Gaddam Shiva Kumar MM22D014

Parallel Scientific Computing (ID5130)

Assignment-1

Due Date: 5 Mar 2024 Maximum Marks: 100

Problem 1:

The given system of linear equations are

$$\begin{aligned} 3x_1 - x_2 &= 2 \\ -x_1 + 3x_2 - x_3 &= 1 \\ -x_2 + 3x_3 - x_4 &= 1 \\ -x_3 + 3x_4 &= 2. \end{aligned}$$

Upon rewriting the above system of equations

$$3x_1 - x_2 = 2 \implies x_1 = 2/3 + x_2/3 \tag{1}$$
$$-x_1 + 3x_2 - x_3 = 1 \implies x_2 = 1/3 + x_1/3 + x_3/3 \tag{2}$$
$$-x_2 + 3x_3 - x_4 = 1 \implies x_3 = 1/3 + x_2/3 + x_4/3 \tag{3}$$
$$-x_3 + 3x_4 = 2 \implies x_4 = 2/3 + x_3/3. \tag{4}$$

By **using recursive-doubling algorithm**, equation (1) is rewritten by substituting equation (2) in place of x_2. equation (2) is rewritten by substituting equations (1) and (3) in place of x_1 and x_3 respectively. equation (3) is rewritten by substituting equations (2) and (4) in place of x_2 and x_4 respectively. equation (4) is rewritten by substituting equation (3) in place of x_3. The resulting equations are as follows.

$$\begin{aligned} (1) &\implies x_1 = 2/3 + (1/9 + x_1/9 + x_3/9) \\ (2) &\implies x_2 = 1/3 + (2/9 + x_2/9) + (1/9 + x_2/9 + x_4/9) \\ (3) &\implies x_3 = 1/3 + (1/9 + x_1/9 + x_3/9) + (2/9 + x_3/9) \\ (4) &\implies x_4 = 2/3 + (1/9 + x_2/9 + x_4/9) \end{aligned}$$

$$\begin{aligned} (1) &\implies x_1 - x_1/9 = 7/9 + x_3/9 \\ (2) &\implies x_2 - x_2/9 - x_2/9 = 6/9 + x_4/9 \\ (3) &\implies x_3 - x_3/9 - x_3/9 = 6/9 + x_1/9 \\ (4) &\implies x_4 - x_4/9 = 7/9 + x_2/9 \end{aligned}$$

$$
\begin{aligned}
(1) &\implies 8x_1/9 = 7/9 + x_3/9 \\
(2) &\implies 7x_2/9 = 6/9 + x_4/9 \\
(3) &\implies 7x_3/9 = 6/9 + x_1/9 \\
(4) &\implies 8x_4/9 = 7/9 + x_2/9
\end{aligned}
$$

$$
\begin{aligned}
&\implies x_1 = 7/8 + x_3/8 && (5) \\
&\implies x_2 = 6/7 + x_4/7 && (6) \\
&\implies x_3 = 6/7 + x_1/7 && (7) \\
&\implies x_4 = 7/8 + x_2/8 && (8)
\end{aligned}
$$

equation (5) is rewritten by substituting equation (7) in place of x_3. equation (6) is rewritten by substituting equation (8) in place of x_4. equation (7) is rewritten by substituting equation (5) in place of x_1. equation (8) is rewritten by substituting equation (6) in place of x_2. The resulting equations are as follows.

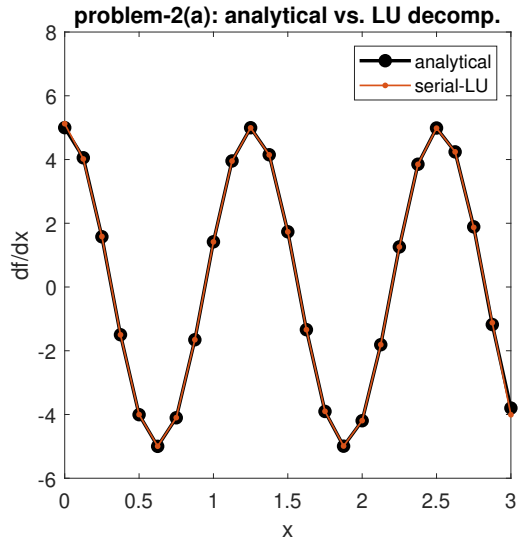
$$
\begin{aligned}
(5) &\implies x_1 = 7/8 + (6/7 + x_1/7)/8 \\
(6) &\implies x_2 = 6/7 + (7/8 + x_2/8)/7 \\
(7) &\implies x_3 = 6/7 + (7/8 + x_3/8)/7 \\
(8) &\implies x_4 = 7/8 + (6/7 + x_4/7)/8
\end{aligned}
$$

$$
\begin{aligned}
(5) &\implies x_1 = 7/8 + 6/56 + x_1/56 \\
(6) &\implies x_2 = 6/7 + 7/56 + x_2/56 \\
(7) &\implies x_3 = 6/7 + 7/56 + x_3/56 \\
(8) &\implies x_4 = 7/8 + 6/56 + x_4/56
\end{aligned}
$$

