# Multiple Correlation Function Approach Library

## Introduction

Multiple Correlation Function Approach Library (MCFAL) is developed for numerical calculation of the NMR signal attenuated by restricted diffusion of spins in three confining domains: a slab, a cylinder and a sphere. This library is implemented as a set of functions for Matlab software on the base of the Multiple Correlation Function (MCF) approach by D. S. Grebenkov[1].

The MCF approach allows one to study restricted diffusion under *arbitrary magnetic field*. The moments of a random phase accumulated by a diffusing spin were found in a matrix form involving the Laplace operator eigenbasis in a confining domain. Spatial inhomogeneities and time dependence of the magnetic field enter as functionals and weight factors to the multiple correlation functions. Although the primary aim of this approach was a theoretical study of diffusive NMR phenomena, it is an efficient numerical technique. This manual is intended to briefly explain the functionality of the MCFAL and give several examples of its use.



[1] D. S. Grebenkov, *NMR survey of the reflected Brownian motion*, Rev. Mod. Phys. (submitted).

**Brief mathematical description**

In this section, we briefly explain the mathematical basis of the numerical technique[1]. This is an extension of the approach suggested originally by Robertson and further developed by Barzykin[2].

The signal attenuation due to restricted diffusion in a confining domain Ω can be found in the following way. The magnetization m(**r**,t) is known to satisfy the Bloch-Torrey equation[3],

$$\left(\frac{\partial}{\partial t} - D\Delta + i\gamma\beta f(t)B(\mathbf{r})\right) \mathfrak{m}(\mathbf{r},t) = 0$$

where D is the free diffusion coefficient, Δ the Laplace operator, γ the gyromagnetic ratio, and βf(t)B(**r**) the diffusion-sensitizing magnetic field. Here β is the maximum intensity, while f(t) and B(**r**) are normalized temporal and spatial profiles, respectively. The Neumann boundary condition

$$\frac{\partial}{\partial n}\, \mathfrak{m}(\mathbf{r},t) = 0$$

is imposed on the boundary ∂Ω of the confining domain. This condition can be used if there is no surface relaxation. At t=0, the magnetization density is supposed to be uniform:

$$\mathfrak{m}(\mathbf{r}, t=0) = \frac{1}{V}$$

V being the volume of the confining domain. A more general case of the Fourier (or mixed) boundary condition and non-uniform initial density was considered in Grebenkov (2006). This feature is not implemented yet to the MCFAL.

For the moment, let us consider the case f(t)=1. The magnetization m(**r**,t) can be expanded over the basis of eigenfunctions $u_m$(**r**) of the Laplace operator (with the Neumann boundary condition):

$$\mathfrak{m}(\mathbf{r},t) = \sum_{m'=0}^{\infty} c_{m'}(t)\; u_{m'}(\mathbf{r})$$

The macroscopic signal E would be obtained by integrating m(**r**,T) over Ω:

$$E = \int_{\Omega} d\mathbf{r}\; \mathfrak{m}(\mathbf{r},T) = \sum_{m'=0}^{\infty} c_{m'}(T) \int_{\Omega} d\mathbf{r}\; u_{m'}(\mathbf{r}) = V^{1/2} c_0(T)$$

To find the coefficient $c_0$(T), one substitutes the above expansion into the Bloch-Torrey equation, multiplies it by the eigenfunction $u_m$*(**r**) and integrates over Ω. These operations lead to a set of ordinary differential equations:

---

[2] B. Robertson, *Spin-echo decay of spins diffusion in a bounded region*, Phys. Rev. **151**, 273-277 (1966); A. V. Barzykin, *Theory of Spin Echo in Restricted Geometries under a Step-wise Gradient Pulse Sequence*, J. Magn. Reson. **139**, 342 (1999).

[3] H. C. Torrey, *Bloch Equations with Diffusion Terms*, Phys. Rev. **104**, 563 (1956).

$$\frac{d}{dt}\, c_m(t) + \frac{D\lambda_m}{L^2}\, c_m(t) + i\gamma\beta \sum_{m'=0}^{\infty} \mathcal{B}_{m,m'}\, c_{m'}(t) = 0$$

where the infinite-dimension matrix *B* is defined as

$$\mathcal{B}_{m,m'} = \int_{\Omega} d\mathbf{r}\; u_m^*(\mathbf{r})\; B(\mathbf{r})\; u_{m'}(\mathbf{r})$$

The initial condition implies $c_m(0) = V^{-1/2}\, \delta_{m,0}$.

The coefficients $c_m(t)$ can be thought as components of an infinite-dimension vector C(t), and the above set of equations gives

$$T\frac{d}{dt}\, C(t) = -(p\Lambda + iq\mathcal{B})C(t)$$

where the diagonal infinite-dimension matrix Λ is defined as

$$\Lambda_{m,m'} = \delta_{m,m'}\; \lambda_m$$

The two dimensionless parameters p and q are defined as

$$p = DT/L^2 \qquad\qquad q = \gamma\beta T$$

where L is the characteristic dimension of the confining domain.

