Time Parallelization for Nonlinear Problems Based on Diagonalization

Martin J. Gander and Laurence Halpern

1 Introduction

Over the last decade, an intensive research effort has been devoted to investigate the time direction in evolution problems for parallelization. This is because modern supercomputers have now so many processors that often space parallelization strategies for evolution problems saturate before all available processors can be used. In the relatively recent field of time parallelization, there are four main algorithmic techniques that have been investigated: methods based on multiple shooting (Chartier and Philippe 1993), like the parareal algorithm (Lions et al. 2001) for which a detailed convergence analysis can be found in Gander and Vandewalle (2007) for the linear case and in Gander and Hairer (2008) for the nonlinear case; methods based on space-time decomposition, like classical Schwarz waveform relaxation (Bjørhus 1995; Gander and Stuart 1998; Giladi and Keller 2002) and optimized variants (Bennequin et al. 2009; Gander and Halpern 2005, 2007; Gander et al. 2003), and Dirichlet-Neumann and Neumann-Neumann waveform relaxation (Gander et al. 2016b; Kwok 2014; Mandal 2014); space-time multigrid methods (Emmett and Minion 2012; Gander and Neumüller 2016; Hackbusch 1984; Horton and Vandewalle 1995); and direct time parallelization methods like tensor product methods (Maday and Rønquist 2008), RIDC (Christlieb et al. 2010), and ParaExp (Gander and Güttel 2013); for an up to date overview and a historical perspective of these approaches, see Gander (2015).

M.J. Gander
University of Geneva, Geneva, Switzerland
e-mail: martin.gander@unige.ch

L. Halpern (✉)
University Paris 13, Paris, France
e-mail: halpern@math.univ-paris13.fr

C.-O. Lee et al. (eds.), *Domain Decomposition Methods in Science and Engineering XXIII*, Lecture Notes in Computational Science and Engineering 116, DOI 10.1007/978-3-319-52389-7_15

We have recently proposed and analyzed a new approach to make the tensor product time parallelization technique from Maday and Rønquist (2008) robust. For linear problems of diffusion type, we have derived in Gander et al. (2014) asymptotic estimates of the best choice of the main parameter in these methods, balancing truncation error and roundoff error, and the study for wave equations is in preparation (Gander et al. 2016a). These methods are however only applicable to linear problems. We propose here a new idea which permits these techniques also to be used for nonlinear problems.

2 Scalar Model Problem

We start with the nonlinear scalar model problem

$$u_t = f(u), \quad u(0) = u_0. \tag{1}$$

Discretization using a backward Euler method with variable time step leads to

$$\frac{u_n - u_{n-1}}{\Delta t_n} = f(u_n), \tag{2}$$

and writing this system over several time steps, we obtain

$$B\mathbf{u} := \begin{pmatrix} \frac{1}{\Delta t_1} & & & \\ -\frac{1}{\Delta t_2} & \frac{1}{\Delta t_2} & & \\ & \ddots & \ddots & \\ & & \frac{1}{\Delta t_n} & \frac{1}{\Delta t_n} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} f(u_1) + \frac{1}{\Delta t_1} u_0 \\ f(u_2) \\ \vdots \\ f(u_n) \end{pmatrix} =: \mathbf{f}(\mathbf{u}). \tag{3}$$

Parallelization in time based on diagonalization uses the assumption that B can be diagonalized, $B = S\Lambda S^{-1}$, which is possible if all the time steps are different. One then diagonalizes the system (3) in time,

$$\Lambda \hat{\mathbf{u}} := S^{-1} B S S^{-1} \mathbf{u} = S^{-1} \mathbf{f}(\mathbf{u}). \tag{4}$$

If the right-hand side is linear, $f(u) = au$, we get with $\mathbf{e}_1 := (1, 0, \ldots, 0)^T$

$$S^{-1}\mathbf{f}(\mathbf{u}) = S^{-1}(a\mathbf{u} + \frac{u_0}{\Delta t_1}\mathbf{e}_1) = a\hat{\mathbf{u}} + \frac{u_0}{\Delta t_1} S^{-1}\mathbf{e}_1,$$

and the system is indeed diagonalized in time, and all time steps can be solved in parallel by a diagonal solve,

$$(\Lambda - aI)\hat{\mathbf{u}} = \frac{u_0}{\Delta t_1} S^{-1}\mathbf{e}_1.$$

The solution is then obtained by simply applying S,

$$\mathbf{u} = S\hat{\mathbf{u}}.$$

Since our problem is nonlinear however, it is not possible to directly diagonalize (4).

