Highly Flexible and Modular Simulation Framework for Magnetic Particle Imaging

P. Vogel, M.A. Rückert, T. Kampf, V.C. Behr

1 Department of Experimental Physics 5 (Biophysics), University of Würzburg, 97074, Würzburg, Germany.
2 Department of Diagnostic and Interventional Neuroradiology, University Hospital Würzburg, 97080, Würzburg, Germany.

* Corresponding author, email: Patrick.Vogel@physik.uni-wuerzburg.de

Abstract:
Simulations with high accuracy are an essential part of scientific research to accelerate the innovation process. They are especially useful for finding novel approaches or optimizing existing methods. Today, powerful software tools are available consisting of multiple packages with a wide variety of features, methods, and models meet the requirements for different questions in multiple fields of research. Unfortunately, the complexity and often inflexibility of such unspecific software tools can hinder an optimal workflow. Especially in the case of a novel research fields, e.g., Magnetic Particle Imaging (MPI), the requirements on a software tool are a high degree of flexibility paired with a manageable number of highly specific features to provide fast and easy access. Thus, often research sites generates their own software solution to address their specific demands.

Until now, only few simulation frameworks are available, which partially fulfil most requirements of the young field of MPI. However, the coverage of the entire process of emulating a full MPI experiment from magnetic field and particle dynamic simulations, hardware and sequence programming over signal generation and data processing to final reconstruction and visualization combined with an easy-to-use graphical user interface (GUI) and without the need of complex combination of different software packages cannot be found in the literature or MPI community at the time of this writing.

With the presented modular simulation framework consisting of multiple interconnected software packages to specific purposes, all necessary simulation steps are provided. The high degree of flexibility and modularity allow the simulation and emulation of almost any kind of MPI scanners known in the MPI community. Furthermore, the modular framework allows an easy connection of third-party software using dedicated interfaces between important steps.

Keywords: Magnetic Particle Imaging, MNP, simulation, hardware, software, reconstruction, real-time visualization

I. Introduction
Since the simulation and emulation as well as optimization of signals, hardware parts, processes and many more are an essential part in different fields of research and industry, a huge variety of software solutions for different research problems are available. In theory, almost any imaginable problem can be implemented, simulated, and emulated with the right software tool.

One major drawback is the complexity of such highly flexible software packages covering multiple parts of the entire process, e.g., from the simulation of novel hardware design of imaging devices to the final reconstructed image. This is a result of the different requirements in each processing step, that have to be handled individually.

For Magnetic Particle Imaging (MPI) [1], there are several simulation studies published, which use commercial software tools, but there are also specific software packages available [2-10] as well as frameworks for specific types of MPI scanners or open-source projects for educational purposes [11, 12]. Each software solution solves one or more specific questions in a small area. Unfortunately, these tools are often difficult adapt to the requirements at other research sites, which typically triggers the creation of another specific software tool. However, to overcome this issue, a modular framework offering a highly flexible workflow allowing the simulation and emulation of almost any question in the field of MPI is required. Open-source projects often provide open libraries consisting of a variety of functions and procedures to solve multiple problems, but the use of those libraries often requires additional frameworks or third-party software to execute which hampers its usability.

One solution is a software environment encapsulating all required libraries but with an easy-to-use graphical user interface. To ensure performance, the usage of a high-level programming language is mandatory, providing the highest flexibility but also the highest complexity to build a problem-specific software package.
I.I. Brief introduction to MPI
Magnetic Particle Imaging (MPI) is a young tomographic technology [1], which has shown significant improvements in the last decade in hardware design and development as well as reconstruction methods [13]. MPI was successfully in different fields of research [14, 15] and makes it a promising candidate for future applications in medicine [16, 17]. MPI relies on the nonlinear magnetization response of magnetic nanoparticles (MNPs) to time-varying magnetic fields. For encoding, a strong gradient is generated and moved along specific trajectories through the volume of interest to determine the spatial distribution of the MNPs (tracer).

Several encoding schemes, such as field-free point (FFP) or field-free line (FFL) encoding, have been presented in the past and lead to multiple different designs and approaches for the generation of the required magnetic fields with more or less complex coil structures [13, 16, 18].

The signal acquisition in MPI devices is mostly done by inductive measurement of the MNP magnetization over time.

The induction voltage $u(t)$ within an electromagnetic receive coil (rx) with coil sensitivity $p$ of a spatially distributed MNP ensemble with concentration $c$ inside a given field of view (FOV) $\Omega$ can be described by the MPI signal equation

$$u(t) = -\mu_0 \int_{\Omega} c(r) p(r) \cdot \frac{\partial}{\partial t} \bar{m}(H(r,t)r,t) \, d^3 r,$$

(1)

with $\mu_0$ as the vacuum permeability and $\bar{m}$ as the averaged magnetic moment of the MNP ensemble. The magnetic moment strongly depends on the external magnetic field $H(r,t)$, which makes it difficult to model this parameter appropriately [19]. To overcome this issue, the signal, or kernel of Equ. 1,

$$s(r, t) = -\mu_0 c(r) p(r) \cdot \frac{\partial}{\partial t} \bar{m}(r, t)$$

(2)

is measured or simulated [20, 21] at each position $r$ in the FOV during a time-consuming calibration resulting in a discrete transformation matrix (system matrix) $\hat{S}$, which transforms the spatial distribution of the MNPs into the corresponding signal acquired with the very system the matrix is designed for. The linear system

$$u = \hat{S} \cdot c$$

(3)

consists of the transfer function $\hat{S}$, the induction voltage $u$ and the desired particle distribution $c$ and can be solved by different approaches, e.g., iterative solvers or directly via matrix inversion to determine the spatial distribution of the MNP concentration $c$ [22].

The advantage of the system matrix approach is the high accuracy since all information, e.g., the specific field distortions of the scanner, the MNP characteristics and the receive chain dependent signal distortions, are considered and stored in the system matrix. On the other hand, it is highly inflexible, since even slight changes make a new calibration necessary [23].

Another reconstruction approach is the x-space reconstruction [23-29], which directly works with data in the time domain instead of the Fourier domain. For image reconstruction, the data are directly gridded at their corresponding spatial position on a 2D surface or in a 3D space resulting in a raw-image, where the MNP distribution is convolved by the system-specific point-spread function (PSF) [22, 27].

The advantage of x-space reconstruction is its high flexibility, simple implementation, and short calculation times, but slight distortions due to inaccurate magnetic fields can cause artifacts [23].

However, there are many methods and approaches to overcome these issues, e.g., model-based system matrix, image-based system matrix, hybrid reconstruction approaches, single harmonic methods and many more [21, 30-34].

I.II. Requirements for MPI simulations
The entire process for an adequate simulation of, e.g., a complete MPI device with the goal to emulate the reality as well as possible, requires multiple sophisticated steps. In the past, many software packages have been presented fulfilling specific requirements and offering appropriate simulation and/or emulation results for the specific application [2-12]. However, often the flexibility is limited, and the handling is complex due to the absence of a graphical user interface (GUI). For a complete emulation of an MPI system, multiple features should be provided by the framework.
The **basic workflow** is listed in the following:

- **Magnetic Field Calculator**
  Based on defined coil structures, the magnetic field is calculated for selected points in space. Since the frequency range is below 10 MHz, a Biot-Savart calculator [35] can be used. This simplifies the implementation and reduces the computational effort – even more for multiple points and long sequences (parallelization).

- **Magnetic Nanoparticle (Magnetization) Models**
  For the generation of a realistic behavior of MNP magnetization in (time-varying) magnetic fields, several models are used: From simple single particle model (SPM, Langevin-function) over more realistic models with stochastic terms (Langevin equation) to PPI (particle-particle-interaction) approaches. Depending on the topic and available calculation times highly sophisticated and complex MNP signals can be generated [36-40].

- **Data Acquisition**
  The change in magnetization is measured inductively in real MPI devices, thus the framework offers an induction calculator for custom receive coils following the reciprocity theorem for arbitrary geometries known from NMR [41, 42] (see Sec. II.I.II.).

- **Data Processing**
  Real signals are affected by noise and distortions in amplitude and phase, e.g., due to hardware limitations. Therefore, digital filtering and data processing is essential for signal cleaning and preparation for further steps [7].

- **Reconstruction**
  After data preparation, several reconstruction methods (gridding, deconvolution, system matrix, etc.) can be applied to visualize the data [23].

- **Visualization**
  Appropriate data visualization is essential for analyzing. This often requires sophisticated tools, e.g., for 3D volume rendering [8, 24].

In addition to the listed basic features a simulation framework for MPI should provide, user experience will be improved by:

- **Operation and Handling**
  A fast and intuitive workflow can be essential for innovative solutions and/or optimizing common concepts. An easy-to-use 3D-GUI as well as near real-time visualization and data preview are essential features for that.

- **Adaption and Flexibility**
  Closed all-in-one frameworks often provide good press-button solutions. In novel fields it is more important to replace parts of a data processing pipeline for a more optimized workflow. For that, a modular framework consisting of several packages can be advantageous. Standardized and open interfaces should be available to transfer data between the packages.
  For higher flexibility, a propriety scripting language allows the direct access to functions for data processing, signal generation, etc. and allows automatization of complex processes (see Sec. II.I.VII. Scripting tool). With cross-platform capability, the software packages are available on multiple platforms and operating systems.