The solution of the above differential equation is simply

$$C(t) = \exp[-(p\Lambda + iq\mathcal{B})t/T]C(0)$$

Bringing together the above relations, one can write the macroscopic signal as the first diagonal element of the matrix

$$E = \left[\exp[-(p\Lambda + iq\mathcal{B})]\right]_{0,0}$$

We stress that *this is an exact result* for the case f(t)=1.

If the temporal profile f(t) is not constant, the time interval [0,T] can be divided into a large number K of subintervals of duration τ=T/K. On the $k^{th}$ subinterval, the function f(t) is approximated by a constant f(kτ). The signal can be numerically found with

$$E \simeq \left[\prod_{k=0}^{K} \exp[-\tau(p\Lambda + iq\; f(k\tau)\; \mathcal{B})]\right]_{0,0}$$

The last relation is used to calculate the signal in the MCFAL. For this purpose, one first calculates the matrices *B* and Λ for a chosen confining domain Ω and spatial profile B(**r**). The time dependence of the magnetic field is then approximated.

**Description of the functions**

☛ After each launching Matlab, it is recommended to start the use of MCFAL by the initialization function:

function [] = MCF_ini;

This function initializes several global variables used by other functions. In particular, it sets on the flag ‘warning’ in order to show warning messages.

☛ The main function calculating the signal is declared as

function [E] = MCF(Gradient, Time, Length, Diffusion, Gamma, Domain, Sprofile, Tprofile);

Input parameters:

| | |
|---|---|
| Gradient | Value (or array of values) of the diffusion-sensitizing magnetic field intensity, e.g., the gradient strength (in Tesla per meter, T/m). |
| Time | Echo time (in seconds, s). |
| Length | Characteristic dimension of the confining domain: separation width for a slab geometry and radius for a cylinder and a sphere (in meters, m). |
| Diffusion | Free (self-)diffusion coefficient (in square meters per second, $m^2/s$). |
| Gamma | Nuclear gyromagnetic ratio (in radians per Tesla per second, rad $T^{-1}s^{-1}$). By default, 'Gamma' is equal to $2.675 \cdot 10^8$ rad $T^{-1}s^{-1}$ that corresponds to protons. |
| Domain | Confining domain:<br>– 'p' or 'plane' for a slab geometry;<br>– 'c' or 'cylinder' for a cylinder;<br>– 's' or 'sphere' for a sphere.<br>By default, a slab geometry is considered. |
| Sprofile | Spatial profile of the magnetic field. For the moment, only two profiles are considered:<br>– 'l' or 'linear' for a linear gradient,<br>– 'p' or 'parabolic' for a parabolic magnetic field.<br>By default, the linear gradient is used. |
| Tprofile | Vector representing discretized temporal profile of the magnetic field (see below). By default, a steady profile is considered. |

The function MCF returns the value E (array of values) of the NMR signal calculated for the set of input parameters (number of values in E is determined by the length of the input vector ‘Gradient’).

The signal E is normalized in such a way that E=1 if there is no diffusion-sensitizing magnetic field.

It is MANDATORY to use the SI units as described above.

Examples:

```
Tprofile = MCF_Trectangle(5e-3, 0, 100);
E = MCF(1e-2, 1e-1, 1e-4, 2.3e-9, 2.675e8, 'plane', 'linear', Tprofile)

E =
   0.3561
```

Returns the signal attenuated by restricted diffusion of protons (water molecules) confined between parallel planes of separation 0.1 mm under steady magnetic field of linear gradient 10 mT/m and duration 0.1 s.

The same result could be obtained by writing simply

```
E = MCF(1e-2, 1e-1, 1e-4, 2.3e-9)

E =
   0.3561
```

To study the dependence of the signal on the intensity of the magnetic field, one can write

```
g = 0:0.001:0.01;
E = MCF(g, 1e-1, 1e-4, 2.3e-9)

E =
 Columns 1 through 8

    1.0000    0.9891    0.9573    0.9066    0.8405    0.7632    0.6791    0.5927

  Columns 9 through 11

    0.5081    0.4285    0.3561
```

This function returns a vector E of 11 elements containing the signal for the gradient varying between 0 and 10 mT/m.

☛ Although a number of physical parameters are involved, the signal attenuation is determined by two dimensionless parameters $p=DT/L^2$ and $q=\gamma\beta T$ (or $q=\gamma gLT$ for a linear gradient). The following function calculates the signal for given values of these two parameters

```
function [E] = MCF_pq(p, q, Domain, Sprofile, Tprofile);
```

Input parameters:

| | |
|---|---|
| p | Value (or array of values) of the dimensionless diffusion coefficient. |
| q | Value (or array of values) of the dimensionless magnetic field intensity. |

| | |
|---|---|
| Domain | Confining domain:<br>– 'p' or 'plane' for a slab geometry;<br>– 'c' or 'cylinder' for a cylinder;<br>– 's' or 'sphere' for a sphere.<br>By default, a slab geometry is considered. |
| Sprofile | Spatial profile of the magnetic field. For the moment, only two profiles are considered:<br>– 'l' or 'linear' for a linear gradient,<br>– 'p' or 'parabolic' for a parabolic magnetic field.<br>By default, the linear gradient is used. |
| Tprofile | Vector representing discretized temporal profile of the magnetic field (see below). By default, a steady profile is considered. |

The function MCF_pq returns the value E (or array of values, or matrix) of the NMR signal calculated for the set of input parameters (the size of the matrix E is determined by the lengths of the input vectors ‘p’ and ‘q’). Although this function is a bit more general than MCF, the calculation is identical for both cases.

