

The Simulation and Optimization of NMR Experiments using a Liouville Space Method

Christopher Kumar Anand, Alex D. Bain, Zhenghua Nie
Department of Computing and Software, Department of Chemistry
McMaster University
1280 Main St. West
Hamilton, ON L8S 4M1, Canada
anandc@mcmaster.ca

May 3, 2006

Abstract

We simulate, using symbolic computation in Maple, the dynamics of coupled two-spin systems without relaxation, including the basic elements of Nuclear Magnetic Resonance (NMR) pulse sequences, as used in experimental Chemistry. This extends previous work by other authors on single-spin and weakly-coupled systems. This technique extends directly to n -spin systems, including relaxation, but our aim in this paper is to demonstrate the feasibility of doing these calculations symbolically, and validate our implementation against existing analytic solutions, which are for special cases of two-spin systems. For the purposes of illustration, we simulate a two-parameter family of spin-echo experiments, for a representative spin system, and show graphically that this simulation can be visually optimized. This lays the groundwork for the optimization of more complicated pulse sequences, with many parameters.

1 Introduction

Magnetic resonance is widely used in experimental, clinical and industrial applications, including medical diagnosis, experimental and industrial chemistry, and petroleum exploration. Magnetic Resonance Imaging (MRI) and chemical Nuclear Magnetic Resonance (NMR) are based on the same physical models and this work applies equally to both, although multi-spin systems are not yet part of standard clinical practice. Magnetic resonance arises from the interaction of spins (hydrogen nuclei in the present case) with magnetic fields and radio frequency radiation. MRI which gives diagnostic information of various organs detects signal from protons in water of body tissues so as to obtain medical images of slices or volumes from inside a body. Water, by far the most common

molecule in the body, is a single signal, with a relatively easy to analyze response to imaging pulse sequence. Quantum mechanics is not needed to understand the water signal. In contrast, in NMR and MR Spectroscopy, most chemical compounds have many different protons with detectable signals, some of whose spins are coupled. Coupling of the protons complicates the response, but it can be calculated analytically, using tools such as Maple. In the body, although the effects on the image from other hydrogen nuclei which have coupled spins are small, information about their concentration and distribution is valuable diagnostically [5, 6, 9, 10]. These spins include hydrogen in creatine or n-acetyl aspartate (NAA). This is our original motivation for doing this research: to use Maple software and optimization methods to design sequences which enhance signals from biologically-significant molecules in the brain or body.

Maple and other computer algebra methods have been applied to NMR before [11-29]. An important part of this calculation is the use of angular momentum methods, which are facilitated with Maple [18-25]. This previous work mostly involves simulation of weakly-coupled spins, but little work has yet been done to use them for optimizing NMR experiments, or simulate strong-coupling. We will show how to construct the linear systems which govern the evolution of coupled spin systems in different stages of the experiment. Uncoupled systems correspond to simultaneously block-diagonal systems. Weakly-coupled systems have numerically-small off-diagonal blocks, and strongly-coupled systems require numerically-large off-diagonal elements to reproduce empirically-observed behaviour. We will show how to use Maple to simulate the more complicated strongly-coupled systems.

In quantum mechanics, the density matrix gives all possible information about a system. Operator formalisms are methods to act on the density matrix so as to give reasonable solutions to show the effect of a pulse and a delay in NMR [6]. We use the Liouville space method for pulse NMR as the theory to directly work on the density matrix [1, 2, 3, 4], implementing the linear algebra symbolically in Maple.

In NMR, each spin has four observables $|1_0\rangle$, $|1_{+1}\rangle$, $|1_{-1}\rangle$ and $|0\rangle$. The behaviour of a single spin is straightforward, but two coupled spins offer significant challenges. Two coupled spins have 16 elements, written here in direct product form:

$$\text{Basis} = \begin{bmatrix} |0\rangle|0\rangle \\ |0\rangle|1_{+1}\rangle \\ |0\rangle|1_0\rangle \\ |0\rangle|1_{-1}\rangle \\ |1_{+1}\rangle|0\rangle \\ |1_{+1}\rangle|1_{+1}\rangle \\ |1_{+1}\rangle|1_0\rangle \\ |1_{+1}\rangle|1_{-1}\rangle \\ |1_0\rangle|0\rangle \\ |1_0\rangle|1_{+1}\rangle \\ |1_0\rangle|1_0\rangle \\ |1_0\rangle|1_{-1}\rangle \\ |1_{-1}\rangle|0\rangle \\ |1_{-1}\rangle|1_{+1}\rangle \\ |1_{-1}\rangle|1_0\rangle \\ |1_{-1}\rangle|1_{-1}\rangle \end{bmatrix} \quad (1)$$

A spin system is represented by a 16-element vector. Our manipulation of the spin system is represented by matrices acting on this vector. A pulse sequence of NMR is a series of short radio-frequency pulses interleaved with delays. Behaviour in a delay is defined by a matrix called a Liouvillian which encodes information about chemical structure, frequencies, and couplings. A pulse is equivalent to a rotation of the reference frame. A rotation is defined by a special matrix called a Wigner rotation matrix and is easy to set up for one spin. The direct product provides the rotation for multiple spins.

