

Parallel Computation in Simulating Diffusion and Deformation in Human Brain *

Ning Kang [†] Jun Zhang [‡]

Laboratory for High Performance Scientific Computing and Computer Simulation
Department of Computer Science, University of Kentucky
Lexington, KY 40506-0046, USA

Eric S. Carlson [§]

Department of Chemical Engineering, University of Alabama
P. O. Box 870203
Tuscaloosa, AL 35487-0203, USA

September 9, 2004

Abstract

The purpose of this paper is to survey some recent developments in the application of parallel and high performance computation in simulating the diffusion process in the human brain and in modeling the deformation of the human brain. Computational neuroscience is a branch of biomedical science and engineering in which sophisticated high performance computing techniques can make a huge difference in extracting brain anatomical information non-invasively and in assisting minimal invasive neurosurgical interventions. This paper will demonstrate that there are lots of potential opportunities for computational scientists to work with biomedical scientists to develop high performance computing tools for biomedical applications.

1 Introduction

We present a survey on two types of computer simulations conducted in the human brain involving parallel computation. The first one is to simulate the anisotropic diffusion process in the human brain, making use of the measured diffusion tensor magnetic resonance

^{*}Technical Report No. 418-04, Department of Computer Science, University of Kentucky, Lexington, KY, 2004.

[†]The research work of this author was supported by the U.S. Department of Energy Office of Science under grant DE-FG02-02ER45961. E-mail: nkang2@cs.uky.edu.

[‡]The research work of this author was supported in part by the U.S. National Science Foundation under grants CCR-9988165, CCR-0092532, and ACR-0202934, in part by the U.S. Department of Energy Office of Science under grant DE-FG02-02ER45961, and in part by the University of Kentucky Research Committee. E-mail: jzhang@cs.uky.edu, URL: <http://www.cs.uky.edu/~jzhang>.

[§]The research work of this author was supported by the U.S. Department of Energy Office of Science under grant DE-FG02-02ER45961. E-mail: ecarlson@bama.ua.edu.

imaging (MRI) data. Diffusion tensor MRI technique is the only non-invasive *in vivo* approach available so far to investigate the three dimensional architecture of brain white matter with the potential to generate fiber-tract trajectories in the white matter. The simulation of the diffusion over the brain could play a critical role in the development of improved approaches for the reconstruction of nerve fiber tracts in the human brain. The second type is the simulation of brain deformation during image guided neurosurgery. The challenge that a neurosurgeon has to face is the removal from the brain of as much diseased tissue as possible and meanwhile the removal of normal tissue must be minimized and the disruption of important anatomical structures must be avoided. Therefore, it is crucial to capture the brain deformation during the neurosurgical interventions on patients by aligning preoperatively acquired image data with intraoperative images. To be practical, the brain deformation simulation has to meet real-time constraints as well as achieve robustness and high accuracy.

2 Anisotropic Diffusion Simulation in White Matter Tractography

2.1 Background

The solution of the general unsteady state anisotropic diffusion equation could be of critical importance in the development of improved approaches for the analysis of diffusion tensor magnetic resonance imaging (DT-MRI). DT-MRI is an extension of conventional MRI with the added capability of measuring the random motion of water molecules in all three dimensions, usually referred to as diffusion or “Brownian motion”. Since water diffusion is influenced by the microstructure, architecture, and physical properties of tissues, DT-MRI can render the information about how water diffuses in biological tissues containing a large number of fibers, like muscles or brain white matter, into intricate three-dimensional representations of the tissues architecture. Thus, it can be exploited to visualize and extract information about the brain white matter and nerve fibers by reconstructing the fiber pathways, which has raised promises for achieving a better comprehension of the fiber tract anatomy of the human brain. In combination with functional MRI, it might potentially bring tremendous improvements in deeply understanding the crucial issue of anatomical connectivity and functional coupling between different regions of the brain [36, 18, 27]. Therefore, the neuro-anatomical knowledge on connectivity interpreted from the DT-MRI information has been playing an indispensable role in neurosurgery planning [52] and in tackling a lot of brain diseases and disorders, such as Alzheimer’s disease [45, 49], attention-deficit hyperactivity disorder, and schizophrenia [1, 24].

It is known that water diffusion is anisotropic in brain white matter. The significant anisotropy presented in white matter reveals microscopic properties of the anatomy of the nerve fibers by the fact that water tends to diffuse predominantly along the long axis of the fibers, because the longitudinally oriented structures, the dense packing of axons and the inherent axonal membranes, which are widely assumed to be the main barrier for water diffusion, hinder diffusion perpendicular to the fibers [8]. DT-MRI is sensitive to this structural anisotropy and is able to characterize it by non-invasively quantifying and assessing the self-diffusion of water *in vivo*. The information concerning the local orientation of fibers, extracted from the water anisotropic diffusion in white matter, forms

the basis of utilizing DT-MRI to track fiber pathways and build connectivity mapping *in vivo*. The water diffusion behavior elucidated by the diffusion tensor imaging reflects the directional organization of the underlying white matter microstructure. DT-MRI characterizes the behavior on a voxel by voxel basis and for each voxel, the diffusion tensor yields the diffusion coefficient corresponding to any direction in space [4]. The direction of the greatest diffusion can be determined by evaluating the diffusion tensor in each voxel, which is believed to point along the dominant axis of the white matter fiber bundles traversing the voxel. Thus, the panoramic view of the fastest diffusion direction can be generated to provide a visualization of the white matter pathways and their orientations.

A number of fiber tracking algorithms have been developed since the appearance of DT-MRI. In [9, 40] a variety of these algorithms are described and reviewed. As the measured quantity in DT-MRI is water diffusion, an intuitive way to gain insights from the diffusion tensor data is to carry out a virtual simulation of water diffusion, which is anisotropic and governed by the diffusion equation, over the brain. The white matter fiber bundles are assumed to proceed along the direction where the diffusion is the greatest. The idea of studying brain connectivity by way of simulating the anisotropic diffusion has been explored in [5, 25, 33]. [5] specifies a starting point for tractography where a seed is diffused. A virtual concentration peak of water is spread in [25]. In [33], successive virtual anisotropic diffusion simulations are performed over the whole brain, which are utilized to construct three dimensional diffusion fronts and then the fiber pathways. This technique of solving a diffusion equation makes use of the full information contained in the diffusion tensor and it is not dependent upon a point to point eigenvalue/eigenvector computation along a trajectory, thus in that sense may enhance the robustness and reliability of fiber reconstruction algorithms. It is also intuitively related to the underlying physical-chemical process in the central nervous system [42, 51]. The diffusion process and related transport mechanisms in the brain are discussed in detail in [41].

Simply put, anisotropic systems are those that exhibit a preferential flow direction, in which the flow field does not follow the concentration gradient directly, for the material properties also affect diffusion. Therefore, the diffusion tensor, D , is introduced to fully describe the molecular mobility along each direction and the correlation between these directions. We have

$$D = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix},$$

where the subscripts xx , xy , xz , etc., denote the values of the individual coefficients in the matrix that can be seen as the influence from directions in the input (being the concentration) on the various directions in the output (being the flux). For the brain system and other typical systems, the tensor is symmetric. Fig. 1 shows an axial slice of a diffusion tensor volume from the human brain.

Essentially, the fiber tracking schemes based on diffusion simulation mentioned above are seeking to solve an unsteady state diffusion equation in an anisotropic medium based on the measured diffusion tensor D . The anisotropic diffusion process, due to the conservation of mass, is governed by

$$\frac{\partial C}{\partial t} = \nabla \cdot (D \nabla C), \quad (1)$$

where t is the independent time variable. This equation says that over the time, the rate

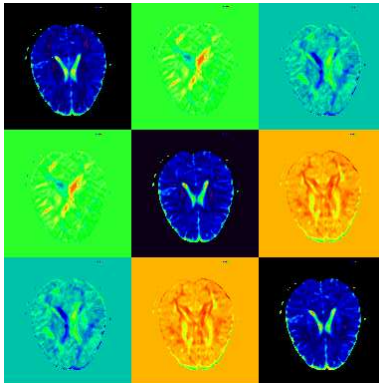


Figure 1: An axial slice of a diffusion tensor volume [35], which shows the individual diffusion tensor components, corresponding to the diffusion tensor matrix D .

of change in concentration is proportional to the divergence of the flux.