The signal E is normalized in such a way that E=1 if there is no diffusion-sensitizing magnetic field (q=0).

Example:

```
Tprofile = MCF_Trectangle(5e-3, 0, 100);
p = 2.3e-9 * 1e-1/(1e-4*1e-4)

p =
    0.0230

q = 2.675e8 * 1e-2 * 1e-4 * 1e-1

q =
    26.7500

E = MCF_pq(p, q, 'plane', 'linear', Tprofile)

E =
    0.3561
```

☛ The function MCF_BL returning the truncated matrices $B$ and $\Lambda$ for a given domain and spatial profile is

```
function [B,Lam] = MCF_BL(Domain, Sprofile, N);
```

Input parameters:

| | |
|---|---|
| Domain | Confining domain: |

| | |
|---|---|
| | – 'p' or 'plane' for a slab geometry;<br>– 'c' or 'cylinder' for a cylinder;<br>– 's' or 'sphere' for a sphere.<br>By default, slab geometry is considered. |
| Sprofile | Spatial profile of the magnetic field. For the moment, only two profiles are considered:<br>– 'l' or 'linear' for a linear gradient,<br>– 'p' or 'parabolic' for a parabolic magnetic field.<br>By default, the linear gradient is used. |
| N | Requested size of matrices $B$ and $\Lambda$. In the case of a cylinder or a sphere, this truncation parameter must be smaller than 60. By default, N is equal to 50. |

The matrices $B$ and $\Lambda$ are formally of infinite dimension, but a rapid decrease of the Laplace operator eigenvalues allows one to truncate these matrices to a moderate size. This is a numerical parameter 'N' that controls the accuracy of calculations: for bigger 'N', the calculation is more accurate. Of course, the increase of 'N' slows down the computation. In the most typical situations, 'N' around 20 is already sufficient for a very precise analysis. Except for a few cases, we never used 'N' bigger than 50. In the case of a cylinder or a sphere, the value of 'N' cannot exceed 60 for the function MCF_BL. This limitation is simply related to the fact that the matrices $B$ and $\Lambda$ were calculated once for 'N'=60 and then stored as binary files. The function MCF_BL merely reads data from these files. In the case of slab geometry, the matrices $B$ and $\Lambda$ are recalculated by the function MCF_BL.

Example:

```
[B,Lam] = MCF_BL('s', 'l', 2)

B =

         0    0.4448
    0.4448         0


Lam =

         0         0
         0    4.3330
```

One obtains the matrices $B$ and $\Lambda$ truncated to sizes 2x2 for a sphere under a linear gradient. Of course, such truncation is shown only for illustrative purpose, and it is not sufficient to calculate the signal.

☛ The rectangular Stejskal-Tanner two-pulse profile is generated by function

```
function [Tprofile] = MCF_Trectangle(Steady, Pause, K);
```

Input parameters:

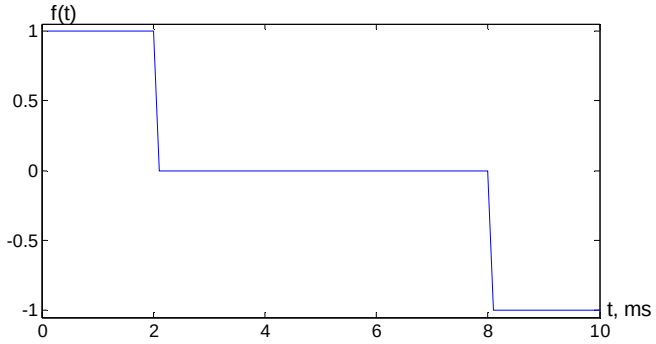
| | |
|---|---|
| Steady | Duration of a rectangular pulse (in seconds, s). |
| Pause | Delay between rectangular pulses (in seconds, s). |
| K | Discretization parameter (must lie between 10 and 10000). |

This function returns a vector of length K containing the discrete version of the rectangular Stejskal-Tanner two-pulse profile. The accuracy is better for bigger value of K, but the computation is slower. For typical calculations, K=100 ensures quite accurate results.