In the Maple implementation, the Liouvillian is expressed in a different basis in which it is block diagonal. Then we do a permutation which brings it into the basis we use. The actual evolution is defined by the exponential of the Liouvillian matrix, \exp^{iLt} . There are many ways of calculating the exponential of a matrix; Maple provides a function to calculate the exponential of a matrix, but it does not give the answer when it works on our Liouvillian matrix. Specifically, the Maple command $\langle \text{MatrixExponential}(L, I^*t) \rangle$ or $\langle \text{MatrixExponential}(I^*L^*t) \rangle$, was left to run half of a day on a PowerMac G5, and didn't complete the computation before we interrupted it. Whereas, explicitly calculating the eigenvalues, eigenvectors and forming the diagonalization executes almost instantaneously. We use the method based on diagonalization and a priori knowledge of eigenvalues and eigenvectors of the Liouvillian matrix

[8].

In Liouville space, we detect observables by taking the dot (scalar) product with the density matrix. A scalar product between two Liouville space vectors is defined as the trace of their product as operators in spin space. This product gives the projection of one vector on the other vector [3].

$$(P|Q) = \text{trace}(PQ) \quad (2)$$

The main steps of the Liouville space method are the following: (1) to define the basis of Liouville space; (2) to construct the Liouvillian matrix; (3) to calculate the equation of the motion; (4) to design NMR experiments such as series of pulses (Wigner rotation matrices) and delays (the equation of the motion); (5) to detect the NMR signal by means of the scalar product.

The main tasks of the present Maple application, which is a specific application of the Liouville space method to spin echo, are: (1) to calculate the exact formula of NMR signal for spin echo (the first pulse is $\frac{\pi}{2}$, the second pulse is π) on a two-spin system; (2) to calculate the general formula of NMR signal for spin echo (the first pulse is $\frac{\pi}{2}$, the second pulse is α), so α will be a variable in the spin-evolution; and then (3) to give an objective function for an optimization problem.

2 Initialization

In this stage, we set up a Liouvillian matrix for an AB (two-spin) system. J is the coupling constant and ω_A and ω_B are their two Larmor frequencies. A Liouvillian matrix, \mathbf{L} , is divided into several blocks in the coherence order basis. Usually, the Liouvillian matrix is built in the coherence order basis and then immediately permuted into the direct product order basis so to keep consistent with other vectors and matrices.

```
> restart:
> with (LinearAlgebra):
> assume(omega[A]>0, omega[B]>0, theta>0, theta<Pi/4, omega[A]>omega[B],
delta>0, J>0, omega[av]>0, omega[av]>delta);
```

One block is for the single quantum [2]:

```
> sqPos := Matrix(4,4, [[omega[A], J/2, 0, -J/2], [J/2, omega[A], -J/2, 0],
[0, -J/2, omega[B], J/2], [-J/2, 0, J/2, omega[B]]]):
```

Another block is for the other single quantum:

```
> sqNeg :=Matrix(4,4, [[-omega[A], -J/2, 0, J/2], [-J/2, -omega[A], J/2, 0],
[0, J/2, -omega[B], -J/2], [J/2, 0, -J/2, -omega[B]]]):
```

The following code is to construct the Liouvillian matrix from its blocks.

```

> L := < <Matrix(1,1,[[omega[A]+omega[B]])|ZeroMatrix(1,15)>,
  <ZeroMatrix(4,1)|sqPos|ZeroMatrix(4,11)> ,
  <ZeroMatrix(1,5)|Matrix(1,6,[[ -omega[A]+omega[B],0,J/2,-J/2,0,0]])
  |ZeroMatrix(1,5)>,<ZeroMatrix(1,16)>,
  <ZeroMatrix(1,5)|Matrix(1,6,[[J/2,0,0,0,0,-J/2]])|ZeroMatrix(1,5)>,
  <ZeroMatrix(1,5)|Matrix(1,6,[[ -J/2,0,0,0,0,J/2]])|ZeroMatrix(1,5)>,
  <ZeroMatrix(1,16)>,
  <ZeroMatrix(1,5)|Matrix(1,6,[[0,0,-J/2,J/2,0,
  omega[A]-omega[B]])|ZeroMatrix(1,5)>,
  <ZeroMatrix(4,11)|sqNeg|ZeroMatrix(4,1)>,
  <ZeroMatrix(1,15)|Matrix(1,1,[[ -omega[A]-omega[B]])>>:

```

PermutationDirect is a 16×16 matrix which is for the permutation between coherence order basis and direct product order basis.

```

> PermutationDirect := < <ZeroMatrix(1,5)|Matrix(1,1,[1])|ZeroMatrix(1,10)>,
  <Matrix(1,2,[[0,1]])|ZeroMatrix(1,14)>,
  <ZeroMatrix(1,9)|Matrix(1,1,[1])|ZeroMatrix(1,6)>,
  <ZeroMatrix(1,4)|Matrix(1,1,[1])|ZeroMatrix(1,11)>,
  <ZeroMatrix(1,6)|Matrix(1,1,[1])|ZeroMatrix(1,9)>,
  <ZeroMatrix(1,7)|Matrix(1,1,[1])|ZeroMatrix(1,8)>,
  <Matrix(1,1,[1])|ZeroMatrix(1,15)>,
  <ZeroMatrix(1,8)|Matrix(1,1,[1])|ZeroMatrix(1,7)>,
  <ZeroMatrix(1,2)|Matrix(1,1,[1])|ZeroMatrix(1,13)>,
  <ZeroMatrix(1,10)|Matrix(1,1,[1])|ZeroMatrix(1,5)>,
  <ZeroMatrix(1,13)|Matrix(1,1,[1])|ZeroMatrix(1,2)>,
  <ZeroMatrix(1,3)|Matrix(1,1,[1])|ZeroMatrix(1,12)>,
  <ZeroMatrix(1,11)|Matrix(1,1,[1])|ZeroMatrix(1,4)>,
  <ZeroMatrix(1,12)|Matrix(1,1,[1])|ZeroMatrix(1,3)>,
  <ZeroMatrix(1,14)|Matrix(1,1,[1])|ZeroMatrix(1,1)>,
  <ZeroMatrix(1,15)|Matrix(1,1,[1])>>:

```

The Liouvillian matrix (**NL**) is in the direct product order basis. To simplify the formulae, ω_A and ω_B are substituted by ω_{av} and δ .

```

> NL := simplify(subs({omega[A]=omega[av]+delta,
  omega[B]=omega[av]-delta},L)):
> NL := Multiply(MatrixInverse(PermutationDirect),
  Multiply(NL,PermutationDirect)):

```

d1 is the Wigner rotation matrix in the spherical tensor basis for one spin. **WignerAlpha** is a 16×16 matrix for two-spin rotation in the direct product order basis. We will use it to calculate the effect of a pulse on a two-spin system. For a multiple spin system, the rotation also can be represented by multiple direct product.

```

> d1 := Matrix(3,3,[[ (cos(alpha/2))^2, -sin(alpha)/sqrt(2), (sin(alpha/2))^2 ],

```

```

[sin(alpha)/sqrt(2), cos(alpha), -sin(alpha)/sqrt(2)],
[(sin(alpha/2))^2, sin(alpha)/sqrt(2), (cos(alpha/2))^2]]):
> WignerA := <<matrix(1,1,[1]|ZeroMatrix(1,3)>, <ZeroMatrix(3,1)|d1>>;
> DirectProduct := (A,B) -> Matrix([seq([seq(ScalarMultiply(B,A[i,j]),
j=1..ColumnDimension(A))], i=1..RowDimension(A))]):
> WignerAlpha := DirectProduct(WignerA,WignerA):

```

$$WignerA := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & (\cos(1/2\alpha))^2 & -1/2 \sin(\alpha)\sqrt{2} & (\sin(1/2\alpha))^2 \\ 0 & 1/2 \sin(\alpha)\sqrt{2} & \cos(\alpha) & -1/2 \sin(\alpha)\sqrt{2} \\ 0 & (\sin(1/2\alpha))^2 & 1/2 \sin(\alpha)\sqrt{2} & (\cos(1/2\alpha))^2 \end{bmatrix} \quad (3)$$

We have now set up the Liouvillian for delays and the rotation matrix for pulses. We need to define detection of the signal. The vector (**TotalXYVector**) which represents the total magnetization in the xy plane is used to detect the signal.

```
> TotalXYVector := <0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0>:
```

