Variable Relaxation Solve for Nonlinear Thermal Conduction*

Jin Chen

Princeton Plasma Physics Laboratory, Princeton, NJ, USA
jchen@pppl.gov

Abstract. Efficient and robust nonlinear solvers, based on Variable Relaxation, is developed to solve nonlinear anisotropic thermal conduction arising from fusion plasma simulations. By adding first and/or second order time derivatives to the system, this type of methods advances corresponding time-dependent nonlinear systems to steady state, which is the solution to be sought. In this process, only the stiffness matrix itself is involved so that the numerical complexity and errors can be greatly reduced. In fact, this work is an extension of implementing efficient linear solvers for fusion simulation on Cray X1E.

Two schemes are derived in this work, first and second order Variable Relaxations. Four factors are observed to be critical for efficiency and preservation of solution's symmetric structure arising from periodic boundary condition: mesh scales, initialization, variable time step, and nonlinear stiffness matrix computation. First finer mesh scale should be taken in strong transport direction; Next the system is carefully initialized by the solution with linear conductivity; Third, time step and relaxation factor are vertex-based varied and optimized at each time step; Finally, the nonlinear stiffness matrix is updated by just scaling corresponding linear one with the vector generated from nonlinear thermal conductivity.

1 Introduction

In plasma physics modeling[1], the steady state of nonlinear anisotropic thermal conduction can be modeled by the following nonlinear elliptic equation

$$\frac{\partial}{\partial x}(\kappa_x \frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(\kappa_y \frac{\partial T}{\partial y}) = s \tag{1}$$

on a 2D rectangular domain ABCD: $[0, L_x] \times [0, L_y]$ with four vertexes at A$(0, 0)$, B$(0, L_x)$, C(L_x, L_y), and D$(0, L_y)$. $L_x < L_y$. The coordinate is given in Cartesian (x, y) system. The magnetic field is directed in the y direction, and accordingly we can set $\kappa_x = 1$ and κ_y as an nonlinear function of the temperature T, parallel to magnetic field line. Therefore we can omit κ_x and denote κ_y by $\kappa_\parallel$ to make its meaning more clear. The periodic boundary condition is set on edges AD and

* This work is supported by DOE contract DE-AC02-76CH03073.

Y. Shi et al. (Eds.): ICCS 2007, Part I, LNCS 4487, pp. 30–37, 2007.

BC, and Dirichlet boundary conditions are set on edges AB and CD. This setup allows us to separate the effects of grid misalignment from the boundary effects. The upper boundary, CD, represent the material surface where the temperature is low, and the boundary condition there is $T_{CD} = 1$. At the lower boundary, AB, the inflow boundary condition is $T_{AB}(x) = 10 + 40e^{(-|x-L_x/2|)}$.

Finite element discretization[2] generates the following nonlinear system

$$(S_{xx} + S_{yy}(T))T = Ms. \tag{2}$$

M is the mass matrix. S_{xx} and $S_{yy}(T)$ are the stiffness matrices contributed by operator $\frac{\partial^2 T}{\partial x^2}$ and $\frac{\partial}{\partial y}(\kappa_\parallel \frac{\partial T}{\partial y})$, respectively. T is the temperature profile to be solved. When $\kappa_\parallel$ is linear, $S_{yy}(T)$ reduced to $\kappa_\parallel S_{yy}$. Newton-Krylov method can be used to solve system (2). But usually it is quite expensive to update Jacobian at each iteration. Although the Jacobian-free variation[3][4] is more efficient, information of the Jacobian is still needed to form the preconditioner and preconditioning is expensive.

In this work we present an alternative way, Variable Relaxation[5], to solve the nonlinear system (1). This is a class of iterative methods which solve the elliptic equations by adding first and/or second order time derivative terms to eq.(1) to convert it to nonlinear parabolic or hyperbolic equation and then marching the system to steady state. In this marching process, only the nonlinear stiffness matrix $S_{yy}(T)$ itself is involved and needs to be updated regularly.