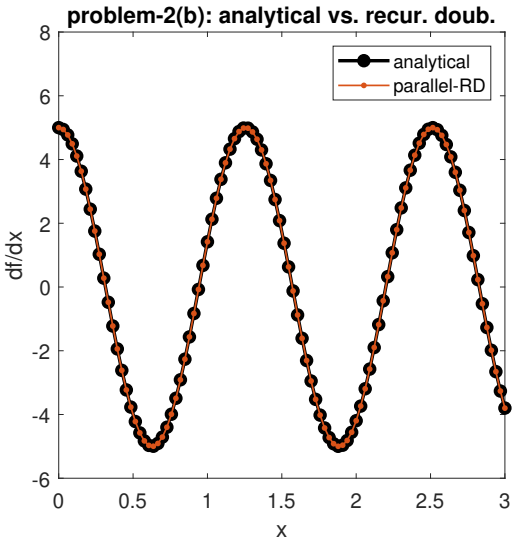
$$
\begin{aligned}
(5) &\implies 55x_1/56 = 55/56 \implies x_1 = 1 \\
(6) &\implies 55x_2/56 = 55/56 \implies x_2 = 1 \\
(7) &\implies 55x_3/56 = 55/56 \implies x_3 = 1 \\
(8) &\implies 55x_4/56 = 55/56 \implies x_4 = 1
\end{aligned}
$$

Therefore, the solution to the given system of equations is $x_1 = 1,\ x_2 = 1,\ x_3 = 1,\ x_4 = 1$.

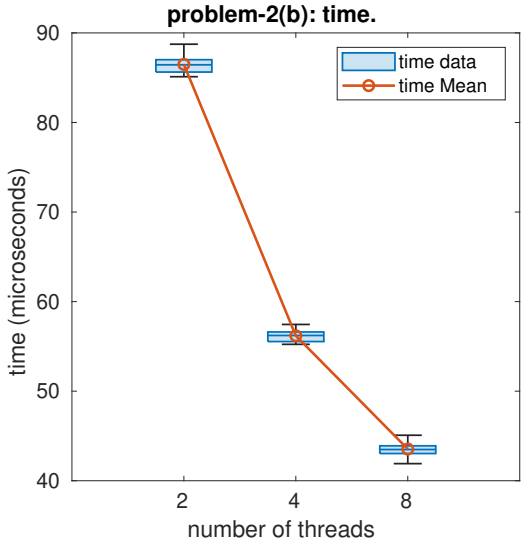
Problem 2:



(a) Problem 2a: analytical solution vs. (serial) LU decomposition solution



(b) Problem 2b: analytical solution vs. (parallel) recursive doubling solution

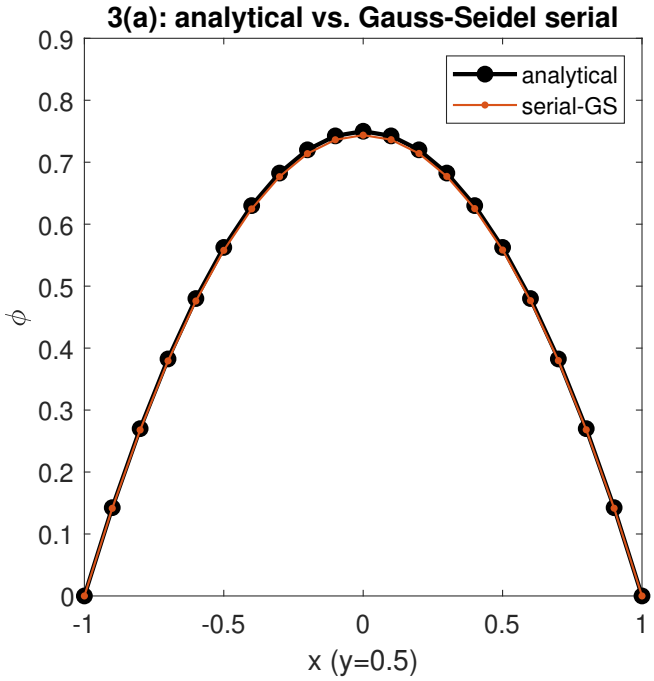


(c) Problem 2b: time taken by no. of threads - 2, 4, 8

The measured time fluctuates with each execution. To present the time taken accurately, the recursive doubling program is run 100000 times, and the time is measured using the `time` command in Linux. The distribution of 10 such measurements is presented for each thread. From plot (c), it is clear that increasing the number of threads decreases the computation time.