A Maple function gives the eigenvalues and eigenvectors of the Liouvillian matrix symbolically. The eigenvectors of **NL** correspond to density matrix elements and the eigenvalue represents a "resonant frequency" of that density matrix element if the spin space basis consists of eigenfunctions of the Hamiltonian. If the observable is one of the basis elements, its expectation value is just the corresponding density matrix element [3]. Currently, Maple 10 could not give the symbolic eigenvectors of a matrix which includes trigonometric elements. After the eigenvectors are calculated, J is substituted by θ and δ .

```
> (V,E) := Eigenvectors(NL):
> E := simplify(subs({J=(sin(2*theta)/cos(2*theta))*2*delta},E)):
```

In order to get the exponential of a matrix, we use the eigen decomposition method. The eigenvectors which are returned by Maple are not normalized, so the required inverse is given by a Maple function. If the eigenvectors are normalized, the inverse can be directly calculated by a transpose.

```
> ExpV := <seq(exp(I*V[i]*t), i=1..Dimension(V))>:
> ExpVdiagonal := Matrix(1..16,1..16,ExpV,shape=diagonal):
> ExpL := MatrixMatrixMultiply(E,MatrixMatrixMultiply(ExpVdiagonal,
simplify(MatrixInverse(E))))):
```

The spectrum of the two-spin system consists of four lines, so only four quantities are observed where they are in the density matrix. **IndexofSQ** stores the index of the eigenvalues vector **V** in which the value is the frequency of the signals. Every time the eigenvectors are calculated, the index needs to be set up again.

```

> GetSQIndex := proc(V)
  local i,j,SQIndex;
  j := 1;
  i := 1;

  while j < 5 do
    if V[i] = omega[av]+1/2*J+1/2*sqrt(J^2+4*delta^2)
  then SQIndex[j] := i; j := j + 1; end if;
    if V[i] = omega[av]+1/2*J-1/2*sqrt(J^2+4*delta^2) then
  SQIndex[j] := i; j := j + 1; end if;
    if V[i] = omega[av]-1/2*J+1/2*sqrt(J^2+4*delta^2) then
  SQIndex[j] := i; j := j + 1; end if;
    if V[i] = omega[av]-1/2*J-1/2*sqrt(J^2+4*delta^2) then
  SQIndex[j] := i; j := j + 1; end if;
    i := i + 1;
  end do;
  [SQIndex[1],SQIndex[2],SQIndex[3],SQIndex[4]]
end proc:
> IndexofSQ := GetSQIndex(V):
> Frequency := V[IndexofSQ];

```

SQEigenvectorsNorm stores the normalized eigenvectors which are associated with the eigenvalues. These eigenvectors represent the transitions.

```

> SQEigenvectors := E[1..16,IndexofSQ]:
> SQEigenvectorsNorm := [seq(SQEigenvectors[1..16,i]
/VectorNorm(SQEigenvectors[1..16,i],2,conjugate=false),
i=1..ColumnDimension(SQEigenvectors))]:

```

$$frequency = \begin{bmatrix} 1/2 J + \omega_{av} + 1/2 \sqrt{J^2 + 4 \delta^2} \\ 1/2 J + \omega_{av} - 1/2 \sqrt{J^2 + 4 \delta^2} \\ -1/2 J + \omega_{av} + 1/2 \sqrt{J^2 + 4 \delta^2} \\ -1/2 J + \omega_{av} - 1/2 \sqrt{J^2 + 4 \delta^2} \end{bmatrix} \quad (4)$$

3 Spin Echo of a Coupled System

We will implement the calculation of the NMR signal of spin echo. This is a series of pulse, delay, and detection. Since each of these manipulations is a matrix operation, we only need the multiply function between matrices and vectors. The main steps are:

1. setting up the equilibrium state of the system;
2. applying the first pulse. In general, Wigner matrix is for any α ; α can be a variable. In this case, the first pulse is $\frac{\pi}{2}$;

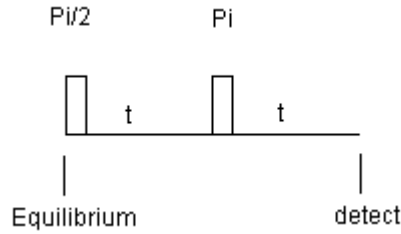


Figure 1: The spin echo NMR experiment

3. evolving during a delay τ ;
4. applying the second pulse. (Normally, $\alpha = \pi$). For optimization problems, we allow the flip angle of the second pulse, α , to vary. In this present case, α is equal to π ;
5. evolving during a delay τ ;
6. detecting.