- **Response, Speed and Maintenance**
  For optimal performance, a high-level programming language permanently maintained over long times and providing embedded compilers for multiple operating systems is required, e.g., Rad Studio (Embarcadero, USA). Using optimized libraries, e.g., FFTW [43] for fast Fourier transformation, LABLAS or LAPACK [44] for fast matrix operations, the fastest possible data processing can be provided for maximum responsiveness of the GUI. In addition, parallelized implementation of parts of the source code and pre-calculation steps (see Sec. II.I.VII. pre-calculation) can dramatically increase performance.

II. Material and methods

The proposed MPI simulation framework consists of three different software packages (see Fig. 1), which can be used individually or together via specific interfaces, e.g., memory mapped files (MMF):

- **Magnetic Field Simulator (MFS)**
  Main simulation environment for the generation of MNP signals of arbitrary magnetic field configurations (Fig. 1a).

- **Reconstruction Framework (RiFe)**
  Framework optimized for fast data processing and near real-time image reconstruction (Fig. 1b) [7].

- **3D Visualization Tool (3DVT)**
  Software package optimized for visualization of 3D data using volumetric rendering (Fig. 1c).
Each software module is programmed in Delphi® (Embarcadero RAD Studio 11 Alexandria, USA), a general-purpose programming language providing an integrated development environment (IDE) for rapid application development on multiple platforms and operating systems.

For data transfer between the software modules, either specific data formats (time data, Fourier data, image data, system-matrix data, etc.) can be used or a fast data exchange through Memory Mapped File technology (MMF) is available.

All modules are user-friendly designed (3D graphical user interface, etc.) and optimized for fast data processing (Magnetic Field Simulator) and for near real-time visualization with low latency (Reconstruction Framework and 3D Visualization Tool). The latter feature in combination with specific data transfer protocols (e.g., Teledyne LeCroy, USA or Tektronix, USA) allows the near real-time reconstruction and visualization of data streams obtained on real MPI scanners (Fig. 1d) [7, 24, 45, 46].

II.I. Magnetic Field Simulator – MFS

The MFS tool is a versatile and flexible simulation environment for magnetic fields in 4D [5]. A 3D-GUI with intuitive controlling (mouse and/or multi-touch) provides a fast and easy navigation around the designed structures. Standard features known from versatile 3D software are available, e.g., specific lighting or different rendering settings.

The MFS software is structured in different types of helper-containers with different assignments. This allows a standardization and thus optimization of data processing.

II.I.I. Conductor-container

For the simulation of quasi-static and dynamic magnetic fields with frequencies below 10 MHz, the law of Biot-Savart is valid, which is sufficient for common MPI experiments [35]

$$d\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} i d\mathbf{l} \times \frac{\mathbf{r} - \mathbf{r}'}{|\mathbf{r} - \mathbf{r}'|^3},$$

with $d\mathbf{B}(\mathbf{r})$ being the magnetic field at position $\mathbf{r}$ in 3D space generated by a flexible current $i$ due to a differential element $d\mathbf{l}$ at position $\mathbf{r}'$ (see Fig. 2 top left) and $\mu_0$ as the vacuum permeability.

Based on this differential concept, arbitrary coil designs for magnetic field generation can be generated element by element (see Fig. 2). Parameterized function-based templates, e.g., from simple single loops over saddle-coil pairs of canted-cosine-theta (CCT) coils to toroidal structures, or sophisticated structured assemblies, e.g., as spherical arrangement are available [47]. Each conductor-container consists of an array of size $N_{el}$ with all element information and a transformation matrix $\tilde{\mathbf{M}}$ (see Sec. VI.S1)

$$\text{conductor}(\tilde{\mathbf{M}}, N_{el}) = \{\tilde{\mathbf{M}}, \{\mathbf{r}, d\mathbf{l}_1\}, \{\mathbf{r}, d\mathbf{l}_2\}, \ldots \{\mathbf{r}, d\mathbf{l}_{N_{el}}\}\}.$$
Fig. 2: **Top left**: magnetic field calculation using Biot–Savart law: for each element $dl$ within a conductor array, the partial magnetic field $dB$ at a specific position $r$ can be calculated. **Top**: With this differential concept, arbitrary structures can be generated element-wise, e.g., solenoidal, toroidal, multipole CCT, saddle coils or free-form shapes. **Bottom**: Additional features such as bending or twisted offers the generation of complex designs.

Standard features, such as translation, rotation and scaling, but also special functions, such as bending or twisting, can be applied to the conductor elements to modify position and shape and thus the resulting magnetic field (Fig. 2 bottom). In addition, for more individual structures, a tool is available for hand-free drawing of conductor structures. This can be helpful for rapid prototyping structures and/or also for educational purposes. By combining multiple conductors to one single element (conductor network list – cnl-element), complex structures can also be stored and used in other projects.

Each conductor can be used in two basic modes: as **transmit coil (tx)** or as **receive coil (rx)**. In **tx mode**, the conductor exclusively serves as magnetic field generator and generates a magnetic field following Equ. 4 & 6 at selected positions with a given arbitrary time-dependent current sequence $I_m(t)$. At each position $r_I$, the magnetic field $B(r_I, t)$ generated by all transmit conductors $N_{cond}$ each consisting of $N_{elm}$ single elements is calculated for every time step $t$ (see Sec. II.VII.)

$$B(r_I, t) = \frac{\mu_0}{4\pi} \sum_{m=1}^{N_{cond}} \sum_{n=1}^{N_{elm}} I_m(t) dl_{m,n} \times \frac{r_I - r_{m,n}}{|r_I - r_{m,n}|^3}.$$  \hspace{1cm} (6)

In **rx-mode**, this conductor is connected over a magnetic nanoparticle array (see Sec. II.I.IV.) to a magnetic field array (see Sec. II.I.IV.) collecting magnetic field and particle information (magnetization) generated by the transmit conductors and particle system at these points.

The dependency graph in Fig. 3 shows a simple MFA, a slice-container and an rx coil. The transmit coils are used for magnetic field calculation at the specific points represented by the MFA-container. According to their function, i.e., slice-container or receive conductor, further data processing is performed either for visualization of the magnetic fields or calculating the induction signal for a given receive coil and a given MNP ensemble.
Fig. 3: The MFA-container is the central part of the simulation. Depending on their function, e.g., MFA visualization, slice visualization or signal generation for a receive coil, further processing steps are performed.

II.I.II. MFA-container

MFA stands for magnetic field array and represents an ensemble of $N$ dedicated points $r'$ in 3D space. The MFA is a container, which can be used as a modular computation unit of all parameters required for calculation and visualization of magnetic fields from different sources. For each position $r'$ several additional information is stored, such as the latest magnetic field vector $B$, the previous magnetic field vector $B'$, a pick-up table $\tilde{C}$ consisting of information about associated conductors

$$\text{MFA}[N] = \left\{ \{r', B, B', \tilde{C}\}_1, \{r', B, B', \tilde{C}\}_2, \ldots, \{r', B, B', \tilde{C}\}_N \right\}. \hspace{1cm} (7)$$

The MFA-container has an optimized data structure for fast data processing and calculation, as well as interfaces for data visualization. E.g., several parts of the entire process are parallelized and can be pre-calculated in the background to ensure rapid data generation and highly responsive interaction experience of the user (see Sec. II.I.VII.).

Fig. 4: Different methods for MFA visualization: (a)-(c) several types of magnetic field vector length rendering, displayed as unit vectors (d) or with specific thresholding (e) in 2D or 3D (g) & (h). Using marching cube algorithm, isosurfacing can be used for detailed visualization of magnetic field geometries.
For visualization of MFAs, the framework offers different methods as indicated in Fig. 4. From simple magnetic field vector renderings (Fig. 3 a to d) to solid renderings (iso-surfacing) with specific constraints, e.g., thresholding or unit vectors, in 2D or 3D (Fig. 4 e to i). In addition to magnetic field vector information, the visualization of magnetic field gradients ($\frac{dB}{dr}$ or $\frac{dB}{dt}$) or special information about the magnetic field behavior over time, e.g., circular, elliptic or linear magnetic fields, is available.

II.III. Slice-container
A specific type of the MFA-type is the slice-container, which is a 2D plane with a given size (width and height). It can be positioned and rotated arbitrarily in the 3D space to visualize, e.g., the magnetic field or gradient with a given number of voxels $N_x$ and $N_y$ spanned by the plane vectors $u$ and $v$. The slice-container consists of an MFA-container and additional parameters, e.g., affine transformation matrix $\mathbf{M}$ (see Sec. VI.S1) and more (PAR)

$$\text{slice}(\mathbf{M}, N_x, N_y) = \{ \mathbf{M}, u, v, MFA[N_x \cdot N_y], PAR \}. \quad (8)$$

In Fig. 5 several different visualization methods are shown indicating various interesting parameters of static and dynamic magnetic fields generated a given conductor array. Fig. 5b&c shows an example of two perpendicular solenoids driven by the same sinusoidal current but with a phase shift of 90° generating a rotating magnetic field. While Fig. 5b shows the absolute values of the magnetic field vector, Fig. 5c shows: the magnetic field gradient $\frac{dB}{dr}$ (top left), the magnetic field derivative $\frac{dB}{dt}$ (top middle), absolute values of phase (top right), absolute phase values along x-direction (bottom left), specific contour plot (bottom middle) and special magnetic field parameters, e.g., circular behavior over time (bottom right). Fig. 5d-e shows the absolute magnetic field of a static spiral-shaped conductor. With profile plots more specific information can be extracted (Fig. 5f). In Fig. 5g-i a contour plot visualization with increasing slice resolution shows the increasing image quality at the expense of prolonged calculation times.