Example:

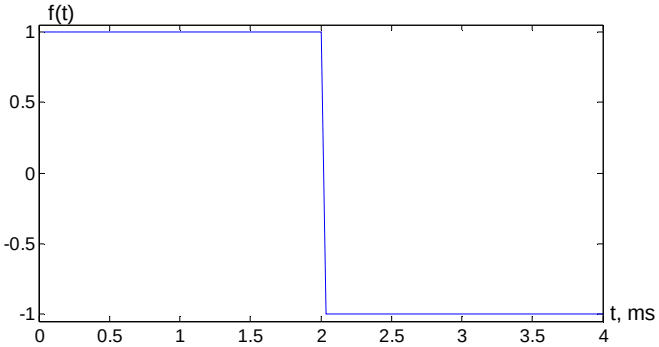
Tprofile = MCF_Trectangle(2e-3, 6e-3, 100);

generates the temporal profile f(t) with two rectangular pulses of duration 2 ms with time inverval 6 ms.




Tprofile = MCF_Trectangle(2e-3, 0, 100);

generates a steady profile.



☛ The trapezoidal Stejskal-Tanner two-pulse profile is generated by function

function [Tprofile] = MCF_Ttrapeze(Up, Steady, Down, Pause, K);

Input parameters:

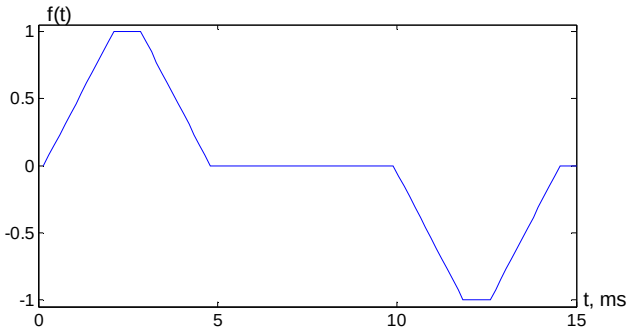
| | |
|---|---|
| Up | Ramp increasing time (in seconds, s). |
| Steady | Duration of plateau (in seconds, s). |
| Down | Ramp decreasing time (in seconds, s). |
| Pause | Delay between trapezoidal pulses (in seconds, s). |
| K | Discretization parameter (must lie between 10 and 10000). |

This function returns a vector of length K containing the discrete version of the trapezoidal Stejskal-Tanner two-pulse profile. The accuracy is better for bigger value of K, but the computation is slower. For typical calculations, K=100 ensures quite accurate results.

Example:

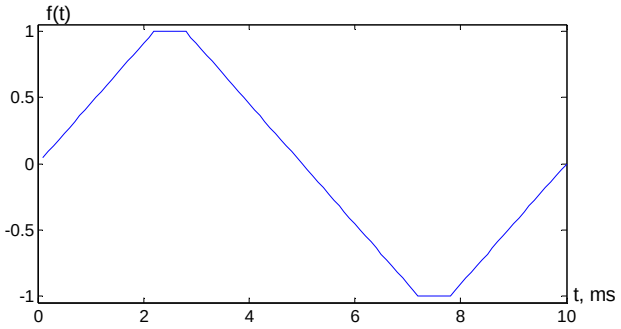
Tprofile = MCF_Ttrapeze(2e-3, 1e-3, 2e-3, 5e-3, 100);

generates the temporal profile f(t) with two trapezoidal pulses of total duration 5 ms with time inverval 5 ms. Note that the last linear segment (14-15 ms) is an artifact of rounding. This artifact is suppressed at higher discretization (e.g., K=1000).




Tprofile = MCF_Ttrapeze(2.2e-3, 0.6e-3, 2.2e-3, 0, 100);

generates the temporal profile f(t) with two trapezoidal pulses of total duration 5 ms without time inverval.



Tprofile = MCF_Ttrapeze(0, 2e-3, 0, 6e-3, 100);

generates the same rectangular profile as previous function MCF_Trectangular.

☛ Coefficients $\zeta_k$ containing averaged information about the spatial profile of the magnetic field in a confining domain are defined as

$$\zeta_k = \sum_{m=1}^{\infty} \mathcal{B}_{0,m}\ \lambda_m^k\ \mathcal{B}_{m,0}$$

and given by

function [z] = MCF_zeta(Domain, Sprofile, k);

Input parameters:

| | |
|---|---|
| Domain | Confining domain:<br>– 'p' or 'plane' for a slab geometry;<br>– 'c' or 'cylinder' for a cylinder;<br>– 's' or 'sphere' for a sphere.<br>By default, slab geometry is considered. |
| Sprofile | Spatial profile of the magnetic field. For the moment, only two profiles are considered:<br>– 'l' or 'linear' for a linear gradient,<br>– 'p' or 'parabolic' for a parabolic magnetic field.<br>By default, the linear gradient is used. |
| k | Index (an integer number which must not exceed 1). |

For three basic domains, the coefficients $\zeta_k$ are convergent for k≤1. For k = 1, 0, -1, or –2, the values of $\zeta_k$ are simply tabulated for a slab, a cylinder and a sphere with linear gradient and parabolic magnetic fields. When k<-2, these coefficients are computed numerically, and they may be not very accurate. The function is used to theoretically study the slow diffusion and motional narrowing regimes.

