An Improved Parallel AGE Method to Solve Incomplete Blow-up Problem through High Performance Computing System

Norma Alias[1], Md. Rajibul Islam[2], Nurul Ain Zhafarina Muhamad[3], Noriza Satam[3] and Norhafiza Hamzah[3]

[1]Ibnu Sina Institute, Faculty of Science, University Technology Malaysia, 81310 Johor
e-mail: norma@ibnusina.utm.my
[2]Faculty of Information Science and Technology, Multimedia University, Malaysia
e-mail: md.rajibul.islam05@mmu.edu.my
[3]Mathematic Department, Faculty of Science, University Technology Malaysia, 81310 Johor
e-mail: {nurul.ain.zhafarina, norizasatam, norhafizahamzah}@gmail.com

Abstract

Incomplete blow-up is a condition under the quasilinear heat equation. The Porous Medium Equation (PME) with power source are admitting incomplete blow-up. It is used as one of the filtration process in the industry. This filtration process has been used globally in the medical and laboratory applications. Previously, the standard numerical procedure was Gauss Seidel method to solve this problem. We propose a new variance of the Alternating Group Explicit Scheme (AGE) algorithms to solve incomplete blow-up problem through High performance computing (HPC). HPC systems include of multiple (usually mass-produced) processors linked together in a single system with commercially available interconnects. This is in contrast to mainframe computers, which are generally monolithic in nature. Four important terms that are, convergent rate by the number of iteration, execution time, computational complexity and stability are considered in this study to evaluate the performances of this approach.

Keywords: *High performance computing (HPC), blow-up problem, Alternating Group Explicit Scheme (AGE), Gauss Seidel method.*

1   Introduction

Temperature distribution is a physical model may be imagined in which heat is considered to be a fluid inside matter, free to flow from one position to another. The amount of fluid present is measured in some unit such as the calorie (cal) or BTU (British Thermal Unit). Evidence of its presence in matter is the temperature thereof; it is assumed that the more heat presents the higher the temperature, and that it flows from places of high temperature to places of low temperature. Temperature can be measured directly by a thermometer; the quantity of heat present is inferred indirectly.

Five types of blow-up patterns were illustrated for the 4th-order semilinear parabolic equation of reaction-diffusion type by Galaktionov [1] and Noriko Mizoguchi [2] presented multiple blow-ups to solve a semilinear heat equation problem. R. Natalini, C. Sinestrari and A. Tesei [3] presented an incomplete blowup of entropy solutions to first-order quasilinear hyperbolic balance laws. They specified a general procedure to continue solutions beyond the blowup time, which made use of monotonicity methods. The continuations thus obtained were possibly unbounded and satisfied suitable generalized entropy and Rankine-Hugoniot conditions. Then they proved the uniqueness of continuations satisfying such conditions as well. José M. Arrieta and Aníbal Rodríguez-Bernal [4] showed that blow-up occurred only on the boundary while they analyzed the existence of solutions that blow-up in finite time for a reaction-diffusion equation. Noriko Mizoguchi and Juan Luis Vazquez [5] demonstrated multiple blow-ups for semilinear heat equations at different places and different times and also solutions for a semilinear heat equations II described by Noriko Mizoguchi [6]. Nonlinear Volterra integral equations of the second kind with solutions that blow-up or quench had analyzed by Catherine A. Roberts [7].

1.1   Incomplete Blow-up

In general, quasilinear heat equation is a natural problem which is to find conditions for complete and incomplete blow-up in terms of the constitutive functions. In principle, the alternative will also depend on the initial data. It is clear that for flat initial data (u_0 constant) blow-up is always flat, hence complete.

Incomplete blow-up is an admitting of power source of Porous Medium Equation (PME). The incomplete blow-up has some properties to consider. But, in this study, only one of the properties will be discussed.

The nonnegative solutions *u=u(x, t)* of the quasilinear heat equation

$$u_t = \left(u^m\right)_{xx} + u^p, \tag{1}$$

in $S = \Re \times \Re_+,\ m > 0, p > 1,$

may blow up in finite time for some initial data $u(x,o) = u_0(x) \geq 0, \ u \neq 0$. For $1 < p \leq m + 2$, solutions blow up for arbitrarily small initial data u_0, while for $p > m + 2$ blow up always occurs for sufficiently large initial function.

Let $u(x, t)$ be the unique global proper (minimal) solution constructed by monotone increasing approximations, and $T = T(u_0)$ be its finite blow up time. If the continuation of the solution beyond blow-up is trivial, i.e., $u(x,t) \equiv \infty$ for $t>T$, we say that the blow-up is complete, otherwise, if $u(x,t) \equiv \infty$ for $t>T$, it is incomplete.

The blow-up set $B[u](t)$ is defined for every $t>T$ by the formula

$$B[u](t) = \{x \in \Re : \exists \{x_k\} \to x, \{t_k\} \to t^- \text{ with } u(x_k, t_k) \to \infty\}, \qquad (2)$$

and in the case of incomplete blow-up $B[u](t) \neq \Re$ at least for $i \approx T^+$. This corresponds to the idea of burnt zone in the theory of flame propagation, while the boundary of this set $\partial B[u]$ corresponds to the flame zone.
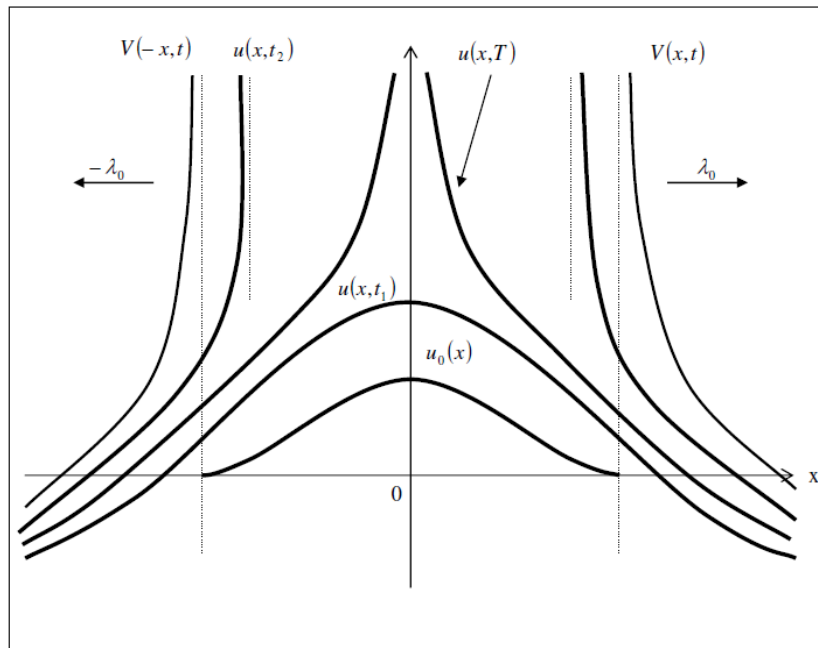

