Parallel Implementations of Broyden’s Method

F. A. M. Gomes and J. M. Martinez, Campinas

Received January 14, 1991; revised July 22, 1991

Abstract — Zusammenfassung

Parallel Implementations of Broyden’s Method. Broyden’s method is one of the most effective algorithms for solving nonlinear systems of equations. When the number of equations and unknowns is very large, memoryless implementations of this method are frequently used. We analyze one of this implementations (Gomes-Ruggiero, Martinez and Moretti, SIAM J. Sci. Stat. Comput. 1991, to appear) and we show that calculations may be organized in such a way that parallelism can be exploited.

AMS Subject Classification: 65H10

Key words: Nonlinear systems, Broyden’s method


1. Introduction

We consider the problem of solving

\[ F(x) = 0 \quad (1) \]

where \( F: \mathbb{R}^n \rightarrow \mathbb{R}^n \), \( F \) is differentiable and \( n \) is large.

One of the most effective ways of solving (1) is Broyden’s “good” method (Broyden [1965], Dennis and Schnabel [1983]). Given an initial estimate \( x_0 \in \mathbb{R}^n \) and an initial nonsingular \( n \times n \) matrix \( B_0 \), this method is defined by:

\[
x_{k+1} = x_k - \lambda_k B_k^{-1} F(x_k), \quad (2)
\]

\[
B_{k+1} = B_k + \frac{(y_k - B_k s_k)s_k^T}{s_k^T s_k}, \quad (3)
\]

where

\[
s_k = x_{k+1} - x_k \quad (4)
\]

Work supported by FAPESP (grant 90/3724-6), CNPq, FINEP and FAEP-UNICAMP.
and

$$y_k = F(x_{k+1}) - F(x_k)$$  \hfill (5)

for \( k = 0, 1, 2, \ldots \). \( \lambda_k \) is a real parameter used to improve the global convergence properties of the method.

Even if the Jacobian matrix \( F'(x) \) is sparse, the successive Jacobian approximations \( B_k \) may be dense. However, it is currently accepted that implementations of Broyden's method that proceed storing a finite number of past update vectors are very effective for saving computer time and memory when we are solving (1) by means of (2)–(3). (See Matthies and Strang [1979], Griewank [1986], Gomes-Ruggiero, Martinez and Moretti [1992]).

The implementation of Gomes-Ruggiero, Martinez and Moretti is described by the following algorithm.

**Algorithm 1**

Given \( x_0, B_0 \), compute \( x_{k+1}, k = 0, 1, 2, \ldots \) performing the following steps:

**Step 1.** If \( k = 0 \), compute an \( L - U \) factorization of \( B_0 \) and solve

$$B_0 s_0 = -F(x_0).$$

Go to Step 3.

**Step 2** (Complete the computation of \( s_k \))

Compute

$$s_k = (I + u_{k-1}s_{k-1})s_{k-1}$$

**Step 3** (Save the step and compute the new point)

$$\tilde{s}_k = s_k$$

$$s_k = \lambda_k s_k$$

(where \( \lambda_k \in \mathbb{R} \) is chosen by global criteria)

$$x_{k+1} = x_k + s_k.$$  

**Step 4** (Compute \( u_k \))

Execute Steps 4.1–4.3

**Step 4.1** (Compute \( \tilde{s}_k = -B_k^{-1}F(x_{k+1}) \))

Execute Steps 4.1.1–4.1.2

**Step 4.1.1** (Compute \( w_k = -B_0^{-1}F(x_{k+1}) \))

Solve

$$B_0 w_k = -F(x_{k+1}).$$

If \( k = 0 \) set \( \tilde{s}_k = w_k \) and go to Step 4.2.

**Step 4.1.2**

Compute

$$\tilde{s}_k = (I + u_{k-1}s_{k-1}^T)\ldots(I + u_0s_0^T)w_k.$$