Examples:

MCF_zeta('c', 'l', 1)

returns 1 since $\zeta_1$ is equal to 1 in any confining domain under a linear gradient.

MCF_zeta('p', 'l', -1)

returns 0.0083 (=1/120) for a slab geometry.

☛ The f-weighted time average $<(t_2-t_1)^{\alpha}>_2$ is defined as

$$< (t_2 - t_1)^{\alpha} >_2 = \int_0^1 dt_1 \; f(t_1) \int_{t_1}^1 dt_2 \; f(t_2) \; (t_2 - t_1)^{\alpha}$$

and numerically calculated for a given temporal profile f(t) by

function [Ta] = MCF_Taverage(Tprofile, alpha);

Input parameters:

| | |
|---|---|
| Tprofile | Vector representing discretized temporal profile of the magnetic field (see below). By default, a steady profile is considered. |
| Alpha | Arbitrary positive exponent. |

The f-weighted time average is approximated by a sum over discretized temporal profile. This calculation is more accurate for bigger length of Tprofile. Please note that the result of this function is *dimensionless*, the usual factor $T^3$ is not taken into account. Since $<(t_2-t_1)^{\alpha}>_2$ is negative, the function MCF_Taverage explicitly changes its sign for convenience.

Example:

```
Tprofile = MCF_Trectangle(5e-3, 0, 100);
Ta = MCF_Taverage(Tprofile, 1)

Ta =

   0.0834
```

This is a good approximation to $<(t_1-t_2)>_2=1/12$ for a steady temporal profile.

☛ The integral of squared temporal profile

$$\int_0^1 dt \; f^2(t)$$

is calculated by

function [Ta] = MCF_T2average(Tprofile);

Input parameters:

| | |
|---|---|
| Tprofile | Vector representing discretized temporal profile of the magnetic field. By default, a steady profile is considered. |

This integral is approximated by a sum over discretized temporal profile. This calculation is more accurate for bigger length of Tprofile. Please note that the result of this function is *dimensionless*, the usual factor $T^3$ is not taken into account.

Example:

```
Tprofile = MCF_Trectangle(2e-3, 6e-3, 100);
Ta = MCF_T2average(Tprofile)

Ta =

    0.4000
```

This is an exact result for two rectangular pulses of duration 2 ms with delay 6 ms.

☛ The second moment $\boldsymbol{E}\{\phi^2/2\}$ of the accumulated phase $\phi$ for steady temporal profile is defined as

$$\mathbb{E}\{\phi^2/2\} = \sum_{m=1}^{\infty} \mathcal{B}_{0,m}\mathcal{B}_{m,0}\left(\frac{1}{p\lambda_m} - \frac{e^{-p\lambda_m} - 4e^{-p\lambda_m/2} + 3}{p^2\lambda_m^2}\right)$$

and calculated by

function [E] = MCF_phi2(p, Domain, Sprofile);

Input parameters:

| | |
|---|---|
| p | Value (or array of values) of the dimensionless diffusion coefficient. |
| Domain | Confining domain:<br>– 'p' or 'plane' for a slab geometry;<br>– 'c' or 'cylinder' for a cylinder;<br>– 's' or 'sphere' for a sphere.<br>By default, slab geometry is considered. |
| Sprofile | Spatial profile of the magnetic field. For the moment, only two profiles are considered:<br>– 'l' or 'linear' for a linear gradient,<br>– 'p' or 'parabolic' for a parabolic magnetic field.<br>By default, the linear gradient is used. |

The function returns the value (or array of values) of the second moment. Since this calculation is based on the exact formula, the result is very accurate.

Example:

```
p = [0.1, 1, 10];
E = MCF_phi2(p, 'cylinder', 'linear')

E =

   0.0064    0.0235    0.0066
```

Three values of the second moment are shown for p=0.1 (slow diffusion), p=1 (intermediate regime) and p=10 (fast diffusion).

☛ In the previous function, the second moment $\boldsymbol{E}\{\phi^2/2\}$ was found for steady temporal profile by means of an exact formula, whatever the value of p is. Although similar formula could be in principle derived for any temporal profile, its practical realization is difficult. A theoretical approximation may be then useful to find the second moment for a given temporal profile. In the slow diffusion regime (p<<1), the second moment can be approximated as

$$\mathbb{E}\{\phi^2/2\} \simeq \sum_{k=0}^{K_{max}} c_{1+k/2} < (t_2 - t_1)^{1+k/2} >_2 p^{1+k/2}$$

where coefficients $c_{1+k/2}$ depend on the confining domain and the magnetic field spatial profile. The approximate relation is implemented by

function [E] = MCF_phi2slow(p, Domain, Sprofile, Tprofile, Kmax);

Input parameters:

| | |
|---|---|
| p | Value (or array of values) of the dimensionless diffusion coefficient. |
| Domain | Confining domain:<br>– 'p' or 'plane' for a slab geometry;<br>– 'c' or 'cylinder' for a cylinder;<br>– 's' or 'sphere' for a sphere.<br>By default, slab geometry is considered. |
| Sprofile | Spatial profile of the magnetic field. For the moment, only two profiles are considered:<br>– 'l' or 'linear' for a linear gradient,<br>– 'p' or 'parabolic' for a parabolic magnetic field.<br>By default, the linear gradient is used. |
| Tprofile | Vector representing discretized temporal profile of the magnetic field. By default, a steady profile is considered. |
| Kmax | Number of correction terms (Kmax=0 for the leading term alone; Kmax=1 for the classical $p^{3/2}$-correction; Kmax≥2 for higher order terms). |

The function returns the value (or array of values) of the approximate second moment. There is no verification whether the input parameter p is small enough or not. If this condition failed, the result may be incorrect or even unphysical (e.g., negative).

