CSRD (2009) 23: 177–183
DOI 10.1007/s00450-009-0080-x

SPECIAL ISSUE PAPER

Parallel scalable PDE-constrained optimization: antenna identification in hyperthermia cancer treatment planning

Olaf Schenk · Murat Manguoglu · Ahmed Sameh · Matthias Christen · Madan Sathe

Published online: 6 May 2009

Abstract We present a PDE-constrained optimization algorithm which is designed for parallel scalability on distributed-memory architectures with thousands of cores. The method is based on a line-search interior-point algorithm for large-scale continuous optimization, it is matrix-free in that it does not require the factorization of derivative matrices. Instead, it uses a new parallel and robust iterative linear solver on distributed-memory architectures. We will show almost linear parallel scalability results for the complete optimization problem, which is a new emerging important biomedical application and is related to antenna identification in hyperthermia cancer treatment planning.

Keywords PDE-constrained optimization · Large-scale parallel optimization · Biomedical application · Saddle-point matrices · Sparse liner solver

Mathematics subject classification (2000) 90C06 · 90C51 · 90C26 · 65F10 · 65F05 · 68W10

O. Schenk (✉) · M. Christen · M. Sathe
Computer Science Department, University of Basel,
Klingelbergstrasse 50,
4056 Basel, Switzerland
e-mail: olaf.schenk@unibas.ch

M. Christen
e-mail: m.christen@unibas.ch

M. Sathe
e-mail: madan.sathe@unibas.ch

M. Manguoglu · A. Sameh
Computer Sciences, University of Purdue,
305 N. University Street,
West Lafayette, IN 4790, USA

M. Manguoglu
e-mail: mmanguog@cs.purdue.edu

A. Sameh
e-mail: sameh@cs.purdue.edu

1 Introduction

Biomedical hyperthermia cancer treatment is a promising therapeutical option in oncology [12]. Various types of cancer can be treated by heating the tumor to about 45 °C using non-ionizing radiation (microwaves). It makes the tumor more susceptible to an accompanying radio or chemotherapy. Heating tumors above a temperature of about 45 °C results in preferential killing of tumor cells and makes them more susceptible to an accompanying radio or chemo therapy. Modern hyperthermia applicators operating at around 100 MHz provide a larger number of antennas for which the amplitude and phase can be controlled independently which permits shifting the focus and preventing hotspots. The optimal hyperthermia treatment planning can be formulated as a nonlinear optimization problem, in which the Pennes Bio-heat equations [13] appear as important constraints – a mathematical task known as *PDE-constrained optimization*. Solving the PDE constrained optimization problem presents a frontier problem in scientific computing. The size, complexity and infinite-dimensional nature of PDE-constrained optimization problems present significant challenges for general purpose optimization algorithms and, typically, Tikhonov regularization, iterative solvers, preconditioning, inexactness and parallel implementations are necessary to cope with the numerical challenges.

For designing an individually optimal therapy, amplitudes and phases of the antennas have to be selected such that the tumor temperature is maximized up to a target therapeutical temperature T_{ther} of 43 °C. In order not to damage healthy tissue, certain temperature constraints have to be respected. The induced temperature distribution is essentially described by the elliptic bio heat transfer equation [13]. The aim is to predict the temperature distribution T (*state variables*) which depends on the complex control of the anten-

nas u_i and E_i (*control variables*). This leads to the following large-scale nonlinear optimal PDE control problem and the goal is to minimize the objective function [12]:

$$F = \int_{x\in\Omega_t} (T_{\text{ther}} - T)^2 \,\mathrm{d}\Omega + \int_{x\notin\Omega_t, T>T_{\text{health}}} (T - T_{\text{health}})^2 \,\mathrm{d}\Omega \tag{1a}$$

subject to

$$-\nabla\cdot(k\nabla T) + \rho_b\rho\omega(T - T_b) = \frac{\rho\sigma}{2}\left|\sum_i u_i E_i\right|^2 \quad \text{in} \quad \Omega \tag{1b}$$

$$k\partial_n T = q_{\text{const}} \quad \text{on} \quad \partial\Omega \tag{1c}$$

$$T|_{\Omega/\Omega_t} < T_{\text{lim}} \tag{1d}$$

$$\min\{|u_i|\} \geq \alpha \max\{|u_i|\}\,. \tag{1e}$$

Here, k is the thermal conductivity, ρ the density (ρ_b: blood density), ω the perfusion rate, σ the electrical conductivity, T the temperature and T_b the arterial blood temperature and these tissue parameters depend also on the temperature. Ω is the part of the patient's body that is affected, $\Omega_t \subset \Omega$ is the domain occupied by tumor tissue. The complex control of antenna is defined by u_i and E_i and the temperature is $T_{\text{ther}} = 43\,^\circ\text{C}$, $T_{\text{health}} = 42\,^\circ\text{C}$ and $T_{\text{lim}} = 44\,^\circ\text{C}$. α depends on the HTP applicator that is used in the therapy.