Since the discretized system (3) is nonlinear, we will have to apply an iterative method to solve it, e.g. we can apply Newton's method to

$$\mathbf{F}(\mathbf{u}) := B\mathbf{u} - \mathbf{f}(\mathbf{u}) = 0.$$

This leads with some initial guess $\mathbf{u}^0$ to the iteration

$$\mathbf{u}^m = \mathbf{u}^{m-1} - (F'(\mathbf{u}^{m-1}))^{-1}\mathbf{F}(\mathbf{u}^{m-1}).$$

Now the Jacobian is

$$F'(\mathbf{u}) = B - \mathrm{diag}(f'(u_1), f'(u_2), \ldots, f'(u_n)) =: B - D(\mathbf{u}).$$

The Newton iteration can thus be rewritten as

$$\begin{aligned}(B - D(\mathbf{u}^{m-1}))\mathbf{u}^m &= (B - D(\mathbf{u}^{m-1}))\mathbf{u}^{m-1} - (B\mathbf{u}^{m-1} - \mathbf{f}(\mathbf{u}^{m-1})) \\ &= \mathbf{f}(\mathbf{u}^{m-1}) - D(\mathbf{u}^{m-1})\mathbf{u}^{m-1}, \end{aligned} \tag{5}$$

and for a given iteration step $m-1$, $\mathbf{u}^{m-1}$ is known. Denoting by $\tilde{B}^{m-1} := B - D(\mathbf{u}^{m-1})$ and $\tilde{\mathbf{f}}^{m-1} := \mathbf{f}(\mathbf{u}^{m-1}) - D(\mathbf{u}^{m-1})\mathbf{u}^{m-1}$, we have to solve at each iteration step of Newton the evolution problem

$$\tilde{B}^{m-1}\mathbf{u}^m = \tilde{\mathbf{f}}^{m-1}.$$

This can be done by diagonalization now, since it is a linear problem: having $\tilde{B}^{m-1} = \tilde{S}\tilde{\Lambda}\tilde{S}^{-1}$, we can solve

$$\tilde{\Lambda}\hat{\mathbf{u}}^m := \tilde{S}^{-1}\tilde{B}^{m-1}\tilde{S}\tilde{S}^{-1}\mathbf{u}^m = \tilde{S}^{-1}\tilde{\mathbf{f}}^{m-1}$$

for all $\hat{u}_j^m, j = 1, 2, \ldots, n$ in parallel.

A major disadvantage that is brought in by the nonlinear term is that one has to compute a factorization of the time stepping matrix $\tilde{B}^{m-1}$ at each Newton iteration. This could be avoided if we do not use the exact Jacobian at each Newton iteration, but an approximation which uses for example a scalar approximation of the diagonal matrix by averaging,

$$D(\mathbf{u}) \approx \frac{1}{n}\sum_{j=1}^{n} f'(u_j)I.$$

Now we can use the old factorization of the time stepping matrix B and solve in parallel at each quasi Newton step

$$(\Lambda - \frac{1}{n}\sum_{j=1}^{n} f'(u_j^{m-1})I)\hat{\mathbf{u}}^m = \tilde{S}^{-1}\mathbf{f}(\mathbf{u}^{m-1}) - \frac{1}{n}\sum_{j=1}^{n} f'(u_j^{m-1})\mathbf{u}^{m-1}. \tag{6}$$

Using this approximate Jacobian, the quasi Newton method will then however only converge linearly in general, and we will compare in the numerical section the two approaches to see how much is lost due to this approximation.

3 A PDE Model Problem

Suppose we want to solve the time dependent semi-linear heat equation

$$u_t = \Delta u + f(u), \quad u(0,x) = u^0(x), \tag{7}$$

with homogeneous Dirichlet boundary conditions. Using a standard five point finite difference discretization in space over a rectangular grid of size $J = J_1 J_2$, we obtain the discrete problem

$$\frac{\mathbf{u}_n - \mathbf{u}_{n-1}}{\Delta t_n} = \Delta_h \mathbf{u}_n + f(\mathbf{u}_n), \tag{8}$$

where now $\mathbf{u}_n$ and $\mathbf{u}_{n-1}$ are vectors in $\mathbb{R}^J$. As in the scalar case, we need to introduce an iteration to solve this nonlinear problem, but here the system has to be treated also by tensor products to separate space and time. Let I_t be the $N \times N$ identity matrix associated with the time domain and I_x be the $J \times J$ identity matrix associated with the spatial domain. Setting $\mathbf{u} := (\mathbf{u}_1, \ldots, \mathbf{u}_N)$, $\mathbf{f}(\mathbf{u}) := (f(\mathbf{u}_1) + \frac{1}{\Delta t_1}\mathbf{u}_0, f(\mathbf{u}_2), \cdots, f(\mathbf{u}_N))$, and using the Kronecker symbol, we can rewrite (8) as one large nonlinear system,

$$(B \otimes I_x)\mathbf{u} = (I_t \otimes \Delta_h)\mathbf{u} + \mathbf{f}(\mathbf{u}). \tag{9}$$