We have been using this type of idea on Cray X1E to design efficient linear elliptic solvers for M3D code[6]. Although It takes longer to converge, each iteration is much cheaper than other iterative solvers[7] so that it still wins on vector architecture machines.

The nonlinear iteration can be completed in two steps:

Step 1: solve eq.(1) with linear conductivity $10^0 \leq \kappa_\parallel \leq 10^9$.
Step 2: solve eq.(1) with nonlinear conductivity $\kappa_\parallel = T^{5/2}$.

The solution from "**Step 1**" is used as an initial guess for "**Step 2**". Experiments will show that this is a very powerful strategy to accelerate convergence. We will also demonstrate how to choose artificial time step from CFL condition and relaxation factor from dispersion relation to achieve optimization. An efficient way to generate the stiffness matrix is also to be discussed in order to preserve the symmetry structure of the solution as a result of periodic boundary condition.

2 First Order Relaxation and Numerical Schemes

The so called first order relaxation is obtained by adding a first order time derivative term to eq. (1)

$$\frac{\partial u}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial}{\partial y}(\kappa_\parallel \frac{\partial T}{\partial y}). \tag{3}$$

Discretizing it in temporal direction by finite difference and spatial directions as in system (2), we have

$$(\frac{1}{\delta t}M - \theta S_{non})T^{k+1} = [\frac{1}{\delta t}M + (1-\theta)S_{non})]T^k - Ms. \tag{4}$$

$0 \leq \theta \leq 1$. When $\theta = 0$, the system is fully explicit; when $\theta = 1$, the system is fully implicit; when $\theta = \frac{1}{2}$, the system is stable and has smallest truncation error as well. $S_{non} = S_{xx} + S_{yy}(T)$. δt is the artificial time step which should be chosen to be small enough to make the scheme stable and big enough to allow the system approach steady state quickly. According to CFL condition, δt is related to mesh scales δx in x direction and δy in y direction by

$$\delta t = \frac{1}{2}\frac{1}{\frac{1}{\delta x^2} + \kappa_\| \frac{1}{\delta y^2}} = \frac{\delta x \delta y}{4}\frac{2}{\frac{\delta y}{\delta x} + \kappa_\| \frac{\delta x}{\delta y}} \equiv \frac{\delta x \delta y}{4}\bar{\delta t}. \tag{5}$$

Obviously, when $\kappa_\| = 1$, $\bar{\delta t}$ is symmetric in $(\delta x, \delta y)$ and gets maximized at $\delta x = \delta y$. More can be derived if we different $\bar{\delta t}$ with respect to δx and δy

$$\frac{\partial \bar{\delta t}}{\partial \delta x} = -2\frac{-\frac{\delta y}{\delta x^2} + \kappa_\| \frac{1}{\delta y}}{(\frac{\delta y}{\delta x} + \kappa_\| \frac{\delta x}{\delta y})^2} = 2\frac{1}{\delta y}\frac{\frac{\delta y^2}{\delta x^2} - \kappa_\|}{(\frac{\delta y}{\delta x} + \kappa_\| \frac{\delta x}{\delta y})^2},$$

$$\frac{\partial \bar{\delta t}}{\partial \delta y} = -2\frac{\frac{1}{\delta x} - \kappa_\| \frac{\delta x}{\delta y^2}}{(\frac{\delta y}{\delta x} + \kappa_\| \frac{\delta x}{\delta y})^2} = 2\frac{\delta x}{\delta y^2}\frac{\kappa_\| - \frac{\delta y^2}{\delta x^2}}{(\frac{\delta y}{\delta x} + \kappa_\| \frac{\delta x}{\delta y})^2}.$$

When $\kappa_\| > 1$, most likely we will have $\frac{\partial \bar{\delta t}}{\partial \delta x} < 0$ and $\frac{\partial \bar{\delta t}}{\partial \delta y} > 0$. This suggests that δx should be taken as large as possible, while δy as small as possible.

