Adaptive Parareal for systems of ODEs

D. Guibert[1] and D. Tromeur-Dervout[1]

CDCSP/UMR5208 University Lyon 1-CNRS, 69622 Villeurbanne cedex, France

Summary. The parareal scheme (resp. PITA algorithm) proposed in [4] (resp. [1]) considers two levels of grids in time in order to split the domain in time-subdomains. A prediction of the solution is computed on the fine grid in parallel. Then at each end boundaries of time subdomains, the solution makes a jump with the previous initial boundary value (IBV) of the next time-subdomain . A correction of the IBV for the next fine grid iteration is then computed on the coarse grid in time. In this paper, it is proposed to investigate adaptivity in the time slice decomposition based on an a posteriori numerical estimation obtained from the time step behavior on coarse grids. The outline of this paper is as follows: in section 1, the original parareal method is reminded and it is shown that the latter is a particular case of the multiple shooting method of Deuflhard [2]. Then in section 2, the definition of the fineness of the grids is slightly modified in order to introduce adaptivity within the parareal algorithm for the time stepping, the number of subdomains, and the time decomposition. This adaptivity leads to an improvement of the method so as to solve moderately stiff nonlinear ODEs problems. Nevertheless for very stiff problems as the Oregonator model, it fails even with the introduced adaptivity. This leads to develop in section 3 an adaptive parallel extrapolation method, based on a posteriori numerical assessment, which obtains results on this stiff problem.

1 Parareal and the BVPSOL multiple shooting method

The principle of the parareal algorithm to solve

$$\frac{dy}{dt} = f(y,t), \quad \forall t \in \Omega_t =]T^0, T^f], \quad \text{with} \quad y(t^0) = y_0 \tag{1}$$

consists in splitting the time domain in m time-slices $\left\{S^i = [t^i, t^{i+1}]\right\}$ of different sizes with $t^0 = T^0$ and $t^m = T^f$. Let Y^i denote the values of the exact

[1] This work was funded by the thema "mathématiques" of the Région Rhône-Alpes thru 2003-2006 the project: "Développement de méthodologies mathématiques pour le calcul scientifique sur grille". The second author is partially supported by the GDR-CNRS MOMAS thru the project: "Simulation et solveurs multidomaines"

solution of problem (1) at the beginning of the time-slice S^i. The principle of the parareal algorithm consists in defining an approximation Y_k^i of these Y^i on a coarsest grid. Y_k^i known, the solution $y_k^i(t)$ on the m time-slices S^i can be computed as:

$$\frac{dy_k^i}{dt} = f(y_k^i, t), \ \ \forall t \in S^i, \ \text{ with } \ y_k^i(t^i) = Y_k^i. \tag{2}$$

These solutions exhibit jumps $\Delta_k^i = y_k^{i-1}(t^i) - Y_k^i, \quad 1 \leq i \leq m-1$ at the time-instances t^i. A correction function c_k piecewise C^1 in Ω_t is introduced to update the Y_k^i values with a Newton-type linearized method around y_k,

$$\frac{dc_k}{dt} = F_y(y_k, t)c_k, \text{ with } c_k(t^0) = 0, \text{ and} \tag{3}$$

$$c_k(t^{i+}) = c_k(t^{i-}) + \Delta_k^i, \ 1 \leq i \leq m-1. \tag{4}$$

It exists a link between the parareal method and the multiple shooting method (BVPSOL , [2]). As in the parareal method, the multiple shooting method uses a Newton process to vanish the jumps of the solution at the end of time slices. If $\left\{T^0 = t^1 < t^2 < ... < t^m = T^f\right\}$ where $m > 2$, represents a decomposition of the time interval and x_j estimates the unknown values at the nodes t_j, then the solution with the initial value x_j on the time slice $S^j = [t^j, t^{j+1}]$ can be written as $y_j(t) = \Phi^{t,t^j} x_j, t \in S^j, j = 1, ..., m-1$ where Φ^{t,t^j} represents the flow trajectory starting from t^j.