Fig. 1: The idea of incomplete blow-up is comparison: *u(x, t)* cannot overtake the singular Traveling Waves (TWs) *V(x, t)* for all t>0. Here $0 < t_1 < T < t_2$.

1.2   The application to industry

Incomplete blow-up is one of a special Porous Medium Equation (PME) with power source. In industrial world, the porous medium is used as one of the process such as in the filtration process.

As we can see, filtration is the process of passing a flow containing suspended solids through a porous medium, the fabric. It has been widely accepted as an effective, reliable and economic method for solid-liquid separation. Some critical applications include sterilization of pharmaceutical fluids; control of sub micron contaminants is deionized water for integrated circuit manufacture and purification of a variety of chemicals and solvents. Greater performance demands are being placed on filtration systems with particular reference to increase security, improved economy and enhanced removal efficiency. A good knowledge of the technique is, therefore, essential for the process industry personnel.

For a number of reasons, the medical and laboratory product industries cannot rely solely on the current forms of filtration and separations materials for a new application.

The incomplete blow up Heat Equation (HE) will be discretized by using Partial Differential Equation (PDE) in this study. This discretization must be done to see whether the function is satisfying or not the condition that have been stated.

The key purpose on this study is to implement the new variance of AGE method. The accuracy of the AGE method is comparable. This method also employs the fractional splitting strategy and the implicit form. The second endeavor is to run the AGE method with the HPC platform. Through the Linux Programming, the algorithms of the AGE method will be coded and run by the HPC.

The final target is to analyze the equation. This step is important because there will be an example that can be run to find the error residual. The error will be compared through two methods which are Gauss-Seidel method and AGE method. The best method which can give the lowest error is the best method.

The scope of this study is to develop the code of AGE algorithms and to determine the performance of the coding on the HPC. On the other hand, the equation of the incomplete blow-up which is one of the subtopic in HE will be solved by the AGE algorithms. For the comparison, AGE method will be compared by Gauss-Seidel method in numerical analysis to get the error residual.

This paper describes some previous research work relevant to the incomplete blow-up equation in this section. Also discusses about its application in the industry and introduces two dimensional heat equations. In section 2, the iterative methods i.e. Gauss Seidel and AGE methods are exposed that are used in this study, as a numerical approximation under consideration. Section 3 introduces the High Performance Computer (HPC) systems to solve the mathematical model. A brief clarification about parallel architecture and Parallel Virtual Machine (PVM)

which is used as a communication platform will be discussed in this section. The discretization of the incomplete blow-up and the performance analysis of Gauss Seidel and AGE programming will be discussed and the result will be compared in order to evaluate the performance in section 4. Finally, Section 5 will conclude the paper.

2.0   Numerical Method under Consideration

There are two methods that have taken under consideration for this study which are the Alternating Group Explicit Scheme (AGE) method and the Gauss-Seidel method in solving the equation.

2.1   Alternating Group Explicit Scheme (AGE) Method

Through the Alternating Direction Implicit (ADI), Alternating Group Explicit Scheme (AGE) methods with Peaceman-Rachford variation are created to be more extremely powerful, flexible and these offer users many advantages. The accuracy of these methods are comparable if not better than that of the GE class of problem as well as other existing schemes presented by Evans and Abdullah [8]. These methods employ the fractional splitting strategy and the implicit form is as follows,

$$u^{\left(k+\frac{1}{2}\right)} = \left(G_1 + rI\right)^{-1}\left[\left(rI - G_2\right)u^{(k)} + f\right],\tag{3}$$

$$u^{(k+1)} = \left(G_2 + rI\right)^{-1}\left[\left(rI - G_1\right)u^{\left(k+\frac{1}{2}\right)} + f\right].\tag{4}$$

We have

$$A = G_1 + G_2.\tag{5}$$

if we resume m to be odd then $\hat{G}$ could be written as,

$$\hat{G}\begin{bmatrix} r_2 & b \\ c & r_2 \end{bmatrix}_{2\times 2},\tag{6}$$

where $r_2 = r + \dfrac{a}{2}$. The alternating implicit nature of the (2×2) groups where the implicit and explicit values are given on the forward and backward levels for sweeps on the $\left(k+\dfrac{1}{2}\right)^{th}$ and $(k+1)^{th}$ levels, with $r_1 = r - \dfrac{a}{2}, r_2 = r + \dfrac{a}{2}$ end $\Delta = r_2^2 - bc$, k=1,2,…,n,

   i.   at the $\left(k+\dfrac{1}{2}\right)^{th}$ iterate

$$u_1^{\left(k+\frac{1}{2}\right)} = \frac{r_1 u_1^{(k)} - b u_2^{(k)} + f_1}{r_2}, \tag{7}$$

$$u_i^{\left(k+\frac{1}{2}\right)} = \frac{A u_{i-1}^{(k)} + B u_i^{(k)} + C u_{i+1}^{(k)} + D u_{i+2}^{(k)} + E_i}{\Delta}, \tag{8}$$

$$u_{i+1}^{\left(k+\frac{1}{2}\right)} = \frac{A u_{i-1}^{(k)} + B u_i^{(k)} + C u_{i+1}^{(k)} + D u_{i+2}^{(k)} + E_i}{\Delta}, \tag{9}$$

with $i=2,4,6,.....,m-1,$

$$\left.\begin{array}{l} A = -cr_2, B = r_1 r_2, C = -br_1 \\ E_i = r_2 f_i - b f_{i+1}, D = \begin{cases} 0, i = m-1 \\ b^2, i \neq m-1 \end{cases} \end{array}\right\} \tag{10}$$