The convergence of scheme (4) can be analyzed in the following way. Given the form of transient solution of eq.(3) as $\tilde{u} = e^{-\gamma t} \sin \frac{m\pi x}{L_x} \sin \frac{n\pi y}{L_y}$, the operator $\frac{\partial^2}{\partial x^2} + \frac{\partial}{\partial y}(\kappa_\| \frac{\partial}{\partial y})$ has eigenvalues $\lambda_{mn} = \pi^2(\frac{m^2}{L_x^2} + \kappa_\| \frac{n^2}{L_y^2})$. m and n are the mode numbers in x and y directions, respectively. Then the decaying rate is $-\lambda_{11}$ and the corresponding decaying time can be found by

$$t = \frac{1}{\lambda_{11}} = \frac{1}{\pi^2}\frac{1}{\frac{1}{L_x^2} + \kappa_\| \frac{1}{L_y^2}}.$$

The number of iterations needed for convergence can be predicted by

$$N_{its} \equiv \frac{t}{\delta t} = \frac{2}{\pi^2}\frac{\frac{1}{\delta x^2} + \kappa_\| \frac{1}{\delta y^2}}{\frac{1}{L_x^2} + \kappa_\| \frac{1}{L_y^2}} = \frac{2}{\pi^2}\frac{\frac{N_x^2}{L_x^2} + \kappa_\| \frac{N_y^2}{L_y^2}}{\frac{1}{L_x^2} + \kappa_\| \frac{1}{L_y^2}}.$$

When $\kappa_\| \to \infty$

$$N_{its} \to \frac{2}{\pi^2}N_y^2 \approx \frac{1}{5}\frac{N_y}{N_x}(N_x N_y) \equiv c(N_x N_y).$$

(N_xN_y) is the number of unknowns. After some experiments, we found the optimized coefficient should be $c = 0.64$ for the problem we are studying. Also from the following expression we found the number of iterations increases as $\kappa_\|$ gets larger

$$\frac{dN_{its}}{d\kappa_\|} = \frac{2}{\pi^2}\frac{(N_y^2 - N_x^2)}{(\frac{L_y}{L_x} + \kappa_\| \frac{L_x}{L_y})^2} > 0$$

as long as $\delta y \leq \delta x$.

3 Second Order Relaxation and Numerical Schemes

Besides the addition of the first order derivative term in eq. (3), the second order relaxation is obtained by adding a relaxation factor, τ, and a second order time derivative term to eq. (1)

$$\frac{\partial^2 u}{\partial t^2} + \frac{2}{\tau}\frac{\partial u}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial}{\partial y}(\kappa_\| \frac{\partial T}{\partial y}). \tag{6}$$

Again it can be discretized and rearranged as

$$\begin{gathered}[(1 + \tfrac{\delta t}{\tau})M - \theta S_{non}]T^{k+1} = \\ -(1 - \tfrac{\delta t}{\tau})MT^{k-1} + [2M + \delta t^2(1-\theta)S_{non}]T^k - \delta t^2 Ms.\end{gathered} \tag{7}$$

The CFL condition can be expressed as $\delta t^2(\frac{1}{\delta_x^2} + \kappa_\| \frac{1}{\delta_y^2}) \leq 1$. Therefore,

$$\delta t \leq \frac{1}{\sqrt{\frac{1}{\delta x^2} + \kappa_\| \frac{1}{\delta_y^2}}} = \frac{\sqrt{\delta x \delta y}}{\sqrt{\frac{\delta y}{\delta x} + \kappa_\| \frac{\delta x}{\delta y}}} = \frac{\sqrt{\delta x \delta y}}{\sqrt{2}} \frac{\sqrt{2}}{\sqrt{\frac{\delta y}{\delta x} + \kappa_\| \frac{\delta x}{\delta y}}}. \tag{8}$$

