Nonlinear Network Tearing Algorithm for Transputer System Implementation

Dr A. Bargiela

Department of Computing, Nottingham Polytechnic
Burton Street, Nottingham NG1 4BU, U.K.

Abstract : This paper presents a new algorithm for simulation of large scale nonlinear systems. The algorithm is a development and generalisation of Kron's ideas on tearing linear systems and it combines Newton-Raphson iterative process with diakoptical calculation of state increments in each iteration. The nonlinear network tearing algorithm has been implemented in 3L Parallel Fortran 77 and run on a 5 transputer system hosted in a Unix workstation. Computational results, obtained for a realistic nonlinear system, are provided and discussed.

1. INTRODUCTION

Decomposition can provide a useful basis for the optimisation or control of many large scale systems. Physical structure of the problem often facilitates its partitioning into a number of smaller subproblems which can be solved individually, and the results recombined to achieve the solution of the overall problem. To take into account interactions between the subproblems, the overall solution proceeds as a sequence of subproblem solutions which are then modified by the coordinating task. The way in which the coordination is achieved characterises different decomposition methods.

Kron's method of decomposing large scale linear systems [1], referred to in the literature as network tearing or diakoptics, is a particularly efficient way of calculating a coordinated solution as it is a direct analytical technique avoiding any iterative re-calculation of subsystem solutions. Most importantly however, the form of decomposition resulting from the application of this technique maps directly onto MIMD parallel processing hardware (such as transputer arrays), enabling an approximately n-fold improvement in the computational efficiency. The major limitation of the Kron's technique is its requirement of system linearity which serves to ensure that the magnitude of corrections to subsystem solutions is not a limiting factor in the solution process.

While the application of diakoptics to large scale linear systems are found in such diverse fields as electrical systems modelling, structural analysis, neutron scattering etc.; decomposition of nonlinear systems, features less prominently in the literature.

This paper presents a development and generalisation of Kron's method of diakoptics applied to nonlinear systems. The nonlinear systems need to be solved subject to a constraint that the corrections to partial solutions have restricted magnitude so that the local mathematical models of subsystems are not invalidated in the process. The system is solved as a sequence of Newton-Raphson iterations with each iterative correction calculated as diakoptical coordinated solution to linearised subsystem. In order to facilitate an efficient use of a parallel processing hardware, the algorithm has been designed to minimise data traffic between subsystem solvers and the coordinating task.

2. THE ALGORITHM

Without detracting from the generality of the algorithm, in order to focus our discussion, the algorithm is described here in the in the context of water distribution networks, which are large scale, nonlinear systems. Let’s consider a system composed of two subnetworks, as illustrated in Figure 1a. The numerical simulation of such a system involves the solution of a set of mass balance equations for each node in the system.

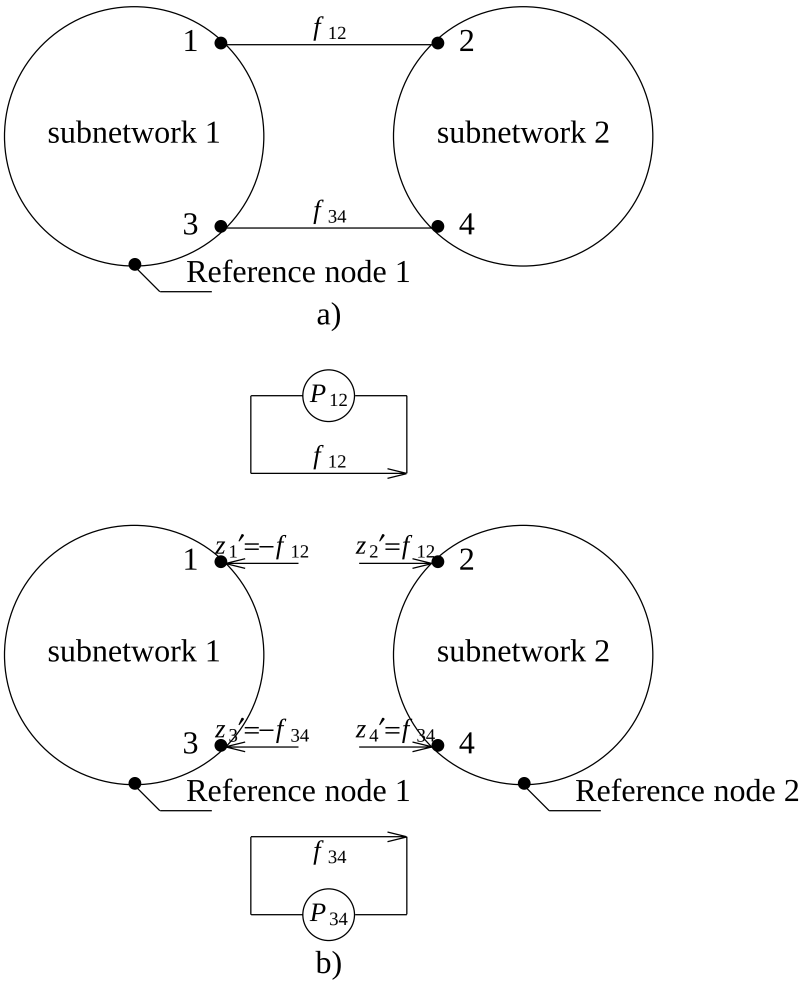

Figure 1 a) Original system
b) Equivalent partitioned system.

