Institute of Scientific Computing
Technische Universität Braunschweig
Prof. Hermann Matthies
Dr. Dishi Liu

Summer Term 2018
4.06.2018

Partitioned Methods for Multifield Problems:
Assignment 4:Block Newton methods

Suppose we have two coupled m-dimensional systems:

$$\begin{aligned} \mathbf{x} \circ \mathbf{y} - \frac{1}{3}\langle \mathbf{x}, \mathbf{y} \rangle + 6 = 0 \\ \mathbf{x} + 2\mathbf{y} - 6 = 0, \end{aligned} \tag{1}$$

with $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$. The $\mathbf{x} \circ \mathbf{y}$ denotes an element-wise product (Hadamard product), and $\langle \mathbf{x}, \mathbf{y} \rangle$ inner product. This can also be re-written in a way suitable for a partitioned fixed point iteration:

$$\begin{aligned} \mathbf{x} &= \mathbf{x} - \mathbf{x} \circ \mathbf{y} + \frac{1}{3}\langle \mathbf{x}, \mathbf{y} \rangle - 6 &&= \boldsymbol{f}(\mathbf{x}, \mathbf{y}) \\ \mathbf{y} &= -\mathbf{y} - \mathbf{x} + 6 &&= \boldsymbol{g}(\mathbf{x}, \mathbf{y}) \end{aligned}$$

Exercise 1: **(10 points)**

Write and run a Matlab program to solve the problem with a block Newton method (as introduced in the downloadable lecture notes), starting with an initial guess $\mathbf{x}_0 = \mathbf{y}_0 = [1, 1]^T$. The function $\boldsymbol{f}$ and $\boldsymbol{g}$ and their Jacobi are given in the downloadable Matlab code for this assignment. Note that in this task no derivative approximation is required.

Exercise 2: **(26 points)**

If we are going to use individual solvers respectively specified to the two partitioned problems, the cross-derivative Jacobi $\boldsymbol{f}_y$ and $\boldsymbol{g}_x$ are no longer available.

(a) In the code for block Newton method, replace $\boldsymbol{g}(\mathbf{x}, \mathbf{y}) + \mathbf{g}_x \mathbf{q}$ by a finite difference approximation to avoid $\mathbf{g}_x$. (4 points)

(b) Write a Matlab function $\mathbf{fun1}(\boldsymbol{v})$ that returns an approximate product of the Jacobi matrix $\boldsymbol{f}_y$ and an input vector $\boldsymbol{v}$, use a finite difference approximation to avoid using $\boldsymbol{f}_y$. (6 points)

(c) Write a Matlab function "run_bicgstab()" that solves for $\Delta\mathbf{y}$ (the 2nd step in the block Newton procedure) using Matlab built-in routine "bicgstab" . This function should not use the Jacobi $\boldsymbol{f}_y$ and $\boldsymbol{g}_x$. For some hints check the given template code and the comment there. (10 points)

(d) Use the above approximation to finish an approximate block Newton program that solves the coupled system. (6 points)