and

$$\left.\begin{array}{l} A = -cr_2, B = r_1 r_2, C = -br_1 \\ E_i = r_2 f_i - b f_{i+1}, D = \begin{cases} 0, i = m-1 \\ -b^2, i \neq m-1 \end{cases} \end{array}\right\} \tag{11}$$

ii. at the $(k+1)^{th}$ iterate

$$u_i^{(k+1)} = \frac{P u_{i-1}^{\left(k+\frac{1}{2}\right)} + Q u_i^{\left(k+\frac{1}{2}\right)} + R u_{i+1}^{\left(k+\frac{1}{2}\right)} + S u_{i+2}^{\left(k+\frac{1}{2}\right)} + T_i}{\Delta}, \tag{12}$$

$$u_{i+1}^{(k+1)} = \frac{P u_{i-1}^{(k)} + Q u_i^{(k)} + R u_{i+1}^{(k)} + S u_{i+2}^{(k)} + T_i}{\Delta}, \tag{13}$$

$$u_m^{(k+1)} = \frac{-c u_{m-1}^{\left(k+\frac{1}{2}\right)} + r_1 u_m^{\left(k+\frac{1}{2}\right)} + f_m}{\Delta}, \tag{14}$$

With

$$\left.\begin{array}{l} Q = r_1 r_2, R = -br_1, S = b^2, \\ T_i = r_2 f_i - b f_{i+1}, P = \begin{cases} 0, i = 1 \\ c^2, i \neq 1 \end{cases} \end{array}\right\} \tag{15}$$

and

$$\left.\begin{array}{l} Q = -cr_2, R = Q = r_1 r_2, \\ S = -br_2, T_i = -cf_{i-} + r_2 f_{i+1}, \end{array}\right\} \tag{16}$$

All the equations are not dependent on every point *i-1*, *i* and *i+1* event for every time step. This is the advantages to create the parallel algorithm for AGE. Computational complexity of the new variance of AGE method i.e.

IADE_BRIAN and IADE_DOUGLAS for one dimensional problem is presented in table 1 below.

Table 1: computational complexity for sequential IADE_BRIAN and
IADE_DOUGLAS for one-dimensional problem

Methods	Multiplication	Addition
Constant of IADE_BRIAN	*18*	*7*
IADE_BRIAN	*10m + 5*	*8m + 5*
Constant of IADE_DOUGLAS	*21*	*11*
IADE_DOUGLAS	*12m + 7*	*8m + 5*
Gauss Seidel Red Black	*6m + 5*	*4m + 3*

2.2   Gauss-Seidel Method

The Gauss-Seidel method attempts to solve the equations in a non-linear system within each period by a series of iterations which do not involve linear approximations. In the descriptions below the superscript represents the value calculated in the "*n*" th iteration and the superscript *0* indicates the starting value for the variable.

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}}{a_{ii}}, \tag{17}$$

$$\forall i = 1,2,3,......,n.$$

The Gauss-Seidel Method is more efficient than the Jacobi Method in terms of the rate of convergence.

2.3   The Discretization

In previous sections, we have been overviewed the incomplete blow-up equation. In this subtopic, we will discuss the discretization of the incomplete blow-up equation. The incomplete blow-up is a two dimensional parabolic equation. From equation (1), we have

$$u_t = \frac{u_{i+1,j} - u_{i,j}}{\delta t}, \tag{18}$$

$$(u^m)_{xx} = \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\delta x^2}, \tag{19}$$

$$u^p = u_{i,j}. \tag{20}$$

The discretization is based on partial differential equation as shown below.

Substitute equation (18), (19), (20) into equation (1),

$$\frac{u_{i+1,j} - u_{i,j}}{\delta t} = \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\delta x^2} + u_{i,j}, \tag{21}$$

$$u_{i+1,j} - u_{i,j} = \frac{\delta t}{\delta x^2}\left(u_{i,j-1} - 2u_{i,j} + u_{i,j+1} + \delta x^2 u_{i,j}\right), \tag{22}$$

Hence,

$$u_{i+1,j} = \frac{\delta t}{\delta x^2} u_{i,j-1} + \left(1 + \delta t - \frac{2\delta t}{\delta x^2}\right) u_{i,j} + \frac{\delta t}{\delta x^2} u_{i,j+1}, \tag{23}$$

The equation (23) can be written in the matrix form:

$$\begin{bmatrix} u_{i+1,1} & u_{i+1,2} & \cdots & \cdots & u_{i+1,N-1} \end{bmatrix}$$
$$= \begin{bmatrix} b & c & & & \\ c & b & c & & \\ & \ddots & \ddots & \ddots & \\ & & c & b & c \\ & & & c & b \end{bmatrix} \begin{bmatrix} u_{i,1} & u_{i,2} & \cdots & \cdots & u_{i,N-1} \end{bmatrix}, \tag{24}$$

where $b = 1 + \delta t - 2\left(\dfrac{\delta t}{\delta x^2}\right)$ and $c = \dfrac{\delta t}{\delta x^2}$.

2.4 Gauss Seidel Programming

The classical Gauss Seidel has been discussed in the previous section. The incomplete blow-up equation has been discrete according to the classical Gauss Seidel. The one dimensional incomplete blow-up:

$$\frac{\delta t}{\delta x^2} = \frac{\delta u^m}{\delta x^2} + u^p, \tag{25}$$

and the discretization of the form,

$$\frac{\delta t}{\delta x^2} = \frac{\delta u^m}{\delta x^2} + u^p, \tag{26}$$

$$\frac{u_{i+1,j} - u_{i,j}}{\delta t} = \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\delta x^2} + u_{i,j}, \tag{27}$$

Giving the explicit formula,

$$u_{i+1,j} = \frac{\delta t}{\delta x^2} u_{i,j-1} + \left(1 + \delta t - \frac{2\delta t}{\delta x^2}\right) u_{i,j} + \frac{\delta t}{\delta x^2} u_{i,j+1}, \tag{28}$$

The Gauss Seidel programming has been done by using C Programming. The graph of the Gauss Seidel approximation can be seen in the Fig. 2.