Fig. 5: (a) Sketch of a slice-container with specific size (width and height) positioned in 3D space with $p$ as support vector and $u$ and $v$ as span vectors. (b) The absolute magnetic field values at a point in time of two perpendicular solenoids running with a phase shift of 90° generating a rotating magnetic field. (c) Visualization of different parameters: top left: magnetic field gradient $\frac{dB}{dr}$, top middle: magnetic field derivative $\frac{dB}{dt}$, top right: absolute phase values, bottom left: absolute phase values along x-direction, bottom middle: specific contour plot, bottom right: magnetic field circulation behavior over time. Absolute magnetic field values of a spiral-shaped conductor with different slice-resolutions: 51×51 px² (d) and 501×501 px² (e). (f) 2D profile plots along the center shows more information about the magnetic field. (g)-(i) Contour plots for different slice sizes (51×51 px², 501×501 px², 2001×2001 px²).
II.I.IV. MNP-container

For MPI simulations, a mandatory feature is the particle ensembles (MNPs) and their behavior as well as their spatial distribution. Like the slice-container, the MNP-container consists of an ensemble of multiple discrete points in space representing the positions of particle ensembles \( \{\vec{M}, \text{MFA}[N]\} \) following a specific magnetization model \( \vec{m} \)

\[
\text{MNP}(\vec{M}, N) = \{\vec{M}, \vec{m}, \text{MFA}[N], \text{PAR}\}.
\]  

(9)

For calculating the nonlinear behavior of the magnetization \( \vec{m} \) of MNP ensembles to a time-varying magnetic field \( \vec{H}(r, t) \) multiple models are available [37-40] The mostly used model in the MPI community is the Langevin function \( L(\xi) \) (Eqn. 10) with \( \mu_0 \) as the vacuum permeability, \( k_B \) as the Boltzmann constant and \( T \) as temperature. With the Langevin parameter \( \xi \) this model describes paramagnets in external magnetic fields semi-classically [37]

\[
L(\xi) = \coth(\xi) - \frac{1}{\xi} \text{ with } \xi = \frac{\mu_0 m H}{k_B T}.
\]  

(10)

The entire magnetization \( \vec{m} \) of the ensemble can be determined by \( \vec{m} = N m L(\xi) \) with \( N \) as the number of particles within the volume elements of the simulation. This model follows the single particle model (SPM), which assumes, that the magnetic moments of the particles follow the external magnetic field instantaneously [37, 48]. This makes it easy to implement and quite fast.

The second model is a naive implementation of MNP relaxation using a lag-effect (Eqn 11). For that, the magnetization of the MNPs is interpolated from the current magnetization vector and the last \( N \) magnetization vectors. This results in a relaxation type, which requires less computational effort as the third one

\[
m_t = \frac{1}{N + 1} \sum_{i=0}^{N} m_{t-i},
\]  

(11)

A third model follows the Langevin equation (Eqn. 12), which overcomes limitations of the Langevin function approximation, since it considers the dependency of the magnetic response on viscosity of the suspending liquid and changes in the particle’s hydrodynamic diameter [49, 50]

\[
\frac{dm}{dt} = \frac{1}{\xi} (m \times H) \times m + \frac{2k_B T}{\xi} \lambda \times m,
\]  

(12)

with \( \lambda \) as normally distributed random vectors with expectation value \( \langle \lambda_1(t) \rangle = 0 \) and \( \zeta = \kappa \eta R^3 \) as the Stokes-Einstein diffusion coefficient (viscosity-coefficient) which depends on the viscosity \( \eta \) of the surrounding liquid, the particle diameter \( R \) and a shape factor \( \kappa \). The Langevin equation consists of a deterministic part and a stochastic part. This means that for the calculation of the magnetization multiple single magnetic moments have to be tracked over time. This results in a more complex implementation and requires more computational effort. This model is also only valid for particles that show no Néel relaxation, i.e., the magnetization is fixed inside the particle.

In Fig. 6 the magnetization for three different models is shown in a simple experiment with parallel transmit and receive coils. A rectangular shaped current pulse generates a strong magnetic field. The Langevin function instantaneously follows the shape of the provided current. A naive relaxation model, which simply interpolates the last \( N \) magnetization vectors, shows a symmetric relaxation behavior [50]. The Langevin equation model, here calculated from 50 \( k \) single particles, shows a more realistic relaxation. The calculation times for the Langevin function model and the Langevin equation model differ by four orders of magnitude.
Fig. 6: Plot of different magnetization models demonstrating the behavior of MNPs during a rectangular shaped magnetic field pulse. The Langevin function follows instantaneously the external magnetic field, the naïve relaxation model shows a symmetric magnetization behavior for induced magnetization and free relaxation and the magnetization following the Langevin equation shows the most realistic behavior [50].

-Alternative magnetization calculator – dynamic Bloch solver
Furthermore, the magnetization response of a spin ensemble or isochromate in dynamic magnetic fields can be calculated using a dynamic Bloch-solver [51].

\[
\frac{dm}{dt} = \gamma m \times H - \frac{m_\perp}{T_2} - \frac{m_\parallel}{T_1}
\]

with \( m = m_\perp + m_\parallel \), \( m_\parallel \parallel H \), \( m_\perp \perp H \) \hspace{1cm} (13)

This allows not only the simulation of NMR (nuclear magnetic resonance) or MRI (magnetic resonance imaging) signals with static \( B_0 \) fields, but also the magnetization response on arbitrarily magnetic field dynamics.

II.I.V. Receive coil-container
The receive coil-container is a sub-structure of the conductor-container consisting of an additional MNP-container (see Fig. 3)

\[
rx(\hat{M}, N_{el}, N) = \left\{ \text{conductor} (\hat{M}, N_{el}), \text{MNP} (\hat{M}, N), \{p_1, p_2, \ldots, p_N\} \right\}. \hspace{1cm} (14)
\]

This MNP-container, or better the dependent MFA-container serves as array, which defines the position of sensitivity map or points \( p \) (cf. Eqn. 1) of the selected coil. Following the NMR reciprocity theorem, which state that the magnetic field generated by a coil at specific points in space are equivalent to its sensitivity in the same spot [41, 42], the calculation of the induced voltage \( u(t) \) in this receive coil can be performed by summing up the magnetization changes over time \( \frac{d}{dt} \hat{m}(r, t) \) in every point generated by a transmit coil (Faraday’s law).

Fig. 7 shows the projection calculation for a single point in space defined by the MNP-container. The magnetic field \( B \) generated by the transmit coil is used to calculate the magnetization \( m \) at this point following the selected magnetization model (MNP-container). The induced voltage \( u \) is the change of the projected magnetization \( m \) on the sensitivity map \( p \).

Instead of the induced voltage, also the magnetization can be directly extracted as signal as well as the vectorial signal of the magnetic field generated of the transmit coils.
**II.I.VI. 3D model-container**

This container is a passive container used for visualization of 3D models as well as for calculation of MNP distributions based on 3D volumes. The container can import stl files (standard triangle language), which is a standard file format of various 3D software, e.g., for rapid prototyping, 3D printing and computer-aided manufacturing. It uses lists of triangles to describe the surface of the 3D model, which can be rendered as triangle mesh (see Fig. 8 a).

For calculation of 3D points, that are located inside the volume of the 3D model, a slicing algorithm (slicer) is used, which is well known in 3D printing. Based on a given start orientation multiple parallel 2D slices are generated at a given spacing. For each slice, which serves as 2D gridding field with given distances, the intersection with the 3D model is calculated in form of a closed 2D contour. Using a point-in-polygon algorithm, all 2D points on this slice lying inside the contour can be determined as indicated in Fig. 8 b. These 3D points can be transferred to an MNP-container, where the sample emulation can be used as MNP distribution with specific magnetization models.

**II.I.VII. Calculation process**

An important feature of any simulation software with a GUI is the response time, which means the update time, between consecutive simulation time steps to ensure a fluid presentation and visualization of dynamics and results, e.g., time-varying magnetic fields. Furthermore, an optimized calculation process is important for fast generation of huge data sets, which can be used for further processing.

In the MFS software, several calculation methods are available, which separate independent steps allowing to update them on demand only or combine similar steps to optimize calculation time through parallelization.