In our application there is great interest in solving optimization problems of extremely large sizes. Since the constraints of the problem correspond to a discretized biomedical PDE, the accuracy of the optimization solution with respect to this infinite-dimensional problem is directly related to size of the largest discrete approximate problem that can be solved. One possible alternative for large-scale optimization is to reduce the original problem into one of a smaller size through a process of *nonlinear elimination* [9, 23]. This process involves an iteration for determining an optimal set of *control variables*. For each set of controls, the equality constraints in (1b) are solved for the remaining *state variables*, and an auxiliary system may be solved for the sensitivities of the state variables with respect to the controls. The remainder of the iteration involves only the computation of a displacement in the controls. Algorithms of this type, however, suffer from a number of various setbacks. For example, if a large number of iterations are required to find an optimal set of control variables, then such a procedure requires a large number of exact solutions of the equality constraints (a PDE) and the adjoint equations (another set of PDEs).

The challenge is thus to design a constrained optimization algorithm that emulates an efficient nonlinear programming approach. The algorithm may utilize matrix-vector products with the constraint Jacobian, its transpose, and the Hessian of the Lagrangian together with appropriate preconditioners – quantities that are computable for many large-scale applications of interest – but must overcome the fact that exact factorizations of derivative matrices are impractical to obtain. Iterative linear system solvers present a viable alternative to direct factorization methods, but the benefits of these techniques are only realized if inexact step computations are controlled appropriately in order to guarantee global convergence of the algorithm.

Figure 1 shows the parallel biomedical framework of our application. In a first step, the Maxwell equation has to be solved in order to determine the electric magnetic fields. The overall EM fields will be used in the Pennes-Bio heat equation (1b) and the optimal temperature distribution can be computed by changing the phase and amplitude of each antenna of the hyperthermia applicator. Interior point optimization methods will be used for the nonlinear optimization problem. They have been proven to be a very efficient class of methods for inequality constrained finite dimensional optimization. An inexact Newton pathfollowing algorithm is constructed on the base of an efficient inexact scheme in the primal-dual barrier interior-point optimizer `IPOPT` [22] and the parallel linear system solver `PSPIKE`. The framework of inexact Newton methods yields accuracy requirements on the discretization error used to control the Newton iteration.

Recent advances in optimization algorithms [4–7, 22] combined with scalable robust appropriate preconditioners will result in PDE-constrained optimization applications that can often scale to millions of optimization variables.

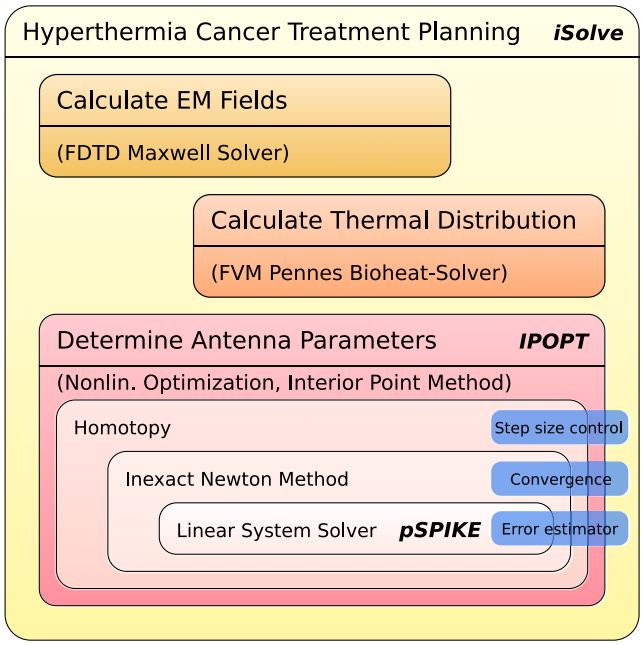

Fig. 1 Overview of the hyperthermia treatment planning process

We note that our method has much in common with the algorithms in [2, 3, 10] as we follow a *full-space*, or *all-at-once*, approach for PDE-constrained problems. The major difference, however, is that we have presented in [7] conditions that guarantee the global convergence of the inexact optimization algorithm. In this paper, we will use this inexact primal-dual barrier interior-point optimizer `IPOPT` [22] and add a new scalable and robust solver – `PSPIKE`. It will be used to solve a large-scale biomedical PDE-constrained optimization application in parallel on distributed-memory architectures. Figure 2 shows different temperature distributions for two different hyperthermia patient models during the optimization process. The results have been computed using the parallel algorithms that are described in the next sections. The underlying framework of the new solver `PSPIKE` is based on the well-known shared-memory parallel sparse direct solver `PARDISO` [18, 19], which typically scales up to eight to sixteen cores. In order to address scalability up to thousands of cores, the new solver `PSPIKE` has been very recently developed and represents a significant extension to existing parallel methods. In this paper, we will show almost linear scalability results for the complete optimization application up to 256 cores.

