Preconditioning Techniques for the Bidomain Equations

Rodrigo Weber Dos Santos[1], G. Plank[2], S. Bauer[1], and E.J. Vigmond[3]

[1] Dept. of Biosignals, Physikalisch-Technische Bundesanstalt, Berlin, Germany
email: rodrigo.weber@ptb.de
[2] Inst. für Medizinische Physik und Biophysik, Universität Graz, Austria
[3] Dept. of Electrical and Computer Engineering, University of Calgary, Canada

Summary. In this work we discuss parallel preconditioning techniques for the bidomain equations, a non-linear system of partial differential equations which is widely used for describing electrical activity in cardiac tissue. We focus on the solution of the linear system associated with the elliptic part of the bidomain model, since it dominates computation, with the preconditioned conjugate gradient method. We compare different parallel preconditioning techniques, such as block incomplete LU, additive Schwarz and multigrid. The implementation is based on the PETSc library and we report results for a 16-node HP cluster. The results suggest the multigrid preconditioner is the best option for the bidomain equations.

1 Introduction

The set of bidomain equations (see Keener and Sneyd [1998]) is currently the mathematical model that best reflects the electrical activity of the heart. The non-linear partial differential equations (PDEs) model both the intracellular and extracellular domains of cardiac tissue from an electrostatic point of view. The coupling of the two domains is done via non-linear models describing the current flow through the cell membrane. Such models are based on experimental data that quantify different ionic contributions, as first proposed by Hodgkin and Huxley [1952].

Unfortunately, the bidomain equations are computationally very expensive. Modern membrane models involve more than 20 non-linear equations. One way to avoid the solution of a large non-linear PDE system at every time step is to use an operator splitting approach (Vigmond et al. [2002], Sundnes et al. [2001], Keener and Bogar [1998], Weber dos Santos [2002]). The numerical solution reduces to the solution of a parabolic equation, a non-linear system of ordinary differential equations, and an elliptic system. It is the latter that dominates computation (Vigmond et al. [2002]).

An efficient way of solving the large linear algebraic system that arises from the discretization of the bidomain equations has been a topic of re-

search since 1994 (see Hooke et al. [1994]). Many different approaches have been employed with the preconditioned conjugate gradient method (CG) becoming the standard choice for an iterative method. Diagonal preconditioner was used for the bidomain equations by Eason and Malkin [2000], Skouibine and Krassowska [2000]. Incomplete LU factorization (ILU) was implemented by Pavarino and Franzone [2004], Street and Plonsey [1999], Vigmond et al. [2002]. The Symmetric-Successive-Over-Relaxation (SSOR) preconditioner was investigated by Pennacchio and Simoncini [2002], Weber dos Santos [2002]. In Weber dos Santos [2002] it was shown that incomplete factorization and SSOR achieved comparable results and were at least two times faster than the diagonal preconditioner.

Another way of reducing the computation time is cluster computing. The use of such parallel environment is supported by standard communication libraries such as the Message Passing Interface library [1994] (MPI). The solution of the bidomain equations has been efficiently implemented with MPI on small clusters (16 to 32 processors), see Pavarino and Franzone [2004], Pormann [2000], Yung [2000], Weber dos Santos [2002].

In this work we focus on the solution of the linear algebraic system associated with the elliptic part of the bidomain model. We employ CG and compare different parallel preconditioning techniques, such as block incomplete LU (ILU), the additive Schwarz method (ASM), and multigrid (MG). Both overlapping and non-overlapping domain decomposition techniques are investigated. The implementation is based on the PETSc C library (Balay et al. [2002]), which uses MPI. The results taken from a 16-node HP-Unix cluster indicate that the multigrid preconditioner is at least 13 times faster than the single-level Schwarz-based techniques.