EquVector represents the equilibrium state, in which most experiments begin:

```
EquVector := <0,0,1,0,0,0,0,0,1,0,0,0,0,0,0>:
```

The first pulse, which is a $\frac{\pi}{2}$ pulse, acts on the system:

```
RotateEqu1 :=  
MatrixVectorMultiply(eval(WignerAlpha,[alpha=Pi/2]),EquVector):
```

The first delay:

```
Evaluation1 :=  
simplify(simplify(simplify(MatrixVectorMultiply(ExpL,RotateEqu1)))):
```

The second pulse (π) acts on the system:

```
> WignerAlpha2 := eval(WignerAlpha,[alpha=Pi]):  
> RotateEqu2 := MatrixVectorMultiply(WignerAlpha2,Evaluation1):
```

The second delay:

```
Evaluation2 := MatrixVectorMultiply(ExpL,RotateEqu2):
```

Detection:

The vector **Intensity** and **IntensityOriginal** will return the intensity of the NMR signal with respect to the frequency vector **Frequency**.

```

> Evaluation := simplify(eval(Evaluation2,[t=tau])):
> scalar1 := [seq(Multiply(Transpose(Evaluation),SQEigenvectorsNorm[i]),
i=1..ColumnDimension(SQEigenvectors))]:
> scalar2 := [seq(Multiply(Transpose(SQEigenvectorsNorm[i]),TotalXYVector),
i=1..ColumnDimension(SQEigenvectors))]:
> IntensityOriginal := <seq(simplify(scalar1[i]*scalar2[i]),
i=1..ColumnDimension(SQEigenvectors))>:
> Intensity := factor(subs({cos(2*theta)^2=1-sin(2*theta)^2},
IntensityOriginal)):

```

Display one of the intensities.

```
> Intensity[1];
```

$$\frac{1}{4} \left(-e^{i\tau(J+\sqrt{J^2+4\delta^2})} \left(1 - (\sin(2\theta))^2\right) + e^{i\tau J} \left(1 - (\sin(2\theta))^2\right) - e^{i\tau(J+\sqrt{J^2+4\delta^2})} \sin(2\theta) + e^{i\tau(J+\sqrt{J^2+4\delta^2})} \right) \sqrt{2}$$

The above result is the same as [4].

For the weak coupling limit $\theta \rightarrow 0$, we observe a simple phase modulation at frequency J [4]. This result is also the same as [4].

```

> WeakIntensity := simplify(eval(IntensityOriginal,[theta=0])):
> 'WeakIntensity[1]' = WeakIntensity[1];
> plot(Re(eval(WeakIntensity[1],[J=8])),tau=0..3);

```

$$\text{WeakIntensity} = \frac{1}{4} e^{i\tau J} \sqrt{2} \quad (5)$$

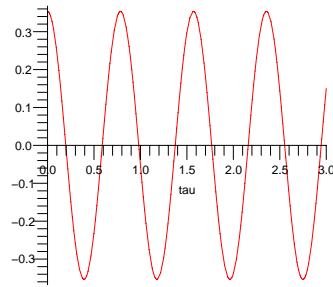


Figure 2: a weak coupling system

For a strong AB system, i.e. $\tan(2\theta) \approx 1$, the behaviour is more complex, but still matches [4].

```

> StrongIntensity := simplify(eval(IntensityOriginal,[theta=Pi/8])):
> 'StrongIntensity[1]' = StrongIntensity[1];
> plot(Re(eval(StrongIntensity[1],[J=8,delta=4])),tau=0..3);

```

$$StrongIntensity = -1/8 \left(-e^{i\tau(J+\sqrt{J^2+4\delta^2})} - e^{i\tau J} + e^{i\tau(J+\sqrt{J^2+4\delta^2})}\sqrt{2} \right) \sqrt{2} \quad (6)$$

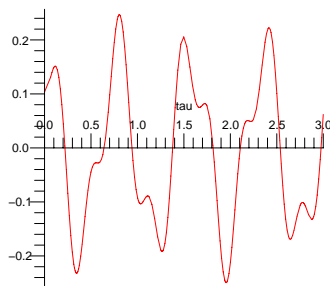


Figure 3: a strong coupling system

At this stage, the difference between these plots illustrates the complications that strong coupling introduces. Maple provides a reliable way of dealing with these complications.

4 Optimization Spin Echo Detection

We choose the sum of the squares of the real parts of the spectrum as our objective function. This will give us the strongest spectrum. This is a simple function and we optimize it visually. The capabilities of Maple will support much more complicated objective functions corresponding to longer pulse trains and multiple chemical species in the future, but visualization of two-dimensional subspaces will still be important in understanding the design spaces.