2 Primal-dual barrier interior-point optimization

After applying a finite-difference discretization, the PDE-constrained optimization problem (1) can be transformed into a nonlinear programming problem (NLP) given by

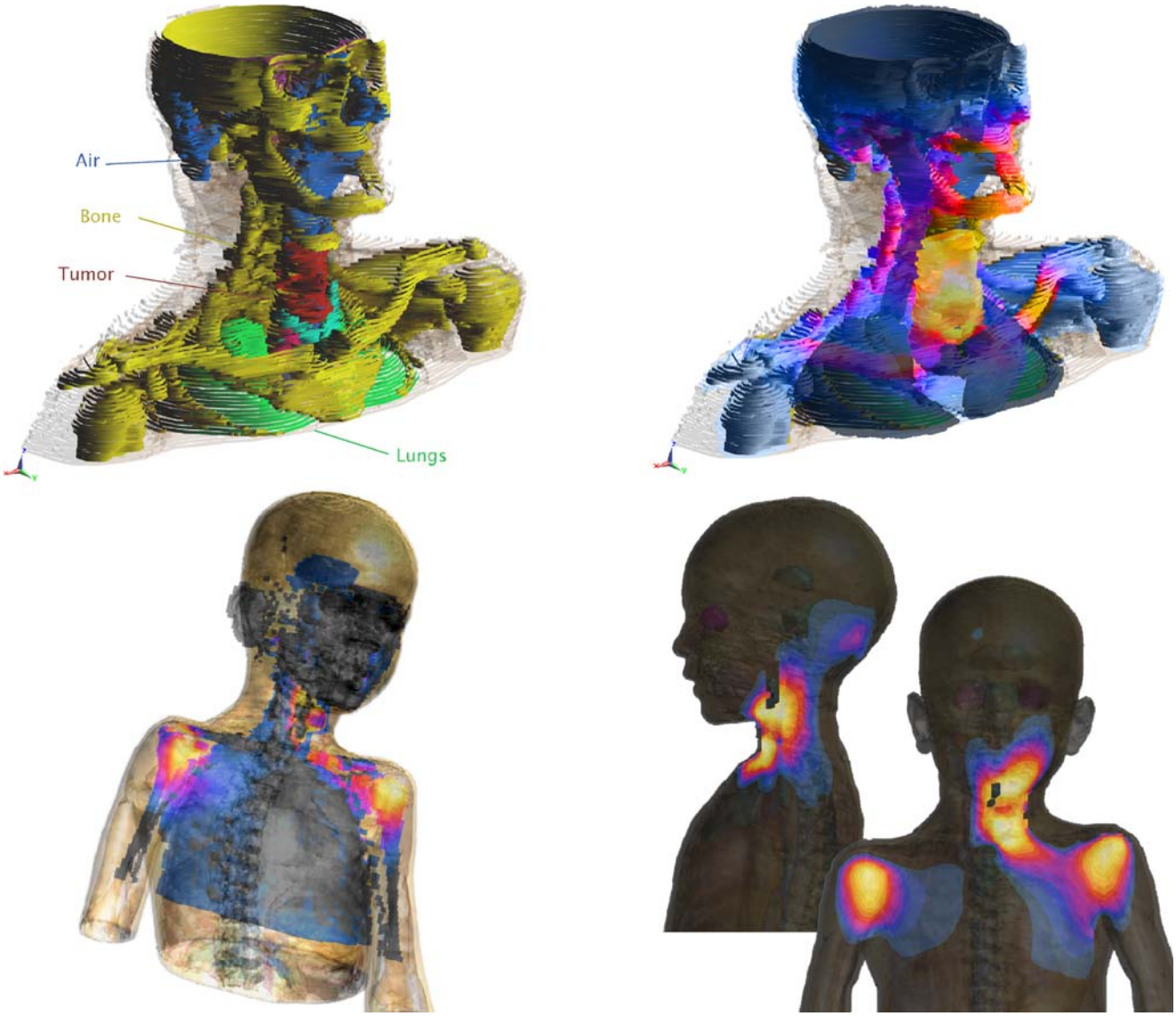

Fig. 2 Patient model and simulated energy absorption temperature distribution. The images show different temperature distributions within the iteration of our parallel PDE-constrained optimization process

a nonconvex objective function $f(x) : \mathbb{R}^n \to \mathbb{R}$ and constraint functions $c(x) : \mathbb{R}^n \to \mathbb{R}^m$, which are both assumed to be twice continuously differentiable. The objective is to find a local solution of the optimization problem.