2 Mathematical formulation

We simulated a two-dimensional piece of cardiac tissue in contact with a perfusing bath as previously described by Weber dos Santos et al. [2003], Weber dos Santos and Dickstein [2003]. The model that we present here was successfully used to reproduce an in-vitro experiment which explored the effects of lapine cardiac tissue micro-structure (see Weber dos Santos et al. [2003]). The simulated square region $\Omega = \Omega_t \cup \Omega_o$ with $\bar{\Omega}_t \cap \bar{\Omega}_o = \Gamma_{to}$, where Ω_o accounts for the perfusing bath and Ω_t for the cardiac tissue sample. The geometric information of Ω_t was extracted by image processing techniques and is represented by a mask vector M, $M_{i,j} = 1.0$, for $(i,j) \in \Omega_t$, $M_{i,j} = 0.0$, otherwise. In Figure 1 the tissue regions are represented by the gray color.

The bath was modeled as an isotropic conductor with conductivity σ_o. The electric potential $\phi_o : \Omega_o \times [0, t_f] \rightarrow \Re$ satisfies $\sigma_o \Delta \phi_o = 0$. The cardiac tissue is modeled by the bidomain equations.

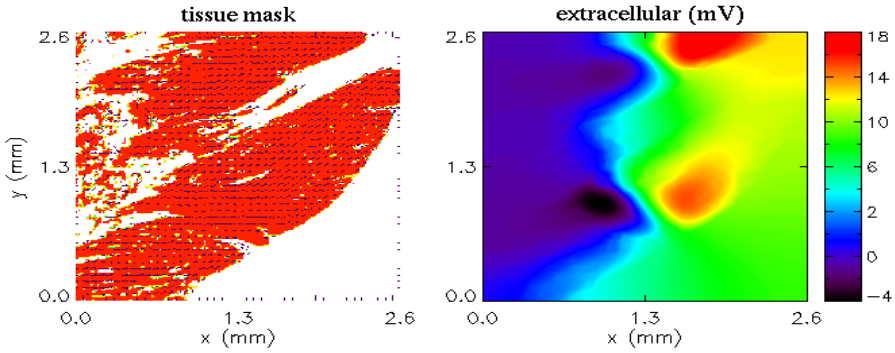

Fig. 1. Tissue geometry and fiber orientation extracted from a histological picture (courtesy of E. Hofer and D. Sanchez-Quintana) (left). Solution of the extracellular potential (ϕ_e and ϕ_o) at time 2.6ms (right).

$$\chi\left(C_m\frac{\partial\phi}{\partial t}+f(\phi,\upsilon)\right)=\nabla.(\boldsymbol{\sigma_i}\nabla\phi)+\nabla.(\boldsymbol{\sigma_e}\nabla\phi_e), \tag{1}$$

$$\nabla.((\boldsymbol{\sigma_e}+\boldsymbol{\sigma_i})\nabla\phi_e)=-\nabla.(\boldsymbol{\sigma_i}\nabla\phi), \tag{2}$$

$$\frac{\partial\upsilon}{\partial t}=g(\phi,\upsilon),\phi=\phi_i-\phi_e, \tag{3}$$

where ϕ_e and ϕ_i: $\Omega_t\times[0,t_f]\rightarrow\Re$ are the extracellular and intracellular potentials; ϕ: $\Omega_t\times[0,t_f]\rightarrow\Re$ is the transmembrane voltage; υ: $\Omega_t\times[0,t_f]\rightarrow\Re^m$ represents the ionic current variables; $\boldsymbol{\sigma_i}$ and $\boldsymbol{\sigma_e}$ are conductivity tensors of the intracellular and extracellular spaces; C_m and χ are capacitance per unit area and surface to volume ratio respectively; f: $\Re\times\Re^m\rightarrow\Re$ and g: $\Re\times\Re^m\rightarrow\Re^m$ model ionic currents and specify the cell membrane model. We used the rabbit atrial model ($m=27$) of Lindblad et al. [1996].