In a Cartesian coordinate system, the Equation (1) is expressed as

$$\begin{aligned} \frac{\partial C}{\partial t} = & \frac{\partial}{\partial x} \left(D_{xx} \frac{\partial C}{\partial x} + D_{xy} \frac{\partial C}{\partial y} + D_{xz} \frac{\partial C}{\partial z} \right) \\ & + \frac{\partial}{\partial y} \left(D_{yx} \frac{\partial C}{\partial x} + D_{yy} \frac{\partial C}{\partial y} + D_{yz} \frac{\partial C}{\partial z} \right) \\ & + \frac{\partial}{\partial z} \left(D_{zx} \frac{\partial C}{\partial x} + D_{zy} \frac{\partial C}{\partial y} + D_{zz} \frac{\partial C}{\partial z} \right). \end{aligned} \quad (2)$$

This equation could be very difficult to solve under the circumstance of the human brain for a couple of reasons. First, since the brain structure is heterogeneous where anisotropy requires full tensor representation, the second order cross derivatives must be calculated. Second, the diffusion tensor changes drastically between adjacent small regions in the brain tissues. Thus, fine gridding must be used to avoid a crude approximation to the true geometry of interesting structures and this leads to large systems of equations. The third challenge is that time plays a crucial role in the real environment, such as clinical diagnosis, surgical planning, and neurosurgery [52]. In the sense to be practical, the solution must meet the real-time constraints and achieves good reliability and robustness as well.

2.2 Simulating the Diffusion Process

As mentioned above, the anisotropic diffusion process is simulated in [5, 25, 33], each of which employs different tensor data set and uses different approach to solve the governing diffusion equation. The diffusion tensor data is extracted and estimated from the collected echo-planar images with different diffusion weighting and at least six independent gradient directions and one unweighted image by solving the Stejskal-Tanner equation for anisotropic diffusion, since the symmetric tensor matrix has at least six independent components [3].

In [5], the diffusion tensor data set used is obtained from the acquired 25 slices with voxel size being $2 \times 2 \times 2 \text{ mm}^3$, and the diffusion equation is discretized in space with Galerkin finite element method and in time with finite difference method. The semi-implicit system is then solved by using a Crank-Nicholson scheme. Initially, the concen-

tration distribution in the brain, which is the seed, is described as

$$C(x, 0) = \begin{cases} 1 & \text{in starting region,} \\ 0 & \text{elsewhere,} \end{cases} \quad (3)$$

and the simplified anisotropic diffusion equation being solved is

$$\nabla \cdot D \nabla C(x, t) = -\frac{\partial C(x, t)}{\partial t}, \quad (4)$$

where $C(x, t)$ is the concentration function. The Neumann boundary condition is enforced in the computation, i.e., $D \nabla C \cdot \mathbf{n} = \mathbf{0}$ on the boundary, in which \mathbf{n} is the direction normal to the surface of the boundary.

Both a two dimensional and a three dimensional diffusion tensor data set are applied to solve the anisotropic diffusion equation in [25], with the resolution of the tensor data being $128 \times 128 \times 43$ corresponding to a uniform voxel size of 1.8 mm^3 . The equation is solved using a finite element method from the software package ‘‘FEMLAB’’ (www.femlab.com). The anisotropic diffusion equation being solved is also simplified as

$$\frac{\partial C}{\partial t} = \sum_{i,j=1}^3 D_{ij} \frac{\partial^2 C}{\partial x_i \partial x_j}, \quad (5)$$

with an initial water concentration peak specified by a Gaussian function with a standard deviation of 2. The time cost for the finite element computations in a two dimensional grid with 128×128 voxels is reported in the order of minutes on a PC with a 700 MHz Pentium III CPU and 512 MB memory. The three dimensional simulation runs about 4 minutes before completion, using a mesh with 19651 voxels generated by a Delaunay algorithm.

In [33], successive virtual anisotropic diffusion simulations are performed over the whole brain, where the resolution of the tensor data set is $128 \times 128 \times 48$ with each voxel size being $2.5 \times 2.5 \times 2.5 \text{ mm}^3$ defined on the Cartesian mesh. The anisotropic diffusion equation (2) on the Cartesian grid is discretized using a finite difference approximation. The central difference in space and backward differentiation formula in time are applied to approximate the spatial derivative and time derivative terms in Equation (2), respectively. On the boundaries of the heterogeneous system, it is assumed to be insulated, i.e., $D \nabla C \cdot \mathbf{n} = \mathbf{0}$, which corresponds to the Neumann condition as well. This condition means that the normal part of the gradient of the concentration on the boundary is zero. Nothing diffuses outside of the brain image boundary. The initial condition uses Equation (3), where the start region is a single voxel and it changes with the growth of the fiber pathways.

In another related work to white matter connectivity [43], a steady state concentration distribution is created in the diffusion tensor field by solving

$$\nabla \cdot (D \nabla C) = \mathbf{0}. \quad (6)$$

As for the boundary and initial conditions, they simulate a sink at one point of interest and a source at another, in order to estimate the steady-state flow between the regions. [6] further extends the diffusion equation to a diffusion-convection equation by adding a convection term

$$\frac{\partial C}{\partial t} = \nabla \cdot (D \nabla C) + \mathbf{f} \cdot \nabla C, \quad (7)$$

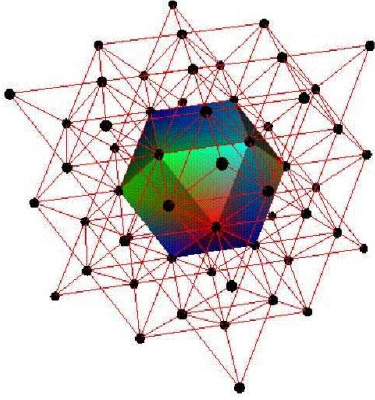


Figure 2: The 3D FCC grid (the center unit is a cuboctahedron) [34].

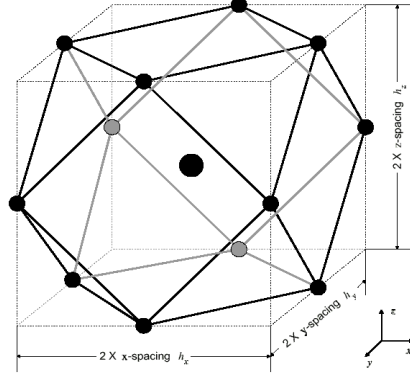


Figure 3: The 13-point cuboctahedral stencil in a 3D FCC grid [34].

where \mathbf{v} is a vector field and \mathbf{f} is a coefficient to control the convection term and could be chosen to rely on anisotropies. Equation (7) is discretized by way of a finite element method with Crank-Nicholson scheme.

The anisotropic diffusion equation is also solved in the environment of the human brain with measured DT-MRI data in [34], where the resolution of the tensor data set is $128 \times 128 \times 16$ with each voxel size being $2.5 \times 2.5 \times 7.5 \text{ mm}^3$ defined on the Cartesian mesh. A different yet efficient gridding scheme, the face-centered cubic (FCC) grids are employed to do the discretization [53]. In the case of 3D, Fig. 2 displays the three dimensional FCC mesh and the grid has a thirteen-point cuboctahedral stencil, as shown in Fig. 3. The node at the center of Fig. 3 is symmetrically surrounded by 12 equidistant nodes. The external 12 nodes are the 12 vertices of a cuboctahedron. Thus, the actual number of computational nodes in the FCC grid is half the size of that in the Cartesian grid, in terms of the way of generating the FCC grid described in [34]. A three dimensional Gaussian function with an equal standard deviation of $\sigma = 0.2$ is selected to be the initial distribution profile of the water concentration in the brain

$$C \Big|_{t=0} = \frac{1}{\sigma^3 \sqrt{8\pi^3}} e^{-[(x-\mu_x)^2 + (y-\mu_y)^2 + (z-\mu_z)^2] / 2\sigma^2}, \quad (8)$$

where μ_x , μ_y , and μ_z are the mean in the x , y , and z direction, respectively.

The discretization of the Equation (2) and its boundary conditions on the FCC grid or the Cartesian grid generates a large scale system of semi-explicit differential-algebraic equations (DAEs) with the form

$$\mathbf{F}(t, \mathbf{y}, \mathbf{y}') = \mathbf{0}, \quad (9)$$

where \mathbf{y} and \mathbf{y}' are N -dimensional vectors corresponding to the discretized values of \mathbf{C} and \mathbf{C}' . In both [33] and [34], a high performance differential-algebraic equation (DAE) solver, the IDA solver in the SUNDIALS suite, one of the software packages contained in the ACTS Toolkit (acts.nersc.gov). The name IDA stands for Implicit Differential-Algebraic solver, which is a general purpose solver for the initial value problem for systems

of DAEs. SUNDIALS stands for SUite of Nonlinear and DIfferential/ALgebraic equation Solvers. Detailed explanations of the DAEs theories and its numerical solution methods on initial-value problems are given in [10]. More detailed information about the SUNDIALS suite and the IDA solver can be found in [31] and at the web site of SUNDIALS (<http://acts.nersc.gov/sundials/>).

For the solution of the linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{10}$$

resulted from Equation (9), the IDA includes both direct and iterative methods. Due to the large-scale nature of the 3D simulation problem, direct solution methods based on Gaussian elimination are not suitable for their prohibitive CPU and memory costs. Another thing about solving the time-dependent system is that it is often possible to use a preconditioner over more time steps when using iterative methods than it would be possible to keep an iteration matrix in the direct method, because the iterative methods do the rest of the work in solving the system. Therefore, both [33] and [34] choose the preconditioned iterative methods based on the Krylov subspaces.