$$\min_{x \in \mathbb{R}^n} f(x) \tag{2a}$$

$$\text{s.t.} \quad c(x) = 0 \quad \text{and} \quad x_L \le x \le x_U \,. \tag{2b}$$

Growing interest in efficient optimization methods has led to the development of interior-point or barrier methods for large-scale nonlinear programming. In particular, these methods provide an attractive alternative to active set strategies in handling problems with large numbers of inequality constraints. Over the past 15 years, there has also been a much better understanding of the convergence properties of interior-point methods and efficient algorithms and robust software codes e.g. [22] have been developed with desirable global and local convergence properties. Given the original problem in the form (2), a sequence of corresponding *barrier problems*,

$$\min_{x \in \mathbb{R}^n} \varphi_\mu(x) = f(x) - \mu^i \sum_{k=1}^{n} \ln(x^{(k)}) \tag{3a}$$

$$\text{s.t.} \quad c(x) = 0 \,, \tag{3b}$$

is solved to increasingly tighter tolerances, while again the barrier parameter μ is driven to zero. Interior-point method computes the solution equivalently by first solving a smaller, symmetric, indefinite linear system

$$\begin{bmatrix} \tilde{W}_k A_k \\ A_k^T \; 0 \end{bmatrix} \begin{pmatrix} \Delta x_k \\ \Delta \lambda_k \end{pmatrix} = - \begin{pmatrix} \nabla \varphi_\mu(x_k) + A_k \lambda_k \\ c(x_k) \end{pmatrix} . \tag{4}$$

The vectors λ and z are the Lagrangian multipliers for the equality and bound constraints, and X and Z denote the diagonal matrices with the vector elements of x and z on the diagonal. Here $A_k = \nabla c(x_k)$, and W_k denotes the Hessian $\nabla_{xx} \mathcal{L}(x, \lambda, z)$ of the Lagrangian function for the original problem (2),

$$\mathcal{L}(x, \lambda, z) := f(x) + c(x)^T \lambda - z \,. \tag{5}$$

The search directions are obtained from solving the linear system (4) where $\tilde{W}_k$ is an approximation of the Hessian of the Lagrangian for the barrier problem (3).

Therefore, the computational efficiency of all PDE constrained optimization applications strongly depends on the efficiency of the numerical linear algebra kernel to solve the symmetric indefinite Karush–Kuhn–Tucker systems of optimality (Eq. 4). In recent years, a large amount of work has been devoted to the problem of solving large symmetric indefinite systems (Eq. 4) in saddle-point form efficiently. One reason for this surge in interest is due the success of interior-point methods in nonlinear programming, which at their core require the solution of a series of linear systems in saddle-point form. In this work, we will focus on a new method – the `PSPIKE` algorithm – which represents a highly scalable parallel solver on distributed memory architectures.

3 The scalable PSPIKE algorithm

3.1 The basic SPIKE algorithm

Let $Ax = f$ be a nonsymmetric diagonally dominant system of linear equations where A is of order n and bandwidth $2m+1$. Unlike classical banded solvers such as those in Lapack which are based on LU-factorization of A, the spike algorithm [1, 8, 14, 16, 17] is based on the factorization $A = D \times S$, in which D is a block-diagonal matrix and S is the spike matrix as shown in Fig. 3 for three partitions. Note that the block diagonal matrices A_1, A_2, and A_3 are nonsingular by virtue of the diagonal dominance of A. For the example in Fig. 3, the basic Spike algorithm consists of the following stages.

- Stage 1: Obtain the LU-factorization (without pivoting) of the diagonal blocks A_j (i.e. $A_j = L_j U_j$, $j = 1, 2, 3$).
- Stage 2: Forming the spike matrix S and updating the right hand side
  (i) solve $L_1 U_1 [V_1, g_1] = [\binom{0}{B_1}, f_1]$
  (ii) solve $L_2 U_2 [W_2, V_2, g_2] = [\binom{C_2}{0}, \binom{0}{B_2}, f_2]$
  (iii) solve $L_3 U_3 [W_3, g_3] = [\binom{C_3}{0}, f_3]$
  f_j, $i \le j \le 3$, are the corresponding partitions of the right hand side f.
- Stage 3: Solving the reduced system,