An image processing technique (see Weber dos Santos et al. [2003]) was applied to extract the cardiac fiber orientation θ: $\Omega_t\rightarrow\Re$. Conductivity tensors were derived from the curved cardiac fibers based on θ by applying

$$\boldsymbol{\sigma_*}=\begin{pmatrix}\sigma_{*l}\cos^2\theta+\sigma_{*t}\sin^2\theta & (\sigma_{*l}-\sigma_{*t})\cos\theta\sin\theta\\ (\sigma_{*l}-\sigma_{*t})\cos\theta\sin\theta & \sigma_{*t}\cos^2\theta+\sigma_{*l}\sin^2\theta\end{pmatrix},$$

where σ_{*l} and σ_{*t} are longitudinal and transversal fiber conductivities ($*=i,e$). The boundary conditions for the bath to tissue interface were set to (see Krassowska and Neu [1994])

$$\phi_e=\phi_o, \tag{4}$$

$$\boldsymbol{\sigma_e}\nabla\phi_e.\eta=\sigma_o\nabla\phi_o.\eta, \tag{5}$$

$$\boldsymbol{\sigma_i}\nabla\phi_i.\eta=0,\text{ on }\Gamma_{to}\times[0,t_f]. \tag{6}$$

The other boundaries are assumed to be electrically isolated, modeled by imposing homogeneous Neumann-like conditions

$$\boldsymbol{\sigma_i}\nabla\phi_i.\eta = \boldsymbol{\sigma_e}\nabla\phi_e.\eta = \sigma_o\nabla\phi_o.\eta = 0, \text{ on } \partial\Omega \times [0, t_f]. \tag{7}$$

The numerical results presented in the next sections were obtained with the following parameters: $C_m = 1\,\mu\text{F/cm}^2$, $\chi = 2000\,/\text{cm}$, $\sigma_{il} = 3$ mS/cm, $\sigma_{it} = 0.31$ mS/cm, $\sigma_{el} = 2$ mS/cm, $\sigma_{et} = 1.35$ mS/cm and $\sigma_o = 20$ mS/cm. Ω is a square of sides equal to 2.6mm.

3 Operator splitting and boundary conditions

We approached the non-linear system of PDEs with an operator splitting technique (Strang [1968]). The solution reduced to a three-step scheme that involved the solution of a parabolic PDE, an elliptic system, and a non-linear system of ordinary differential equations at each time step. Since the CFL condition of the parabolic PDE severely restricted the time step, we solved this equation via the Crank-Nicolson method. The large non-linear ODE system was solved via a forward-Euler scheme:

$$1.\quad (1 - \frac{\Delta t}{2}\, A_i)\varphi^{k+1/2} = (1 + \frac{\Delta t}{2}\, A_i)\varphi^k + \Delta t\, A_i(\varphi_e)^k, \tag{8}$$

$$2.\quad \varphi^{k+1} = \varphi^{k+1/2} - \Delta t\, f(\varphi^{k+1/2}, \nu^k)/(\chi C_m) \tag{9}$$

$$\nu^{k+1} = \nu^k + \Delta t\, g(\varphi^{k+1/2}, \nu^k) \tag{10}$$

$$3.\quad (A_i + A_e)(\varphi_e)^{k+1} = -A_i\varphi^{k+1}, \tag{11}$$

$$A_o{\varphi_o}^{k+1} = 0, \tag{12}$$

where A_i, A_e and A_o are the $\nabla.(\boldsymbol{\sigma_i}\nabla)/(\chi C_m)$, $\nabla.(\boldsymbol{\sigma_e}\nabla)/(\chi C_m)$ and $\sigma_o\Delta$ operators; Δt is the time step; φ^k, φ_e^k, φ_o^k and ν^k are time discretizations of ϕ, ϕ_e, ϕ_o and υ, respectively, for time equal to $k\Delta t$, with $0 \le k \le t_f/\Delta t$. A von Neumann analysis of the linearized system shows that the above scheme is unconditionally stable.