Since the preconditioning of the linear iteration is essential and beneficial for both robustness and efficiency, [34] applies a number of different preconditioning techniques, based on incomplete LU (Lower-Upper) factorizations of the coefficient matrix, to the GMRES [46] method embedded in the IDA solver. The basic idea of preconditioning is as follows. Let \mathbf{P} be a matrix which approximates the coefficient matrix \mathbf{A} in some way. Preconditioning in an iterative method for solving the linear system, Equation (10), means applying the iterative method instead to the equivalent system $\mathbf{P}^{-1}\mathbf{A}\mathbf{x} = \mathbf{P}^{-1}\mathbf{b}$. The preconditioned system is expected to become easier to solve than the original problem, i.e., the matrix $\mathbf{P}^{-1}\mathbf{A}$ is better conditioned than \mathbf{A} or $\mathbf{P}^{-1}\mathbf{A}$ has a more favorable eigenvalue distribution than \mathbf{A} does. In order for the preconditioning to be efficient, \mathbf{P} should be in some sense close to \mathbf{A} and its construction should be inexpensive.

Numerical experiments are performed in [34] to investigate the efficiency and effectiveness of several standard ILU preconditioners in solving the linear systems arising at each Newton iteration, which are conducted on a Sun Blade 100 workstation with a single 500 MHz UltraSPARC-IIe processor and 128 MB memory. The overall time cost runs in several hundred seconds, including the preconditioner setup time and iteration time. The ILU preconditioners based on the static pattern scheme, $\text{ILU}(\mathbf{k})$, are found to be inefficient for solving the current problem. The best choice among all the preconditioners that are investigated is the dynamic pattern scheme based ILUT, with optimum choices of its two thresholding parameters \mathbf{p} and $\boldsymbol{\tau}$. The composite preconditioner in which $\text{ILU}(0)$ uses the ILUT data pattern shows better performance than $\text{ILU}(0)$, but no better than ILUT. The numerical tests in [34] illustrate that, regardless of the thresholding strategies applied in the ILU preconditioning, the choice of the corresponding parameters has direct and distinct influences on the accuracy and the construction cost of the preconditioner, the convergence rate of the iterative solution, and the total computational efforts. Generally speaking, the more entries kept in the factorization, the higher quality of the ILU preconditioner, which makes the iterative solution more robust but with the price of more construction time, and higher per iteration cost.

Shown in Fig. 4 are the simulated concentration distribution profiles of the anisotropic diffusion in the human brain from [34]. The simulated result on the 3D Cartesian grid

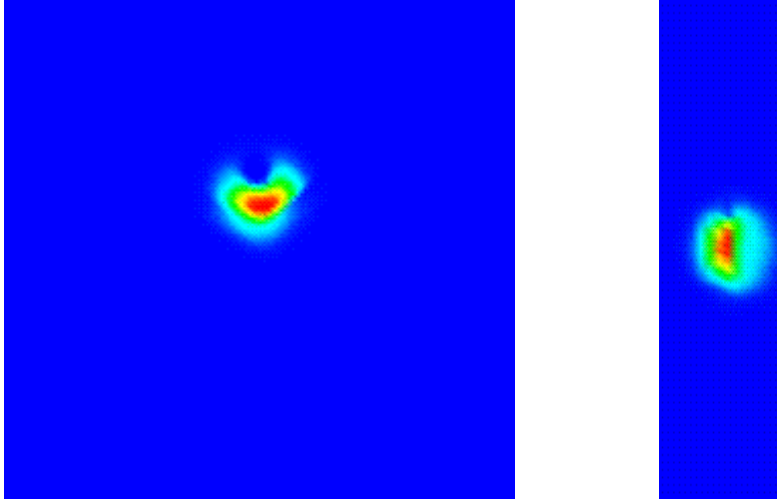


Figure 4: The concentration distribution profiles of the anisotropic diffusion in the human brain [34]. Left: the profile on the cutting plane of $z = 8$. Right: the profile on the cutting plane of $y = 75$.

is derived from interpolation when the solution on the FCC grid is mapped back to the Cartesian grid.

2.3 A Parallel Implementation

All the above mentioned implementations of solving the anisotropic diffusion equation in a human brain with the measured DT-MRI data are done in a serial computing environment. In order to perform diffusion simulations over the whole brain with sufficient accuracy and acceptable computational time and memory cost, a parallel implementation of the solution is presented in [35]. The tensor data set used in the simulation is the same as that used in [34], where the resolution of the data set is $128 \times 128 \times 16$ with each voxel size being $2.5 \times 2.5 \times 7.5 \text{ mm}^3$ defined on the Cartesian mesh. The 3D diffusion equation (2) is discretized on the Cartesian grid using finite difference approximation. The central difference in space and backward differentiation formula in time are applied to approximate the spatial derivative and time derivative terms in Equation (2), respectively. [35] applies the same boundary and initial condition as those used in [34]. The resulting coefficient matrix after the discretization process is distributed to different processors with the scheme of row-wise block striping, i.e., the matrix is divided into groups of complete rows and each group is assigned to one processor. Then the parallel version of the IDA solver in the ACTS Toolkit is exploited as the primary integration tools. For the large scale sparse linear system arising from each integration step, the Krylov subspace method, preconditioned GMRES, is used and a number of highly efficient and robust parallel preconditioners are applied as well to achieve speedy solutions with good accuracy.

The parallel version of IDA uses a revised version of the vector module NVECTOR in its package to achieve parallelism and the MPI (Message Passing Interface) library for all interprocessor communication. The NVECTOR module is the key to make possible the shift from the serial computing environment to the parallel computing environment. It

contains a set of mathematical operations on \mathbf{N} -vectors (\mathbf{N} -dimensional vectors), including vector linear combinations, scaling, vector norms, scalar products, and so forth. By separating these operations from the rest of the code, all operations in IDA with significant potential for parallel computation are isolated, which allows parallel computation to be neatly implemented with these codes. Because the parallel form of IDA is intended for an SPMD (Single Program Multiple Data) programming model with distributed memory, all \mathbf{N} -vectors are identically distributed across processors such that each processor is solving a contiguous subset of the DAE system. For any given vector operation, each processor performs the operation on its contiguous elements of the input vectors, followed by a global reduction operation where needed. In this way, vector calculations can be done simultaneously with each processor working on its own segment of the vector.

The incomplete LU preconditioners studied in [34] are not suitable for implementations on parallel computers. In the parallel computing environment, it is desirable that both the preconditioner construction phase and the preconditioned solution phase possess a high degree of parallelism. In [35], two classes of preconditioners suitable for parallel implementation are investigated. The first parallel preconditioning technique is a class of sparse approximate inverse preconditioners. The sparse approximate inverse preconditioner, as its name implies, is an approximation to \mathbf{A}^{-1} , the inverse of a matrix \mathbf{A} . Both its construction and its application in the iterative solution, which requires nothing but matrix-by-vector products, allow a high degree of parallelism and can be implemented in parallel without much difficulty. The sparse approximate inverse preconditioning technique discussed in [35] is based on the idea of the least squares (Frobenius norm) minimization [26], using *a priori* sparsity patterns [14], where people seek to approximate the inverse of a matrix \mathbf{A} (usually sparse) by a sparse matrix \mathbf{P} , such that $\mathbf{AP} \approx \mathbf{I}$ in some sense, where \mathbf{I} is the identity matrix. Another class of preconditioners that are involved in [35] is the block-diagonal preconditioning, which is also suitable for the parallel architecture. Actually, close attention is paid to the banded-block-diagonal preconditioners. This class of preconditioners is based on the block Jacobi method where a preconditioner can be derived by a partitioning of the variables. The basic idea is to isolate the preconditioning so that it is local to each processor. In fact, on parallel computers it is natural to let the partitioning coincide with the division of the variables over the processors.