For each container type an individual update-function is implemented, which allows to perform necessary updates selectively:
• **Update conductor-container (conductorlist)**
  This procedure updates on demand all parameters of each conductor, such as position, rotation, scaling (affine transformation matrix values), shape, type, etc. Depending on this, a pre-calculation process is started to calculate values and parameter, which have not necessarily to be updated during each time step, such as affine transformation matrix values, currents, etc. and/or dependent MNP/MFA-containers (see Sec. II.I.VII.). Additional pre-calculation are performed for generating the triangle mesh for 3D visualization.

• **Update receive coil-container (receive-coils)**
  A special sub-calculation process is the calculation of receive coils. If a conductor-container is defined as a receive coil, an additional pre-calculation step is performed: a sensitivity map (sens-maps) is created for each receive coil based on the dependent MNP-container, which is mandatory for fast calculation of signals over long time (data sets).

• **Update MNP-container (mnplist)**
  Depending on the selected magnetization type additional pre-calculation to the affine transformation matrix values of connected MFA-container are mandatory, such as initialization of random vectors. Additional pre-calculation are performed for 3D visualization.

• **Update slice-container (slicelist)**
  Updating the slice-container means triggering updating the connected MFA-container with new vector positions and directions required for 3D visualization.

• **Update MFA-container (mfalist)**
  The main function of the MFA-container is the management of the magnetic field vectors at dedicated positions required for MNP-container, slice-container, and receive coil-container. Thus, the update is often triggered by these containers. In addition, the MFA-container is used for 3D visualization of magnetic field parameters, which requires among other the pre-calculation of triangle-mesh models for visualization of 3D arrows.

- **Generation of data sets**
  This special procedure works independently of the above-mentioned calculation-for-visualization process. For the generation of data sets consisting of multiple time steps, an encapsulated data processing step has been implemented, which provides several methods for time step calculation, especially for parallelization:

1. The first parallelization method is used for the generation of a single data set with multiple MNP-containers connected with MFA-containers. For each individual MFA-entry (3D point) a separate thread is generated calculating the full data set coming from this specific point in space. At the end, all single MNP data sets are summed up.

2. The second parallelization method is similar to the first one and used for generating multiple data sets at once. This is useful for the calculation of system matrices, which can consist of multiple positions in space where a data set has been calculated for a point-like sample. The created data sets can be stored separately or further processed to create the final system matrix.

3. The third parallelization method is used for time-dependent calculation processes like flow simulations or particle-particle-interaction (PPI) simulations, and tries to parallelize for each time point all required data of conductors, MNPs, MFAs, etc.

4. A fourth parallelization method has been implemented within the calculation step of Langevin-equation. Since for each time step multiple single magnetization vectors $N_{\text{stoch}}$ are used for calculating the stochastic part of the signal, it can be useful to parallelize these calculations by dividing all vectors in $M$ threads consisting of $N_{\text{stoch}}/M$ vectors each. This method can be combined with any other method.

There are additional parts of the source code, where parallelization can be implemented, e.g., the pre-calculation steps. All threads are monitored and managed by a thread-pool, which is specifically set up for the dedicated hardware, the software is running on.

The generated data sets can be further processed within the MFS software (see Sec. II.I.VII. scripting tool), directly stored in a proprietary data format (*.mpi & *.dbl – see Sec. VI.S2) or provided to other software packages, e.g., to RiFe (Sec. II.II.), via fast MMF interface.

- **Pre-calculation**
  Imagine a conductor structure with $N_{\text{cond}}$ conductors with $N_{\text{el}}$ elements each, which is positioned and oriented arbitrarily in space represented by matrix $\tilde{M}$. For calculation of the magnetic field $\mathbf{B}(\mathbf{r}_i, t)$ at $N$ specific points in space $\mathbf{r}_1$ and time $t$ following Equ. 6, the position $\mathbf{r}_{m,n}$ as well as the direction $\mathbf{d}l_{m,n}$ of each single element must be calculated. A naive estimation results in the requirement of $4 \cdot 4 = 16$ multiplications and $4 \cdot 3 = 12$ additions per vector. In total, $N \cdot N_{\text{cond}} \cdot N_{\text{el,m}} \cdot 2 \cdot 16$ multiplications and $N \cdot N_{\text{cond}} \cdot N_{\text{el,m}} \cdot 2 \cdot 12$ additions are required. Since
each mathematical operation requires computational time for calculation (addition requires typically 2 cycles and multiplication 7 cycles [52]), \( N \cdot N_{\text{cond}} \cdot N_{\text{el,lm}} \cdot 272 \) cycles are required per time point. This does not sound like much in times of GHz processor, but for large datasets, e.g., with length \( 2 \cdot 10^6 \), a 3D volume consisting of \( 10 \times 10 \times 10 \) points in space and 10 conductors with 100 single elements each, at the end \( 272 \cdot 10^{12} \) cycles are required. Often, there is no need to perform all calculations at each time step. E.g., when using spatially fixed conductors, the matrix calculations mentioned above only have be performed once and can be used for all time steps. This pre-calculation step checks for several processes whether a new calculation is required or not. If so, then the data are calculated and stored in separate pickup lists, which are updated on demand, to be used in further calculations. The mentioned example with pre-calculating the 3D points for different tasks requires the major part in computation time in the software and is used in different situations, e.g., position and direction vector calculations, determining the sensitivity map or preparing a current look-up table over a dataset with given data length.

Further time-saving pre-calculation can be made for calculations of Fourier transforms by pre-calculating plans for specific data length [43].

**-scripting tool**

A built-in scripting tool allows to run customizable scripts for automatization within the software, which can be used for optimization, visualization, and demonstration of magnetic fields and MPI or MRI signals. All parameters of the containers, e.g., changing the positioning or particle properties, can be manipulated within the scripting area. Furthermore, signal processing commands, such as high-, low- or bandpass filtering, Fourier or Hilbert transformations, mathematical operations, or reconstruction commands, such as gridding processes are available to process data sets. The scripts can be stored in a proprietary format (see Sec. VI.S2).

**II.I.VIII. Useful MPI specific features**

As mentioned at the beginning, there are multiple software frameworks available, which can be used for calculation and visualization of magnetic fields. However, especially specific questions, e.g., calculating system matrices or calculating the magnetization response of dedicated magnetic materials within a given receive coil, often are difficult to implement. Specific modes implemented within the MFS software allow the visualization of the magnetic fields and gradients and provides a highly responsive environment to investigate different feature, such as FFP or FFL surface rendering and 3D tracking, etc.

In the following, a selection of MFS features especially designed for MPI is shown:

- **Calculation of non-linear magnetization response**
  The basic concept of MPI is the non-linear magnetization response of magnetic material to time varying magnetic fields [1]. Thus, a basic feature is an intrinsic and fast calculation for signals of that kind. In Fig. 9, time signals and hysteresis curves for different kind of MNP magnetization types (see Sec. II.I.IV) are shown, from pure Langevin function particles without Hysteresis to signal simulations with relaxation times (Langevin equation).

![Fig. 9: Left: Time signals showing the derivative of non-linear magnetization responses for different magnetization algorithms. Right: the corresponding hysteresis curves.](image-url)
• **Calculation of system matrices**

System matrices can be used for data reconstruction in MPI. Especially for real MPI scanners, a point-by-point acquisition of the corresponding system matrix is useful for black-box reconstruction. Since the acquisition time of real system matrices with high spatial resolution using a robot is quite long [1], hybrid system matrices using appropriate simulation models as well as pure model-based system matrices [21] are of high interest. With a highly optimized data processing, e.g., parallelization, the calculation time for huge system matrices can be dramatically reduced allowing fully automatic parameter optimization with short feedback times. For fast calculation within the MFS software, there is a specific function implemented providing calculation of time signals, Fourier transformed signals as well as dedicated system matrices: the user defines the number of system matrix points by setting up an MNP-container with, e.g., 21×15×1 for a rectangular 2D slice or 21×21×21 for a cubic 3D volume. For each MNP entry, a separate data set is calculated using an optimized parallelization process.

In Fig. 10, an example is shown for generating a 2D system matrix for real-time visualization with RiFe. First, the size of the desired system matrix area is set (MNP-container – 25×25). Afterwards, the system matrix (ARSM – arbitrary system matrix) is generated with the specific parameters the MPI scanner is operating. Using an interactive script (see Sec. II.I.VII. scripting tool), the system function can be visualized and investigated before using the system matrix for image reconstruction (see supplementary video 2: system matrix).

![Fig. 10: Process of generating a system matrix for virtual Gleich/Weizenecker-MPI scanner [1]: Top left: the area the FFP can cover is visualized with a slice-container and is used for setting up the system matrix size using an MNP-container with 25×25 entries and a spatial resolution of 0.50×0.25 mm² (top middle). Top right: setting up the system matrix type with dedicated information about the scanners frequencies (here \(f_1=10\) kHz and \(f_2=11\) kHz). Using an interactive script, the system function can be investigated (bottom left & middle) before image reconstruction within the RiFe software using the generated system matrix (bottom right).](image)

• **Visualization of magnetic field parameters**

For understanding of given scanner approaches or for investigating novel encoding schemes, a highly flexible visualization portfolio of magnetic fields and magnetic field gradients are important (see Sec. II.I.III. & Sec. II.I.II.). In Fig. 11, several different magnetic field renderings of an FFL MPI scanner [53] are shown.
Fig. 11: Renderings of magnetic field configurations for an FFL MPI scanner [53]. (a) starting with four saddle coils creating an FFL along the symmetry axis over magnetic field vector visualization of the FFL for different current settings (b) to surface renderings of the FFL (c) rotating about the symmetry axis (d) and displacing along the x-y slice (e).