In order to optimize the pulse sequence, we need to control the second pulse and the delay time, which gives two variables: α and τ . Maple gives the general formulae with α and τ . Maple can differentiate the objective functions we give, which is important for rapid convergence, but for the moment we simply plot the objective function. In order to simulate the variation in magnetic field present in real experiments, we average ω_{av} over a Gaussian spread of values. The steps to calculate the formulae are the same as before.

EquVector represents the equilibrium state:

```

> EquVector := <0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0>:

```

The first pulse ($\frac{\pi}{2}$) acts on the system:

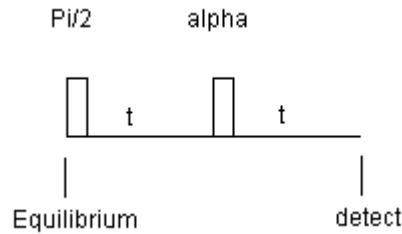


Figure 4: The spin echo NMR experiment, the flip angle of the second pulse is a variable.

```
> RotateEqu1 :=
MatrixVectorMultiply(eval(WignerAlpha, [alpha=Pi/2]), EquVector):
```

The first delay:

```
> Evaluation1 :=
simplify(simplify(simplify(MatrixVectorMultiply(ExpL, RotateEqu1))))):
```

The second pulse (α) acts on the system, we leave α as a variable:

```
> WignerAlpha2 := eval(WignerAlpha, [alpha=alpha]):
> RotateEqu2 := MatrixVectorMultiply(WignerAlpha2, Evaluation1):
```

The second delay:

```
> Evaluation2 := MatrixVectorMultiply(ExpL, RotateEqu2):
```

Detection:

The vector **IntensityOriginal** will return the intensity of the NMR signal with respect to the frequency vector Frequency.

```
> Evaluation := simplify(eval(Evaluation2, [t=tau])):
> scalar1 := [seq(Multiply(Transpose(Evaluation), SQEigenVectorsNorm[i]),
i=1..ColumnDimension(SQEigenVectors))]:
> scalar2 := [seq(Multiply(Transpose(SQEigenVectorsNorm[i]), TotalXYVector),
i=1..ColumnDimension(SQEigenVectors))]:
> IntensityOriginal := <seq(simplify(scalar1[i]*scalar2[i]),
i=1..ColumnDimension(SQEigenVectors))>:
> Intensity := factor(subs({cos(2*theta)^2=1-sin(2*theta)^2},
IntensityOriginal)):
```

We now have an expression for a spin echo using a general value of α . This would have been prohibitive with hand calculation, but Maple handles it easily. In order to optimize the system, we need an objective function.

The objective function we choose is **ObjFunc** which has two variables α and τ (without a Gaussian distribution).

```

> SumofRealIntensity := (Re(IntensityOriginal[1]))^2+
  (Re(IntensityOriginal[2]))^2 +(Re(IntensityOriginal[3]))^2
  +(Re(IntensityOriginal[4]))^2:
> ObjFunc := evalf(eval(SumofRealIntensity, [J=8,delta=4,omega[av]=0,
  theta= arctan(1/1)/2])):
> plot3d(ObjFunc,tau=0..1,alpha=0..2*Pi,axes=normal);

```

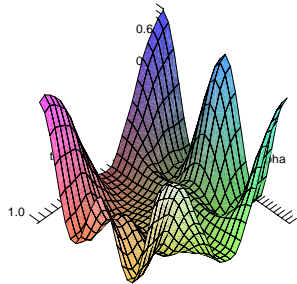


Figure 5: The objective function without a Gaussian distribution

The following code simulates field variation by integrating ω_{av} taken from a Gaussian distribution.

Case 1: $\omega_{av} = 0$, $J = 8hz$, $2\delta = 20J$;

```

> GaussianSignal := <seq(int(eval(IntensityOriginal[i]
  *exp(-omega[av]^2/(2*(2)^2))/(2*(sqrt(2*Pi))),
  [J=8,delta=10*8,theta=arctan(1/(20))/2]),
  omega[av]=-infinity..infinity), i=1..Dimension(IntensityOriginal))>:
> ObjFunc := (Re(GaussianSignal[1]))^2+(Re(GaussianSignal[2]))^2
  +(Re(GaussianSignal[3]))^2+(Re(GaussianSignal[4]))^2:
> plot3d(ObjFunc,tau=0..1,alpha=0..2*Pi,axes=normal);

```

Case 2: $\omega_{av} = 0$, $J = 8hz$, $2\delta = J$;

```

> GaussianSignal := <seq(int(eval(IntensityOriginal[i]
  *exp(-omega[av]^2/(2*(2)^2))/(2*(sqrt(2*Pi))),
  [J=8,delta=4,theta=arctan(8/(2*4))/2]),
  omega[av]=-infinity..infinity), i=1..Dimension(IntensityOriginal))>:
> ObjFunc := (Re(GaussianSignal[1]))^2+(Re(GaussianSignal[2]))^2
  +(Re(GaussianSignal[3]))^2+(Re(GaussianSignal[4]))^2:
> plot3d(ObjFunc,tau=0..1,alpha=0..2*Pi,axes=normal);

```

With several thousand basic operations, an expression for the derivatives of this objective are clearly beyond our ability to compute by hand, but simple enough for Maple or Maple Code Generation.