Consequently, this is the user who has to pay special attention to the accuracy and applicability of this function.

The accuracy of the slow diffusion approximation depends on the number of correction terms considered in the series of powers $p^{n/2}$. The parameter Kmax determines the highest considered power $p^{1+Kmax/2}$. Using different values of Kmax, one can investigate the role of these corrections. In general, the leading term alone (Kmax=0) is very rough approximation, insufficient for typical values of p. The classical $p^{3/2}$-correction (Kmax=1) considerably improves it. Higher order terms can still be required when p is not small enough. For the moment, the corrections are limited[4] to Kmax=3. Note that bigger Kmax does not increase the computational time, so that the maximum value can be always used.

Example:

```
Tprofile = MCF_Trectangle(5e-3, 0, 100);
p = [0.1, 1, 10];
E = MCF_phi2slow(p, 'cylinder', 'linear', Tprofile, 2)

E =

   0.0064    0.0121   -2.4866
```

The comparison with previous exact calculation for a steady temporal profile shows that this function is accurate for the slow diffusion (p=0.1), but it gives invalid values for p=1 and p=10 (where the slow diffusion approximation apparently fails).

☛ In the motional narrowing regime (p>>1), the second moment $\boldsymbol{E}\{\phi^2/2\}$ can be found as

$$\mathbb{E}\{\phi^2/2\} \simeq \frac{\zeta_{-1}}{p} \int_0^1 dt\ f^2(t) + O(p^{-2})$$

This approximate relation is implemented by

```
function [E] = MCF_phi2fast(p, Domain, Sprofile, Tprofile);
```

Input parameters:

| | |
|---|---|
| p | Value (or array of values) of the dimensionless diffusion coefficient. |
| Domain | Confining domain:<br>– 'p' or 'plane' for a slab geometry;<br>– 'c' or 'cylinder' for a cylinder; |

[4] For slab geometry under linear gradient, the series of corrections in powers $p^{n/2}$ is naturally truncated to the classical $p^{3/2}$-correction (higher correction is exponential). In this case, Kmax cannot exceed 1. Similarly, for slab geometry under parabolic field, Kmax is limited to 2 [see (Grebenkov, 2006) for details].

| | |
|---|---|
| | – 's' or 'sphere' for a sphere.<br>By default, slab geometry is considered. |
| Sprofile | Spatial profile of the magnetic field. For the moment, only two profiles are considered:<br>– 'l' or 'linear' for a linear gradient,<br>– 'p' or 'parabolic' for a parabolic magnetic field.<br>By default, the linear gradient is used. |
| Tprofile | Vector representing discretized temporal profile of the magnetic field. By default, a steady profile is considered. |

The function returns the value (or array of values) of the approximate second moment. There is no verification whether the input parameter p is big enough or not. If this condition failed, the result may be incorrect. Consequently, this is the user who has to pay special attention to the accuracy and applicability of this function. In contrast with the function MCF_phi2slow, only the leading term is calculated by this function since the form of correction terms sensitively depends on the temporal profile.

Example:

```
Tprofile = MCF_Trectangle(5e-3, 0, 100);
p = [0.1, 1, 10];
E = MCF_phi2fast(p, 'cylinder', 'linear', Tprofile)

E =

    0.7292    0.0729    0.0073
```

The last value (for p=10) is relatively close to the exact value 0.0066 of the second moment (see function MCF_phi2). In opposite, the first two values (for p=0.1 and p=1) are invalid since the condition p>>1 is failed.

☛ The result of the previous function can be significantly improved by correction term. For the steady temporal profile, the approximate relation

$$\mathbb{E}\{\phi^2/2\} \simeq \frac{\zeta_{-1}}{p} - \frac{3\zeta_{-2}}{p^2}$$

is implemented by

```
function [E] = MCF_phi2fast_steady(p, Domain, Sprofile);
```

Input parameters:

| | |
|---|---|
| p | Value (or array of values) of the dimensionless diffusion coefficient. |
| Domain | Confining domain:<br>– 'p' or 'plane' for a slab geometry;<br>– 'c' or 'cylinder' for a cylinder;<br>– 's' or 'sphere' for a sphere. |

| | |
|---|---|
| | By default, slab geometry is considered. |
| Sprofile | Spatial profile of the magnetic field. For the moment, only two profiles are considered:<br>– 'l' or 'linear' for a linear gradient,<br>– 'p' or 'parabolic' for a parabolic magnetic field.<br>By default, the linear gradient is used. |

The function returns the value (or array of values) of the approximate second moment. There is no verification whether the input parameter p is big enough or not. If this condition failed, the result may be incorrect or even unphysical (e.g., negative). Consequently, this is the user who has to pay special attention to the accuracy and applicability of this function. In contrast with the function MCF_phi2f, the correction term is added to improve the accuracy.