- **(Virtual) phantom generation**

  For investigation of imaging results or for comparison with realistic phantoms, it is useful to create virtual models of them. As described above (see Sec. II.I.VI.), an array of MNPs is calculated lying inside a given mesh model to obtain a filled 3D model with MPI generating signals, e.g., for testing realistic vessel structures within a virtual MPI scanner.

  In Fig. 12, an entire process of data extraction from 3D MRI or CT data sets over MPI simulation to measuring with realistic models is shown [54]. The desired structure has been extracted in a first step using a home-built framework providing an easy-to-use 3D GUI (Fig. 12 (1)+(2)) [55, 56]. The generated mesh mode can directly be used within the MFS software for virtual phantom generation (Fig. 12 (V1)+(V2)) and furthermore for creating real phantoms (Fig. 12 (R1)..(R4)), for example through 3D printing, coating, and washing [57]. Finally, both reconstructions can be compared (Fig. 12 (3)).

Fig. 12: Sketch of the processing steps for generation of virtual&real phantoms for comparison: starting with the data extraction of, e.g., desired 3D vessel structures resulting in a 3D mesh model, the stl-file can be directly imported in MFS software for further virtual signal simulation (V1&V2). By 3D printing the stl-file followed by additional processing steps, a realistic phantom can be measured in a real MPI scanner (R1..R4).

**II.II. Reconstruction Framework – RiFe**

The reconstruction framework (RiFe) is a software package, which provides a fast data processing of data sets either coming directly from an analog-digital converter like an MPI scanner or a simulation tool like MFS (see Sec. II.I.) via MMF or a proprietary data format (see Sec. VI.S2). It is optimized for latency times belows 100 ms between starting the sequence for collecting the data and visualizing them (see Fig. 13) [7]. This is mandatory for medical applications under real-time conditions, such as PTA or stenting [45, 46], to ensures fast visual feedback for the medical operator.
Fig. 13: Sketch of the important steps starting from pressing-the-button to the final reconstructed image. The latency time is the sum of all time lags of the corresponding steps [8].

RiFe offers fast data manipulation as required, i.e., for receive-chain correction (RCC) of real MPI scanners, or frequency peak picking and filtering as preparation step for multiple reconstruction methods, such as image-based (re-gridding, direct deconvolution, image-based system matrix) and Fourier-based peak picking system matrix reconstruction [23, 25, 28, 29].

Different algorithms are available for the reconstruction process (see Fig. 14): Wiener filter, Kaczmarz algorithm, singular-value decomposition (SVD), etc., all optimized for near real-time calculation. The system matrix data can be generated directly within the MFS environment (see Sec. II.I.) for multiple scanner and reconstruction types.

As indicated in Fig. 14, the entire reconstruction process consists of several parts:

- **Raw data**
  The raw data are generated either within the simulation software, e.g., MFS (see Sec. II.I.), or within a real MPI scanner and provided to the RiFe software.

- **Data correction**
  From dedicated signal filtering in time and Fourier domain to data correction with received correction data sets (receive chain correction – RCC) there are multiple implementations available optimized for real-time data processing.

- **Gridding**
  Based on the trajectory, the real (or virtual) MPI scanner has acquired the data, a raw-image or raw-volume can be generated (x-space).

- **Reconstruction**
  Using deconvolution algorithms, 2D raw images can be reconstructed using, e.g., Wiener filter, or dedicated system matrices (model based or measured) for iterative image reconstruction.
Fig. 14: Flow chart indicating the entire reconstruction process, which is separated into several sections: pre-calculation step, raw data step, correction step, gridding step, reconstruction step, and visualization step [7].

A built-in graphic tool allows the direct visualization of time-signal, Fourier transformed spectra, Hysteresis, peak-picking and tracking, 2D raw-images and 2D reconstructed images with data transfer rates up to 67 frames per second depending on the data size and reconstruction settings [7]. For 3D visualization, data can be transferred via MMF data transfer to the 3D visualization tool (Sec. II.III.).
II.III. 3D Visualization Tool – VT
Since the visualization of 2D data can be performed directly within the RiFe software, the visualization of 3D data requires a more sophisticated rendering. The 3D visualization tool is a fast OpenGL volume renderer for direct visualization of 3D data sets using shader technology [58]. As data pipelines, the open-source data format NiFTI [59] is used, or a direct data transfer via MMF from the reconstruction framework (see Sec. II.II.) is provided. For the latter data interface, additional information has been provided to map and visualize the 3D data correctly. Since the data stream is sent linearly, as a 1D array consisting of double values representing the brightness level, for each value the correct position in space must be provided, e.g., as an additional array of vectors. As indicated in Fig. 15, a 3D data set is reconstructed within the RiFe software (see Sec. II.I.) resulting in a 2D image. In this case, the 3D data are projected onto the 2D space, row-wise, where each row can be seen as slice [24]. With an appropriate coordinate look-up-table, the 2D data can be mapped in the 3D space [24, 60].

![Fig. 15: Example of a 3D data set reconstructed and projected onto a 2D image. As indicated, each row of the 2D image represents a slice in the 3D volume. Using an appropriate coordinate look-up-table the 2D data can be mapped in a 3D space [24].](image)

III. Results

III.I. 3D GUI MFS
Initially, a first overview of the features of the 3D GUI of the MFS software is given in Fig. 16, which shows several screenshots of the tutorial video (supplementary video 1: 3D-GUI). The fast and intuitive handling allows the creation and manipulation of different objects directly in 3D with fast feedback providing an easy-to-use environment for high flexibility and creativity.

![Fig. 16: Selected screenshots of a tutorial video for demonstrating the easy-to-use 3D GUI of the MFS software. Top: On the left, the starting window of the software is shown. Middle: after adding some solenoids, two (red) in Maxwell configuration creating a strong magnetic field gradient in their center, indicated as profile plot of the magnetic field (right), and additional two solenoid pairs in Helmholtz configuration for moving the resulting field free point (FFP) on a Lissajous trajectory through the FOV. Bottom left: the slice-container shows the stored trajectory. Bottom middle: an additional solenoid coil pair allows the displacement of the FFP along a 3D trajectory (FFP is rendered as blue 3D surface). Bottom right: with a receive coil (cyan), the signal can be rapidly calculated and showed as time-signal as well as Fourier transformed signal indicating the higher harmonics.](image)
III.II. Benchmark

The processes for calculating the magnetic field information have been optimized for a fast and intuitive user-experience with high response. For testing and quality demonstration, a benchmark has been implemented to test the performance of the MFS software on different platforms. The test consists of 12 solenoid conductors with 64 elements arranged as a dodecahedron each drive with a sinusoidal current. Using a single solenoid receive coil with 64 elements, a data set with 10k data points is generated of the 5569 MNP-container elements following the Langevin magnetization model (SPM). In Fig. 17, a screenshot of the test environment is shown, visualizing the magnetization at different time points. The results in Tab. 1 show the required calculation time on different machines.

![Fig. 17: Screenshot of the test environment consisting of 12 solenoids and 5569 MNP-container visualized at different time points. The areas with low magnetic field are shown as blue arrows.](image)

| CPU                              | st – calc. time [ms] (dl=1k) | mt – Calc. time [ms] (dl=1k) | mt – Calc. time [ms] (dl=10k) |
|----------------------------------|-------------------------------|------------------------------|-------------------------------|
| Intel i7-8650U (4 cores/8 threads) | 12533                         | 715                          | 7275                          |
| Intel i7-1065G7 (4 cores/8 threads) | 13299                         | 649                          | 6442                          |
| Intel i5-9500 (6 cores)          | 12944                         | 597                          | 5875                          |
| 2×Intel XEON E5-2670 (8 cores/16 threads) | 18588                         | 2003                         | 19195                         |

III.III. MFS projects

With the proposed software framework several different projects have been realized in the past, which are shortly introduced in the following. Starting with projects for educational purpose, e.g., to introduce different encoding schemes in MPI, additional tutorials are available to present the features of this software, especially the entire working process of the MFS+RiFe+VT framework. Finally, in the last section, a selection of scientific projects is shown, which are published in peer-reviewed journals.

- **Educational projects**
  One major goal of this framework was the visualization of magnetic fields for educational purpose. Especially the idea of MPI can be demonstrated in more detail as the selected projects demonstrated in the following.

  - **First MPI scanner à la Gleich&Weizenecker (G/W MPI)**
    In Fig. 18, an early simulation for a 2D MPI scanner à la Gleich and Weizenecker [1] is shown with an implemented 2D Lissajous trajectory scanning the FOV.
Fig. 18: Renderings of the first time steps showing the FFP moving along a 2D Lissajous trajectory through the FOV covering the ‘MPI’ logo. This concept was the starting point of MPI in 2005 [1].