The relaxation factor can be found again by looking for the transient solution of eq.(6). The decay rates satisfy $\gamma^2 - \frac{2}{\tau}\gamma + \lambda^{mn} = 0$, or $\gamma = \frac{1}{\tau} \pm (\frac{1}{\tau^2} - \lambda^{mn})^{1/2}$. For optimal damping, we choose $\tau^2 = \frac{1}{\lambda^{11}} = 1/[(\frac{L_y^2}{L_x^2} + \kappa_\|)\frac{\pi^2}{L_y^2}]$, i.e.,

$$\tau = \frac{1}{\pi\sqrt{\frac{1}{L_x^2} + \kappa_\| \frac{1}{L_y^2}}} \tau = \frac{\sqrt{L_xL_y}}{\sqrt{2}\pi} \frac{\sqrt{2}}{\sqrt{\frac{L_y}{L_x} + \kappa_\| \frac{L_x}{L_y}}} \tag{9}$$

and the number of iterations for convergence can be predicted by

$$\begin{aligned} N_{its} &\equiv \tfrac{\tau}{\delta t} \\ &= \frac{1}{\pi}\frac{\sqrt{\frac{1}{\delta x^2} + \kappa_\| \frac{1}{\delta_y^2}}}{\sqrt{\frac{1}{L_x^2} + \kappa_\| \frac{1}{L_y^2}}} = \frac{1}{\pi}\frac{\sqrt{\frac{N_x^2}{L_x^2} + \kappa_\| \frac{N_y^2}{L_y^2}}}{\sqrt{\frac{1}{L_x^2} + \kappa_\| \frac{1}{L_y^2}}} = \frac{1}{\pi}\frac{\sqrt{\frac{25}{9}N_x^2 + \kappa_\| N_y^2}}{\sqrt{\frac{25}{9} + \kappa_\|}} = \frac{1}{\pi}\frac{\sqrt{\frac{25}{9}\frac{N_x}{N_y} + \kappa_\| \frac{N_y}{N_x}}}{\sqrt{\frac{25}{9} + \kappa_\|}}\sqrt{N_xN_y}\end{aligned}$$

When $\kappa_\| \to \infty$

$$N_{its} \to (\frac{1}{\pi}\sqrt{\frac{N_y}{N_x}})\sqrt{N_xN_y} \equiv c\sqrt{N_xN_y}.$$

Experiments show that the optimal coefficient would be $c = 0.6$. The number of iteration increases as the conductivity $\kappa_\parallel$ increases. This can be understood from the following expression.

$$\frac{dN_{its}}{d\kappa_\parallel} = \frac{1}{\pi}\frac{\sqrt{3N_y^2 + \kappa_\parallel N_y^2} - \sqrt{3N_x^2 + \kappa_\parallel N_y^2}}{3 + \kappa_\parallel} > 0.$$

4 Variable Relaxations

When $\kappa_\parallel$ is an nonlinear function of T, $\kappa_\parallel$ changes as T_{ij}^k changes at every vertex ij and every time step k. Therefore, time step and relaxation factor changes as well. This is why the name "Variable" is given. From now on, schemes (4) is called VR(4), scheme (7) is called VR(7), and $\kappa_\parallel$ is rewritten as κ_{ij}^k in nonlinear case. From the analysis given in the previous two sections, we have

$$\delta t_{ij}^k = \frac{1}{2}\frac{1}{\frac{1}{\delta x^2} + \kappa_{ij}^k \frac{1}{\delta y^2}} = \frac{\delta x \delta y}{4}\frac{2}{\frac{\delta y}{\delta x} + \kappa_{ij}^k \frac{\delta x}{\delta y}} \tag{10}$$