In [35] a number of numerical results are presented to compare the performance of sparse approximate inverse (SAI) and banded-block-diagonal (BBD) preconditioners on the simulation of the anisotropic diffusion in the human brain. The numerical tests are conducted on a 32-processor (HP PA-RISC 8700 processors running at 750 MHz) subcomplex of an HP superdome supercomputer at the University of Kentucky. Each processor has two gigabytes local memory. The running time reported in all cases is less than 100 seconds. The experimental results show that the SAI preconditioners based on *a priori* sparsity pattern provides a more robust and efficient parallel preconditioning technique than the BBD preconditioners for the brain diffusion simulation problem. It is the SAI preconditioner whose convergence performance is not affected by the number of processors employed, although both the SAI and BBD preconditioners demonstrate a good speedup which is close to linear. The SAI preconditioners take more CPU time to construct, but need less memory space to store, than the BBD preconditioners. The numerical tests also illustrate that the best performance of the preconditioners can be obtained by choosing optimum values for their corresponding parameters, τ_1 and τ_2 in SAI, and \mathbf{w}_1 and \mathbf{w}_2

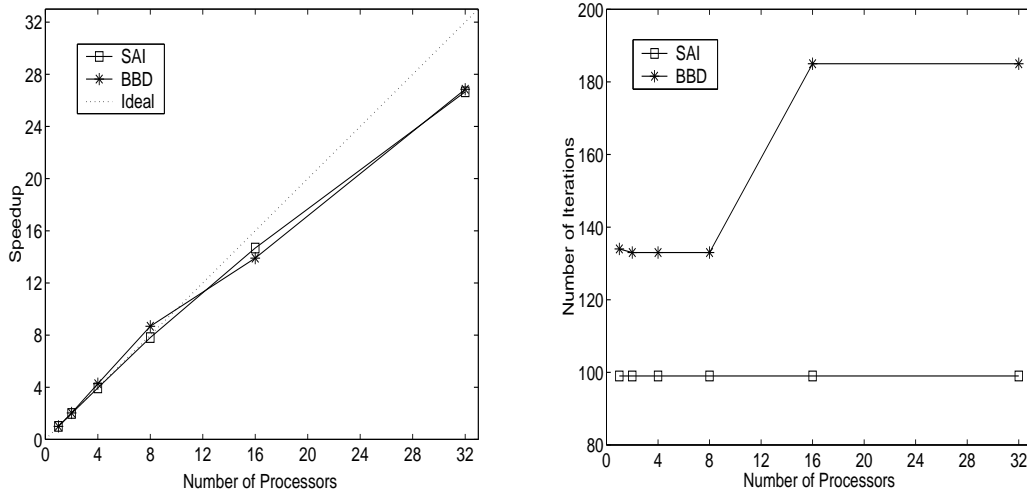


Figure 5: The diagram for scalability comparison of the SAI and BBD preconditioners [35]. Parameters are selected as, for SAI, $\tau_1 = 0.05$, $\tau_2 = 0.001$; for BBD, $w_1 = 8$, $w_2 = 5$. Left: speedup obtained v.s. the number of processors used. Right: the number of iterations v.s. the number of processors used.

in BBD, which have direct and distinct influences on the quality and the construction expense of the preconditioners, the convergence rate of the iterative solutions, and the total computational efforts. The comparison of scalability between the SAI and BBD preconditioners is given in Fig. 5, where there exists superlinear speedups for the BBD preconditioner. This can be attributed to the caching effects. When the problem is dispatched onto multiple processors, the subproblems are obviously a fraction of the original problem size. With a smaller problem size, it is most likely to get a higher cache hit rate, and the result, even after considering the communication time, is still better than the time on a single processor with more cache misses.

3 Brain Deformation Simulation in Image Guided Neurosurgery

3.1 Background

During the past decade, the development of image guided surgery techniques has been playing a crucial role in minimizing invasive neurosurgery. The neurosurgeon is allowed to acquire new images during surgery by exploiting these newly developed and advanced techniques, which are performed in operating rooms equipped with imaging devices for special purposes. These images, as acquired by intraoperative ultrasound or intraoperative magnetic resonance imaging, can provide improved contrast between normal and diseased tissue and enable people to gain better understanding of the brain by exposing deeper structure of the brain through the surface, since the major difficulty a neurosurgeon has to confront is the removal from the brain of as much tumor tissue as possible while minimizing the removal of healthy tissue and preventing the disruption of important

anatomical structures like whiter matter fiber tracts [52].

Image guided neurosurgery heavily relies upon the visualization of the underlying brain tissue structure. Recent reviews by [32] and [52] point out the importance and great potential impact of the image guidance on more invasive surgical procedures. Efforts have been largely focused on the issues concerned with image acquisition, visualization, and registration of intraoperative and preoperative data, in order to supply the surgeon with a better visualization environment from which to derive quantitative assessment of intraoperative imaging data as well as qualitative judgments. Intraoperative image guidance is based on functional integration of the previously acquired and processed three dimensional image-based visual information and the corresponding anatomy of the patient within the same frame of reference, which requires matching of the two frames of reference [32]. The actual coordinates used by the models of surgical planning and simulation must be mapped or registered onto the physical space of the patient. These two ingredients are connected together by combining image-to-patient registration and by tracking instruments within the surgical field. They are the crucial components of frameless stereotactic targeting methods, which capitalize on the interactive control of image planes and explore the full information contained in the perceived three dimensional space. Reliable and completely automated registration techniques are desired to integrate the image-based information with the patient's anatomy. However, as noticed in [32] and [52], the existing methods for rigid registration and visualization impose limitations on image guided treatment process in deep brain structures and with large resections. The restrictions motivate the development of improved approaches for visualization and better algorithms for registration, which can capture the non-rigid deformation of the brain during operation.

During the process of neurosurgery, the shape of the brain gets deformed in response to mechanical and physiological changes related to the surgery. Many different approaches have been proposed to characterize these changes and to construct integrated visualizations of preoperative data in the architecture of the deformed brain, which can be roughly divided into three types. Techniques classified in the first category rely on some form of physics-oriented deformation models, which assume that the image space deformation subjects to the theory of physical laws, such as elasticity in continuum mechanics. The second type of techniques is purely image-based approaches based on some image related constraint criteria [29, 30, 50]. The third category is to use physical deformation models to constrain the deformation field derived from image data using elastic [2, 19] or viscous fluid deformation models [13, 15, 16, 17]. The third category is different from the techniques in the first category in that it does not take into consideration the actual material characteristics of the brain, where the matching is trying to minimize an energy measure that includes a weighted sum of an image similarity term and a relaxation term representing the potential energy of a physical body. This survey will only focus on the techniques in the first category, which takes into account the physical properties and material characteristics of the underlying depicted objects and thus are believed to be able to drastically improve the robustness and accuracy of deformable models [23, 52].

A very good review can be found in [52] for physics-based models in support of the image guided neurosurgery, which range from less physically plausible but very fast models to extremely accurate biomechanical models demanding hours of computational time to get the solution and are developed and implemented under serial computing environments. We first briefly go through a couple of the typical physics-based models for capturing the brain

deformations. Then in the sections followed, the elasticity theory based biomechanical models and simulations are surveyed in more detail, with the focus being put on the ones which have been implemented in parallel under different parallel environments.

3.2 Physics-Based Brain Deformation Models

In [28], a biomechanical model for brain deformation simulation based on linear elasticity theory is proposed to deal with the deformations rising during surgical interventions, by using a finite element discretization method. Two-dimensional implementation of the model is carried out with two different materials being distinguished, namely brain tissue and skull bone. In stead of applying forces in the model, a set of manually determined correspondences is used to drive the deformation of the preoperative image. The finite element mesh is constructed by using four-node quadrilateral finite elements, in accordance with the underlying pixel grid of 2D images. The registration experiments are performed for the case of synthetic images as well as for real brain image data, with a preoperative MR image of the human brain and a corresponding postoperative image simulating an intraoperative image. The tests from both cases yield physically plausible deformation results. The time cost to process an image with a size of 256×256 pixels is around 45 minutes on a Sun Ultra 2/1300 machine with a 300MHz CPU.

An algorithm attempting to track the intraoperative tissue deformation is presented in [21], which is based on a simple 2D three component model consisting of rigid, deformable, and fluid regions. The deformable region is allowed to have a number of energy characteristics, such as stiffness and tension energies, and a fold avoidance strategy is used in the algorithm. The clinical data are utilized to carry out the tests. Their multigrid implementation on 2D images with a size of 128×128 pixels takes about 120-180 minutes to obtain the solution when the test is run on a Sun Sparc 20 machine.

The Kelvin solid model, a simplified homogeneous brain tissue material model, is exploited by the model proposed in [48] for the interconnection of mass nodes to capture real-time brain shift under gravity, with boundary conditions to simulate the interaction of the brain with the skull. The surface displacement is calculated by tracking the brain surface points in this approach. In the discretized deformation model, there are 2088 nodes, 11733 connections, and 1521 brick elements, and the time required to simulate the system typically runs in 10 minutes on an SGI Octane R10000 machine with one 250MHz processor.

[22] presents an integrated approach that uses surface-based correspondences to drive a biomechanical model of the brain in stead of using estimates of forces, which allows the analysis of the characteristics of the deformed tissues, such as stress measures. More specifically, the biomechanical model is driven by imposing displacements to critical boundary surfaces. The displacements at the boundary surfaces are calculated using an active surface matching algorithm, which determines the correspondences between landmark surfaces in the preoperative and intraoperative images. The finite element method is used for discretization which results in approximately one hundred thousand tetrahedra in the finite element mesh. Under such a size of the mesh, the running time of the entire deformation algorithm is around 30 minutes on a Sun Ultra 10 workstation with one 440MHz processor, while with four CPUs, the computation time is about 10 minutes.