First FFL MPI design
In 2008, Weizenecker et al. proposed a novel kind of encoding scheme for Magnetic Particle Imaging. Instead of using an FFP, a field free line (FFL) can be used to acquire projection information of the sample under dedicated angles [61]. But the suggested approach was technically too challenging for realization. Erbe et al. could demonstrate an enhancement of the coil configuration, which allows a realistic implementation [53]. In Fig. 19, a simulation of the initially proposed FFL scanner is shown, the enhanced system can be seen in Fig. 11.

Fig. 19: Simulation of the first approach of FFL MPI scanner proposed by Weizenecker et al. [61]. The yellow surface rendering represents the FFL. The concept was too complicated to realize and would require multiple times more power than an FFP scanner with an effective field of view of similar size.

Mech Halbach MPI scanner
The introduction of Halbach rings into MPI opened a door to new and promising scanner designs providing the advantage of generating strong magnetic field gradients without electrical power and thus heat dissipation [32, 62-65]. By constructing scanners composed of multiple different types of Halbach rings with individual axial rotations, different encoding schemes in form of FFPs or FFLs can be generated and moved through the FOV. A novel concept uses additional degrees of freedom, i.e., the poloidal and toroidal rotation of the Halbach rings to provide a higher flexibility. In Fig. 20, the renderings of the concept of such mechanically driven MPI scanner is shown.
Fig. 20: Renderings of the basic idea behind mechanically driven MPI scanner using Halbach rings. Top: poloidal and toroidal rotation of each magnet provides more degrees of freedom. Bottom: the combination of poloidal and toroidal rotated Halbach rings creates an FFP traveling along a 3D trajectory through the FOV.

- **Additional tutorials**
  For training purposes of new users of the entire framework, multiple tutorial videos have been created (see Sec. VI.S3. Tutorial videos). A special tutorial video demonstrates the interaction between all proposed software packages (MFS+RiFe+VT) starting with the simulation of a novel type of TWMPI scanner, followed by the generation of an appropriate 3D system matrix using peak-picking method (MFS), followed by data processing, reconstruction (RiFe), and visualization (VT). In Fig. 21, several screenshots of the key moments of the tutorial video are shown (see supplementary video 3: real-time visualization).

Fig. 21: Several screenshots of the tutorial video demonstrating a fully virtual MPI FFL [66] scanner simulation, from data generation, system matrix creation, reconstruction and visualization with the proposed framework.

- **Scientific projects**
  Using a powerful simulation software, such as MFS, several interesting approaches and novel ideas can be investigated, visualized, and understood before transfer into the real world.

  - **Traveling Wave Magnetic Particle Imaging – TWMPI**
    The first scientific simulation project using MFS software was the Traveling Wave MPI system, which is an alternative scanner approach using a dynamic linear gradient array (dLGA) [67] for the generation and
movement of the FFP [25]. In first studies, the scanner geometry and size could be investigated before real implementation (see Fig. 21).

**MPI meets CT**

This project combined for the first time an MPI scanner with a CT system. For that a novel design has been used based on Halbach arrays for the generation of a static field free line (FFL) [63]. With additional solenoids, the FFL can be moved within the imaging slice (See Fig. 22). In combination with a mechanical rotation of the sample, a full 2D reconstructed image can be provided simultaneously to the CT experiments [62].

**Parallel MPI – pMPI**

The Traveling Wave Magnetic Particle Imaging scanner [25] allows data acquisition with multiple encoding schemes (here multiple FFPs) simultaneously using multiple receive coils. In initial simulation studies emulating the data acquisition, reconstruction process and visualization, the pMPI approach could be validated in simulations (see Fig. 23) and real experiments [68].

*Fig. 21: Renderings of the different coil parts the TW MPI scanner consists of: the dLGA consisting of 20 solenoids combined in two channels (top), two perpendicular saddle-coil pairs and a receive coil (bottom).*

*Fig. 22: renderings of the basic concept behind the FFL MPI scanner based on Halbach rings. (a) using two Halbach rings, a static FFL can be generated in their center. (b) additional solenoids can move the FFL rapidly through a 2D slice for imaging.*
Fig. 23: Result of the simulation study of the parallel MPI approach using two field free points (FFPs) for data acquisition simultaneously [68].

- **Zoom MPI**
  
The dynamic linear gradient array (dLGA [67]) of a TWMPI scanner allows an adjustable gradient strength yielding a ‘zoom’ effect enhancing the spatial resolution within a specific region of the FOV [69]. All simulation studies have been performed with the mentioned MFS software (see Fig. 24).

Fig. 24: Simulation of the ‘zoom’ effect provided by TWMPI scanners through adjusting the phase between the main gradient system [69].
○ **iMPI**

The goal of the iMPI (interventional Magnetic Particle Imaging) project is the implementation of a first human-sized MPI scanner for a human leg dedicated for interventional treatment, such as real-time PTA or stenting [70, 71]. In prior simulation studies, the performance of such a novel kind of scanner has been investigated [72].

![Fig. 25](image)

**Fig. 25:** Evolution from a FFP MPI design (a) to a novel kind of FFL scanner providing a compact system for human-sized MPI-guided treatment (f) [72].

○ **COMPASS**

The idea behind COMPASS (Critical Offset Magnetic PArticle SpectroScopy) is the measurement of slightly changes in the hydrodynamic diameter of functionalized MNPs, caused by binding for example SARS-CoV-2 antibodies, providing an MNP-based rapid test with sensitivities comparable to ELISA. Performing simulation studies with MFS software, the basic effect behind this method was investigated in more detail (see Fig. 26) [73].

![Fig. 26](image)

**Fig. 26:** Result of an offset magnetic field sweep simulation of two samples with slightly different \( \zeta \)-parameter using Langevin equation. Both data sets show differences in the position of the crossing points of real and imaginary curves (critical point on the 3rd harmonic), which can be used for highly sensitive measurement of phase differences between both samples.

○ **3D system function visualization**

For understanding the reconstruction process in MPI, the specific system functions for MPI scanner can be investigated [74]. Using the proposed framework (MFS+RiFe+VT), the system functions for a specific sequence for TWMPI scanner can be investigated and visualized (see Fig. 27) [24].
IV. Discussion

Fast calculation processes are of high interest in many fields, especially for real-time applications such as visualization with low latency. The proposed framework consists of three software packages, each optimized for a specific task: MFS for simulation of magnetic fields and non-linear magnetization response, RiFe for signal processing and image reconstruction, and VT for volume rendering of 3D data. In combination, the framework is optimized for real-time visualization. An example is RiFe & VT in combination with a real MPI scanner [24, 45, 46, 72].

There are many possibilities for optimizing the source-code for better performance as discussed in [7] for RiFe software, e.g., using optimized libraries for matrix calculation, Fourier transformation or outsourcing on GPUs (graphic processing unit). For VT software, regular updating the 3D hardware and driver can boost the performance of the visualization dramatically, but in the end, the bottleneck here are the previous calculation processes (MFS & RiFe). Despite several optimizations, the generation of data sets within MFS software can take a while and especially when generating system matrices with a high number of data points, the waiting times can be quite long (c.f. tutorial video – Fig. 21). To speed up the calculation process, an outsourcing of specific calculation steps on the GPU is possible, which provides a massive parallelization capability [75].

Further optimizations have to be performed to use the provided software packages on computers with more than 100 cores, such as memory optimization to reduce CPU-RAM transfer rates, clustering calculation packages for NUMA cores, etc.

For a higher flexibility and user experience, a server/client system can be implemented to entangle the calculation process from the visualization process. Since these processes did not work on the same machine anymore, an easy-to-use environment for fast reconstruction visualization without the need of powerful computers are possible. To extend the possibilities of the simulation framework (MFS), several magnetization models can be added for a more realistic signal calculation, e.g., the mentioned Langevin equation is just valid for Brownian relaxation and does not cover Néel relaxation of MNP ensembles.

Furthermore, realistic descriptions of the used component, here the conductor-container, can be useful for a more realistic simulation of magnetic field generators. Parameters such as self-inductance, inductivity or blind capacities, which yields eigen-frequencies in the simulated systems, are useful to calculate transfer functions or receive chains with a higher precision. This provides a better understanding of signal generation and data processing in MPI.

Finally, fully automated and intelligent algorithms can be used for optimization processes of magnetic fields, magnetic field gradients, conductor distributions, or signal processing to fit a specific question. The latest achievements in AI-based methods, such as machine learning, evolutionary algorithms, etc. show promising results and could help to support the user of such software packages.