for VR(4) and

$$\delta t_{ij}^k \le \frac{1}{\sqrt{\frac{1}{\delta x^2} + \kappa_{ij}^k \frac{1}{\delta_y^2}}} = \frac{\sqrt{\delta x \delta y}}{\sqrt{\frac{\delta y}{\delta x} + \kappa_{ij}^k \frac{\delta x}{\delta y}}} = \frac{\sqrt{\delta x \delta y}}{\sqrt{2}}\frac{\sqrt{2}}{\sqrt{\frac{\delta y}{\delta x} + \kappa_{ij}^k \frac{\delta x}{\delta y}}} \tag{11}$$

$$\tau_{ij}^k = \frac{\sqrt{L_x L_y}}{\sqrt{2}\pi}\frac{\sqrt{2}}{\sqrt{\frac{L_y}{L_x} + \kappa_{ij}^k \frac{L_x}{L_y}}} \tag{12}$$

for VR(7).

5 Numerical Issues

In practical application due to nonuniform meshes and nonlinearity of the problem, δt and the damping factor τ are modified by scaling factors t_{scale} and τ_{scale}. The optimal δt and τ in both cases can be found by tuning these two parameters. This is summarized in the following table:

VR(4) for linear problem	VR(7) for linear problem
$\delta t = \frac{\delta x \delta y}{4}\frac{2}{\frac{\delta y}{\delta x}+\kappa_\parallel \frac{\delta x}{\delta y}} \cdot t_{scale}$	$\delta t = \frac{\sqrt{\delta x \delta y}}{\sqrt{2}}\frac{\sqrt{2}}{\sqrt{\frac{\delta y}{\delta x}+\kappa_\parallel \frac{\delta x}{\delta y}}} \cdot t_{scale}$ $\tau = \frac{\sqrt{L_x L_y}}{\sqrt{2}\pi}\frac{\sqrt{2}}{\sqrt{\frac{L_y}{L_x}+\kappa_\parallel \frac{L_x}{L_y}}} \cdot \tau_{scale}$
VR(4) for nonlinear problem	VR(7) for nonlinear problem
$\delta t_{ij}^k = \frac{\delta x \delta y}{4}\frac{2}{\frac{\delta y}{\delta x}+\kappa_{ij}^k \frac{\delta x}{\delta y}} \cdot t_{scale}$	$\delta t_{ij}^k = \frac{\sqrt{\delta x \delta y}}{\sqrt{2}}\frac{\sqrt{2}}{\sqrt{\frac{\delta y}{\delta x}+\kappa_{ij}^k \frac{\delta x}{\delta y}}} \cdot t_{scale}$ $\tau_{ij}^k = \frac{\sqrt{L_x L_y}}{\sqrt{2}\pi}\frac{\sqrt{2}}{\sqrt{\frac{L_y}{L_x}+\kappa_{ij}^k \frac{L_x}{L_y}}} \cdot \tau_{scale}$

$\frac{\delta x \delta y}{4}$ is the stability criterion for VR(4) when $\kappa_{\|} = 1$. $\frac{2}{\frac{\delta y}{\delta x}+\kappa_{\|}\frac{\delta x}{\delta y}}$ or $\frac{2}{\frac{\delta y}{\delta x}+\kappa_{ij}^{k}\frac{\delta x}{\delta y}}$ is the extra term if $\kappa_{\|}$ is larger than one or nonlinear. $\frac{\sqrt{\delta x \delta y}}{\sqrt{2}}$ is the stability criterion for VR(7) when $\kappa_{\|} = 1$. $\frac{\sqrt{2}}{\sqrt{\frac{\delta y}{\delta x}+\kappa_{\|}\frac{\delta x}{\delta y}}}$ or $\frac{\sqrt{2}}{\sqrt{\frac{\delta y}{\delta x}+\kappa_{ij}^{k}\frac{\delta x}{\delta y}}}$ is the extra term if $\kappa_{\|}$ is larger than one or nonlinear. For the relaxation factor τ, $\frac{\sqrt{L_x L_y}}{\sqrt{2}\pi}$ is the criterion for VR(7) when $\kappa_{\|} = 1$ and $\frac{\sqrt{2}}{\sqrt{\frac{L_y}{L_x}+\kappa_{\|}\frac{L_x}{L_y}}}$ or $\frac{\sqrt{2}}{\sqrt{\frac{L_y}{L_x}+\kappa_{ij}^{k}\frac{L_x}{L_y}}}$ is the extra term when $\kappa_{\|}$ is larger than one or nonlinear.