The elliptic equations of the third step are coupled by the boundary conditions (4), (5) and (6). The implementation of these and the other homogeneous Neumann-like conditions deserve further discussion. Without loss of generality, we initially focus on equation (11). By writing the boundary condition in the form $-(\boldsymbol{\sigma_i} + \boldsymbol{\sigma_e})\nabla\varphi_e^{k+1}.\eta = g$, the compatibility condition is

$$\int_{\partial\Omega} \boldsymbol{\sigma_i}\nabla\varphi^{k+1}.\eta = \int_\Omega \nabla.(\boldsymbol{\sigma_i}\nabla\varphi^{k+1}) = -\int_\Omega \nabla.((\boldsymbol{\sigma_i} + \boldsymbol{\sigma_e})\nabla\varphi_e^{k+1})$$
$$= -\int_{\partial\Omega} (\boldsymbol{\sigma_i} + \boldsymbol{\sigma_e})\nabla\varphi_e^{k+1}.\eta, \Rightarrow \int_{\partial\Omega} (\boldsymbol{\sigma_i}\nabla\varphi^{k+1} - g).\eta = 0. \tag{13}$$

Explicit schemes (see Latimer and Roth [1998], Skouibine and Krassowska [2000], Saleheen and Kwong [1998]) have been implemented with boundary conditions $\boldsymbol{\sigma_i}\nabla\varphi^{k+1}.\eta = -\boldsymbol{\sigma_i}\nabla\varphi_e^k.\eta$, $\boldsymbol{\sigma_e}\nabla\varphi_e^{k+1}.\eta = 0$ on $\partial\Omega$, applied to (8) and (11), respectively. Condition (13) becomes $\int_{\partial\Omega} \boldsymbol{\sigma_e}\nabla(\varphi_e^{k+1} - \varphi_e^k).\eta = 0$

and thus does not always hold. Perhaps such violation was not critical for the explicit schemes, since the CFL condition restricts Δt to very small values. In Sundnes et al. [2001], the following conditions were applied to (8) and (11): $\boldsymbol{\sigma_i}\nabla\varphi^{k+1}.\eta = 0$, $-(\boldsymbol{\sigma_i} + \boldsymbol{\sigma_e})\nabla\varphi_e^{k+1}.\eta = 0$ on $\partial\Omega$, which satisfy (13). These conditions are equivalent to (7) only if $\boldsymbol{\sigma_i} = \alpha\boldsymbol{\sigma_e}$, $\alpha \in \Re$, i.e., when both intra- and extracellular domains have equal anisotropy ratios. Therefore, they were used as approximations of (7), but did not properly implement the electric isolation for the general case of unequal anisotropy ratios, which is the case with cardiac tissue.

We implemented the following boundary conditions

$$\boldsymbol{\sigma_i}\nabla\varphi^{k+1}.\eta = -\boldsymbol{\sigma_i}\nabla\varphi_e^{k}.\eta, \tag{14}$$

$$-(\boldsymbol{\sigma_i} + \boldsymbol{\sigma_e})\nabla\varphi_e^{k+1}.\eta = \boldsymbol{\sigma_i}\nabla\varphi^{k+1}.\eta \,\text{on}\, \partial\Omega, \tag{15}$$

applied to (8) and (11), respectively. (14)-(15) are natural approximations of (7) and satisfy the compatibility condition (13). In addition, since we use the finite element method, conditions (14)-(15) are naturally implemented by the numerical scheme. The same properties apply to our bath-tissue interface conditions and to the homogeneous Neumann condition for ϕ_o:

$$\sigma_o\nabla\varphi_o^{k+1}.\eta = \boldsymbol{\sigma_e}\nabla\varphi_e^{k+1}.\eta, \tag{16}$$

$$\boldsymbol{\sigma_i}\nabla\varphi^{k+1}.\eta = -\boldsymbol{\sigma_i}\nabla\varphi_e^{k+1}.\eta, \,\text{on}\, \Gamma_{to}. \tag{17}$$

$$\sigma_o\nabla\varphi_o^{k+1}.\eta = 0 \,\text{on}\, \partial\Omega. \tag{18}$$