For the solution of the problem the sub-trajectories have to be joined continuously and hence at the intermediate nodes the n continuity conditions $F_j(x_j, x_{j+1}) = \Phi^{t,t^j} x_j - x_{j+1} = 0, j = 1, ..., m-1$, have to hold. In addition the n boundary conditions $F_m(x_1, x_m) = r(x_1, x_m) = 0$ must be verified.

$$x = (x_1, \ldots, x_m)^T \in R^{n.m}, F(x) = (F_1(x_1, x_2), \ldots, F_m(x_1, x_m))^T \tag{5}$$

The BVPSOL finds the zeros of F by means of an ordinary Newton correction

$$F'(x_k)\delta x_k = -F(x_k), x_{k+1} = x_k + \Delta x_k, k = 0, 1, ... \tag{6}$$

The corresponding Jacobian matrix has the cyclic block structure:

$$J = F'(x) = \begin{bmatrix} G_1 & -I & & \\ & \ddots & \ddots & \\ & & G_{m-1} & -I \\ A & & & B \end{bmatrix} \tag{7}$$

where A et B are the derivatives of the boundary conditions r with respect to the boundary values (x_1, x_m) and $G_j = \partial \Phi^{t,t_j} x_j / \partial x_j, j = 1, ..., m-1$.

Proposition 1. *if $B = 0$ then one iteration of the multiple shooting method is one iterate of the parareal algorithm in which the correction step is a purely sequential process.*

Proof. the resolution of eq (6) with Jacobian matrix given by eq (7) can be reduced to solve [[2],p.319]:

1. evaluate by recursion over $j = 1, \dots, m-1$ $E := A + BG_{m-1} \dots G_1$,
   $u := r + B[F_{m-1} + G_{m-1}F_{m-2} + \dots + G_{m-1} \dots G_2F_1]$
2. solve the linear (n,n)-system $E\Delta x_1 = -u$
3. Execute explicit recursion

$$\Delta x_{j+1} = G_j \Delta x_j + F_j,\ j = 1, \dots, m-1 \tag{8}$$

If $B = 0$ then $E = A$ and $u = r$ thus eq (8) is reduced to parareal eq (4).

A simple way to transform an IBV problem to a BVP is to consider the following statement. Instead of considering the problem on the $[0, T]$ time span, one can consider a time forward integration from 0 to T and then a time backward integration from T to 0. Then eq (7) becomes:

$$J = F'(x) = \begin{bmatrix} G_1 & -I & & & & & \\ & \ddots & \ddots & & & & \\ & & G_{m-1} & -I & & & \\ & & & \bar{G}_{m-1} & -I & & \\ & & & & \ddots & \ddots & \\ & & & & & \bar{G}_1 & -I \\ A & & & & & & A \end{bmatrix} \tag{9}$$

where $\bar{G}_j$ is related to the integration with time step $-h$.

Proposition 2. *Consider the IBV problem (1) with $f(t, y) = -\alpha y$, $\alpha > 0$, then the error when solving the BVP (9) with the first order Euler explicit scheme is of the same order than the consistency error of the scheme. More precisely the error produced at the end boundary value is of order $O(h)$.*

Proof. the Euler forward time integration scheme on $]0, T]$ with $h = T/n$ and IBV y_0 gives $y(T) \simeq y_n = (1 - h\alpha)^n y_0$ then the Euler backward time integration on]T,0] with IBV y_n gives $y_{2n+1} = (1 + h\alpha)^n y_n = (1 - (h\alpha)^2)^n y_0$ instead of y_0. Thus the error with the IBV y_{2n+1} is $O(n\alpha^2T^2/n^2) = O(h)$ when n is sufficiently large.

2 Parareal revisited

Test problems with increasing stiffness

Consider three classical IBV problems with increasing stiffness to validate and to give the limitation of the given methodologies. The first model (eq (10)) is a linear problem with oscillations. Thus the linearization of the correction step is exact as the jacobian matrix is constant. The second (eq (11)) is the prey-predator model with the lotka-Volterra 2×2 system of nonlinear ODEs. One can increase the solution oscillations by playing with the multivalues parameter μ. The third ((eq (12))is the Oregonator model with a 3×3 system of nonlinear ODEs associated with the Belousov-Zhabotinskii (BZ) reaction. This problem can be very stiff for some ranges of the parameter μ.