A three-dimensional physical model adapted from consolidation theory in soil mechan-

ics is developed in [37, 38, 39, 44], which treats the brain as a solid tissue filled with interstitial fluid where the tissue displacement is characterized by an instantaneous deformation at the contact area followed by subsequent additional deformation due to interstitial fluid draining in the direction of strain-induced pressure gradients. [39] augments the technique to incorporate the simulation of brain tissue retraction and resection. This computational model, combined with the intraoperatively acquired data which is usually incomplete and sparse, is exploited to update high resolution preoperative images and then to estimate the brain deformation in vivo through a series of controlled repeat-experiments during surgery. The finite element method is applied to discretize the model and the intraoperative computation with the real human brain image data takes about 5-18 minutes to get the numerical solutions in a 3D tetrahedra mesh with 19000 nodes.

3.3 Biomechanical Models with Parallel Implementation

3.3.1 Elastic Deformation of Volumetric Solid

A real-time model for surgery simulation to calculate the elastic deformation of a volumetric solid is developed and described in [12]. The mesh-based 3D finite element scheme with linear elastic deformation is exploited to model the general elastic volumetric deformation. The strain energy of a linear elastic solid Ω is defined as:

$$E(\mathbf{u}) = \frac{1}{2} \int \int \int_{\Omega} \boldsymbol{\epsilon}^T \boldsymbol{\sigma} d\mathbf{x}, \quad (11)$$

where \mathbf{u} is the displacement of particle $\mathbf{x} \in \Omega$, $\boldsymbol{\epsilon}$ is the strain vector, and $\boldsymbol{\sigma}$ is the stress vector which is related to the strain vector through Hook's law by $\boldsymbol{\sigma} = \mathbf{D} \cdot \boldsymbol{\epsilon}$, where \mathbf{D} is a symmetric 6×6 material stiffness matrix. Then the domain Ω of the volumetric solid is discretized into a number of finite elements in the form of four-node tetrahedrons with linear interpolation of the displacement field between the nodes. The solution to the deformation problem can be found when the potential energy of the system takes its minimum value, which happens when $\delta E(\mathbf{u}) = \mathbf{0}$, i.e., the first variation of the functional E vanishes. This results in a large sparse linear system

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \quad (12)$$

where \mathbf{K} is the stiffness matrix and \mathbf{f} is the force vector. The linear system (12) models the behavior of the solid object and includes both surface nodes as well as the internal nodes of the model. Since this work only concerns speed instead of precision and memory cost, in the actual simulation, condensation is also used to convert the volumetric model into a model with only surface nodes, which is achieved by removing the internal nodes from the linear equation system (12). And numerical experiments are performed with explicit inversion, Conjugate Gradient method with and without preconditioner, Gauss elimination, and several factorization techniques such as QR and Cholesky. Mass and damping are added to the model in order to have a physically correct model of the solid, which leads to a Lagrangian dynamic system. An alternative simulation method, Selective Matrix Vector Multiplication method, is also developed which is considerably faster for sparse force vectors based on the idea that uses the original static linear system (12) instead of the Lagrangian dynamic system and exploits the sparse structure of the force vector.

The simulation system is implemented on a Silicon Graphics ONYX with four MIPS R4400 processors using SGI Performer graphics library. The actual deformation simulation system runs on a single processor although the entire system is a parallel system and the parallel feature is only used to separate rendering from simulation. A model of the lower leg is used as an example to perform the real-time simulation of solid volumetric deformable models. Performance of the dynamic simulation methods are determined only by the size of the linear system, while it is more difficult to predict the performance of the methods using the Selective Matrix Vector Multiplication scheme [12]. By way of the dynamic simulation scheme, the rate of 20 frames/second is achieved with up to 250 nodes in the system equation. The model presented in [12] is applicable for real-time surgical simulation where video frame rates are required.

[11] also employs the finite element model of linear elasticity to simulate the elastic deformation of a volumetric solid object in real-time. Condensation is also used to remove the internal nodes from the linear matrix system (12), which has the same size as would result from a finite element surface model. The condensed problem shows exactly the same behavior for the surface nodes as the original solid volumetric system. The Selective Matrix Vector Multiplication approach is applied to solve the linear system (12), which is based on the fact that only a few positions of the force vector \mathbf{f} are nonzero. Since (12) is solved by using the inverted stiffness matrix \mathbf{K} as $\mathbf{u} = \mathbf{K}^{-1}\mathbf{f}$, the scheme will reduce the complexity to $\mathcal{O}(n/N)$ times the time of a normal matrix vector multiplication if n of the N positions in \mathbf{f} are nonzero. The solution of the linear system (12) is implemented in parallel using domain decomposition, where the domain of the solid Ω is decomposed into a number of non-overlapping sub-domains Ω_i with a common boundary Γ . The nodes of the global stiffness matrix \mathbf{K} are ordered in a way such that the nodes on the boundary Γ are ordered first, followed by sections of nodes corresponding to the subdomains Ω_i . Then the condensation technique is used to separate the computation of the individual subdomains, each of which can be handled by one processor with the result reduced to a root processor. The simulation system is implemented on the same platform with the same physical organ data set, the lower leg as used in [12], but no result or performance data is reported under the parallel environment.

3.3.2 Elastic Deformation Model Integrated with Optical Flow

In [23], an integrated hybrid approach is proposed to implicitly compute the forces applied to the 3D deformation model by constraining the deformation field to satisfy both the elasticity model and the model based on optical flow methods, where the objects are modeled as elastic bodies. The optical flow methods for recovering image deformation are largely based on local image structure [7, 20], which simulates the deformation field between images to satisfy some pre-chosen smoothness constraints by minimizing local similarity criteria. The satisfaction with both models is achieved in [23] by embedding an image similarity constraint on the deformation field into the minimization scheme which results in the constitutive equations of the deformation model. The main difference between the hybrid technique and the purely image-based optical flow method is that the former technique takes into account the actual physical properties of the deformed objects to be deformed and allows the thorough investigation of the characteristics of the deformed objects by providing information about the physically realistic deformation field, while the

latter one does not. Another difference is that the hybrid technique computes a global solution to the deformation field instead of a collection of local solutions as the optical flow method does. The integrated model could be very useful for the inspection of stresses induced by the deformation of certain objects on their surroundings, which might be used to predict deformations due to the growth of tumors and brain shift during neurosurgery. We take a closer look at the hybrid model [23] in the next paragraphs.

The fundamental step behind the integrated approach is to formulate the elastic matching of two images as an energy minimization procedure, where the energy consists of two terms. One term is used to simulate the physical behavior of the object to be deformed, while the other term drives the model in order to make both images match. The deformation field minimizes the sum of the squared differences between the images to be matched and at the same time is constrained by the biomechanical properties of the different objects represented by the image.

Under the assumption of a linear elastic continuum with no initial stresses or strains, the potential energy of an elastic body subject to externally applied forces can be described as

$$E = \int_{\Omega} \boldsymbol{\sigma}^T \boldsymbol{\epsilon} d\Omega + \int_{\Omega} \mathbf{F} \mathbf{u} d\Omega, \quad (13)$$

where \mathbf{F} is the force vector applied to the elastic body, \mathbf{u} the displacement vector, and Ω the elastic body. The strain vector, $\boldsymbol{\epsilon}$, is defined as

$$\boldsymbol{\epsilon} = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial u}{\partial z}, \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}, \frac{\partial u}{\partial y} + \frac{\partial u}{\partial z}, \frac{\partial u}{\partial z} + \frac{\partial u}{\partial x} \right)^T = \mathbf{L} \mathbf{u}. \quad (14)$$

The constitutive equation of the material regulates the relationship between the stress vector $\boldsymbol{\sigma}$ and the strain vector $\boldsymbol{\epsilon}$, and in the case of linear elasticity without initial stresses or strains, it is expressed as

$$\boldsymbol{\sigma} = (\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}, \tau_{xz})^T = \mathbf{D} \boldsymbol{\epsilon}, \quad (15)$$

where \mathbf{D} is the elasticity matrix characterizing the properties of the material.

The deformation field governed by the elastic model is first initialized to be the optical flow field since the external forces \mathbf{F} can be calculated as a typical optical flow field between the images to be matched (\mathbf{I}_1 and \mathbf{I}_2). The initial estimate can then be iteratively refined until an equilibrium is reached by a semi-implicit method. In [23], the deformation field is computed to simultaneously satisfy both the elasticity constraint and a local image similarity constraint between the images \mathbf{I}_1 and \mathbf{I}_2 , to avoid the separate computation of the forces \mathbf{F} , the elastic deformation, and the matching criterion. Thus Equation (13) can be expressed as

$$E = \int_{\Omega} \boldsymbol{\sigma}^T \boldsymbol{\epsilon} d\Omega + \int_{\Omega} (\mathbf{I}_1(\mathbf{x} + \mathbf{u}(\mathbf{x})) - \mathbf{I}_2(\mathbf{x}))^2 d\Omega, \quad (16)$$

where $\mathbf{I}_1(\mathbf{x} + \mathbf{u}(\mathbf{x}))$ is approximated by the first order Taylor expansion since in [23] the deformation field is assumed to be small relative to the variation of \mathbf{I}_1 smooth.