δx and δy are chosen based on the guidelines discussed in the previous sections so that as an example we have $N_x = (16-1)*2+1$ is 3 times less than $N_y = (51-1)*2+1$. N_x and N_y are the number of corresponding grid points in x and y directions. In this case VR(4) converged in 29708 number of iterations at optimal $t_{scale} = 0.174$; while VR(7) converged in 1308 number of iterations at optimal $t_{scale} = 0.41, \tau_{scale} = 0.87$. From here we can say that VR(7) is more than 20 times faster than VR(4). Hence from now on we will only use VR(7). Although iteration numbers seems to be large, each iteration is very cheap even compared to JFNK which requires preconditioning.

Next let's study the impact of initializing on convergence. As mentioned before, the nonlinear process can be initialized by the solution from the linear system with constant $\kappa_{\|}$. Given the linear solution with different size of $\kappa_{\|}$, the number of iterations for the nonlinear system to reach steady state is given in the following table. We found as long as the linear solution has $\kappa_{\|} \geq 2$, the nonlinear convergence doesn't have much difference. It only diverges when a guess has $\kappa_{\|} = 1$.

$\kappa_{\|}$	1	2	3	4	5	6	7,8,9,$10^1 \sim 10^9$
N_{its}	diverge	1313	1310	1309	1309	1309	1308

The marching process is even accelerated by varying δt and τ at each vertex ij and every time step k. We found the iteration won't even converge if uniform δt and τ are used.

Finally we give an efficient approach to update the nonlinear stiffness matrix $S_{yy}(T)$ at each time step. The numerical integration has to be carefully chosen in order to keep the symmetric structure as a result of periodic boundary condition. Generally

$$S_{yy}(T) = -\int\int \kappa_{\|}\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}d\sigma$$

where N_i and N_j are the ith and jth base functions in finite element space. On each triangle, assuming n is the index running through all of the collocation points, then one way to formulate $S_{yy}(T)$ at kth time step would be

$$S_{yy}^{ij}(T) = \sum_n w(n)\kappa^k(n)\frac{\partial N_i}{\partial y}(n)\frac{\partial N_j}{\partial y}(n)J(n)$$

where $w(n)$, $\kappa^k(n)$, and $J(n)$ are the corresponding weight, conductivity, and Jacobian at nth point. $\frac{\partial N_i}{\partial y}(n)$ and $\frac{\partial N_j}{\partial y}(n)$ are also valued at these points as well. As a function of T, $\kappa^k(n)$ can be found by

$$\sum_l (T_l^k)^{5/2} N_l(n) \quad \text{or} \quad \sum_l [T_l^k N_l(n)]^{5/2}$$

where l is the index running through all of the vertexes on each triangle. But experiments show that the symmetric structure is destroyed by the above two formulations. Then we worked out the following formula

$$S_{yy}^{ij}(T) = \kappa_{ij}^k \sum_n w_n \frac{\partial N_i}{\partial y}(n) \frac{\partial N_j}{\partial y}(n) J(n)$$

which leads to

$$S_{non} = S_{xx} + B^k S_{yy}$$

where B^k is a vector with component $B_{ij} = \kappa_{ij}^k$ at each vertex given by ij. Therefore, we conclude that the nonlinear stiffness matrix S_{yy} can be updated by just scaling the linear stiffness matrix S_{yy} using nonlinear vector B. This approach not only saves computation complexity, but also preserves the symmetric structure of the periodic solution. The nonlinear solution is shown in Fig. 1 again in (x, y) coordinate system. The linear initial guess with $\kappa_{\parallel} = 2 \times 10^4$ given in the left plot is applied.