From Fig. 2, we can see that the Gauss Seidel method gives the parabolic curve to the value of x_i. The value of $x_1 = x_9 = 0.415383, x_2 = x_8 = 0.676786$, $x_3 = x_7 = 0.880472$, $x_4 = x_6 = 0.999940$, and $x_5 = 1.040007$. From the programming that has been run, the number of iteration for Gauss Seidel method is 150 iterations with the error drop to 0.
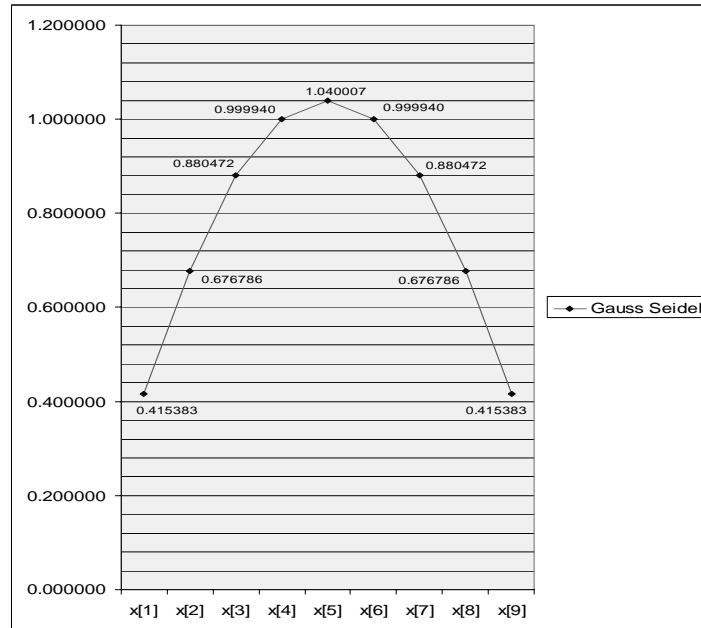

Fig. 2: Gauss Seidel Approximation Graph

2.5   AGE Method Programming

The algorithm for the AGE method has been discussed in the previous section. This method employs the fractional splitting strategy and the implicit form is as follows,

$$u^{\left(k+\frac{1}{2}\right)} = \left(G_1 + rI\right)^{-1}\left[\left(rI - G_2\right)u^{(k)} + f\right], \tag{29}$$

$$u^{(k+1)} = \left(G_2 + rI\right)^{-1}\left[\left(rI - G_1\right)u^{\left(k+\frac{1}{2}\right)} + f\right], \tag{30}$$

with k=1,2,…..,n and

$$A = G_1 + G_2. \tag{31}$$

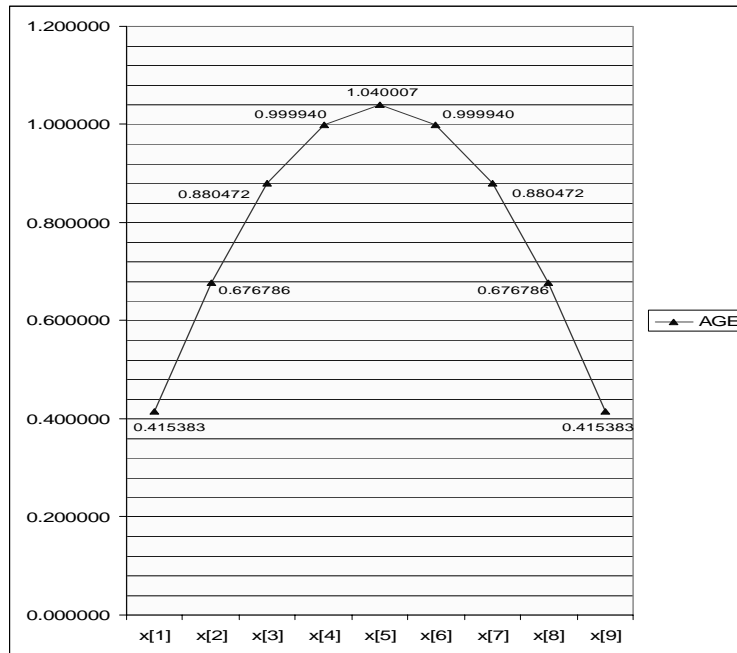By using AGE approximation, we got the result as in Fig. 3.

Fig. 3: AGE Approximation Graph

From Fig. 3, we can see that the AGE method gives the parabolic curve to the value of x_i. The value of $x_1 = x_9 = 0.415382$, $x_2=x_8=0.676786$, $x_3=x_7=0.880472$, $x_4=x_6=0.999939$, and $x_5=1.040007$. From the programming that has been run, the number of iteration for AGE method is 200 iterations with the error drop to 0.

3.0   Performance Analysis

We can compare both Gauss Seidel and AGE approximation method to get the four important components which are convergent rate by the number of iteration, execution time, computational complexity and stability.

As we can see from Fig. 4, Gauss Seidel and AGE method got the parabolic curve which means that the both approximations are valid.
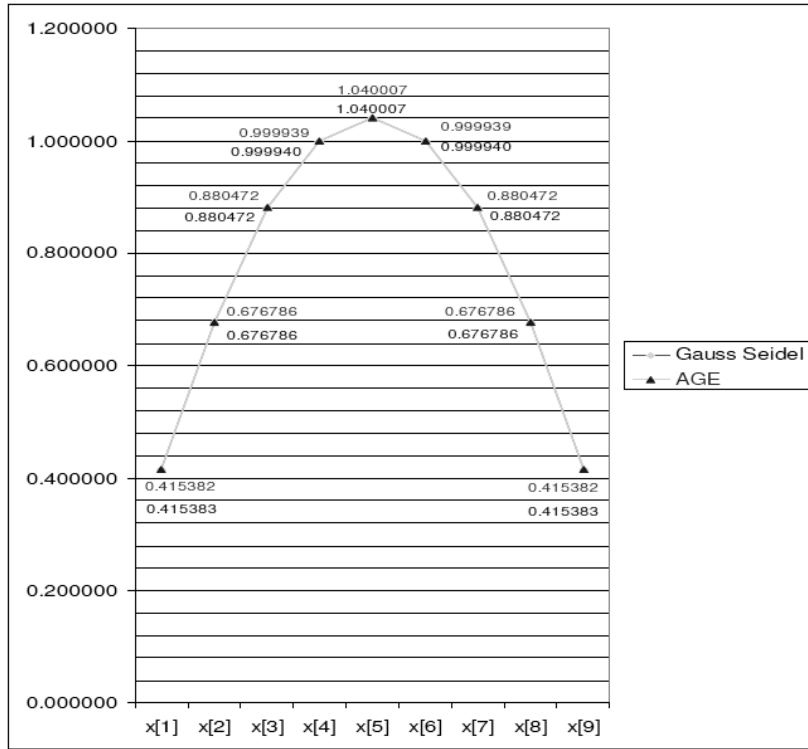
Fig. 4: Gauss Seidel and AGE Approximation Graph

