(Ω 1*D1)
   -- 90 (X)+x -->
2x(2)x(3): +cos(Ω 2*D1)cos(Ω 3*D0)sin(Ω 3*D0)+cos(Ω 2*D1)cos(Ω 3*D0)sin(Ω 3*D0)
2x(2)z(3): +cos(Ω 2*D1)sin(Ω 3*D0)sin(Ω 3*D0)-cos(Ω 2*D1)cos(Ω 3*D0)cos(Ω 3*D0)
2y(2)x(3): -sin(Ω 2*D1)cos(Ω 3*D0)sin(Ω 3*D0)-sin(Ω 2*D1)cos(Ω 3*D0)sin(Ω 3*D0)
2y(2)z(3): +sin(Ω 2*D1)cos(Ω 3*D0)cos(Ω 3*D0)-sin(Ω 2*D1)sin(Ω 3*D0)sin(Ω 3*D0)
x(1): +sin(Ω 1*D1)
y(1): +cos(Ω 1*D1)
   -- D1 -->
2x(2)x(3): +cos(Ω 3*D0)sin(Ω 3*D0)cos(Ω 3*D1)+cos(Ω 3*D0)sin(Ω 3*D0)cos(Ω 3*D1)
2x(2)y(3): +cos(Ω 3*D0)sin(Ω 3*D0)sin(Ω 3*D1)+cos(Ω 3*D0)sin(Ω 3*D0)sin(Ω 3*D1)
y(2): +sin(Ω 3*D0)sin(Ω 3*D0)-cos(Ω 3*D0)cos(Ω 3*D0)
y(1): +1
   -- AQ(+x) -->
x(2): +sin(Ω 2*AQ)cos(Ω 3*D0)cos(Ω 3*D0)-sin(Ω 2*AQ)sin(Ω 3*D0)sin(Ω 3*D0)  +R
y(2): +cos(Ω 2*AQ)sin(Ω 3*D0)sin(Ω 3*D0)-cos(Ω 2*AQ)cos(Ω 3*D0)cos(Ω 3*D0)  +I
x(1): -sin(Ω 1*AQ)  +R
y(1): +cos(Ω 1*AQ)  +I

        ########################################

SIGNAL - Real Part:
+2sin(Ω 2*AQ)cos(Ω 3*D0)cos(Ω 3*D0)
-2sin(Ω 2*AQ)sin(Ω 3*D0)sin(Ω 3*D0)
-4cos(Ω 2*AQ)cos(Ω 3*D0)sin(Ω 3*D0)

SIGNAL - Imaginary Part:
+2cos(Ω 2*AQ)sin(Ω 3*D0)sin(Ω 3*D0)
-2cos(Ω 2*AQ)cos(Ω 3*D0)cos(Ω 3*D0)
-4sin(Ω 2*AQ)cos(Ω 3*D0)sin(Ω 3*D0)
```

Figure 2. Portion of the output obtained with the input from Fig. 1. The result from the first step of the phase cycle and the final signal after the full phase cycle are shown.

EVOLVE will "decouple" any spin if an appropriate statement is added on the same line of any delay.

EVOLVE was written specifically for sequences in which some delays are inversely proportional to certain scalar coupling constants as in heteronuclear correlation experiments. Proper space is provided in the input file for defining such cases.

The user is then asked to indicate if evolution under both coupling and chemical shift should be taken into account or if the latter should be neglected. Finally, the user chooses whether all the resulting operators or just the observable ones should be printed out after the acquisition step.

The program runs through the pulse sequence as many times as required by the phase cycle. If only the observables are chosen as output, the final intensities of the signal are written in two separate files, one for the real part and one for the imaginary part. At the end of the phase cycle, these files are read and final simplifications are carried out.

In Fig. 2, a portion of the output obtained with the input file of Fig. 1 is reported, *i.e.*, the result from the first of the four steps of the phase cycle and the signal obtained at the end of the four step cycle. It can be seen that the term **y(2)** generated by the first 90° pulse gives rise only to antiphase terms at the end of the first **D1**, because $D1 = 1/2J_{2,3}$.

The signal from the proton not coupled to the X-nucleus is present in the first acquisition, but is canceled out at the end of the four step cycle while the single quantum terms from the proton coupled to the X-nucleus add up to generate the final signal.

The BASIC language does not have built-in routines for the simplification of algebraic expressions. A sizable portion of the program is devoted to such routines. Beside the more trivial expressions containing $\pi/2$, EVOLVE is able to handle terms containing $sin(\pi/4)$ or $cos(\pi/4)$, most commonly encountered in heteronuclear sequences. The program does not evaluate these expressions; rather, it simplifies them according to the rules it knows. This entails a longer computation time but with the advantage of eliminating all numerical coefficients. This compromise was found satisfactory.

The only terms that would be useful to have simplified and that the program is unable to handle at this stage are those including $sin(2\theta)$ or $cos(2\theta)$ terms, encountered for example when there is a 180° pulse in the middle of a delay. This is not a serious limitation of the BASIC language, especially considering that even in *Mathematica* this simplification must be explicitly requested by the user.

3 Conclusions

EVOLVE has been applied to numerous complicated pulse sequences avoiding lengthy and monotonous calculations and providing the results in a way suitable for straightforward analysis.

A copy of the program, including the source code, can be obtained by sending a 5.25" or 3.5" diskette and return postage to the author.

4 References

[1] O. W. Sørensen, G. W. Eich, M. H. Levitt, G. Bodenhausen, and R. R. Ernst, *Prog. NMR Spectrosc.* **16**, 163 (1983).

[2] See for example H. Kessler, M. Gehrke, and C. Griesinger, *Angew. Chem. Int. Ed. Engl.* **27**, 490 (1988).

[3] T. T. Nakashima and R. E. D. McClung, *J. Magn. Reson.* **70**, 187 (1986).

[4] J. W. Shriver, *J. Magn. Reson.* **94**, 612 (1991).

[5] A. Bax, R. H. Griffey, and B. L. Hawkins *J. Magn. Reson.* **55**, 301 (1983).