Example:

```
p = [0.1, 1, 10];
E = MCF_phi2fast_steady(p, 'cylinder', 'linear')

E =
    -5.7161    0.0085    0.0066
```

The last value (for p=10) is equal to the exact value 0.0066 of the second moment (see function MCF_phi2). In opposite, the first two values (for p=0.1 and p=1) are invalid since the condition p>>1 is failed.

☛ In the slow diffusion regime (p<<1), the signal can be approximated by

function [E] = MCF_slow(Gradient, Time, Length, Diffusion, Gamma, Domain, Sprofile, Tprofile);

Input parameters:

| | |
|---|---|
| Gradient | Value (or array of values) of the diffusion-sensitizing magnetic field intensity, e.g., the gradient strength (in Tesla per meter, T/m). |
| Time | Echo time (in seconds, s). |
| Length | Characteristic dimension of the confining domain: separation width for a slab geometry and radius for a cylinder and a sphere (in meters, m). |
| Diffusion | Free (self-)diffusion coefficient (in square meters per second, $m^2/s$). |
| Gamma | Nuclear gyromagnetic ratio (in radians per Tesla per second, rad $T^{-1}s^{-1}$). By default, 'Gamma' is equal to $2.675 \cdot 10^8$ rad $T^{-1}s^{-1}$ that corresponds to protons. |
| Domain | Confining domain:<br>– 'p' or 'plane' for a slab geometry;<br>– 'c' or 'cylinder' for a cylinder;<br>– 's' or 'sphere' for a sphere. |

| | |
|---|---|
| | By default, a slab geometry is considered. |
| Sprofile | Spatial profile of the magnetic field. For the moment, only two profiles are considered:<br>– 'l' or 'linear' for a linear gradient,<br>– 'p' or 'parabolic' for a parabolic magnetic field.<br>By default, the linear gradient is used. |
| Tprofile | Vector representing discretized temporal profile of the magnetic field (see below). By default, a steady profile is considered. |

The slow diffusion approximation of the signal is based on the computation of the second moment by function MCF_phi2s. There is no verification whether the input parameter p is small enough or not. In additional, the input parameter q should be small. If one of these conditions failed, the result may be incorrect. Consequently, this is the user who has to pay special attention to the accuracy and applicability of this function.

Example:

```
Tprofile = MCF_Trectangle(5e-3, 0, 100);
E = MCF_slow(1e-2, 1e-1, 1e-4, 2.3e-9, 2.675e8, 'plane', 'linear', Tprofile)

E =
   0.3348
```

This value is smaller than the exact result 0.3561 by function MCF (see above).

☛ In the motional narrowing regime (p>>1), the signal can be approximated by

function [E] = MCF_fast(Gradient, Time, Length, Diffusion, Gamma, Domain, Sprofile, Tprofile);

Input parameters:

| | |
|---|---|
| Gradient | Value (or array of values) of the diffusion-sensitizing magnetic field intensity, e.g., the gradient strength (in Tesla per meter, T/m). |
| Time | Echo time (in seconds, s). |
| Length | Characteristic dimension of the confining domain: separation width for a slab geometry and radius for a cylinder and a sphere (in meters, m). |
| Diffusion | Free (self-)diffusion coefficient (in square meters per second, $m^2/s$). |
| Gamma | Nuclear gyromagnetic ratio (in radians per Tesla per second, rad $T^{-1}s^{-1}$). By default, 'Gamma' is equal to $2.675 \cdot 10^8$ rad $T^{-1}s^{-1}$ that corresponds to protons. |
| Domain | Confining domain:<br>– 'p' or 'plane' for a slab geometry;<br>– 'c' or 'cylinder' for a cylinder;<br>– 's' or 'sphere' for a sphere.<br>By default, a slab geometry is considered. |

| | |
|---|---|
| Sprofile | Spatial profile of the magnetic field. For the moment, only two profiles are considered:<br>– 'l' or 'linear' for a linear gradient,<br>– 'p' or 'parabolic' for a parabolic magnetic field.<br>By default, the linear gradient is used. |
| Tprofile | Vector representing discretized temporal profile of the magnetic field (see below). By default, a steady profile is considered. |

The slow diffusion approximation of the signal is based on the computation of the second moment by function MCF_phi2f. There is no verification whether the input parameter p is big enough or not. In additional, the input parameter q should be small. If one of these conditions failed, the result may be incorrect. Consequently, this is the user who has to pay special attention to the accuracy and applicability of this function.