V. Conclusion

The presented framework covers the entire procedure from simulation of a 3D MPI scanner (design, hardware, magnetic field simulation) using MFS software over the reconstruction software (RiFe) offering real-time reconstruction and visualization to 3D visualization (VT software) using volume rendering. The easy-to-use interface, the short calculation times, and the modular processing chain, allows fast prototyping of novel scanner approaches as well as the comparison with real data of implemented MPI systems.
VI. Supplementary material

S1. 3D affine transformation matrix

Any combination of translation \( T \), rotation \( R \), and scaling \( S \) of a 3D object can be combined within a single \( 4 \times 4 \) affine transformation matrix \( \tilde{M} \).

\[
\tilde{M} = R_x(\theta) \cdot R_y(\theta) \cdot R_z(\theta) \cdot S \cdot T
\]

\[
R_x(\theta) = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos(\theta) & \sin(\theta) & 0 \\
0 & -\sin(\theta) & \cos(\theta) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
R_y(\theta) = \begin{bmatrix}
\cos(\theta) & 0 & -\sin(\theta) & 0 \\
0 & 1 & 0 & 0 \\
\sin(\theta) & 0 & \cos(\theta) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
R_z(\theta) = \begin{bmatrix}
\cos(\theta) & -\sin(\theta) & 0 & 0 \\
\sin(\theta) & \cos(\theta) & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
S = \begin{bmatrix}
s_x & 0 & 0 & 0 \\
0 & s_y & 0 & 0 \\
0 & 0 & s_z & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
T = \begin{bmatrix}
1 & 0 & 0 & t_x \\
0 & 1 & 0 & t_y \\
0 & 0 & 1 & t_z \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

S2. Used data formats

There are multiple data formats used within the framework:

- **Format for MFS project files (*.mfs5)**
  Proprietary file format for MFS software for storing the settings for an MFS project (readable ASCII text-file).

- **Format for MFS-script files (*.mf5s)**
  Proprietary file format for MFS scripting command lists (readable ASCII text-file).

- **Format for connected conductor files (*.cnl)**
  Proprietary file format for MFS software for storing connected conductor vector information (readable ASCII file).

- **Format for 3D vectors (*.3dv)**
  Proprietary file format for MFS software to store MNP array (binary file: 4 Byte header with number \( N \) of entries, followed by \( N \times 3 \times \text{double} \) (8 byte) data array for \([x,y,z]\)).

- **Format for 3D mesh models (*.stl)**
  Open-source file format for 3D mesh models (standard triangle language).

- **Format for data sets (*.mpi & *.dbl)**
  Proprietary file format for storing data sets: the *.mpi file is a readable ASCII file consisting of the name of the data set, the samplingrate, and the name of the connected file (*.dbl), which consists of the data set information (binary file: no header with \( N \times \text{double} \) (8 bytes) values; \( N \) can be determined by the size of the *.dbl-file).

- **Format for system matrices (*.sm, *.2da, *.1da & *.txt)**
  Several proprietary file formats for storing a full system matrix (*.sm) or 2D and 1D arrays (*.2df and *.1df). The text-files (*.txt) consists of all required system matrix information, such as matrix sizes and type. The *.sm and *.2df file are binary files with header: \( \text{sizeX} \) (integer) and \( \text{sizeY} \) (integer) followed by...
sizeX×sizeY×double (8 bytes) data array consisting of the values. The *.1df file stores a 1D array (binary file with header: size (integer) followed by size×double (8 bytes) data array.

- **Format for 3D data sets (*.nii)**
  This data format is an open-source data format and follows the NiFTi specifications [59].

- **Formats used for RiFe (*.2df, *.txt, *.png, *.bmp)**
  The *.2df file is a proprietary file format for storing 2D field information (binary file: header (32 bytes) with sizeX (integer – 4 bytes), sizeY (integer – 4 bytes), maxValue (double – 8 bytes), minValue (double – 8 bytes), maxCoordX (integer – 4 bytes), maxCoordY (integer – 4 bytes) followed by sizeX×sizeY×double (8 bytes) data array.
  The *.txt file consists of settings information for RiFe software.
  The *.png and *.bmp file formats are used as standard formats for pictures.

**S3. Tutorial videos**
Multiple tutorial videos are available for training purposes introducing the key features of the proposed framework. A complete list can be found in the latest version of the MFS software.

- 1. Tutorial video 1: first steps in MFS (link)
- 2. Tutorial video 2: transmit&receive (link)
- 3. Tutorial video 3: conductors (link)
- 4. Tutorial video 4: coil-paint (link)
- 5. Tutorial video 5: conductor wizard (link)
- 6. Tutorial video 6: combine conductors (link)
- 7. Tutorial video 7: MPI scanner (FP) (link)
- 8. Tutorial video 8: MPI scanner (FFL) (link)
- 9. Tutorial video 9: TWMPI scanner (link)
- 10. Tutorial video 10: 3D mesh to MNP distribution (link)
- 11. Tutorial video 11: MFSS script (link)
- 12. Tutorial video 12: Rodin-coil (link)
- 13. Tutorial video 13: Halbach MPI (link)
- 14. Tutorial video 14: Halbach MPI (MPI meets CT) (link)
- 15. Tutorial video 15: real-time visualization (german) (link)
- 16. Tutorial video 16: trajectory trend (link)
- **Supplementary video 1**: Introduction to the 3D-GUI
- **Supplementary video 2**: Introduction to system matrices
- **Supplementary video 3**: Real-time visualization using MFS+RiFe+VT
Fig. S3: Several screenshots of the tutorial videos available so far for MFS software and combination of MFS+RiFe+VT (tutorial videos 1..16).

AUTHOR'S STATEMENT
Research funding: This work was partially funded by the DFG (BE 5293/1-1, BE 5293/1-2, VO 2288/1-1, VO 2288/2-1, VO 2288/3-1) and the IDEA project of the 7th framework programme of the European Union (project reference 279288). Conflict of interest: Authors state no conflict of interest.