All boundary conditions cancel each other in the variational formulation:

$$\int_{\Omega_t} v\nabla.((\boldsymbol{\sigma_i} + \boldsymbol{\sigma_e})\nabla\varphi_e^{k+1}) + \int_{\Omega_o} v\sigma_o\Delta\varphi_o^{k+1} = -\int_{\Omega_t} v\nabla.(\boldsymbol{\sigma_i}\nabla\varphi^{k+1}),$$

$$\int_{\Omega_t} \nabla v(\boldsymbol{\sigma_i} + \boldsymbol{\sigma_e})\nabla\varphi_e^{k+1} - \int_{\partial\Omega+\Gamma_{to}} v(\boldsymbol{\sigma_i} + \boldsymbol{\sigma_e})\nabla\varphi_e^{k+1}.\eta + \int_{\Omega_o} \nabla v\sigma_o\nabla\varphi_o^{k+1}$$

$$-\int_{\partial\Omega+\Gamma_{to}} v\sigma_o\nabla\varphi_o^{k+1}.\eta = -\int_{\Omega_t} \nabla v\boldsymbol{\sigma_i}\nabla\varphi^{k+1} + \int_{\partial\Omega+\Gamma_{to}} v\boldsymbol{\sigma_i}\nabla\varphi^{k+1}.\eta,$$

where v is a test function. Using (15)-(18), all boundary integrals vanish:

$$\int_{\Omega_t} \nabla v(\boldsymbol{\sigma_i} + \boldsymbol{\sigma_e})\nabla\varphi_e^{k+1} + \int_{\Omega_o} \nabla v\sigma_o\nabla\varphi_o^{k+1} = -\int_{\Omega_t} \nabla v\boldsymbol{\sigma_i}\nabla\varphi^{k+1},$$

which is used to generate the finite element numerical approximation

$$Ax = b, \tag{19}$$

where A is the stiffness matrix, b is the load vector and x is the discretization of ϕ_e and ϕ_o. A uniform mesh of squares and bilinear polynomials were used. Spatial discretization was set to $\Delta x = 3.3\mu$m and the time step to $\Delta t = 10\mu$s.

4 Parallel preconditioners

We used CG to solve the linear system (19). We compare different preconditioners: ILU, additive Schwarz method (ASM) and multigrid (MG). The solution of (19) is implemented in parallel using the PETSc C library v2.1.5 (Balay et al. [2002]), which uses MPI. CG is parallelized via a linear domain decomposition. The spatial domain is decomposed into *proc* domains with equal sizes, where *proc* is the number of processors involved in the simulation.

The non-overlapping parallel version of the ILU preconditioner uses block Jacobi, i.e., an incomplete factorization is performed over the main diagonal block of the local part of the matrix A, thus avoiding extra communication. ILU has as parameter the level of fill-in, $fill$.

The ASM preconditioner implements an overlapping decomposition of the spatial domain Ω. Each processor block overlaps to the neighboring domain block by the amount ovl. An ILU is performed over each processor block. A greater ovl means more communication is necessary between the processors. The way the overlapping regions affect the residual can be controlled by the parameter *method* ($basic$, $restrict$, $interpolate$ or $none$, see Balay et al. [2002], Cai and Sarkis [1999]).

The MG preconditioner performs a few iterations of PETSc's native geometric multigrid method. The parameter $levels$ indicates the number of different spatial grids involved in the solution. Based on the finest regular grid G_0, coarser regular grids were successively generated with half of the nodes on each direction (G_l, $l = 0$ to $levels - 1$). For each grid pair, G_l and G_{l+1}, a prolongation rectangular matrix, P_l, was generated using a bilinear interpolation scheme. The restriction operators were set to P_l^T and used to generate coarser tissue masks and conductivity tensors. For every grid level, a matrix A_l was generated by applying the finite element method. The *smoother* used for all but the coarsest level was a number of iterations, $smooth$, of the CG method which was, in turn, also preconditioned by ILU. For the coarsest level, we used a direct LU solver with nested dissection reordering. This was not done in parallel, i.e., it was repeated on every processor, avoiding any communication. In addition to the parameters $levels$, $smooth$ and $fill$, we could control the *type* of multigrid ($multiplicative$, $additive$, $full$ and $kaskade$) and the cycle (v or w) (Balay et al. [2002]).