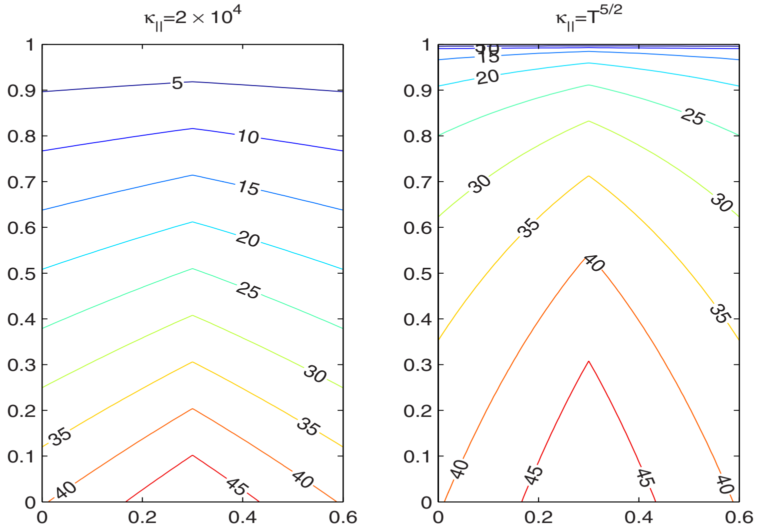

Fig. 1. Nonlinear solution at Nx=31, Ny=101, $t_{scale} = 0.41, \tau_{scale} = 0.87$. VR(7) is stable when $t_{scale} \leq 0.41$; VR(4) is stable when $t_{scale} \leq 0.174$.

6 Conclusions

As an extension of developing efficient linear elliptic solvers for fusion simulation on Cray X1E, nonlinear solver, based on Variable Relaxation, is constructed by by adding first and/or second order time derivative to the nonlinear elliptic equation and marching the resulting time-dependent PDEs to steady state. Instead of Jacobian, Only the stiffness matrix itself is involved and needs to be updated at each iteration.

Two schemes has been given, first and second order Variable Relaxations. four numerical issues has been fully discussed: The mesh scale ratio, nonlinear process initialization, variable time step and relaxation factor, efficient calculation of the nonlinear stiffness matrix. In summary, the mesh needs to be finer in direction with strong conductivity; convergence can be sped up by using the solution from corresponding linear system as an initial guess; time step and relaxation factor has to be varied at each grid point and every time step as well; only the nonlinear vector, used to update the nonlinear stiffness matrix, needs to be updated regularly. Therefore, the only computation consists of renewing δt_{ij}^k, τ_{ij}^k, and B^k at each iteration, and apparently these approaches give an efficient and robust algorithm to solve nonlinear systems.

References

1. W Park et al, Nonlinear simulation studies of tokamaks and STs, Nucl. Fusion **43** (2003) 483.
2. J Chen, S C Jardin, H R Strauss, *Solving Anisotropic Transport Equation on Misaligned Grids*, LNCS 3516, pp. 1076-1079 (2005).
3. D A Knoll, D E Keyes, Jacobian-free Newton-Krylov methods: a survey of approaches and applications, J comp. Phys. 193(2004) 357-397.
4. A Ern, V Giovangigli, D E Keyes, M D Smooke, Towards polyalgorithmic linear system solvers for nonlinear elliptic problems, SIAM J Sci. Comput. 15(1994) 681-703.
5. Y T Feng, On the discrete dynamic nature of the conjugate gradient method, J comp. Phys. 211(2006) 91-98.
6. J Chen, J Breslau, G Fu, S Jardin, W Park, *New Applications of Dynamic Relaxation in Advanced Scientific Computing*, proceedings of ISICS'06 Conference held at Dalian, China, Aug 15-18, 2006.
7. Y Saad, *Iterative Methods for Sparse Linearsystems*, PWS Publishing Company, (1996).