From the Fig. 4, we cannot see any different between Gauss Seidel and AGE approximation. But by the value of each x_i, we can see the different at x_1 , x_4 , x_6 and x_9. The different is 0.000001 at all point x_1 , x_4 , x_6 and x_9. From Table 2, we can see that the number of iteration for Gauss Seidel method is 600 and AGE method is 250 with the lower error rate that Gauss Seidel method. The rate of convergence of AGE is better than Gauss Seidel method. The degree of accuracy of AGE method is higher than Gauss Seidel method. This proved by $t_{execution}$ (Execution time), iteration and |r|=absolute errors, generated by AGE method have a lowest value than Gauss Seidel method. Table 3 shows the parallel computational complexity and communication cost for AGE and GS methods and here it's shown that computational complexity and communication cost of AGE are lower than GS method. Hence, it is showed that AGE method is better than Gauss Seidel (GS) method. It is because the AGE method converges faster than GS method.

Table 2: Analysis of AGE_BRAIN and Gauss-Seidel methods

Parameters	AGE	GS
$t_{execution}$	48.743	156.4432
Iteration	250	600
rmse	$1.5921E^{-9}$	$1.5921E^{-9}$
$|r|$	$1.11022E^{-16}$	$1.1123E^{-16}$
rmse.maks	$1.9846E^{-7}$	$1.9846E^{-7}$
ave_rmse	$5.3374E^{-17}$	$5.3374E^{-17}$
m	720003	720003
Δx	$1.3889E^{-6}$	$1.3889E^{-6}$
Δt	$9.6450E^{-13}$	$9.6450E^{-13}$
t	$4.8225E^{-11}$	$4.8225E^{-11}$
λ	0.5	0.5
θ	1.0	1.0
Level	50	50
r	0.8	-
ε	$1.0E^{-15}$	$1.0E^{-15}$

rmse=root mean square error, |r|=absolute error, r.maks=maximum error and
ave_rmse=average of rmse

Table 3: Computational complexity and communication cost of AGE

Method	Computational complexity	communication cost
AGE	$\left(\dfrac{2250m}{p}+1257\right)T +\left(\dfrac{2250m}{p}+1268\right)D$	$3000t_{data}+1500(t_{start}+ t_{idle})$
GS	$\left(\dfrac{2400m}{p}+1800\right)T +\left(\dfrac{3600m}{p}+3000\right)D$	$7200t_{data}+3600(t_{start}+ t_{idle})$

D=multiplication, T= addition

4.0   Parallel Architectures

The classification of the parallel computer architecture can be divided into three categories: Flynn's taxonomy, Quinn classification, Cheong classification. The PVM system supports the message passing, shared memory, and hybrid paradigms, thus allowing applications to use the most appropriate computing model, for the entire application or for individual sub-algorithms. Processing elements such as scalar machines, distributed-and shared-memory multiprocessors, vector supercomputers and special purpose graphics engines, permitted the use of the best suited computing resource for each component of an application. This versatility is valuable for several large and complex applications including global

environmental modeling, fluid dynamics simulations, and weather prediction applications.

PVM system is implemented on a hardware base which is consists of different machine architectures, including single CPU systems, vector machines, and multiprocessors. This computing element is interconnected by one or more networks, which may themselves be different like one implementation of PVM operates on Ethernet, Internet and a fiber optic network [9].

C, C++ and FORTRAN are all languages that can be used to write PVM codes. This project is done by using C languages by UNIX as an operating function. To execute an application, a user typically starts one copy on one task from a machine within the host pool.

Task-to-task communication in PVM is done with message passing. Message passing is a set of tasks that use their own local memory during computation. Multiple tasks can reside on the same physical machine as well as across an arbitrary number of machines. Tasks exchange data through communications by sending and receiving messages. Data transfer usually requires cooperative operations to be performed by each process. For example, a send operation must have a matching receive operation.

5.0 Performance Analysis of PVM

There are a master task and a number of worker tasks in the PVM implementation of the modeling codes. Master task is responsible to divide the model domain into sub domains and distribute them to worker tasks. Then, the workers tasks perform time marching and communicate after each time step. Time execution, speedup, efficiency, effectiveness and temporal performance will be analyzed by looking at the performance of the parallel algorithm.