Finite-element based scheme is exploited in [23] to solve Equation (16) and compute the deformation field which corresponds to the global minimum of the total energy. Within their finite element discretization framework, the physical domain is discretized by using

tetrahedral elements that are simpler than other elements according to the shape function and data structure, with linear interpolation of the displacement field. The displacements are a function of the displacement at the element's nodal points weighted by the element's shape function. The elastic body can then be approximated as an assemblage of discrete tetrahedral elements interconnected at nodal points on the element boundaries. The mesh structure is constructed in the way such that for images containing multiple objects, a fully connected and consistent tetrahedral mesh is generated for each cell, with a given label corresponding to the object the cell belongs to. By this way various biomechanical properties are assigned without difficulty to the different cells or objects composing the mesh.

After the implementation of finite element algorithm, the linear system is then solved in parallel by using the PETSc (Portable, Extensible Toolkit for Scientific Computation) library, which provides a collection of numerical software tools for solving large-scale sparse nonlinear systems of equations in parallel and relies on the Message Passing Interface (MPI) for interprocessor communication (<http://acts.nersc.gov/petsc/main.html>). The timing of the matching algorithm, which uses a tetrahedra mesh with the size of approximately a hundred thousand, is reported to be around a couple of minutes when the simulation is carried out on a parallel machine with 20 Ultra Sparc II 250MHz CPUs. Synthetic image data as well as real medical data are used in the experiments to test the proposed integrated model, where the objects are considered to be homogeneous elastic bodies. The tested medical applications include the muscle exercise imaging and ventricular deformation in multiple sclerosis obtained from 3D brain MRI images. In these examples, the hybrid model is well suited for the small biomechanical deformations which deform the shape of objects over time.

3.3.3 Linear Elastic Model

[52] presents an algorithm to create enhanced visualization of tumor and critical brain structure during neurosurgery by combining both preoperatively generated surface models of the corticospinal tract from an anatomical atlas with intraoperative MRI and the biomechanical model inferring the 3D volumetric deformation of the brain and ventricles of the patient. The algorithm tracks the surface changes of key structures in intraoperatively acquired brain images by aligning preoperatively acquired image data with intraoperative images of the patient's brain as its shape changes during the neurosurgical procedure. A volumetric deformation field is derived from the shape changes to capture the non-rigid deformations of the shape of the brain in response to mechanical and physiological changes associated with the surgery. An enhanced and integrated visualization is then constructed by exploiting the solution to the deformation field simulated by the biomechanical model and the projection of a preoperatively built model of the brain into the configuration of the deformed brain.

In [52], the biomechanical simulation of the volumetric brain deformation is conducted by modeling the brain as an elastic object with homogeneous linear elasticity. When there is no initial stresses or strains, the deformation energy of this elastic body submitted to externally applied forces is modeled as

$$\mathbf{E} = \frac{1}{2} \int_{\Omega} \boldsymbol{\sigma}^T \boldsymbol{\epsilon} d\Omega + \int_{\Omega} \mathbf{F}^T \mathbf{u} d\Omega, \quad (17)$$

where \mathbf{F} is the vector denoting the forces applied to the elastic body, \mathbf{u} the unknown displacement vector field, Ω the elastic body, $\boldsymbol{\epsilon}$ the strain vector, and $\boldsymbol{\sigma}$ the stress vector. $\boldsymbol{\epsilon}$ and $\boldsymbol{\sigma}$ are related to each other by the constitutive equations of the material, Equation (15). Strain is linked to displacement by the assumption $\boldsymbol{\epsilon} = \mathbf{L}^T \mathbf{u}$ where \mathbf{L} is a linear operator.

The finite element method is used to generate a volumetric unstructured tetrahedral mesh in the image domain, on which the biomechanical simulation of deformation is performed. The volumetric deformation of the brain is obtained by solving the displacement field that minimizes the deformation energy described by Equation (17). This yields a linear equation system, which is solved for the displacements due to the forces applied to the elastic body

$$\mathbf{K} \mathbf{u} = -\mathbf{F}, \quad (18)$$

where \mathbf{K} is the stiffness matrix. Equation (18) is solved in a way such that the derived deformation field over the entire mesh matches the prescribed displacements at the boundary surfaces by fixing the displacements at the boundary surface nodes to match those generated by the active surface model. The entries of the rows in \mathbf{K} corresponding to the nodes at which a displacement is to be imposed are set to zero and the diagonal elements of these rows are set to one. The force vector \mathbf{F} is set to be the displacement vector to be imposed at the boundary nodes.

In order to achieve the timing constraint set by the real-time situation of neurosurgery as well as robustness and high accuracy in the matching of brain image data, the linear equation system (18) is implemented and solved in parallel with the PETSc (Portable, Extensible Toolkit for Scientific Computation) package using the GMRES (Generalized Minimal Residual) solver with a block Jacobi preconditioning scheme. The rows of the global stiffness matrix \mathbf{K} are divided equally among the different processors such that each processor has an equal number of rows to compute and it assembles the local \mathbf{K}^e matrix for each element in its subdomain. After the global matrix \mathbf{K} is assembled in parallel, it is adjusted to reflect the enforced boundary conditions determined by the surface matching. The total number of mesh nodes is 43584, which specifies 214035 tetrahedral elements and represents a system of 130752 unknown displacements to be identified. The brain deformation simulation during neurosurgery is carried out on a Sun Microsystems StarFire 6800 symmetric multiple-processor machine with 12 UltraSPARC-III processors running at 750MHz and 12 GB of RAM.

A set of parallel scaling experiments are performed to demonstrate the scaling characteristics of the implementation with the timings for solving the biomechanical model under serial and parallel environment reported as well as good simulation results of volumetric brain deformation. In the case of serial computing environment, i.e., on a single processor, it takes 17 seconds to assemble the system of equations, 37 seconds to get the solution of the linear system of equations, and 57 seconds for the overall solution time which includes the time for initialization. Under the parallel environment, 12 processors are used and the parallel assembly of system of equations costs 3 seconds, solving the linear system costs 9 seconds, and the overall solution time requires 15 seconds. In [52], a context is provided to compare the timings for the biomechanical simulation with the other intraoperative image analysis and acquisition tasks that are intraoperatively indispensable. The results shows that the most computationally demanding task, the deformation simulation, becomes the

fastest component while the complete execution time for the intraoperative image analysis is less than 10 minutes. Their experiments demonstrate the capability of the algorithm to meet the real-time constraints of neurosurgery and even a relatively complicated biomechanical simulation of brain deformation could be simulated sufficiently fast at a rate that is practical for routine use during neurosurgery.

3.3.4 Volumetric Biomechanical Model with *a priori* Knowledge

A patient-specific biomechanical model based on block-matching to register MRI images of the same patient with a parallel implementation is illustrated in [47], which combines a viscoelastic mechanical regularization with a correlation-based iconic energy term. It relies on a dynamical law of motion of a tetrahedral mesh, which leads to iteratively solving a linear system of equations.

In the registration algorithm of [47], *a priori* anatomical and mechanical knowledge is integrated through a volumetric biomechanical model of the brain. In the deformable model framework, a balance between mechanical and iconic influences is considered during the full registration process through an energy minimization. The internal energy which manipulates the displacement field is based upon linear elasticity in continuum mechanics. The volumetric external energy based on 3D block matching is used in their biomechanical model. With such a volumetric external energy, full motion of the organ can be registered, more complicated deformation can be estimated inside the model, and the quality of the registration does not count on the surface segmentation accuracy. The external energy exploited is relied on the assumption that the intensity around a given vertex of the mesh should be similar in the two images with the difference only coming from noise and anatomical changes. A surrounding block of voxels from the reference image is attached to each vertex and the most similar block around the current position is searched in the target image. The center of the block is chosen to be the matching voxel, which maximizes the prescribed similarity criterion. The external force calculated for each voxel is then proportional to the distance to the matching voxel and its orientation is from the vertex current position to the position of the matching voxel.

The inertia effect is also taken into consideration in the brain deformation model by using a dynamic approach, which is efficient and stable to find a balance between external and internal energies. The dynamic equation of motion is described as

$$\mathbf{M} \frac{d^2 \mathbf{U}}{dt^2} + \mathbf{C} \frac{d\mathbf{U}}{dt} + \mathbf{K}\mathbf{U} = \mathbf{F}, \quad (19)$$

where \mathbf{M} is the mass density matrix, \mathbf{C} the damping matrix, \mathbf{K} the stiffness matrix, \mathbf{U} the displacement from rest position ($\mathbf{P}(t) = \mathbf{P}(0) + \mathbf{U}(t)$, with $\mathbf{P}(t)$ the position at time t), and \mathbf{F} the external force vector computed from the image matching.