$$\begin{cases} dx/dt = -\mu_1 x + \mu_2 y \\ dy/dt = \mu_3 x + -\mu_4 y \end{cases} \tag{10}$$

$$\begin{cases} dx/dt = \mu_1 x - \mu_2 xy \\ dy/dt = \mu_3 y - \mu_4 xy \end{cases} \tag{11}$$

$$\begin{cases} \mu_2 dx/dt = \mu_1 y - xy + x(1-x), \\ \mu_3 dy/dt = -\mu_1 y - xy + \mu_4 z, \\ dz/dt \quad = x - z. \end{cases} \tag{12}$$

Introducing adaptivity

A new definition for the grids fineness is introduced below. The fine and coarse grids are not defined by the size of the time steps but by the relative tolerance of the time integrator. The advantage is that adaptivity in the time step can occur in order to overcome the strong nonlinearities. For two approximations of order p and $\hat{p}$ of solution y_1 and $\hat{y}_1$ the estimation error for the less precise is $y_1 - \hat{y}_1$. The time step h is chosen to give $|y_{1i} - \hat{y}_{1i}| \leq max(y_{0i}, y_{1i}) Rtol = sc$. The new time step h_{new} is obtained as follows $h_{new} = h.min(facmax, max(facmin, fac.(1/err)^{1/(q+1)}))$ where $q = min(p, \hat{p})$ and $fac, facmin, facmax$ are constant factors to avoid too fast decrease/increase of the time step, and $err = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{y_{1i}-\hat{y}_{1i}}{sc})^2}$. The computational complexity of the modified Rosenbrok(2,3) method (implemented in the ode23s matlab procedure for the test IBV problems eqs (11) and (12)) has a number of function evaluations that increases nearby 2.3 when rtol is divided by 10. Let α be the reduction cost coefficient of the elapsed time of the time integrator between run on the grid defined by rtol and those on the grid defined by rtol/10. This coefficient α can be considered as constant. This assumption seems reasonable considering the previous result, excepted for the rejected step which increases nonlinearly when rtol decreases. Let $TIcost(rtol)$ be the elapsed time of the time integrator which solves the problem with a relative tolerance rtol. Let $PararealCost(rtol, rtol/10^l)$ be the computational cost to solve with the modified with the parareal method. This method is applied on P processors using two grids defined by $rtol$ and $rtol/10^l$. Then to perform it iterations, the cost can be roughly approximated by $it \times (1/P + \alpha^l) \times TIcost(rtol)$.

Techniques to evaluate the error and linear problem results

The numerical process to give a measure on the error quality is as follows: first, a fine grid solution is computed with the time integrator using an accuracy $rtol = 10^{-d}$, $d = 8$. Then the solution of the modified parareal is computed with a fine grid defined by $rtol = 10^{-l}, l < d$. As the time integrator can use different step sizes, the two solutions do not match on the same time interpolation points. The solution of reference is then interpolated to the time mesh generated by the parareal algorithm.

Table 1 (top) shows the effect of the interpolation techniques (linear, cubic, spline) on the measure of error for the norm of maximum for the linear problem with 10 subdomains and $rtol = 10^{-7}$ for initializing and for correction. Convergence is guaranteed in at most 10 iterations as the exact IBV is propagated. Good results are obtained with cubic interpolation which will be used to give all results that follow. Table 1 (bottom) shows the convergence of the modified parareal for linear problem with $rtol = \left\{10^{-4}, 10^{-5}, 10^{-6}\right\}$ accuracy for the initial guess and the correction, and $rtol = 10^{-7}$ for fine grid solution.

	linear	cubic	spline
error (parareal iterates)	-4 (2)	-6 (2)	-6 (2)

coarse grid 10^d	d=-4	d=-5	d=-6
error (parareal iterates)	-6(6)	-6(4)	-6(3)

Table 1. Effect of the interpolation technique for the error measurement (top) and convergence of the modified parareal algorithm on eq (10) with $\mu = (-1e-3, 1, 10, -1e-3)$ on 10 subdomains.

Numerical results for non-linear test problems

Figure 1 shows the convergence of the method for the Lotka-Volterra problem. For 10 time subdomains, the method blows up for $rtol = 10^{-3}, 10^{-4}$ and finally converges at the 10^{th} iterate. Even with $rtol = 10^{-7}$ convergence takes 7 iterates. For this number of subdomains the method has no interest. Nevertheless, if the number of equal size subdomains is increased to 200, the convergence is reached between 5 and 7 iterates for $rtol = 10^{-6}$ and 10^{-3}, providing speed up. For 1168 subdomains convergence is obtained between 2 and 7 iterates for $rtol = 10^{-6}$ to 10^{-3}. Let us notice that the correction can converge to 10^{-14} but the convergence on solution is limited by the fine grid solver (here $rtol = 10^{-7}$. It is not necessary to reach the machine accuracy for correction to have the effective convergence on fine grid.