$$\begin{bmatrix} I & V_1^{(b)} & 0 & 0 \\ W_2^{(t)} & I & 0 & V_2^{(t)} \\ W_2^{(b)} & 0 & I & V_2^{(b)} \\ 0 & 0 & W_3^{(t)} & I \end{bmatrix} \begin{bmatrix} x_1^{(b)} \\ x_2^{(t)} \\ x_2^{(b)} \\ x_3^{(t)} \end{bmatrix} = \begin{bmatrix} g_1^{(b)} \\ g_2^{(t)} \\ g_2^{(b)} \\ g_3^{(t)} \end{bmatrix} \tag{6}$$

where $(V_i^{(b)}, W_i^{(b)})$ and $(V_i^{(t)}, W_i^{(t)})$ are the bottom and top $m \times m$ blocks of (V_i, W_i), respectively. Similarly, $g_i^{(b)}$, $g_i^{(t)}$ and $x_i^{(b)}$, $x_i^{(t)}$ are the bottom and top m elements

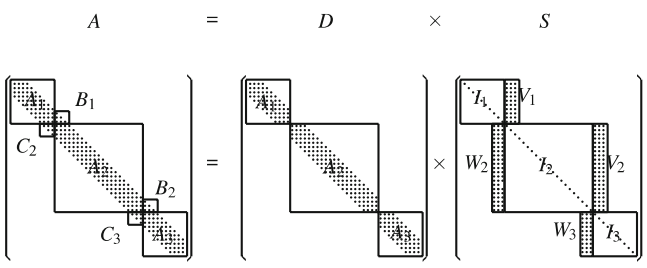

Fig. 3 Decomposition where $A = D \times S$, $S = D^{-1} A$, $B_j, C_j \in \mathbb{R}^{m \times m}$

of each g_i and x_i, respectively. Once the much smaller reduced system is solved, via a dense system solver, say, the three partitions x_i, $i = 1, 2, 3$, of the solution x are obtained as follows:

(i) $x_1 = g_1 - V_1 x_2^{(t)}$
(ii) $x_2 = g_2 - W_2 x_1^{(b)} - V_2 x_3^{(t)}$
(iii) $x_3 = g_3 - W_3 x_2^{(b)}$.

Clearly, the above basic scheme may be made more efficient if in stage 2 one can generate only the bottom and top $m \times m$ tips of the spikes V_i and W_i, as well as the corresponding bottom and top tips of g_i. In this case, once the reduced system is solved, solving the system $Ax = f$ is reduced to solving the independent systems:

(i) $L_1 U_1 x_1 = f_1 - B_1 x_2^{(t)}$
(ii) $L_2 U_2 x_2 = f_2 - C_2 x_1^{(b)} - B_2 x_3^{(t)}$
(iii) $L_3 U_3 x_3 = f_3 - C_3 x_2^{(b)}$.

If the matrix A is not diagonally dominant, we cannot guarantee that the diagonal blocks A_i are nonsingular. However, if we obtain the LU-factorization, without pivoting, of each A_i using diagonal boosting (perturbation), then

$$L_i U_i = (A_i + \delta A_i) \tag{7}$$

in which $||\delta A_i|| = \mathcal{O}(\epsilon ||A_i||)$ where ϵ is the unit roundoff. In this case, we will need to solve $Ax = f$ using an outer iterative scheme with the preconditioner being the matrix M which is identical to A except that each diagonal block A_i is replaced by $L_i U_i$ in (7). Solving systems of the form $My = r$ is accomplished via the Spike scheme outlined above.

3.2 The PSPIKE scheme

The PSPIKE scheme can be used for solving general sparse systems as follows. First, the sparse matrix A is reordered via a nonsymmetric reordering scheme that assures none of the diagonal elements is zero, followed by a weighted reordering scheme which brings as many of the largest elements as possible inside a narrow central band M. Using a Krylov subspace method, e.g. BiCGStab [21], for solving $Ax = f$, with the extracted banded preconditioner M. The major operations in each iteration are: (i) matrix-vector multiplication, and (ii) solving systems of the form $My = r$. The systems $My = r$ are solved using our proposed algorithm: PSPIKE as follows:

The LU-factorization of each diagonal block partition M_i (banded and sparse within the band) is obtained using PARDISO [18,19] without pivoting but utilizing diagonal boosting. The most recent version of PARDISO is also capable of obtaining the top and bottom tips of the left and right spikes $\hat{W}_i$ and $\hat{V}_i$, as well as the corresponding tips of the updated right hand side subvectors. Further, having the largest elements within the band, whether induced by the reordering or occur naturally, allows us to approximate (or truncate) the resulting reduced system by its block diagonal, $\hat{S} = \text{diag}(\hat{S}_1, \hat{S}_2, \ldots, \hat{S}_p)$, where p is the number of partitions and,

$$\hat{S}_j = \begin{bmatrix} I & \hat{V}_j^{(b)} \\ \hat{W}_{j+1}^{(t)} & I \end{bmatrix}. \tag{8}$$

This also enhances parallelism especially when M is of a large bandwidth. For a more detailed discussion of the decay rate of spikes and the truncated Spike algorithm we refer the reader to [11]. In the following section we will demonstrate the suitability of PSPIKE solver for implementation on clusters consisting of several nodes in which each node is of multicore architecture. Thus, while PARDISO is scalable on a single node only, PSPIKE is scalable across multiple nodes.