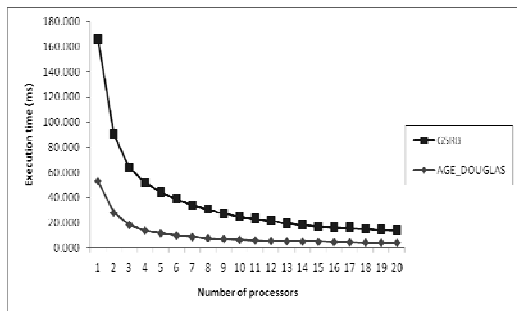*Parallel Analyses of the basic PDE*

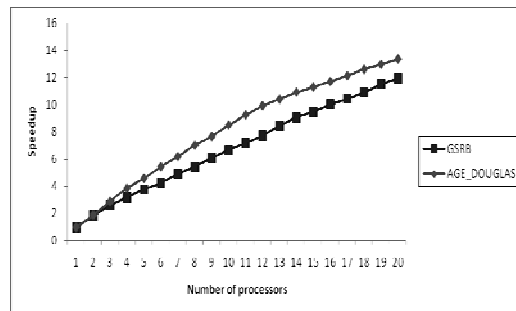Fig. 5: Execution time vs. number of processors   Fig. 6: Speedup vs. number of processors
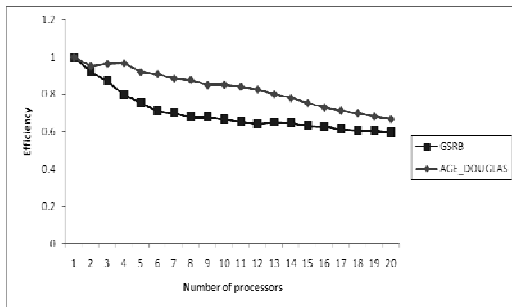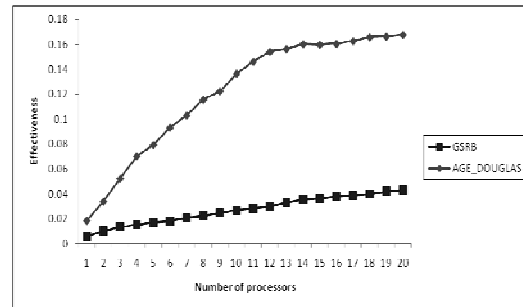
Fig. 7: Efficiency vs. number of processors

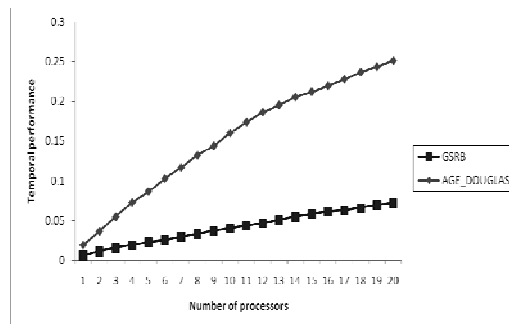Fig. 8: Effectiveness vs. number of processors

Fig. 9: Temporal performance vs. number of processors

From Fig. 5,6,7,8 and 9, we can see that AGE method is providing better results than Gauss Seidel method on parallel environment in terms of time of execution, speedup, efficiency, effectiveness and temporal performance depending on the number of processors. One dimensional PDE was applied for this analysis.

Parallel Analyses to solve incomplete blow-up

The Time Execution

The time execution has been determine when running the parallel algorithm. The result of the time execution is as below.

Table 4 and Fig. 10 shows the time execution of one dimensional parabolic equation model implemented to the parallel computing. The time execution has been determined by using two different method which are AGE and Gauss Seidel (GS) method. The size of the matrices; m=70000 has been used to see the time execution. The time execution of AGE method is lower than GS method for all number of processors. It means that AGE method is the better than GS method while running the incomplete blow-up parallel computing system.

Table 4: The Time Execution of Parallel Computing

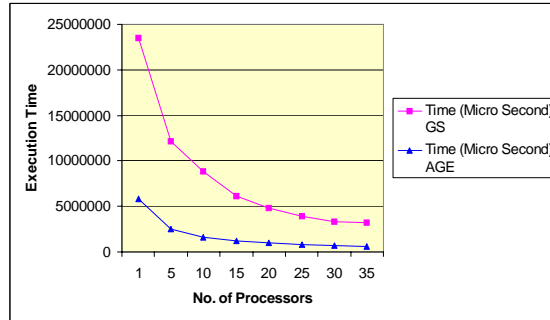| Number of Processors	Time (Micro Second)	
	GS	AGE
1	23483497	5791478
5	12186957	2499727
10	8799795	1653622
15	6169910	1221648
20	4785296	980992
25	3926748	809072
30	3346007	706581
35	3222810	644276

Fig. 10: Time Execution vs. Number of Processors

The Speedup

The Amdahl's law state that the speed of a program is the time to execute the program while speedup is defined as the time it takes a program to execute in serial (with one processor) divided by the time it takes to execute in parallel (with many processors).

Table 5: The Speedup of Parallel Computing

| No. of Processors	Time (Micro Second)		Speedup	
	GS	AGE	GS	AGE
1	23483497	5791478	1	1
5	12186957	2499727	1.926936888	2.316844199
10	8799795	1653622	2.668541372	3.502298591
15	6169910	1221648	3.806132829	4.740709271
20	4785296	980992	4.907428297	5.903695443
25	3926748	809072	5.98039319	7.158173809
30	3346007	706481	7.018364576	8.19764155
35	3222810	644276	7.286652642	8.989125778
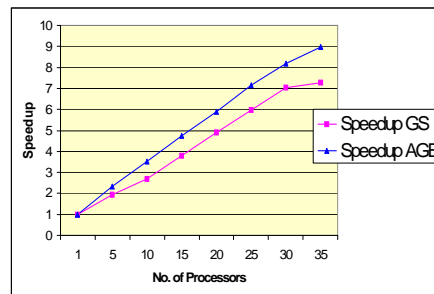
Fig. 11: Speedup vs. Number of Processors

Table 5 and Fig. 11 show that the speedup of parallel computing system while running the AGE and GS method. From Fig. 6, the speedup for both AGE and GS methods are increased. According to the Amdahl's law, the speedup increases with the number of processors increase up to the certain level. For this problem, the speedup of the level from 15 to 35 processors increases slower than the speedup of lower number of processors. However, the parallel computing has been used to show that AGE method has higher than GS for the number of processors.

The Efficiency

The efficiency of a parallel algorithm is a measure of processor utilization. Efficiency is the speedup divided by the number of processors that has been used.

Table 6: The Efficiency of Parallel Computing

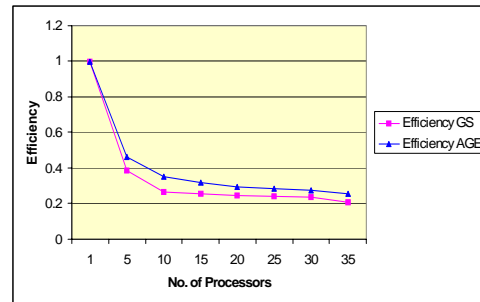| No. of Processors	Time (Micro Second)		Efficient	
	GS	AGE	GS	AGE
1	23483497	5791478	1	1
5	12186957	2499727	0.385387	0.463369
10	8799795	1653622	0.266864	0.350230
15	6169910	1221648	0.253742	0.316047
20	4785296	980992	0.245371	0.295185
25	3926748	809072	0.239216	0.286327
30	3346007	706481	0.233945	0.273255
35	3222810	644276	0.208190	0.256832

Fig. 12: Efficiency vs. Number of Processors

Table 6 shows that the efficiency decreases with the increasing of the number of processors, *p*. Poor load balance when imbalance workload distributed among the different processors is the factor that the decreasing of efficiency happened. It is also contributed by idle time, time startup and waiting for all processors to complete the computations. However, from the Fig. 12, the AGE method still more efficient method than GS method.