The behavior of step size is the same for the rtol grids. Moreover, the reduction of the step size indicates directly the stiffness on solution. The behavior of the time integrator adaptivity on the coarse grid can be useful to introduce adaptivity in the time decomposition. This decomposition is based

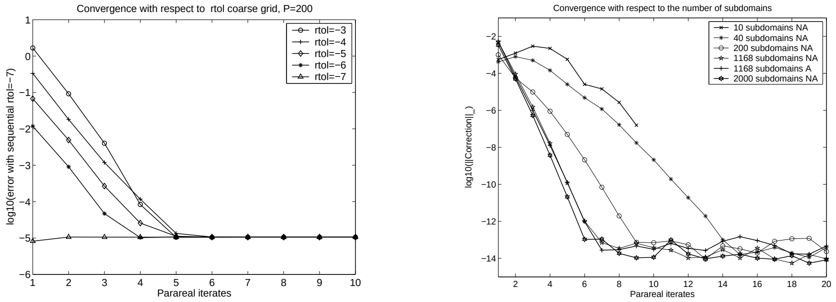

Fig. 1. Modified parareal convergence for Lotka-Volterra Problem on $\mu = (1.5, 1, 3, 1)$ with 200 subdomains with respect to different rtol for the initialization and the correction (left). The convergence of the correction in parareal with respect to the number of subdomains with and without adaptivity and $rtol = 10^{-5}$ for the correction (right)

on the time steps of the time integrator during the coarse grid initialization. Then the size of the subdomain is adapted with respect to strong variations of the step size. Figure 1 shows that the number of subdomains 1168 defined by the adaptivity (A) gives better results. Nevertheless, this is not the optimal number of subdomains, 2000 regular subdomains (NA) lead to a faster convergence, and 1168 regular subdomains give quite the same convergence. For the Oregonator problem, the convergence blows up even with 1000 subdomains for $rtol = 10^{-6}$ to 10^{-3} and even with time decomposition adaptivity. Theses experiments show that new solutions are needed for stiff and very stiff problems with keeping the two major features of parareal algorithm: split the time domain in slice then provide a first good initial boundary value at each slice. The correction of the solution based on linearizing and solving a problem with the jacobian of f seems to be sensitive to the behavior of the solution notably when strong nonlinearity effects occur.

3 Adaptive Parallel Extrapolation

The same behavior for the time step adaptation provided by the solver on coarse grids suggests that we can use some combination of solutions like "Richardson extrapolation" based on the solver. Moreover, as the first subdomain has the exact solution, on can compute from the different grid levels exact extrapolation coefficients based on some control points values. Then the extrapolation coefficients can be broadcasted to the others subdomains where extrapolations are performed. Let us describe the algorithm.

Adaptive Parallel Extrapolation Algorithm:

A) Define some decomposition of the time domain and in each subdomain add some control points $(t_{i,k})_{0\leq k\leq l}$ which are points of the time slice $[t_{i,0}, t_{i,l}] = [T_i, T_{i+1}]$

B) Evaluate the solution on coarse grids $rtol_1 > ... > rtol_l$.

C) Initialize IBV of time slices for the finest grid $rtol_f$ with a Richardson extrapolation based on coarse grids and the first time slice of the fine grid $rtol_f$ as it gives the exact solution. Use the value of the solution at the control points (without the first control point which is a given data and consequently is not a result provided by the time integrator approximation scheme) on this first time slice to define the operators of extrapolation. Let $y^k(t_1, j), 1 \leq j \leq lP, 1 \leq k \leq l$ be the computed solution values on grid $rtol_k$ at the control point $t_{1,j}$. The extrapolation operator can be computed with the formula as follows:

$$\begin{pmatrix} y^1(t_{1,1}) & y^2(t_{1,1}) & \cdots & y^l(t_{1,1}) \\ y^1(t_{2,1}) & y^2(t_{2,1}) & \cdots & y^l(t_{2,1}) \\ \vdots & \vdots & \ddots & \vdots \\ y^1(t_{l,1}) & y^2(t_{l,1}) & \cdots & y^l(t_{l,1}) \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_l \end{pmatrix} = \begin{pmatrix} y^{l+1}(t_{1,1}) \\ y^{l+1}(t_{2,1}) \\ \vdots \\ y^{l+1}(t_{l,1}) \end{pmatrix} \tag{13}$$