All complete and incomplete factorizations were performed only once, in the first time step of the simulation, so that the cost was amortized over the whole simulation. For the global preconditioned CG algorithm associated with (19) the stop criterion adopted was based on the unpreconditioned and absolute L_2 residual norm, $||Ax_i - b||_2 < tol$, where x_i was the solution at iteration i and tol was a tolerance which was set to 10^{-3}. Although this is not the most efficient stop criteria for the CG, it is the fairest one when comparing different preconditioning methods.

5 Results

We performed several comparisons of the different preconditioners on a 16-node HP Unix cluster, each node equipped with McKinley 900 MHz processors and 2 GB of DRAM and connected by a 1Gbit/s Ethernet switch. The electrical activity was initiated on the left side of the 2.6×2.6 mm tissue-bath preparation (640,000 nodes) and propagated to the right (see figure 1). The performance measurements reported in this section are relate to the solution of the elliptic system which is responsible for around 70% of the whole simulation time. We simulate 10ms of electrical activity (100 time steps). It is interesting to note that if the parabolic equation (8) was solved with an explicit method such as the forward-Euler method, the CFL condition would severely restrict Δt. An approximation of the CFL condition can be derived by assuming equal anisotropy ratios ($\boldsymbol{\sigma_i} = \alpha \boldsymbol{\sigma_e}$) and straight fibers:

$$\Delta t \leq \frac{\chi C_m \Delta x^2}{2(\sigma_l + \sigma_t)} = 0.07 \mu s, \tag{20}$$

where $\sigma_* = \sigma_{i*}\sigma_{e*}/(\sigma_{i*} + \sigma_{e*})$, $(* = l, t)$. Numerically we verified that with $\Delta t = 0.07\mu s$ the forward-Euler scheme already did not converge. Thus, the semi-implicit based scheme allows Δt to be more than 100 times greater than one restricted by an explicit based method.

5.1 Parameter tuning

Several preconditioner parameters were tuned: $fill$ for ILU; *method*, *ovl* and $fill$ for ASM; *levels*, *type*, *cycle*, *smooth* and $fill$ for MG. Table 1 shows for different numbers of processors, the optimal parameter values, those combinations yielding the fastest execution time. The parameter $fill$ was set to 0, 1, 2, 5, 10, and 15; *method* to *basic*, *interpolate*, *restrict* and *none*; *ovl* to 1, 2, 4, 6, 8 and 10; *levels* to values from 2 to 7; *type* to *multiplicative*, *additive*, $full$ and *kaskake*; *cycle* to v and w; and *smooth* to values from 1 to 3. In addition, all parameters were tuned for best execution time on 1, 8 and 16 processors. A total of 3042 simulations were performed during more than two weeks of computation time.

	ILU	ASM			MG			
proc	$fill$	*method*	*ovl*	$fill$	*levels*	*type*	*smooth*	$fill$
1	15				3	*kaskade*	1	0
8	5	*basic*	4	10	6	*kaskade*	1	0
16	5	*basic*	4	10	6	*kaskade*	1	0

Table 1. Values of parameters leading to the quickest solution time as a function of the number of processors (*proc*).