The Effectiveness

The effectiveness of the method by using parallel algorithm has been determined by calculate the speedup and efficiency.

Table 7: The Effectiveness of Parallel Computing

| No. of Processors	Time (Micro Second)		Effectiveness	
	GS	AGE	GS	AGE
1	23483497	5791478	$4.258310E^{-08}$	$1.726675E^{-07}$
5	12186957	2499727	$3.162294E^{-08}$	$1.853678E^{-07}$
10	8799795	1653622	$3.032618E^{-08}$	$2.117956E^{-07}$
15	6169910	1221648	$4.112575E^{-08}$	$2.587057E^{-07}$
20	4785296	980992	$5.127612E^{-08}$	$3.009044E^{-07}$
25	3926748	809072	$6.091955E^{-08}$	$3.538955E^{-07}$
30	3346007	706481	$6.991781E^{-08}$	$3.867828E^{-07}$
35	3222810	644276	$6.459893E^{-08}$	$3.986369E^{-07}$
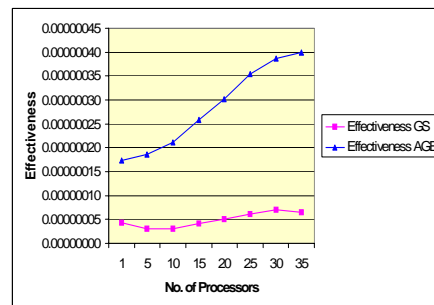
Fig. 13: Effectiveness vs. Number of Processors

Table 7 and Fig. 13 show the effectiveness for both AGE and GS method by using parallel computing system. The formula of the effectiveness depending on the speedup; when speedup increases, the effectiveness is also increase. From the graph we noticed that the AGE method is more effective than GS method.

The Temporal Performance

Temporal performance is a parameter to measure the performance of a parallel algorithm which is

Temporal=1/Time (p)
Where *Time P* = time execution using *p* processor.

Table 8: The Temporal Performance of Parallel Computing

| No. of Processors	Time (Micro Second)		Temporal Performance	
	GS	AGE	GS	AGE
1	23483497	5791478	$4.258310E^{-08}$	$1.889831E^{-07}$
5	12186957	2499727	$8.205494E^{-08}$	$3.704078E^{-07}$
10	8799795	1653622	$1.136390E^{-07}$	$6.047331E^{-07}$
15	6169910	1221648	$1.620769E^{-07}$	$7.863811E^{-07}$
20	4785296	980992	$2.089735E^{-07}$	$1.019376E^{-06}$
25	3926748	809072	$2.546637E^{-07}$	$1.235984E^{-06}$
30	3346007	706481	$2.988637E^{-07}$	$1.456705E^{-06}$
35	3222810	644276	$3.102882E^{-07}$	$1.552130E^{-06}$
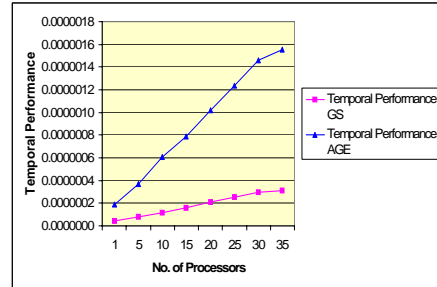
Fig. 14: Temporal Performance vs. Number of Processors

The temporal performance of parallel computing of AGE and GS method can be determined from Fig. 14. The Fig. shows increasing temporal performance for both methods. As showed above, the AGE method has higher temporal performance than GS method.

5.1   Granularity

Migration from sequential to parallel promises the existence of communication between working processors. Thus, the execution time measured for parallel implementation will be totally different with sequential. Parallel execution time will consider the time consumed for computation as well as time consumed for communication process. Communication cost involves time spent during sending and receiving messages. As computation and communication can be separately measured, granularity can be taken into account parallel performance evaluation. In [10], Kwiatkowski defines granularity as

$$G = \frac{T_{comp}}{T_{comm}}$$

Where T_{comp} and T_{comm} each represents computation and communication time. By measuring granularity, the ratio between computations to communication time can be explicitly shown. High rate of granularity shows that computation time holds the higher percentage out of overall execution time.

Table 9x: Granularity Rate, Percentage of Execution, Communication, and
Computation Time for Solution Using GS method

p	Execution time	Computation time	Granularity	Communication time
		Gauss Seidel		
5	44.10	30.99	2.36	13.11
%		70.27		29.73
10	24.88	16.04	1.82	8.831
%		64.47		35.49
15	17.19	10.33	1.51	6.856
		60.09		39.88
20	13.90	7.822	1.29	6.078
%		56.26		43.72

Table 9y: Granularity Rate, Percentage of Execution, Communication, and
Computation Time for Solution Using AGE method

p	Execution time	Computation time	Granularity	Communication time
		AGE		
5	11.00	9.60	6.9	1.40
%		87,27		12.72
10	5.85	4.72	4.2	1.12
%		80.68		19.14
15	4.35	3.18	2.7	1.16
		73.10		36.47
20	3.62	2.39	1.9	1.23
%		66.02		51.46