Example:

```
Tprofile = MCF_Trectangle(5e-3, 0, 100);
E = MCF_fast(1e-2, 1e-1, 1e-4, 2.3e-9, 2.675e8, 'plane', 'linear', Tprofile)

E =
    2.5354e-113
```

This value is obviously invalid since the condition p>>1 is failed (here p=0.023).

☛ Two similar functions involving the dimensionless parameters p and q are introduced to consider the slow diffusion (p<<1) and the motional narrowing (p>>1) regimes:

```
function [E] = MCF_pq_slow(p, q, Domain, Sprofile, Tprofile);

function [E] = MCF_pq_fast(p, q, Domain, Sprofile, Tprofile);
```

Input parameters:

| | |
|---|---|
| p | Value (or array of values) of the dimensionless diffusion coefficient. |
| q | Value (or array of values) of the dimensionless magnetic field intensity. |
| Domain | Confining domain:<br>– 'p' or 'plane' for a slab geometry;<br>– 'c' or 'cylinder' for a cylinder;<br>– 's' or 'sphere' for a sphere.<br>By default, a slab geometry is considered. |
| Sprofile | Spatial profile of the magnetic field. For the moment, only two profiles are considered:<br>– 'l' or 'linear' for a linear gradient,<br>– 'p' or 'parabolic' for a parabolic magnetic field.<br>By default, the linear gradient is used. |
| Tprofile | Vector representing discretized temporal profile of the magnetic field |

| | |
|---|---|
| | (see below). By default, a steady profile is considered. |

These functions return the value E (or array of values, or matrix) of the approximated signal. There is no verification whether the input parameter p is small (big) enough or not. In additional, the input parameter q should be small. If one of these conditions failed, the result may be incorrect. Consequently, this is the user who has to pay special attention to the accuracy and applicability of this function.

Example:

```
Tprofile = MCF_Trectangle(5e-3, 0, 100);
p = 2.3e-9 * 1e-1/(1e-4*1e-4)

p =
      0.0230

q = 2.675e8 * 1e-2 * 1e-4 * 1e-1

q =
      26.7500

E = MCF_pq_slow(p, q, 'p', 'l', Tprofile)

E =
    0.3348

E = MCF_pq_fast(p, q, 'p', 'l', Tprofile)

E =
    2.5354e-113
```

There are several auxiliary functions required to make MCFAL operational. Since these functions are not supposed for an independent use, we only list them without specification:

```
function [tp] = MCF_type(Domain, Sprofile);
function [alpha]=MCF_BL0_cl;
function [alpha]=MCF_BL0_sl;
function [alpha]=MCF_BL0_cp;
function [alpha]=MCF_BL0_sp;
function [s]=MCF_func_steady(x);
```

**Several useful remarks**

The present implementation of the MCF approach has been checked by independent Monte Carlo simulations. Of course, such a numerical test cannot ensure its validity in any situation. It may happen that experimental data do not follow numerical predictions. In this case, four different explanations may be suggested:

– conceptual error or misprint in the MCF approach;

– technical error in the numerical implementation of the MCF approach;

– wrong use of the MCFAL;

– physical artifact in experimental measurements.

The author has certainly tried to avoid the first two causes by numerous tests. If there is a doubt in the functioning of the MCF approach or its numerical implementation, it is suggested to contact the author. The third reason can be related, for instance, to a wrong order of input parameters or non SI-units. It is strongly recommended to read carefully the description of the functions. At last, we should stress that the MCF approach is based on a classical description of restricted diffusion. It is merely a model that *may or may not* be valid for specific experimental conditions. We mention only a few possible deviations from this model: convectional transport, anisotropic diffusivity of the medium[5], bulk relaxation, susceptibility effects, hardware imperfections, residual magnetic field gradients, spin interactions, etc.

Please, do not try to "fake" the functions of MCFAL using deliberately wrong or unphysical parameters. This is not commercial software, there is only a few checks for appropriateness of the input parameters.

For the localization regime ($q>>1$), it is suggested to perform the numerical computation for slab geometry *only*. In the case of a cylinder and a sphere, the truncation of matrices $B$ and $\Lambda$ to maximum available sizes $60\times60$ may be insufficient.[6] This artificial limitation should be suppressed in the following release of the MCFAL.

[5] Please, do not confuse anisotropic diffusivity (dependence of the "free" diffusion coefficient on the coordinates) with anisotropy of the confining domain. The last one is captured by the MCF approach.

[6] There are two reasons to prefer slab geometry. First, there is no limitation to the size of matrices $B$ and $\Lambda$. The second and more important reason is that the Laplace operator eigenvalues $\lambda_m$ increase as $m^2$ for slab geometry, and roughly as m for a cylinder and a sphere. If an accurate computation for slab geometry requires the matrices $B$ and $\Lambda$ of size, for example, $30\times30$, the similar computation for a cylinder and a sphere would require matrices of size around $900\times900$. This technical problem appears only for the localization regime ($q>>1$) since the "oscillating" part $iqB$ in $\exp[-(p\Lambda+iqB)]$ should be dumped by "decreasing" part $p\Lambda$.