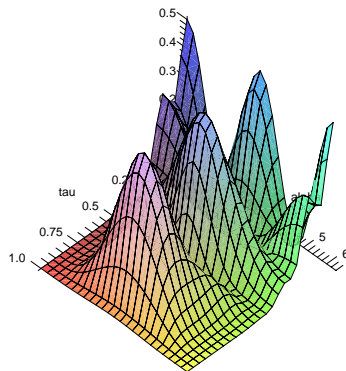


Figure 6: Case 1: The objective function with a Gaussian distribution

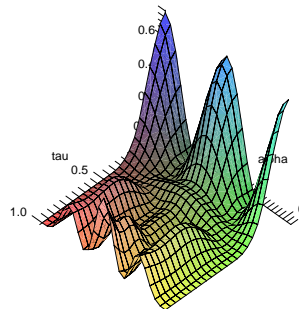


Figure 7: Case 2: The objective function with a Gaussian distribution

```
> codegen[cost](diff(ObjFunc,alpha));
      2405 additions + 10502 multiplications + 5736 functions + 12 divisions
```

5 Conclusion

Maple provides a good way to simulate NMR of strongly coupled nuclear spins, both symbolically and numerically. Using Maple plots, we can quickly understand the design space for two-dimensional families of pulse sequences and, by projection, for more complicated families. Using symbolic computation, we have access to derivatives, which enables future efficient numerical optimization of pulse sequences. In this application, we have defined tools for pulse, delay, and detection. With these tools, we can easily construct any pulse sequence and then observe the signals. The same method can be applied to code the relaxation and phase cycling, and generalize to n-spins. Finally, it is relatively easy to scale up for more than two spins. The computational complexity grows

quickly in Liouville space, since we need to calculate eigenvalues for a system of size 4^n . Most applications are for small numbers of spins, which we can treat in the same way. For the last few percent which involve more than 5 spins, we may calculate it in the single quantum subspace which complexity is reduced to $\binom{2n}{n-1}$.

References

- [1] Alex D. Bain, *Coherence levels and coherence pathways in NMR. A simple way to design phase cycling procedures*, Journal of Magnetic Resonance (1969), vol. 56, no. 3, pp. 418-427, 1984.
- [2] Alex D. Bain, D. A. Fletcher, and Paul Hazendonk, *What is a transition?*, Concepts in Magnetic Resonance, vol. 10, no. 2, pp.85-98, 1998.
- [3] Alex D. Bain, and John S. Martin, *FT NMR of nonequilibrium states of complex spin systems. I. A Liouville space description*, Journal of Magnetic Resonance (1969), vol. 29, no. 1, pp. 125-135, 1978.
- [4] Alex D. Bain, *Modulation of NMR spin echoes in coupled systems*, Chemical Physics Letters, vol. 57, no. 2, pp. 281-284, 1978.
- [5] Hyeonjin Kim, James M. Wild, and Peter S. Allen. *Strategy for the spectral filtering of Myo-Inositol and other strongly coupled spins*, Magnetic Resonance in Medicine, vol. 51, no. 2, pp.263- 272,2004
- [6] Tim Allman, Alex D. Bain, and Joel R. Garbow, *SIMPLTN, a Program for the Simulation of Pulse NMR Spectra*, Journal of Magnetic Resonance, Series A, vol. 123, no. 1, pp. 26-31, 1996.
- [7] Alan Carrington, Andrew D. McLachlan, *Introduction to Magnetic Resonance*, Harper & Row, 1967
- [8] Eric W. Weisstein. *Eigen Decomposition*. From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/EigenDecomposition.html>
- [9] E. Mark Haacke, Robert W. Brown, Michael R. Thompson, Ramesh Venkatesan, *Magnetic Resonance Imaging- Physical Principles and Sequence Design*, John Wiley & Sons, 1999
- [10] Magnetic Resonance - Technology Information Portal, <http://www.mrtip.com/serv1.php?type=db1&dbs=Magnetic%20Resonance%20Imaging%20MRI>
- [11] Christophe Odin, *Calculations of Multipulse Sequence in NQR of Spins*, Journal of Magnetic Resonance, vol. 141, no. 2, pp. 239-255, 1999
- [12] S.Z.Ageev, P.P.Man and B.C.Sanctuary, *Detection of Double and Four Quantum Coherences for SpinExcited by Spin Lock Pulse Sequences*, Journal of Magnetic Resonance, vol. 128, no. 1, pp. 12-20, 1997