Table 9x and 9y shows the granularity rate for red black Gauss Seidel and AGE
method which is the ratio between computation and communication time. The
ratio is decreasing upon increasing number of processor as more processor
causing more communication session during parallel execution. The comparison
of granularity between GS and AGE method is as depicted in Fig. 15. It can be
conclude that AGE is an alternative method that promise better result in
performance evaluation compared to the use of GS classical iterative method. This
is due to the ability of AGE in donating high granularity rate compared to GS
method. The ability of maintaining the ratio between computation and
communication time will be more advantageous as more processors added to the
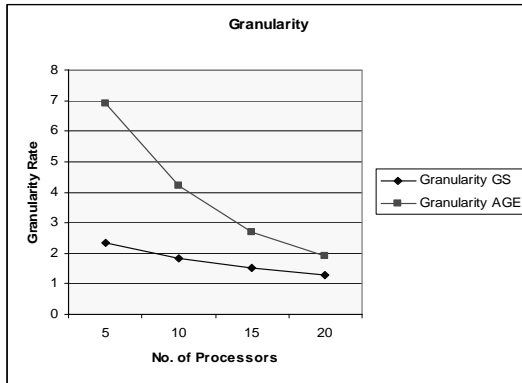parallel cluster.
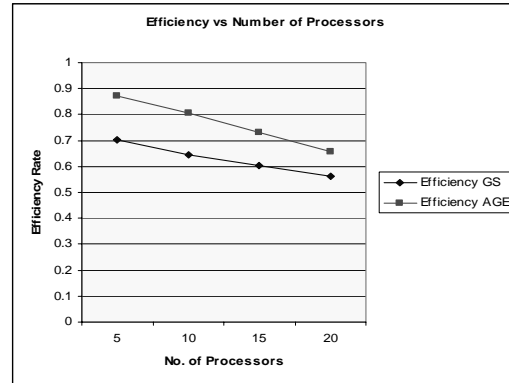
Fig. 15: Granularity vs. Number of Processors   Fig. 16: Efficiency vs. Number of Processors

This evaluation of granularity will be use in measuring parallel efficiency as stated by J. Kwiatkowski [10] where

$$E = \frac{1}{1 + \dfrac{1}{G}} = \frac{G}{G+1}$$

This formula had been used to calculate the rate of efficiency and the result is as shown in Fig. 16. It shows the comparison of efficiency rate for problem solution using GS and AGE method. AGE donates better efficiency rate compared to GS and this shows the benefit of AGE method's usage in computation.

6.0   Conclusions and Open Problems

Porous Medium Equation (PME) has been adopted in one of the important process in the industry that is the filtration process. In recent years, this filtration process is very important to the industry especially in medical and laboratory application. In 1950, Zel'dovich and coworkers developed the heat radiation in plasma by using the PME [11, 12]. Other applications that have been proposed is in mathematical biology, spread of viscous fluids, boundary layer theory, and other fields.

Though, it is still an open problem to discover conditions for complete and incomplete blow-up in terms of the constitutive functions by quasilinear heat equation, in this study, one of the properties (i.e., incomplete blow-up) under PME has been chosen to be solved by using the AGE method. The properties have been stated in the book written by Victor A. Galaktionov [13]. This parabolic equation described the first properties in incomplete blow-up that may blow-up in finite time $u(x,0) = u_0 \geq 0, u_0 \neq 0$.

The mathematical problem in this study is solved by using the HPC system with PVM. The performance analysis of PVM has been done and the five terms which are time execution, speedup, efficiency, effectiveness and temporal performance has been determined. Both AGE and Gauss Seidel method has been compared when determined the graph of each terms. The HPC has been used to run a large scale problem.

At the beginning the parabolic equation incomplete blow-up has been solved by the Gauss Seidel and a new variance of the AGE method. Both equations have been run using C programming to see whether Gauss Seidel or AGE method is the best algorithm to solve the equation by comparing their convergence rate. From that analysis, it's proved that AGE is the best algorithm to solve the parabolic equation incomplete blow-up.

REFFERENCES

[1] Galaktionov, V.A. "Five types of blow-up in a semilinear Fourth-order reaction-diffusion equation: An analytic-numerical approach", Nonlinearity, 22(7), (2009), pp. 1695-1741.

[2] Noriko Mizoguchi, "Multiple blowup of solutions for a semilinear heat equation", Mathematische Annalen, 331, (2005), pp. 461–473.

[3] Natalini, R., Sinestrari, C. and Tesei, A. "Incomplete Blowup of Solutions of Quasilinear Hyperbolic Balance Laws", Arch. Rational Mech, Anal. (Springer-Verlag), 135, (1996), pp 259-296.

[4] José M. Arrieta and Aníbal Rodríguez-Bernal, "Localization on the Boundary of Blow-up for Reaction–Diffusion Equations with Nonlinear Boundary Conditions", Communications in Partial Differential Equations, 29(7&8), (2004), pp 1127–1148.

[5] Noriko Mizoguchi and Juan Luis Vazquez, "Multiple blow-ups for semilinear heat equations at different places and different times", Indiana University mathematics journal, 56(6), (2007), pp 2859-2886.

[6] Noriko Mizoguchi, "Multiple blowup of solutions for a semilinear heat equation II", Journal of Differential Equations, 231, (2006), pp 182–194.

[7] Catherine A. Roberts, "Recent results on blow-up and quenching for nonlinear Volterra equations", Journal of Computational and Applied Mathematics, 205, (2007), pp 736 – 743.

[8] Evans, D. J. and Abdullah, A. R. B. "Group Explicit Method for Parabolic Equations [J]", Inter. J. Comput. Math., 14, (1983), pp 73-105.

[9] Sunderam, V. S., and Geist, G. A. "The PVM System: Supercomputer Level Concurrent Computation on a Network of IBM RS/6000 Powerstations", Proceedings of the Large Scale Analysis and Modeling Conference, Park City, UT, (1991), pp 258-261.

[10] Jan Kwiatkowski, "Evaluation of Parallel Programs by Measurement of Its Granularity", In Parallel Processing and Applied Mathematics, R. Wyrzykowski et al., Eds., (2002), pp 145–153, Springer-Verlag Berlin Heidelberg,.

[11] Ko, Y. "Regularity of the interface for the porous medium equation", Electron. J. Differential Equations, 68, (2000), pp 1-12.

[12] Ya. B. Zel'dovich, and Kompaneets, A.S. "Towards a theory of heat conduction with thermal conductivity depending on the temperature", in Collection of papers dedicated to 70th Anniversary of A. F. Ioffe, Izd. Akad. Nauk SSSR, Moscow, (1950), pp 61-72.

[13] Victor A. Galaktionov, "Geometric Sturmian Theory of Nonlinear Parabolic Equation and Applications", Chapman & Hall/CRC. University of Bath, England, (2004), pp 125-126.