To solve (9) with an iterative method, one could for example apply Newton's method to solve

$$\mathbf{F}(\mathbf{u}) := (B \otimes I_x - I_t \otimes \Delta_h)\mathbf{u} - \mathbf{f}(\mathbf{u}) = 0.$$

To obtain the Jacobian needed, we define the diagonal matrix function

$$J(\mathbf{u}) := \begin{pmatrix} J_s(\mathbf{u}_1) & & \\ & \ddots & \\ & & J_s(\mathbf{u}_N) \end{pmatrix}, \tag{10}$$

where $J_s(\mathbf{u}_n) := \mathrm{diag}(f'(u_n^1), \cdots, f'(u_n^J)) \in \mathcal{M}_J(\mathbb{R})$. We can then write the Jacobian of $\mathbf{F}$ in compact form,

$$\mathbf{F}'(\mathbf{u}) = B \otimes I_x - I_t \otimes \Delta_h - J(\mathbf{u}).$$

Newton's method corresponds then to computing for $m = 1, 2, \ldots$

$$\left(B \otimes I_x - I_t \otimes \Delta_h - J(\mathbf{u}^{m-1})\right)(\mathbf{u}^m - \mathbf{u}^{m-1}) = f(\mathbf{u}^{m-1}) - (B \otimes I_x - I_t \otimes \Delta_h)\mathbf{u}^{m-1},$$

and we see that the linear terms cancel, so we can simplify to obtain

$$\left(B \otimes I_x - I_t \otimes \Delta_h - J(\mathbf{u}^{m-1})\right)\mathbf{u}^m = f(\mathbf{u}^{m-1}) - J(\mathbf{u}^{m-1})\mathbf{u}^{m-1}. \tag{11}$$

In contrast to the scalar case, where one could simply diagonalize at each Newton iteration a modified time stepping matrix $\tilde{B}^{m-1}$ to keep Newton's method without any approximation, this modified $\tilde{B}^{m-1}$ would here also depend on the space dimension now, and one would have to diagonalize a $\tilde{B}^{m-1}$ matrix at each spatial discretization point, which becomes prohibitive. So we perform a similar approximation as in the scalar case: we define

$$\tilde{J}(\mathbf{u}) := \frac{1}{N}\sum_{n=1}^{N} J_s(\mathbf{u}_n),$$

and obtain with this approximation the quasi-Newton algorithm

$$\left(B \otimes I_x - I_t \otimes (\Delta_h + \tilde{J}(\mathbf{u}^{m-1}))\right)\mathbf{u}^m = \mathbf{f}(\mathbf{u}^{m-1}) - (I_t \otimes \tilde{J}(\mathbf{u}^{m-1}))\mathbf{u}^{m-1}. \tag{12}$$

Now we can use the factorization $B = S\Lambda S^{-1}$, and defining

$$\tilde{\mathbf{f}}^{m-1} := \mathbf{f}(\mathbf{u}^{m-1}) - (I_t \otimes \tilde{J}(\mathbf{u}^{m-1}))\mathbf{u}^{m-1},$$

the quasi-Newton step (12) over all time steps can be parallelized in time by solving

$$(\Lambda \otimes I_x - I_t \otimes (\Delta_h + \tilde{J}(\mathbf{u}^{m-1})))\hat{\mathbf{u}}^m = (S^{-1} \otimes I_x)\tilde{\mathbf{f}}^{m-1}, \tag{13}$$

followed by computing $\mathbf{u}^m = (S \otimes I_x)\hat{\mathbf{u}}^m$.