With the flows in network branches (pipes) f_{ij} modelled as

$$f_{ij}(\mathbf{x}) = R_{ij}(x_i - x_j)^{0.54} \tag{1}$$

where R_{ij} is the hydraulic conductivity of the i-j pipe and $\mathbf{x} = [x_1 \cdots x_n]^T$ is the vector of nodal pressures; the mass balance equations in n-1 network nodes and an equation for a reference pressure, are expressed as follows

$$g_i(\mathbf{x}) = \sum_{j\,\varepsilon\,\Omega_i} f_{ij}(\mathbf{x}) \tag{2}$$

$$g_n(\mathbf{x}) = x_n \tag{3}$$

where Ω_i - is a set of nodes adjacent to node i.

The system of nonlinear equations to be solved can therefore be represented in a compact form as

$$\mathbf{Z} = \mathbf{g}(\mathbf{x}) \tag{4}$$

where $\mathbf{Z} = [Z_1 \cdots Z_{n-1}, Z_n]^T$ and $\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}) \cdots g_{n-1}(\mathbf{x}), g_n(\mathbf{x})]^T$.

The solution of (4) involves linearisation of the system of equations and iterative improvement of the initial estimate $\mathbf{x}_0$ of the vector $\mathbf{x}$.

$$\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}_0) + \left.\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right|_{\mathbf{x}_0} .\Delta\mathbf{x} \tag{5}$$

introducing notation $J = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}$ for the Jacobian matrix the linearised equation (4) is

$$\mathbf{Z} = \mathbf{g}(\mathbf{x}_0) + J.\,\Delta\mathbf{x} \tag{6}$$

so

$$\Delta\mathbf{x} = J^{-1}\,(\,\mathbf{Z} - \mathbf{g}(\mathbf{x}_0)\,) \tag{7}$$

and the iterative solution to the nonlinear system is obtained as

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}. \tag{8}$$

With the network partitioned as in Figure 1(b), the subnetworks can be solved concurrently providing that the effect of the removed interconnecting branches is compensated by adding appropriate flows to the corresponding boundary nodes. Thus the vector **Z** becomes

$$\mathbf{Z}' = \mathbf{Z} + \mathbf{z}' \tag{9}$$

where

$$\mathbf{z}' = C_{\alpha\psi}\mathbf{f}_{\psi}(\mathbf{x}) \tag{10}$$

with ψ denoting cut-pipes in the network and $C_{\alpha\psi}$ being a node/cut-pipe incidence matrix defined as follows

$$C_{\alpha\psi} = \begin{cases} \;\;1, & \text{if the node } \alpha \text{ , is a sending node of the cut–pipe } \psi \\ -1, & \text{if the node } \alpha\text{, is a receiving node of the cut–pipe } \psi \\ \;\;0, & \text{if the node } \alpha\text{, is not incident to cut–pipe } \psi. \end{cases}$$

The removal of cut-lines decouples subnetworks so that the Jacobian matrix for the partitioned network assumes block diagonal form and becomes amenable to distributed computation of pressures $\mathbf{x}'$ in subnetworks. By analogy to equation (7) we can write,

$$\Delta\mathbf{x}' = J'^{-1}\,(\,\mathbf{Z} + \mathbf{z}' - \mathbf{g}(x_0)\,) \tag{11}$$

Substituting for $\mathbf{z}'$ and linearising cut-line flows around their values assumed for $\mathbf{x}_0$, the correction to the pressures in the subnetworks $\Delta\mathbf{x}'$ can be calculated as follows,

$$\Delta\mathbf{x}' = J'^{-1}\,(\,\mathbf{Z} + C_{\alpha\psi}.\mathbf{f}_{\psi}(\mathbf{x}_0) - \mathbf{g}(x_0)\,) + J'^{-1}C_{\alpha\psi}.\Delta\mathbf{f}_{\psi} \tag{12}$$

and with $\mathbf{Z}_0' = \mathbf{Z} + C_{\alpha\psi}.\mathbf{f}_{\psi}(\mathbf{x}_0)$ (12) becomes

$$\Delta\mathbf{x}' = J'^{-1}\,(\,\mathbf{Z}_0' - \mathbf{g}(\mathbf{x}_0)\,) + J'^{-1}C_{\alpha\psi}.\Delta\mathbf{f}_{\psi} \tag{13}$$