The finite element method is used to discretize the physical domain, with linear tetrahedral elements in space and the semi-implicit Houbolt scheme in time. The discretization results in a linear system to solve in $\mathbf{U}(t + \Delta t)$ at each iteration. The stiffness matrix \mathbf{K} is chosen to be constant with the assumption of linear elasticity. The numerical integration of the discretized time-dependent system involves two steps, the initialization step and the procedure of solving the linear system. The initialization procedure includes matrix assembly, the effective stiffness matrix calculation, $\bar{\mathbf{K}} = \mathbf{K} + \mathbf{a}_0 \mathbf{M} + \mathbf{a}_1 \mathbf{C}$ with \mathbf{a}_0 and

\mathbf{a}_1 being constants, and the preconditioner matrix construction for $\bar{\mathbf{K}}$ which is performed only once for the overall process. Then at each time step, a linear system of equations, $\bar{\mathbf{K}}\mathbf{U}(t + \Delta t) = \bar{\mathbf{F}}(t + \Delta t)$, is iteratively solved with the computed effective external forces $\bar{\mathbf{F}}(t + \Delta t)$, and then the nodes positions are updated. The parallelization of the registration process uses a distributed memory model, where each processor computes the biomechanical and image forces in a subdomain of the global mesh. The tetrahedral mesh is partitioned using the METIS software (www-users.cs.umn.edu/~karypis/metis) and each vertex of the mesh is assigned to a processor in such a way to minimize the communication cost. Each processor computes the external forces for the vertices corresponding to the associated submesh.

The PETSc library is applied to solve the linear system of equations in parallel. The stiffness matrix is computed, preconditioned, and distributed at the beginning of the process. Several different iterative solvers, CG (Conjugate Gradient), GMRES, and BICGSTAB (BIConjugate Gradient STABilized), with the ILUT and IC (Incomplete Cholesky) preconditioners are tested and compared for a mesh of 2000 nodes on a PC equipped with an Intel Pentium IV 2GHz processor. It is found that BICGSTAB with ILUT has the least number of iterations and the shortest total solution time, while CG with IC has the largest number of iterations and the longest total solution time and GMRES with ILUT sits in between on both the number of iterations and the total solution time.

The performance of the algorithm is evaluated on synthesized images as well as on real brain images. The experiments are conducted on a cluster of PCs consisting of 14 Intel Pentium IV 2 GHz processors interconnected by a Gigabit-Ethernet switch.

In the case of synthesized image data, the underlying tetrahedral mesh of the brain contains around 2000 vertices with block-matching searching range being $5mm$ and searching step $2mm$. The relationship between the number of processors and the speedup of one time step shows that the speedup decreases fast, due to the relatively coarse mesh used in the test. The influence of the damping factor on the convergence of the algorithm is also investigated. It demonstrates that the dynamic effects of the damping factor is critical in the registration computation process, where a small value of it leads to an instability whereas a large value results in a large computation time. The optimum damping factor for this case is found to be 0.015, by which the error tolerance is reached in less than 3 minutes on 14 processors. The analysis of the robustness of the algorithm to noise is presented as well, which shows that it performs well with a high degree of noise (white Gaussian noise on intensity) on images with a similarity measure of the correlation coefficient.

In the case of tests on real brain images, there are 30000 vertices in the tetrahedral mesh with block-matching searching range being $5mm$ and searching step $2mm$. Each image is computed in less than a minute. The results show that the volumetric biomechanical registration is well suited to recover a physical deformation and a good capability of deforming internal structure illustrated in the difference image. The efficiency of the volumetric external energy is also demonstrated by the ability to recover the surface displacement with only internal constraints. However, there are no tests reported on timing, damping factor, and robustness comparisons with the real brain image data.

4 Summary

Parallel computation in simulating diffusion and deformation of the human brain is a new field of exploration. The driving force in use of parallel computing in these biomedical applications is the need to results in real-time speed. The response time is of vital importance in image guided neurosurgery, and more accurate simulations usually demand more computing time.

It can be said that parallel computation in biomedical applications is still in its early development stage, with the gradual realization of the importance of reducing computing time in simulations in the application community. With more sophisticated and complicated simulations to be designed in the future and the need for fast and high definition imaging techniques, parallel computation will undoubtedly find its way in large scale biomedical applications. There are ample opportunities for computational scientists to work with biomedical scientists to develop high performance computing tools and platforms for biomedical applications.

References

- [1] B.A. Ardekani, J. Nierenberg, M.J. Hoptman, D.C. Javitt, and K.O. Lim, “MRI Study of White Matter Diffusion Anisotropy in Schizophrenia”, *Neuroreport*, 14(16):2025-2029, Nov 2003.
- [2] R. Bajcsy and S. Kovacic, “Multi-Resolution Elastic Matching”, *Computer Vision, Graphics, and Image Processing*, 46:1-21, 1989.
- [3] P.J. Basser, “Inferring Microstructural Features and the Physiological State of Tissues From Diffusion-Weighted Images”, *NMR in Biomedicine*, 8:333-344, 1995.
- [4] P.J. Basser, J. Mattiello, and D. Le Bihan, “Estimation of the Effective Self-Diffusion Tensor from the NMR Spin Echo”, *Journal of Magnetic Resonance, Series B*, 103:247-254, 1994.
- [5] P.G. Batchelor, D.L.G. Hill, F. Calamante, and D. Atkinson, “Study of Connectivity in the Brain Using the Full Diffusion Tensor from MRI”, *Information Processing in Medical Imaging*, 17th International Conference, IPMI’01, June 2001, UC Davies, USA. Published by Springer, *Lecture Notes in Computer Science 2082*, pp. 121-133.
- [6] P.G. Batchelor, D.L.G. Hill, D. Atkinson, F. Calamante, and A. Connelly, “Fibre-Tracking by Solving the Diffusion-Convection Equation”, *Proc Intl Soc Mag Res Med*, 2002.
- [7] S. Bauchemin and J.L. Barron, “The Computation of Optical Flow”, *ACM Computing Surveys*, 27(3), 1995.
- [8] C. Beaulieu, “The Basis of Anisotropic Water Diffusion in the Nervous System - a Technical Review”, *NMR in Biomedicine*, 15:435-455, 2002.
- [9] A.B.M. Björnemo, *White Matter Fiber Tracking Using Diffusion Tensor MRI*, Master’s Thesis, Linköping University, Sweden, 2002.

- [10] K.E. Brenan, S.L. Campbell, and L.R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, SIAM, Philadelphia, PA, 1996.
- [11] M. Bro-Nielsen, "Surgery Simulation Using Fast Finite Elements", *Visualization in Biomedical Computing*, Heidelberg, Germany, pp. 529-534, Springer-Verlag, 1996.
- [12] M. Bro-Nielsen and S. Cotin, "Real-Time Volumetric Deformable Models for Surgery Simulation Using Finite Elements and Condensation", *Proc Eurographics'96*, J. Rossignac and F. Sillion (Guest Editors), Blackwell Publishers, 15(3):57-66, 1996.
- [13] M. Bro-Nielsen and C. Gramkow, "Fast Fluid Registration of Medical Images", *Visualization in Biomedical Computing (VBC'96)*, pp. 267-276, 1996.
- [14] E. Chow, "A Priori Sparsity Patterns for Parallel Sparse Approximate Inverse Preconditioners", *SIAM J. Sci. Comput.*, 21:1804-1822, 2000.
- [15] G.E. Christensen, S.C. Joshi, and M.I. Miller, "Volumetric Transformation of Brain Anatomy", *IEEE Transactions on Medical Imaging*, 16(6):864-877, 1997.
- [16] G.E. Christensen, M.I. Miller, M. Vannier, and U. Grenander, "Individualizing Neuroanatomical Atlases Using a Massively Parallel Computer", *IEEE Computer*, 29(1):32-38, 1996.
- [17] G.E. Christensen, R.D. Rabbit, and M.I. Miller, "Deformable Templates Using Large Deformation Kinematics", *IEEE Transactions on Image Processing*, 5(10): 1435-1447, 1996.
- [18] T.E. Conturo, N.F. Lori, T.S. Cull, E. Akbudak, A.Z. Snyder, J.S. Shimony, R.C. McKinstry, H. Burton, and M.E. Raichle, "Tracking Neuronal Fiber Pathways in the Living Human Brain", *Proc. Natl. Acad. Sci. USA*, Vol. 96: 10422-10427, August 1999.
- [19] C. Davatzikos, "Spatial Transformation and Registration of Brain Images Using Elastically Deformable Models", *Computer Vision and Image Understanding*, 66(2):207-222, 1997.
- [20] J. Dengler and M. Schmidt, "The Dynamic Pyramid - a Model for Motion Analysis with Controlled Continuity", *International Journal of Pattern Recognition and Artificial Intelligence*, 2:275-288, 1988.
- [21] P.J. Edwards, D.L.G. Hill, J.A. Little, and D.J. Hawkes, "Deformation for Image Guided Interventions Using a Three Component Tissue Model", in *IPMI'97*, pp. 218-231, 1997.
- [22] M. Ferrant, S.K. Warfield, A. Nabavi, B. Macq, and R. Kikinis, "Registration of 3D Intraoperative MR Images of the Brain Using a Finite Element Biomechanical Model", in *MICCAI 2000: Third Int'l Conf on Medical Robotics, Imaging, and Computer Assisted Surgery*, A.M. DiGioia and S. Delp eds., pp. 19-28, Springer-Verlag, 2000.