4 Numerical Experiments

We first show a numerical experiment for the scalar model problem (1) where we chose either $f(u) = -u^2$ or $f(u) = \sqrt{u}$. We solve these problems on the time interval $(0, T)$ using N time steps on a geometrically stretched grid (Gander et al. 2014)

$$\Delta t_n := \frac{(1+\varepsilon)^n}{\sum_{n=1}^{N}(1+\varepsilon)^n} T,$$

with $T = 1$, $N = 10$, and initial condition $u(0) = 1$. We show in Fig. 1 on the left how the time parallel Newton method (5) and the Quasi-Newton method (6) converge for $\varepsilon = 0.05$. Although the approximation leads only to linear convergence, the first few steps lead already to a high accuracy approximation, like for the true Newton method. On the right in Fig. 1, we show how the accuracy at the end of the time interval is influenced by the stretching of the time grid determined by ε. For a highly anisotropic time grid, ε close to 1, the truncation error is bigger than for a time grid with equal time steps (Gander et al. 2014). When ε becomes too small however, then roundoff errors due to the diagonalization process lead to large errors, and an optimal choice has been determined asymptotically for linear problems in Gander et al. (2014). We can see on the right in Fig. 1 that there is also an optimal choice in the nonlinear case, and it seems to be very similar for the two examples we considered.

We next test the algorithm for the PDE model problem (7) using the same two nonlinear functions as for the scalar model problem, homogeneous boundary conditions and initial condition $u(0, x) = 1$. We discretize the Laplacian using a five point finite difference stencil with mesh size $h = 1/20$ and use the same time grid as for the scalar model problem. We show in Fig. 2 on the left how the

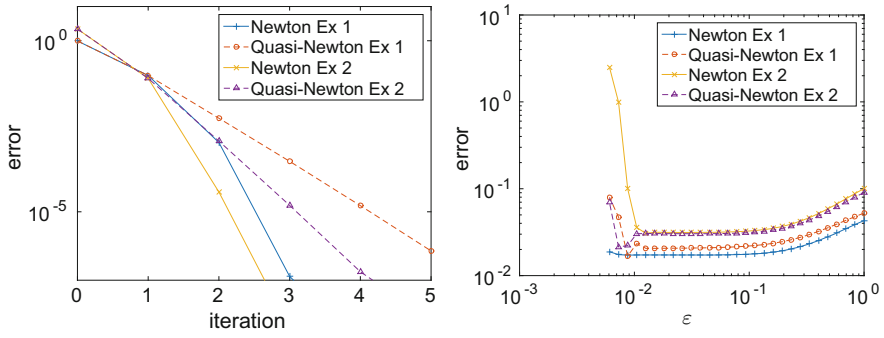

Fig. 1 *Left*: quadratic and linear convergence of the time parallel Newton and Quasi-Newton methods for two scalar model problems. *Right*: accuracy for different choices of the time grid stretching ε

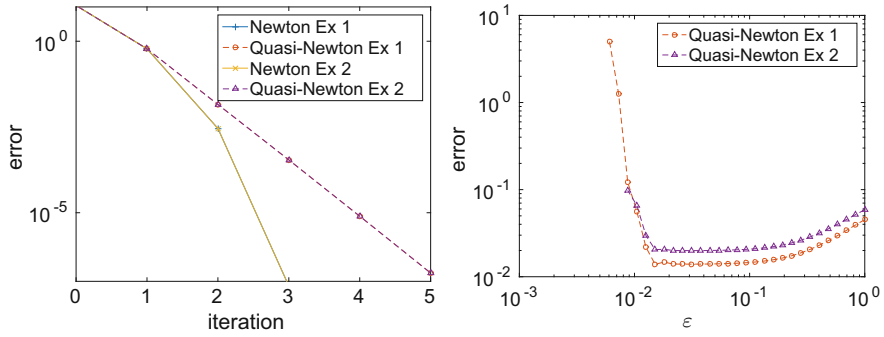

Fig. 2 *Left*: linear convergence of the time parallel Quasi-Newton method for two PDE model problems. *Right*: accuracy for different choices of the time grid stretching ε

Newton method (11) which can only be time parallelized at the cost of many time stepping matrix factorizations, and the Quasi-Newton method (13) that is easily time parallelized converge. Again the approximation still leads to a rapidly converging method. On the right in Fig. 2, we show how the accuracy at the end of the time interval is influenced by the stretching of the time grid in the PDE case, and again we see that there is an optimal choice for the stretching parameter.

5 Conclusion

We have introduced a new method which allows us to use diagonalization for time parallelization also for nonlinear problems. We have shown two variants for nonlinear scalar problems, and one for a nonlinear PDE. Numerical experiments show that the methods converge rapidly, and there is also an optimal choice of the geometric time grid stretching, like in the original algorithm for linear problems (Gander et al. 2014, 2016a). The geometric stretching is only one way to make diagonalization possible: random or adaptive time steps could also be used, but they must be determined for the entire time window before its parallel solve, and they must all be different, otherwise the diagonalization is not possible. In an adaptive setting, one could adaptively determine a macro time step with a larger tolerance as time window, before parallelizing its solve with smaller geometric or random time steps. We are currently investigating such variants, and also the generalization to nonlinear hyperbolic problems.