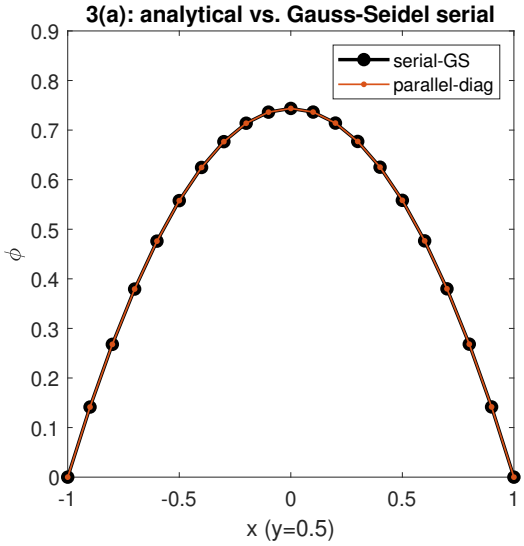
Problem 3:

(a) 191 iterations are required to bring the numerical solution to within 1% of the exact solutions using the serial Gauss-Seidel method.

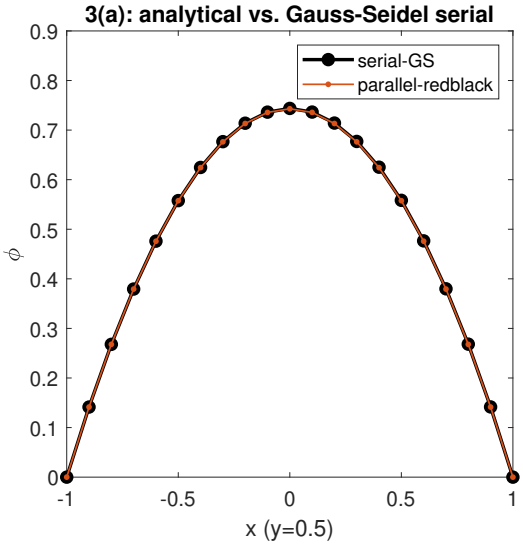


(a) Problem 3a: analytical solution vs. (serial) Gauss-seidel solution at y=0.5

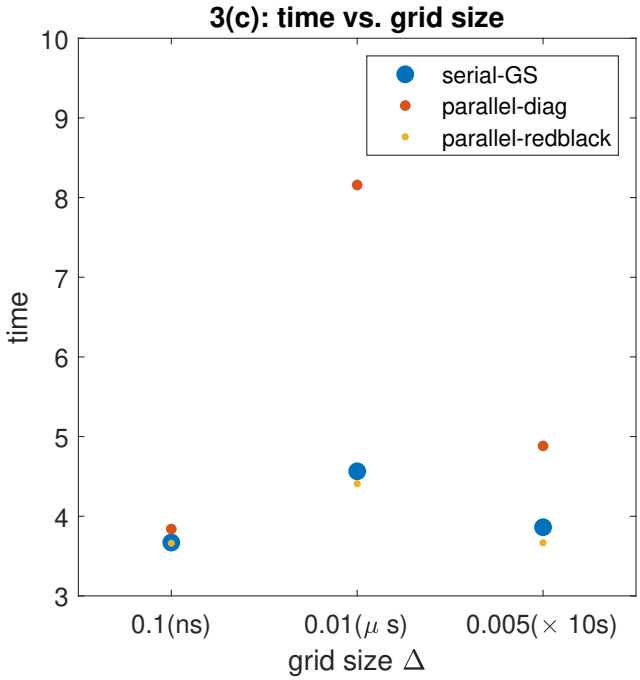
(c)



(a) Problem 3c: (serial) Gauss-seidel vs. (parallel) GS-diagonal



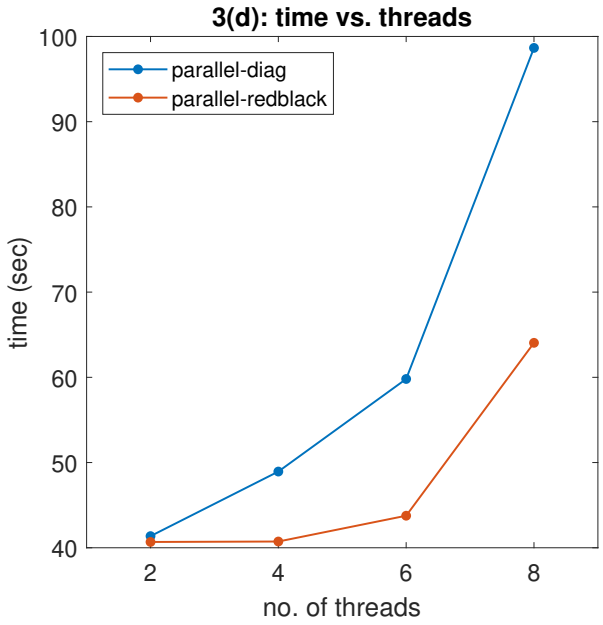
(b) Problem 3c: (serial) Gauss-seidel vs. (parallel) GS-redblack



(a) Problem 3c: time taken by solvers with respect to grid size

The performance improvement is only obtained for the red-black approach. Compared to serial Gauss-Seidel, the diagonal approach takes a lot more time and hence not useful.

(d)



(a) Problem 3d: time taken by solvers with respect to no. of threads

From the plot, it is clear that the red-black method is better than the diagonal approach.