D) Propagate the operator of extrapolation to the other time slices and compute the extrapolated solution with it as follows:

$$y^{l+1}(t_{0,j}) = \sum_{k=1}^{l} \beta_k y^k(t_{l,j-1}),\ 2 \leq j \leq P \tag{14}$$

E) In order to get the time step behavior lost in the extrapolation compute in parallel the solution on each time-slice for the finest grid. The first time-slice has the exact solution for the finest grid (exact IBV).

F) Apply recursively with a new rtolf grid.

Results on Adaptive Parallel Extrapolation

Figure 2 gives the error at the control point between the sequential solution at $rtol = 10^{-10}$ and the Adaptive Parallel Extrapolation with 10 time subdomains and with a fine grid $rtol = 10^{-8}$ for the Oregonator problem. It exhibits that good approximations of the initial guess for each time slice are obtained even with using two grids with $rtol = 10^{-6}$ and 10^{-5}. The computational cost to define a very good approximation of time slice initial guess is reduce by a factor nearby 2.3^2 for the considered time integrator. Notice that the behaviors of the time step are the same between the sequential solution and the first level solution of adaptive parallel extrapolation. The recursive application of the solution obtained with the Adaptive Parallel Extrapolation gives globally better results excepted for some localized region. Improvements should be obtained with a local refinement on the grids used for the extrapolation.

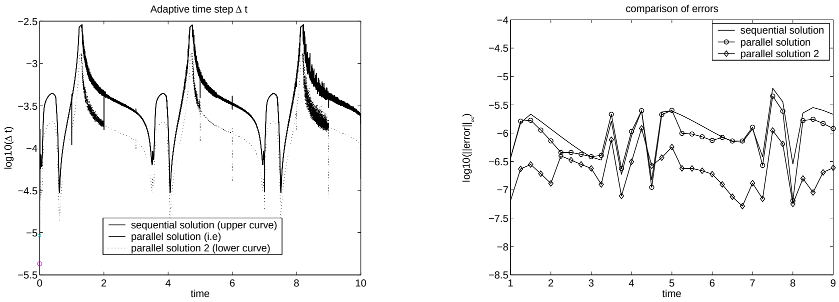

Fig. 2. Comparison of the time step behavior (left) and Error with the solution on a reference grid ($rtol = 10{-}10$) (right) between the sequential algorithm and the Adaptive Parallel Extrapolation with 2 level of extrapolation for the Oregonator problem defined by ($\mu = (1e-2, 1e-3, 1e-2, 1)$.

Conclusions

The equivalence between the parareal method and the multiple shooting method has been established. Then adaptivity in the parareal parallel ODEs solvers has been introduced in order to apply its concepts to stiff ODEs. Some improvements in the method has been shown by defining the fineness of the grids on the relative tolerance of the time slice integrator and by adapting on the number of subdomains. Nevertheless, for very stiff problems, the linearization of the jacobian in the correction steps makes the method very sensitive to blow up. Another parallel solver has been proposed for stiff ODEs based on Richardson Extrapolation. The extrapolation coefficients are based on the time integrator behavior like in the classical Richardson extrapolation but with an a posteriori estimation based on the right solution values of the first subdomain at the control points.

References

1. C. Farhat and M. Chandesris. *Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications*. Int. J. Numer. Meth. Engng., **58**(9), pp. 1397-1434, 2003.
2. P. Deuflhard, *Newton Methods for Non Linear Problems*, Springer Series in Computational Mathematics, Vol. 35, Springer, 2004.
3. E. Hairer, S. P. Norsett and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and DifferentialAlgebraic Problems*, Springer Series in Computational Mathematics, Springer, Berlin, 1991.
4. J.-L. Lions, Y. Maday, and G. Turinici. *Résolution d'EDP par un schéma en temps "pararéel"*, C.R.A.S. Sér. I Math., **332**(7), pp. 661-668, 2000.