References

D. Bennequin, M.J. Gander, L. Halpern, A homographic best approximation problem with application to optimized Schwarz waveform relaxation. Math. Comput. **78**(265), 185–223 (2009)

M. Bjørhus, On domain decomposition, subdomain iteration and waveform relaxation. PhD thesis, University of Trondheim, Norway (1995)

P. Chartier, B. Philippe, A parallel shooting technique for solving dissipative ODEs. Computing **51**, 209–236 (1993)

A.J. Christlieb, C.B. Macdonald, B.W. Ong, Parallel high-order integrators. SIAM J. Sci. Comput. **32**(2), 818–835 (2010)

M. Emmett, M.L. Minion, Toward an efficient parallel in time method for partial differential equations. Commun. Appl. Math. Comput. Sci **7**(1), 105–132 (2012)

M.J. Gander, 50 years of time parallel time integration, in *Multiple Shooting and Time Domain Decomposition Methods* (Springer, Berlin, 2015), pp. 69–113

M.J. Gander, S. Güttel, Paraexp: a parallel integrator for linear initial-value problems. SIAM J. Sci. Comput. **35**(2), C123–C142 (2013)

M.J. Gander, E. Hairer, Nonlinear convergence analysis for the parareal algorithm. in *Domain Decomposition Methods in Science and Engineering XVII*, vol. 60 (Springer, Berlin, 2008), pp. 45–56

M.J. Gander, L. Halpern, Absorbing boundary conditions for the wave equation and parallel computing. Math. Comput. **74**, 153–176 (2005)

M.J. Gander, L. Halpern, Optimized Schwarz waveform relaxation methods for advection reaction diffusion problems. SIAM J. Numer. Anal. **45**(2), 666–697 (2007)

M.J. Gander, M. Neumüller, Analysis of a new space-time parallel multigrid algorithm for parabolic problems. SIAM J. Sci. Comput. **38**(4), A2173–A2208 (2016)

M.J. Gander, A.M. Stuart, Space-time continuous analysis of waveform relaxation for the heat equation. SIAM J. Sci. Comput. **19**(6), 2014–2031 (1998)

M.J. Gander, S. Vandewalle, Analysis of the parareal time-parallel time-integration method. SIAM J. Sci. Comput. **29**(2), 556–578 (2007)

M.J. Gander, L. Halpern, F. Nataf, Optimal Schwarz waveform relaxation for the one dimensional wave equation. SIAM J. Numer. Anal. **41**(5), 1643–1681 (2003)

M.J. Gander, L. Halpern, J. Ryan, T.T.B. Tran, A direct solver for time parallelization, in *22nd International Conference of Domain Decomposition Methods* (Springer, Berlin, 2014)

M.J. Gander, L. Halpern, J. Rannou, J. Ryan, A direct solver for time parallelization of the wave equation. (2016a, in preparation)

M.J. Gander, F. Kwok, B. Mandal, Dirichlet-Neumann and Neumann-Neumann waveform relaxation algorithms for parabolic problems. Electron. Trans. Numer. Anal. **45**, 424–456 (2016b)

E. Giladi, H.B. Keller, Space time domain decomposition for parabolic problems. Numer. Math. **93**(2), 279–313 (2002)

W. Hackbusch, Parabolic multi-grid methods, in *Computing Methods in Applied Sciences and Engineering, VI*, (North-Holland, Amsterdam, 1984), pp. 189–197

G. Horton, S. Vandewalle, A space-time multigrid method for parabolic partial differential equations. SIAM J. Sci. Comput. **16**(4), 848–864 (1995)

F. Kwok, Neumann–Neumann waveform relaxation for the time-dependent heat equation, in *Domain Decomposition Methods in Science and Engineering XXI* (Springer, Berlin, 2014), pp. 189–198

J.L. Lions, Y. Maday, G. Turinici, A parareal in time discretization of PDE's. C.R. Acad. Sci. Paris Ser. I **332**, 661–668 (2001)

Y. Maday, E.M. Rønquist, Parallelization in time through tensor-product space-time solvers. C. R. Math. Acad. Sci. Paris **346**(1–2), 113–118 (2008)

B. Mandal, A time-dependent Dirichlet-Neumann method for the heat equation, in *Domain Decomposition Methods in Science and Engineering, DD21* (Springer, Berlin, 2014)