For the ILU preconditioner, as *proc* increased from 1 to 16, the optimal value for $fill$ decreased from 15 to 5, i.e., as the domain was decomposed, it became less effective to increase $fill$ since the preconditioner became more expensive, but did not speed up the convergence. This was improved by ASM which took advantage of higher values of $fill$ by increasing communication, i.e., increasing ovl. The optimal values were $ovl = 4$ and $fill = 10$. For MG, the optimal value of $levels$ depended on *proc*. On a single processor, $levels = 3$ was fastest. In parallel, since the coarsest grid was solved sequentially, the cost of fewer grid $levels$ rivaled the gains of parallelism. Therefore, as *proc* increased, the optimal $levels$ also increased to 6. The other MG parameters indicate that the cheapest MG method was the best option, i.e, the *kaskade* method with a single iteration of the CG smoother preconditioned by ILU(0). On the other hand, the best ASM method was the expensive *basic* one which included all off-processor values in the interpolation and restriction processes.

5.2 Performance comparison and parallel speedup

Table 2 shows the execution time and number of CG iterations/time step as well as memory consumption (mem(MB))/processor for all preconditioners with the optimal parameters. The ASM preconditioner achieved better performance results than the non-overlapping ILU. ASM was 1.4 (1.5) times faster than ILU but required 20% (55%) more memory than ILU on 8 (16) *procs*. MG was between 15.5 (*proc* = 1) and 20.6 (*procs*=8) times faster than ILU and it required between 44% less memory (*proc*=1) and 7% more memory (*procs*=8) than ILU. Compared to ASM, MG was 14.9 (13.7) times faster than ASM and required 11% (32%) less memory than ASM on 8 (16) processors. All preconditioners achieved low parallel speedup (execution time with

	1 *proc*			8 *procs*			16 *procs*		
	t(s)	it	mem(MB)	t(s)	it	mem(MB)	t(s)	it	mem(MB)
ILU	96.2	98.7	1157.2	28.8	428.5	96.6	20.1	540.8	50.1
ASM				20.9	205.9	116.2	13.7	228.3	77.7
MG	6.3	7.7	649.6	1.4	11.3	103.4	1.0	13.6	52.5

Table 2. Best results of the preconditioners for different numbers of processors. Execution time per time step in seconds, t(s); CG iterations per time step, it; and memory usage per processor in MBytes, mem(MB).

proc=1/execution time) results with *procs*=16: 4.8 for ILU; 7.0 for ASM (related to ILU with *proc*=1); and 6.3 for MG. The reason was mainly due to the increase of the CG iterations with *proc*. The number of iterations was increased by a factor of 5.6 (2.4) by increasing *proc* from 1 to 16 for ILU (ASM). MG suffered less from this problem with an increase of only 1.7 times. Nevertheless, the speedup was poor. We believe the explanation lies for MG in the

sequential direct solver. The cost of this was not reduced by increasing *proc*, and, thus, limited the total parallel speedup. For smaller *proc*, all preconditioners presented reasonable speedup, around 4.5 with *proc*=8.

5.3 Conclusions

In this work, we employed the conjugate gradient algorithm for the solution of the linear system associated with the elliptic part of the bidomain equations and compared different parallel preconditioning techniques, such as ILU, ASM and MG. The results taken from a 16-node HP-Unix cluster indicate that the multigrid preconditioner is at least 13 times faster than the single-level Schwarz based techniques and requires at least 11% less memory.

Acknowledgement. R. Weber dos Santos and S. Bauer acknowledge the support provided by 13N8125 BMFT German Ministry of Research and Technology. G. Plank was supported by the Austrian Science Fund FWF (R21-N04) and by a fund UGP4 of the Austrian Ministry of Education, Science and Culture. E. Vigmond is supported by a grant from the Natural Sciences and Engineering Research Council of Canada.

References

S. Balay, K. Buschelman, W. Gropp, D. Kaushik, M. Knepley, L. McInnes, B. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2002.

X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing*, 21: 239–247, 1999.

J. Eason and R. Malkin. A simulation study evaluating the performance of high-density electrode arrays on myocardial tissue. *IEEE Trans Biomed Eng*, 47(7):893–901, 2000.