- [13] S.Z.Ageev, P.P.Man, J.Fraissard and B.C.Sanctuary, *Determination of quadrupolar spin coupling for spin 7/2 using two pulse sequences*, Molecular Physics vol. 91, no.1, pp. 75-80, 1997.
- [14] D.J.Isbister, M.S.Krishnan and B.C.Sanctuary, *Use of computer algebra for the study of quadrupole spin systems*, Molecular Physics vol. 86, pp. 1517-1535, 1995.
- [15] S.Bahceli and A.Tokatli, *The product operator description of INEPT experiments for IS ($I=1/2$; $S=5/2$) spin systems and its applications to microcline and TMP-AlCl₃ complex in NaX zeolite*, Appl.Magn.Reson. vol. 29, pp. 617-626, 2005.
- [16] A.Tokatli, A.Gencten, M.Sahin, O.Teziel and S.Bahceli, *Product operator descriptions of INEPT and RINEPT NMR spectroscopies for ISn ($I=1/2$, $S=3/2$) spin systems*, J.Magn.Reson. vol. 169, no.1, pp. 68-72, 2004.
- [17] Yuval Zur, *An algorithm to calculate the NMR signal of a multi spin-echo sequence with relaxation and spin-diffusion*, Journal of Magnetic Resonance, vol. 171, no. 1, pp. 97-106, 2004
- [18] S. Fritzsche, *Maple procedures for the coupling of angular momenta I. Data structures and numerical computations*, Computer Physics Communications, vol. 103, no. 1, pp. 51-73, 1997
- [19] S. Fritzsche, S. Varga, D. Geschke & B. Fricke, *Maple procedures for the coupling of angular momenta II. Sum rule evaluation*, Computer Physics Communications, vol. 111, no. 1-3, pp. 167-184, 1998
- [20] G. Gaigalas, S. Fritzsche & B. Fricke, *Maple procedures for the coupling of angular momenta. III. Standard quantities for evaluating many-particle matrix elements*, Computer Physics Communications, vol. 135, no. 2, pp. 219-237, 2001
- [21] T. Inghoff, S. Fritzsche & B. Fricke, *Maple procedures for the coupling of angular momenta. IV. Spherical harmonics*, Computer Physics Communications, vol. 139, no. 3, pp. 297-313, 2001
- [22] S. Fritzsche, T. Inghoff, T. Bastug & M. Tomaselli, *Maple procedures for the coupling of angular momenta. V. Recoupling coefficients*, Computer Physics Communications, vol. 139, no. 3, pp. 314-326, 2001
- [23] G. Gaigalas & S. Fritzsche, *Maple procedures for the coupling of angular momenta. VI. LS-jj transformations*, Computer Physics Communications, vol. 149, no. 1, pp. 39-60, 2002
- [24] S. Fritzsche, T. Inghoff & M. Tomaselli, *Maple procedures for the coupling of angular momenta. VII. Extended and accelerated computations*, Computer Physics Communications, vol. 153, no. 3, pp. 424-444, 2003

- [25] G. Gaigalas, O. Scharf & S. Fritzsche, *Maple procedures for the coupling of angular momenta. VIII. Spin-angular coefficients for single-shell configurations*, Computer Physics Communications, vol. 166, no. 2, pp. 141-169, 2005
- [26] A. Jerschow, *MathNMR: Spin and spatial tensor manipulations in Mathematica*, Journal of Magnetic Resonance, vol. 176, no. 1, pp. 7-14, 2005
- [27] M. P. Barnett, J. F. Capitani, J. von zur Gathen & J. Gerhard, *Symbolic calculation in chemistry: Selected examples*, International Journal of Quantum Chemistry, vol. 100, no. 2, pp. 80-104, 2004
- [28] N.A. Nguyen & T. T. Nguyen-Dang, *Symbolic calculations of unitary transformations in quantum dynamics*, Computer Physics Communications, vol. 115, no. 2-3, pp. 183-199, 1998
- [29] R.P.F. Kanters, B.W. Char & A.W. Addison, *A Computer-Algebra Application for the Description of NMR Experiments Using the Product-Operator Formalism*, Journal of Magnetic Resonance, Series A, vol. 101, no. 1, pp. 23-29, 1993