3.3 The PSPIKE algorithm for solving linear systems arising in biomedical PDE-constrained optimization

The linear systems that are extracted from the nonlinear solver has the following block structure

$$\begin{bmatrix} D & B^{\mathrm{T}} \\ & HC^{\mathrm{T}} \\ BC & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{9}$$

where $D \in \mathbb{R}^{n \times n}$ is diagonal and $D_{ii} > 0$ for $i = 1, 2, \ldots, n$. Furthermore $H \in \mathbb{R}^{k \times k}$ is symmetric positive definite and $C \in \mathbb{R}^{k \times n}$ is dense with $k \ll m$. $B \in \mathbb{R}^{n \times n}$ is nonsymmetric banded and sparse within the band.

Premultiplying the above equation by $\begin{bmatrix} D^{-1} & & \\ & H^{-1} & \\ & & I \end{bmatrix}$ we get

$$\begin{bmatrix} I & \tilde{B}^{\mathrm{T}} \\ & I\,\tilde{C}^{\mathrm{T}} \\ BC & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \\ f_3 \end{bmatrix} \tag{10}$$

where $\tilde{B}^{\mathrm{T}} = D^{-1} B^{\mathrm{T}}$, $\tilde{C}^{\mathrm{T}} = H^{-1} C^{\mathrm{T}}$, $\hat{f}_1 = D^{-1} f_1$ and $\hat{f}_2 = H^{-1} f_2$. Rearranging the rows and columns, we have the system,

$$\begin{bmatrix} I\,\tilde{B}^{\mathrm{T}} & 0 \\ B & 0\ C \\ 0 & \tilde{C}^{\mathrm{T}}\ I \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ x_2 \end{bmatrix} = \begin{bmatrix} \hat{f}_1 \\ f_3 \\ \hat{f}_2 \end{bmatrix}. \tag{11}$$

Observing that

$$\begin{bmatrix} I\,\tilde{B}^{\mathrm{T}} \\ B\ 0 \end{bmatrix}^{-1} = \begin{bmatrix} 0 & B^{-1} \\ \tilde{B}^{-T} - \tilde{B}^{-T} B^{-1} \end{bmatrix} \tag{12}$$

we can see that x_2 can be obtained by solving the small system,

$$(H + J^{\mathrm{T}} D J) x_2 = \left(f_2 - J^{\mathrm{T}} f_1 + J^{\mathrm{T}} D b \right) \tag{13}$$

Table 1 Parallel scalability (total optimization time in seconds) for the PDE-constrained interior-point optimization process. N represents the number of discretization points, "Nodes" is a dual-quad core Intel Harpertown processor, and "Threads" is the number of threads used on each node. "Direct" indicate the optimization process using an exact step computation, whereas "Inexact" indicates step computation based on PSPIKE

N	Threads	Nodes = 1 (direct)	Nodes = 4 (inexact)	Nodes = 8 (inexact)	Nodes = 16 (inexact)	Nodes = 32 (inexact)	Nodes = 64 (inexact)
	1	32 107	2016	1024	507	†	†
75^3	4	10 033	786	262	133	†	†
	8	7830	629	198	114	†	†
	1	‡	60 861	33 812	17 796	9132	6066
150^3	4	‡	20 287	10 821	5393	4072	3881
	8	‡	14 490	7246	4138	1923	1596

The symbol † indicates convergence problems, and ‡ shows that the optimization problem could not be solved due a high memory consumption of the direct solver

where J and b are obtained by solving $BJ = C$ and $Bb = f_3$, respectively. Consequently, x_1 and x_3 can be computed via $x_1 = b - Jx_2$ and $x_3 = B^{-T}(f_1 - Dx_1)$. The solution process requires solving linear systems with the coefficient matrices B^T and B. We use BiCGStab with a banded preconditioner to solve these systems, in which the systems involving the preconditioner are solved via the PSPIKE scheme we described above. In the next section we will illustrate the scalability of the above method for solving those systems that arise in biomedical PDE-constrained optimization problems.