- [23] M. Ferrant, S.K. Warfield, C.R.G. Guttmann, R.V. Mulkern, F.A. Jolesz, and R. Kikinis, “3D Image Matching Using a Finite Element Based Elastic Deformation Model”, *MICCAI 99: Second International Conference on Medical Image Computing and Computer-Assisted Intervention; 1999 Sep 19-22; Cambridge, England*, C. Taylor and A. Colchester eds., Heidelberg, Germany, pp. 202-209, Springer-Verlag, 1999.
- [24] J. Foong, M. Maier, C.A. Clark, G.J. Barker, D.H. Miller, and M.A. Ron, “Neuropathological Abnormalities of the Corpus Callosum in Schizophrenia: a Diffusion Tensor Imaging Study”, *Journal of Neurol Neurosurg Psychiatry*, 68(2):242-4, 2000.
- [25] D. Gembris, H. Schumacher, and D. Suter, “Solving the Diffusion Equation for Fiber Tracking in the Living Human Brain”, *Proc. of the International Society for Magnetic Resonance Medicine (ISMRM)*, 9:1529, Glasgow, Scotland, April 2001.
- [26] M. J. Grote and T. Huckle, “Parallel Preconditioning with Sparse Approximate Inverses”, *SIAM J. Sci. Comput.*, 18:838-853, 1997.
- [27] M. Guye, G.J.M. Parker, M. Symms, P. Boulby, C.A.M. Wheeler-Kingshott, A. Salek-Haddadi, G.J. Barker, and J.S. Duncan, “Combined Functional MRI and Tractography to Demonstrate the Connectivity of the Human Primary Motor Cortex in vivo”, *NeuroImage*, 19:1349-1360, 2003.
- [28] A. Hagemann, K. Rohr, H.S. Stiel, U. Spetzger, and J.M. Gilsbach, “Biomechanical Modeling of the Human Head for Physically Based, Non-Rigid Image Registration”, *IEEE Transactions on Medical Imaging*, 18(10):875-884, 1999.
- [29] N. Hata, A. Nabavi, W.M. Wells, S.K. Warfield, R. Kikinis, P.M. Black, and F.A. Jolesz, “Three-Dimensional Optical Flow Method for Measurement of Volumetric Brain Deformation from Intraoperative MR Images”, *Journal of Computer Assisted Tomography*, 24:531-538, July, 2000.
- [30] D. Hill, C. Maurer, R. Maciunas, J. Barwise, J. Fitzpatrick, and M. Wang, “Measurement of Intraoperative Brain Surface Deformation under a Craniotomy”, *Neurosurgery*, 43:514-526, 1998.
- [31] A.C. Hindmarsh and A.G. Taylor, *User Documentation for IDA, a Differential-Algebraic Equation Solver for Sequential and Parallel Computers*, LLNL Report UCRL-MA-136910, Center for Applied Scientific Computing, LLNL, Livermore, CA, 1999.
- [32] F. Jolesz, “Image-Guided Procedures and the Operating Room of the Future”, *Radiology*, 204:601-612, 1997.
- [33] N. Kang, J. Zhang, E.S. Carlson, and D. Gembris, “White Matter Fiber Tractography via Anisotropic Diffusion Simulation in the Human Brain”, Technical Report No. 410-04, Department of Computer Science, University of Kentucky, Lexington, KY, 2004.
- [34] N. Kang, J. Zhang, and E.S. Carlson, “Performance of ILU Preconditioning Techniques in Simulating Anisotropic Diffusion in the Human Brain”, *Future Generation Computer Systems*, 20(4): 687-698, 2004.

- [35] N. Kang, J. Zhang, and E.S. Carlson, “Parallel Simulation of Anisotropic Diffusion with Human Brain DT-MRI Data”, *Computers and Structures*, in press.
- [36] D. Le Bihan, J.F. Mangin, C. Poupon, C.A. Clark, S. Pappata, N. Molko, and H. Chabriat, “Diffusion Tensor Imaging: Concepts and Application”, *Journal of Magnetic Resonance Imaging*, 13:534-546, 2001.
- [37] M.I. Miga, K.D. Paulsen, P.J. Hoopes, F.E. Kennedy, Jr., A. Hartov, and D.W. Roberts, “In vivo Quantification of a Homogeneous Brain Deformation Model for Updating Preoperative Images During Surgery”, *IEEE Transactions on Biomedical Engineering*, 47(2):266-273, 2000.
- [38] M.I. Miga, K.D. Paulsen, F.E. Kennedy, P.J. Hoopes, A. Hartov, and D. W. Roberts, “In vivo Analysis of Heterogeneous Brain Deformation Computations for Model-Updated Image Guidance”, *Computer Methods in Biomechanics and Biomedical Engineering*, 3(2):129-146, 2000.
- [39] M.I. Miga, D.W. Roberts, F.E. Kennedy, L.A. Platenik, A. Hartov, K.E. Lunn, and K.D. Paulsen, “Modeling of Retraction and Resection for Intraoperative Updating of Images”, *Neurosurgery*, 49:75-85, 2001.
- [40] S. Mori and P.C.M. van Zijl, “Fiber Tracking: Principles and Strategies - a Technical Review”, *NMR in Biomedicine*, 15:468-480, 2002.
- [41] C. Nicholson, “Diffusion and Related Transport Mechanism in Brain Tissue”, *Reports on Progress in Physics*, 64:815-884, 2001.
- [42] C. Nicholson and E. Syková, “Extracellular Space Structure Revealed by Diffusion Analysis”, *Trends in Neurosciences*, 21(5):207-215, 1998.
- [43] L. O’Donnell, S. Haker, C.F. Westin, “New Approaches to Estimation of White Matter Connectivity in Diffusion Tensor MRI: Elliptic PDEs and Geodesics in a Tensor-Warped Space”, *MICCAI*, Tokyo, Japan, September, 2002.
- [44] K.D. Paulsen, M.I. Miga, F.E. Kennedy, P.J. Hoopes, A. Hartov, and D.W. Roberts, “A Computational Model for Tracking Subsurface Tissue Deformation During Stereotactic Neurosurgery”, *IEEE Transactions on Biomedical Engineering*, 46(2):213-225, 1999.
- [45] S.E. Rose, F. Chen, J.B. Chalk, F.O. Zelaya, W.E. Strugnell, M. Benson, J. Semple, and D. Doddrell, “Loss of Connectivity in Alzheimer’s Disease: An Evaluation of White Matter Tract Integrity with Colour Coded MR Diffusion Tensor Imaging”, *Journal of Neurology, Neurosurgery and Psychiatry*, 69:528-530, 2000.
- [46] Y. Saad and M. H. Schultz, “GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems”, *SIAM J. Sci. Stat. Comput.*, 7:856-869, 1986.
- [47] M. Sermesant, O. Clatz, Z. Li, S. Lantéri, H. Delingette, and N. Ayache, “A Parallel Implementation of Non-Rigid Registration Using a Volumetric Biomechanical Model”, J.C. Gee et al. (Eds.): *WBIR 2003 LNCS 2717*, pp. 398-407, 2003.

- [48] O. Skrinjar and J.S. Duncan, “Real Time 3D Brain Shift Compensation”, in *Proc. IPMI'99*, pp. 42-55, Visegrad, Hungary, 1999.
- [49] B. Stieltjes, M. Schluter, H.K. Hahn, T. Wilhelm, and M. Essig, “Diffusion Tensor Imaging: Theory, Sequence Optimization and Application in Alzheimer’s Disease”, *Radiologe*, 43(7):562-565, July 2003.
- [50] S. Tang and T. Jiang, “Nonrigid Registration of Medical Image by Linear Singular Blending Techniques”, *Pattern Recognition Letters*, 25(4):399-405, 2004.
- [51] I. Vorisek and E. Sykova, “Evolution of Anisotropic Diffusion in the Developing Rat Corpus Callosum”, *J. Neurophysiol.*, 78:912-919, 1997.
- [52] S.K. Warfield, F. Talos, A. Tei, A. Bharatha, A. Nabavi, M. Ferrant, P.M. Black, F.A. Jolesz, and R. Kikinis, “Real-Time Registration of Volumetric Brain MRI by Biomechanical Simulation of Deformation during Image Guided Neurosurgery”, *Computing and Visualization in Science*, 5:3-11, 2002.
- [53] J. Xu, *Multidimensional Finite Differencing (MDFD) with Hypersphere-Close-Pack Grids for Numerical Solution of PDE Defined on Irregular Domains*, Ph.D. Thesis, University of Alabama, Tuscaloosa, AL, 2001.