A. Hodgkin and A. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117:500–544, 1952.

N. Hooke, C. Henriquez, P. Lanzkron, and D. Rose. Linear algebraic transformations of the bidomain equations: implications for numerical methods. *Math Biosci*, 120(2):127–45, 1994.

J. Keener and K. Bogar. A numerical method for the solution of the bidomain equations in cardiac tissue. *Chaos*, 8(1):234–241, 1998.

J. Keener and J. Sneyd. *Mathematical physiology.* Springer, 1998.

W. Krassowska and J. Neu. Effective boundary conditions for syncytial tissues. *IEEE Trans. on Biomed. Eng.*, 41:143–150, 1994.

D. Latimer and B. Roth. Electrical stimulation of cardiac tissue by a bipolar electrode in a conductive bath. *IEEE Trans. on Biomed. Eng.*, 45(12): 1449–1458, 1998.

D. Lindblad, C. Murphey, J. Clark, and W. Giles. A model of the action potential and the underlying membrane currents in a rabbit atrial cell. *The American Physiological Society*, (0363-6125):H1666–H1696, 1996.

Message Passing Interface library. MPI, a message-passing interface standard. *Int. J. Supercomp.*, 8:159–416, 1994.

L. Pavarino and P. Franzone. Parallel solution of cardiac reaction-diffusion models. In R. Kornhuber, R. Hoppe, D. Keyes, J. Periaux, O. Pironneau, and J. Xu, editors, *Procedings of the 15th International Conference on Domain Decomposition Methods, Lecture Notes in Computational Science and Engineering*. Springer, 2004.

M. Pennacchio and V. Simoncini. Efficient algebraic solution of reaction-diffusion systems for the cardiac excitation process. *Journal of Computational and Applied Mathematics*, 145(1):49–70, 2002. ISSN 0377-0427.

J. Pormann. Computer simulations of cardiac electrophysiology. In *Proceedings of SC2000*, 2000.

H. Saleheen and Kwong. A new three-dimensional finite-difference bidomain formulation for inhomogeneous anisotropic cardiac tissues. *IEEE Trans. on Biomed. Eng.*, 45(1):15–25, 1998.

K. Skouibine and W. Krassowska. Increasing the computational efficiency of a bidomain model of defibrillation using a time–dependent activating function. *Annals of Biomedical Engineering*, 28:772–780, 2000.

G. Strang. On the construction and comparison of difference scheme. *SIAM Journal on Numerical Analysis*, 5:506–517, 1968.

A. Street and R. Plonsey. Propagation in cardiac tissue adjacent to connective tissue: Two-dimensional modeling studies. *IEEE Transactions on Biomedical Engineering*, 46:19–25, 1999.

J. Sundnes, G. Lines, and A. Tveito. Efficient solution of ordinary differential equations modeling electrical activity in cardiac cells. *Math Biosci*, 172(2): 55–72, 2001.

E. Vigmond, F. Aguel, and N. Trayanova. Computational techniques for solving the bidomain equations in three dimensions. *IEEE Trans Biomed Eng*, 49(11):1260–9, 2002.

R. Weber dos Santos. *Modelling cardiac electrophysiology.* PhD thesis, Federal University of Rio de Janeiro, Mathematics dept., Rio de Janeiro, Brazil, 2002.

R. Weber dos Santos and F. Dickstein. On the influence of a volume conductor on the orientation of currents in a thin cardiac tissue. In I. Magnin, J. Montagnat, P. Clarysse, J. Nenonen, and T. Katila, editors, *Lecture Notes in Computer Science*, pages 111–121. Springer, Berlin, 2003.

R. Weber dos Santos, U. Steinhoff, E. Hofer, D. Sanchez-Quintana, and H. Koch. Modelling the electrical propagation in cardiac tissue using detailed histological data. *Biomedizinische Technik*, 2003.

C. Yung. Application of a stiff, operator-splitting scheme to the computational modeling of electrical propagation of cardiac ventricles. Engineering dept., Johns Hopkins University, Maryland, 2000.