4 Results

The distributed-memory test platform is a cluster with Infiniband interconnection. Each node has 16 Gb of memory and two Intel Harpertown processors where each processor contains four cores that run at 2.8 GHz. The BiCGStab iterations for solving systems involving B^{T} and B are terminated when $||\hat{r}_k||_\infty / ||\hat{r}_0||_\infty < \epsilon_{in}$ where $\hat{r}_0$ and $\hat{r}_k$ are the initial residual and the residual at the k^{th} iteration, respectively, and the systems involving the preconditioners are solved via the PSPIKE scheme described above.

Table 1 shows the parallel scalability (total optimization time in seconds) for the complete interior-point optimization process for an artificial Hyperthermia biomedical model problem described in [20]. N represent the number of spatial discretization points, "Nodes" is a dual-quad core Intel Harpertown processors, and threads is the number of threads used on each node. "Direct" indicate the optimization process using an exact step computation, and "Inexact" indicate step computation based on PSPIKE. It can be observed that almost linear scalability has been reached. The largest PDE-constrained optimization problem contains more than 6 750 000 optimization variables and it can be solved in 1597 s using 512 Intel Xeon cores (64 nodes with each 8 cores).

A real 3D hyperthermia model with over 1.8 million temperature variables in the NLP problem in (1). We consider linear systems extracted from the first, tenth (middle), and twenty first (last) iterations. However, since the results are uniform across these three systems, we present in Fig. 4 the results for the linear system of the tenth Newton iteration.[1] Figure 4 depict the speed improvement for a MPI/OpenMP hybrid implementation (8 cores (threads) per MPI processes) of the PSPIKE scheme. We demonstrate the speed improvement of the PSPIKE scheme for two stopping criteria, 10^{-5} and 10^{-7}. The corresponding final relative residuals are $\mathcal{O}(10^{-5})$ and $\mathcal{O}(10^{-7})$, respectively, which is sufficient to ensure the convergence of the inexact interior-point optimization process. For PARDISO and MUMPS the final relative residuals are $\mathcal{O}(10^{-12})$. Our new solver PSPIKE shows excellent scalability compared to these two state-of-the-art parallel solvers.

5 Conclusion

We have presented a PDE-constrained interior-point algorithm for large-scale hyperthermia cancer treatment planning. The novel aspects of the approach is that we are using a globally convergent optimization method [7] and also using a new scalable linear solver PSPIKE, which is designed to scale-up to thousands of cores. We have demonstrated that PSPIKE is an effective extension of PARDISO regarding parallel scalability on distributed-memory architectures. The experiments illustrate the computational advantages of our algorithm, and it is shown that the PDE-constrained optimization algorithm and implementation outperforms conventional approaches in terms of storage and CPU time for challenging linear systems arising in biomedical applications.

Acknowledgement We thank Andreas Wächter (IBM Research) for useful comments and suggestions, Esra Neufeld (IT'IS, ETH Zurich) for providing clinical data, and David Kuck (Parallel and Distributed Solutions Division, Intel) for financial support of the SPIKE development.

[1] The solution of the linear system within the interior-point optimization process consumes more than > 99% of the overall time and almost linear scalability can be observed up to 256 cores.

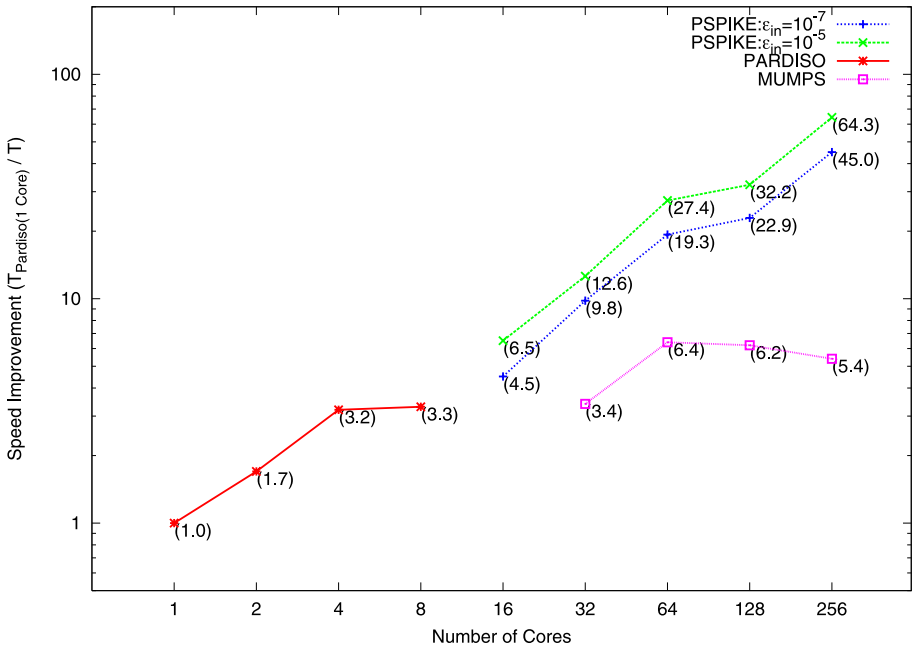

Fig. 4 The speed improvement of PSPIKE compared to PARDISO and MUMPS up to 256 cores. The base-line is the performance of PARDISO using one core (1.338 seconds)