REFERENCES
[1] B. Gleich & J. Weizenecker, Tomographic Imaging using the nonlinear response of magnetic particles, Nature, 435, 1214-7, 2005.
[2] W.T. Smolik et al., Numerical modeling of magnetic field for MPI, Proc. on CIS (Manchester), 12998496, 2012.
[3] W.T. Smolik et al., Magnetic Field Simulation Toolbox for MPI Modeling, Proc. on IWMPI Berlin, p27, 2014.
[4] M. Straub et al., Flexible and Modular MPI Simulation Framework and Its Use in Modeling a μ MPI, IEEE Trans Magn, 51(2):6501204, 2015.
[5] P. Vogel et al., 3D-GUI Simulation Environment for Magnetic Particle Imaging, Proc. on IWMPI (Lübeck), p. 95, 2016.
[6] R. Dhavalikar et al., Finite magnetic relaxation in x-space magnetic particle imaging: comparison of measurements and ferrohydrodynamic models, J Phys D: Appl Phys, vol. 49(30):305002, 2016.
[7] P. Vogel et al., Low Latency Real-time Reconstruction for MPI Systems, IJMPI, 3(2):1707002, 2017.
[8] U. Heinen et al., A versatile MPI System Function Viewer, IJMPI, 3(2):1706006, 2017.
[9] Y. Shen et al., A novel software framework for magnetic particle imaging reconstruction, Int J Imaging Syst Technol, vol. 32, pp. 1119–1132, 2022.
[10] K.N. Quelhas et al., Flexible Software for Rigorous Simulations of Magnetic Particle Imaging Systems, Int J Magn Part Imag, vol 8(1):2203081, 2022.
[11] https://magneticparticleimaging.github.io/
[12] https://os-mpi.github.io/EduSim/
[13] T. Knopp et al., Magnetic Particle Imaging: From Proof of Principle to Preclinical Applications, Phys Med Biol, vol. 62(14), R124, 2017.
[14] P. Vogel et al., Traveling Wave Magnetic Particle Imaging for determining the iron-distribution in rock, Diffusion-Fundamentals.org, vol. 22(12), pp. 1-5, 2014.
[15] M. Kalra, M. Meribout, Modeling and Simulation Study for Solid Particles Detection in Gas–Oil Pipelines Using Field Free Line in Magnetic Particle Imaging and Its Hardware Realization, IEEE Trans Magn, vol. 53(7): 5300408, 2017.
[16] C. Billings et al., Magnetic Particle Imaging: Current and Future Applications, Magnetic Nanoparticle Synthesis Methods and Safety Measures, Int. J. Mol. Sci., vol. 22:7651, 2021.
[17] Z. W. Tay et al., Magnetic Particle Imaging: An Emerging Modality with Prospects in Diagnosis, Targeting and Therapy of Cancer, caners, vol. 13:5285, 2021.
[18] A. Neumann et al., Recent Developments in Magnetic Particle Imaging, JMMM, vol. 550:169037, 2022.
[19] T. Kluth et al., Towards accurate modeling of the multidimensional magnetic particle imaging physics, New journal of physics, vol. 21:103032, 2019.
[20] A. von Gladiss et al., Efficient hybrid 3D system calibration for magnetic particle imaging systems using a dedicated device, Sci Rep, vol. 10:18432, 2020.
[21] X. Chen et al., The Reconstruction of Magnetic Particle Imaging: Current Approaches Based on the System Matrix, diagnostics, vol. 11:773, 2021.
[22] J. Rahmer et al., Signal encoding in magnetic particle imaging: properties of the system function, BMC Med Imaging, vol. 9:4, 2009.
[23] P. Vogel et al., Flexible and Dynamic Patch Reconstruction for Traveling Wave Magnetic Particle Imaging, International Journal on MPI, vol. 2(2):1611001, 2016.
[24] P. Vogel et al., Real-time 3D Dynamic Rotating Slice-Scanning Mode for Traveling Wave MPI. IJMPI, 3(2): 1706001, 2016.
[25] P. Vogel et al., Traveling Wave Magnetic Particle Imaging, IEEE TMI, vol. 33(2), pp. 400-407, 2014.
[26] P. W. Goodwill et al., An x-space magnetic particle imaging scanner, Rev Sci Instrum, vol. 83(3):033708, 2012.
[27] P. W. Goodwill & S. M. Conolly, Multidimensional X-Space Magnetic Particle Imaging, IEEE Trans Med Imaging, vol. 30(9), pp. 1581-1590, 2011.
[28] P. Vogel et al., Superspeed Bolus Visualization for Vascular Magnetic Particle Imaging, IEEE Trans Med Imaging, vol. 39(6), pp. 2133-9, 2020.
[29] P. Vogel et al., First in-vivo Traveling Wave Magnetic Imaging of a beating mouse heart, Phys. Med. Biol., vol. 61(18), pp. 6620-6634, 2016.
[30] P. W. Goodwill et al., Narrowband magnetic particle imaging, IEEE Trans Med Imaging, vol. 28(8), pp. 1231-7, 2009.
[31] K. J. Janssen et al., Single harmonic-based narrowband magnetic particle imaging, Meas. Sci. Technol., vol. 33:095405, 2022.
[32] Baggeri et al., A mechanically driven magnetic particle imaging scanner, Appl. Phys. Lett., vol. 113:183703, 2018.
[33] S.-M. Choi et al., A novel three-dimensional magnetic particle imaging system based on the frequency mixing for the point-of-care diagnostics, Sci Rep., vol. 10:11833, 2020.
[34] A. Tateo et al., Hybrid x-space: a new approach for MPI reconstruction, Phys Med Biol, vol. 61, pp. 4061-77, 2016.
[35] C. Volkmar et al., Arbitrarily shaped coils' inductance simulation based on a 3-dimensional solution of the Biot-Savart law, ISSE (Alba Iulia), 36:210-215, 2013.
[36] X. Batlle et al., Magnetic nanoparticles: From the nanostructure to the physical properties, JMMM, vol. 543:168594, 2022.
[37] W. T. Coffey, Y. P. Kalmykov, J. T. Waldron. The Langevin equation. 2nd ed. Singapore: World Scientific 2004.
[38] S. Ota, Y. Takemura, Characterization of Néel and Brownian Relaxations Isolated from Complex Dynamics Influenced by Dipole Interactions in Magnetic Nanoparticles, J. Phys Chem, vol. 123(47), pp. 28859-66, 2019.
[39] P. Ilg, M. Kröger, Dynamics of interacting magnetic nanoparticles: effective behavior from competition between Brownian and Néel relaxation, Phys. Chem. Chem. Phys., vol. 22, pp. 22244-22259, 2020.
[40] M. A. Martens, R. J. Deissler, Y. W. Wu, L. Bauer, Z. Yao, R. Brown, Modeling the Brownian relaxation of nanoparticle ferrofluids: Comparison with experiment, Med Phys, vol. 40(2):022303, 2013.
[41] D. I. Hoult and R. E. Richards, the signal-to-noise ration of the nuclear magnetic resonance experiment, J. Magn. Reason, vol. 24, pp. 71-85, 1976.
[42] J. J. van der Klink, The NMR Reciprocity Theorem for Arbitrary Probe Geometry - Journal of Magnetic Resonance vol. 148, pp. 147–154, 2001.
[43] M. Frigo and S. G. Johnson. FFTW library, 2004. URL www.fftw.org
[44] E. Anderson et al., LAPACK Users Guide. SIAM, Philadelphia, 1999. doi:10.1137/1.9780898719604.
[45] S. Herz & P. Vogel et al., Magnetic Particle Imaging guided real-time Percutaneous Vascular Angioplasty: A phantom study, Cardiovasc Intervent Radiol, vol. 41(7), pp. 1100-5, 2018.
[46] S. Herz et al., Magnetic Particle Imaging-Guided Stenting, JEVT, vol. 26(4), pp. 512-9, 2019.
[47] Wei Wu et al., Multipole magnets for the HIAF fragment separator using the Canted-Cosine-Theta (CCT) geometry, J. Phys.: Conf. Ser., vol. 1401:012015, 2020.
[48] M.A. Rückert et al., Numerically Efficient Estimation of Relaxation Effects in MPI, Biomed Tech, vol. 58(6), pp. 593-600, 2013.
[49] W. T. Coffey, Y. P. Kalmykov, J. T. Waldron. The Langevin equation. 2nd ed. Singapore: World Scientific 2004.
[50] M.A. Martens et al., Modeling the Brownian relaxation of nanoparticle ferrofluids: Comparison with experiment, Med Phys, vol. 40(2):022303, 2013.
[51] P. Vogel et al., Nuclear magnetic signal generation with variable field amplitudes, Proc on ICMRM (Munich), L062, 2015.
[52] A. Schönhage, A.F.W. Grotefeld, E. Vetter: Fast Algorithms—A Multitape Turing Machine Implementation, BI Wissenschafts-Verlag, Mannheim, 1994.
[53] M. Erbe et al., improved field free line magnetic particle imaging using saddle coils, Biomed Tech, vol. 58(6), pp. 577-582, 2013.
[54] T. Reichl et al., Highly Flexible Human Aneurysm Models for Realistic Flow Experiments with MPI and MRI, Int J Mag Part Imag, vol. 8(1): 2200305, 2022.
[55] K. Guggenberger, et al., High-resolution Compressed-sensing T1 Black-blood MRI – A new multipurpose Sequence in vascular neuroimaging, Clin Neuroradiol, vol. 31(1), pp. 207-16, 2019.
[56] K. Guggenberger & A.J. Krafft et al., Intracranial Vessel Wall Imaging Framework – Data Acquisition, Processing, and Visualization, MRI, vol. 83, pp. 114-24, 2021.
[57] Y. Liu et al., Fabrication of cerebral aneurysm simulator with a desktop 3D printer, Sci Rep, vol. 7:44301, 2017.
[58] Pixar, The RenderMan Interface Specification (v3.1), 1989.
[59] R.W. Cox et al., A (sort of) new image data format standard: NiFTI-1, OHBM (Budapest) #10, 2004.
[60] P. Vogel et al., Rotating Slice Scanning Mode for Traveling Wave MPI, IEEE Trans Magn, vol. 51(2):6501503, 2015.
[61] J. Weizenecker et al., Magnetic Particle Imaging using a Field Free Line, J. Phys. D: Appl. Phys., vol. 41:105009, 2008.
[62] P. Vogel et al., Magnetic Particle Imaging meets Computed Tomography: first simultaneous imaging, Sci. Rep., vol. 9:12627, 2019.
[63] M. Weber et al., Novel Field Geometry featuring a Field Free Line for Magnetic Particle Imaging, International Journal on Magnetic Particle Imaging, vol 4(2):1811004, 2018.
[64] L. Mirzojan et al., Fully Mechanically driven Traveling Wave Magnetic Particle Imaging, Int J Mag Part Imag, vol. 8(1):2203011, 2022.
[65] P. Vogel et al., Magnetic Particle Imaging using Toroidal Vortex Rotation of Halbach Rings, Proc. on ICMRM #15 (Paris), p. 54, 2019.
[66] C. Greiner et al., Traveling Wave MPI utilizing a Field-Free Line, IJMPI, vol. 8(1):2203027, 2022.
[67] P. Vogel and P. Klauer et al., Dynamic Linear Gradient Array for Traveling Wave Magnetic Particle Imaging, IEEE Trans Magn, vol. 54(2): 5300109, 2018.
[68] P. Vogel et al., Parallel Magnetic Particle Imaging, Rev. Sci. Instrum., vol. 91(4):045117, 2020.
[69] P. Vogel et al., Adjustable hardware lens for Traveling Wave MPI, IEEE Trans Magn, vol. 56(11):5300506, 2020.
[70] S. Herz & P. Vogel et al., Magnetic Particle Imaging guided real-time Percutaneous Vascular Angioplasty: A phantom study, Cardiovasc Intervent Radiol, vol. 41(7), pp. 1100-5, 2018.
[71] S. Herz et al., Magnetic Particle Imaging-Guided Stenting, Journal of Endovascular Therapy, vol. 26(4), pp. 512-9, 2019.
[72] P. Vogel et al., iMPI – interventional Magnetic Particle Imaging, Int J Magn Part Imaging, vol. 8(1):220305, 2022.
[73] P. Vogel et al., COMPASS – rapid and highly sensitive medical point-of-care diagnostic, PREPRINT (Version 1) available at Research Square [https://doi.org/10.21203/rs.3.rs-1712353/v1], 2022.
[74] U. Heinen et al., A versatile MPI system function viewer, IJMPI, vol. 3(2):1706006, 2017.
[75] J. Fung et al., Mediated Reality using Computer Graphics Hardware for Computer Vision, In Proceedings of the International Symposium on Wearable Computing 2002 (ISWC2002), pp. 83-89, 2002.