This work was supported by the Swiss National Science Foundation under grant 200021-117745/1, by an IBM Faculty Award on "Modeling, Simulation and Optimization in Hyperthermia Cancer Treatment Planning" and partially supported by a gift from Intel's Parallel and Distributed Solutions Division (PDSD), and grants from NSF (NSF-CCF-0635169), and DARPA/AFRL (FA8750-06-1-0233).

References

1. Berry MW, Sameh A (1988) Multiprocessor schemes for solving block tridiagonal linear systems. Int J Supercomput Appl 1(3):37–57
2. Biros G, Ghattas O (2005) Parallel Lagrange–Newton–Krylov–Schur methods for PDE-constrained optimization. Part I: The Krylov–Schur solver. SIAM J Sci Comput 27(2):687–713
3. Biros G, Ghattas O (2005) Parallel Lagrange–Newton–Krylov–Schur methods for PDE-constrained optimization. Part II: The Lagrange–Newton solver and its application to optimal control of steady viscous flows. SIAM J Sci Comput 27(2):714–739
4. Byrd RH, Curtis FE, Nocedal J (2008) An inexact SQP method for equality constrained optimization. SIAM J Optim 19(1):351–369
5. Byrd RH, Curtis FE, Nocedal J (2009) An inexact Newton method for nonconvex equality constrained optimization. Math Progr A (doi:0.1007/s10107-008-0248-3)
6. Curtis FE, Nocedal J, Wächter A (2008) A matrix-free algorithm for equality constrained optimization problems with rank deficient jacobians. SIAM J Optim (submitted)
7. Curtis FE, Schenk O, Wächter A (2009) An interior-point algorithm for large-scale nonlinear optimization with inexact step computations. IBM Research Report RC 24736. SIAM J Sci Comput (submitted)
8. Dongarra JJ, Sameh AH (1984) On some parallel banded system solvers. Parall Comput 1(3):223–235
9. Fisher M, Nocedal J, Trémolet Y, Wright SJ (2008) Data assimilation in weather forecasting: A case study in PDE-constrained optimization. Optim Eng. doi:10.1007/s11081-008-9051-5
10. Haber E, Ascher UM (2001) Preconditioned all-at-once methods for large, sparse parameter estimation problems. Inverse Problems 17(6):1847–1864
11. Mikkelsen CCK, Manguoglu M (2008) Analysis of the truncated spike algorithm. SIAM J Matrix Anal Appl 30(4):1500–1519
12. Neufeld E (2008) High Resolution Hyperthermia Treatment Planning. PhD thesis, ETH Zurich
13. Pennes HH (1948) Analysis of tissue and arterial blood temperatures in the resting human forearm. J Appl Physiol 1(2):93–122
14. Polizzi E, Sameh AH (2006) A parallel hybrid banded system solver: the spike algorithm. Parall Comput 32(2):177–194
15. Polizzi E, Sameh AH (2007) Spike: A parallel environment for solving banded linear systems. Comput Fluids 36(1):113–120
16. Kuck DJ, Chen SC, Sameh AH (1978) Practical parallel band triangular system solvers. ACM Trans Math Softw 4(3):270–277
17. Sameh AH, Kuck DJ (1978) On stable parallel linear system solvers. J ACM 25(1):81–91
18. Schenk O, Gärtner K (2006) On fast factorization pivoting methods for sparse symmetric indefinite systems. Electron Trans Num Anal 23:158–179
19. Schenk O, Gärtner K (2004) Solving unsymmetric sparse systems of linear equations with PARDISO. Fut Generat Comput Syst 20(3):475–487
20. Schenk O, Wächter A, Weiser M (2008) Inertia revealing preconditioning for large-scale nonconvex constrained optimization. SIAM J Sci Comput 31(2):939–960
21. van der Vorst HA (1992) Bi-cgstab: a fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. SIAM J Sci Stat Comput 13(2):631–644
22. Wächter A, Biegler LT (2006) On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. Math Prog 106(1):25–57
23. Young DP, Huffman WP, Melvin RG, Hilmes CL, Johnson FT (2003) Nonlinear elimination in aerodynamic analysis and design optimization. In: Biegler LT, Ghattas O, Heinkenschloss M, v Bloemen-Waanders B (eds) Large-Scale PDE-Constrained Optimization, Springer, New York, pp 17–44