Since the corrections to subsystem states $\Delta\mathbf{x}'$ are the function of the corrections to cut-line flows $\Delta\mathbf{f}_{\psi}$, additional equations relating partitioned network pressures and cut-line flows need to be found. These are obtained as pressure balance equations for the removed networks, (Figure 1b)

$$\mathbf{h}(\mathbf{f}_\psi)+\mathbf{P}_\psi=\mathbf{O} \tag{14}$$

where $\mathbf{h}(\mathbf{f}_\psi)$ is a functional relationship between the flow and the pressure drop across the pipe, ψ, and

$$\mathbf{P}_\psi=C_{\psi\alpha}.\mathbf{x}'+M_{\psi\delta}.\mathbf{x}_\delta \tag{15}$$

quantifies the pressure drop in terms of the difference of pressures in the end-nodes of the pipe and the difference of reference pressures $\mathbf{x}_\delta$ in the relevant subnetworks. The cut-pipe/subnetwork incidence matrix, $M_{\psi\delta}$, is defined as follows:

$$M_{\psi\delta}=\begin{cases} 1, & \text{if the flow in the cut–pipe } \psi \text{ is directed into subnetwork } \delta \\ -1, & \text{if the flow in the cut–pipe } \psi \text{ is directed away from subnetwork } \delta \\ 0, & \text{if the cut–pipe } \psi \text{ is not incident to subnetwork } \delta \end{cases}$$

and $C_{\psi\alpha}=C^T_{\alpha\psi}$.

Linearising (14) and (15) around the $\mathbf{x}_0$, the pressure balance equations for the removed networks can be written as follows:

$$\mathbf{h}\,(\,\mathbf{f}_\psi(\mathbf{x}_0)\,)+\frac{\partial\mathbf{h}}{\partial\mathbf{f}_\psi}\Delta\mathbf{f}_\psi+C_{\psi\alpha}.\mathbf{x}_0+C_{\psi\alpha}.\Delta\mathbf{x}'+M_{\psi\delta}.\mathbf{x}_\delta+M_{\psi\delta}.\Delta\mathbf{x}_\delta=\mathbf{O}. \tag{16}$$

Assuming that the pressure balance holds for $\mathbf{x}_0$, (16) simplifies to:

$$H.\,\Delta\mathbf{f}_\psi+C_{\psi\alpha}.\Delta\mathbf{x}'+M_{\psi\delta}.\Delta\mathbf{x}_\delta=\mathbf{O} \tag{17}$$

where $H=\left.\dfrac{\partial\mathbf{h}}{\partial f_\psi}\right|_{\mathbf{f}_\psi(x_0)}$.

The corrections to the subnetworks' reference pressures $\Delta\mathbf{x}_\delta$, are related in (17) to the change of flow in cut-pipes $\Delta\mathbf{f}_\psi$ and consequently affect the mass transfers between subnetworks. It is necessary, therefore, to consider equations representing mass balance for the subnetworks:

$$\mathbf{Z}_\delta+M_{\delta\psi}.\mathbf{f}_\psi=\mathbf{O} \tag{18}$$

where $\mathbf{Z}_\delta$ is a sum of $\mathbf{Z}_i$ in all nodes of subnetwork δ, I_δ; $\mathbf{Z}_\delta=\sum_{i\,\varepsilon I_\delta}\mathbf{Z}_i$, and $M_{\delta\psi}=M^T_{\psi\delta}$.

Noticing the definition of $\mathbf{Z}'$ and the equations (9) - (10), the subnetwork mass balance (18) can be linearised around $\mathbf{x}_0$ to give

$$\mathbf{Z}_\delta'+M_{\delta\psi}.\Delta\mathbf{f}_\psi=\mathbf{O} \tag{19}$$

where

$$\mathbf{Z}_\delta'=\mathbf{Z}_\delta+M_{\delta\psi}.\mathbf{f}_\psi(\mathbf{x}_0) \tag{20}$$

Equations (13), (17) and (19) form the basis for the distributed solution of the nonlinear network system. They can be represented in a matrix form

$$\begin{bmatrix} J' & -C_{\alpha\psi} & 0 \\ C_{\psi\alpha} & H & M_{\psi\delta} \\ 0 & -M_{\delta\psi} & 0 \end{bmatrix}\begin{bmatrix}\Delta\mathbf{x}' \\ \Delta\mathbf{f}_\psi \\ \Delta\mathbf{x}_\delta\end{bmatrix}=\begin{bmatrix}\mathbf{Z}_0'-\mathbf{g}(\mathbf{x}_0) \\ \mathbf{O} \\ \mathbf{Z}_\delta'\end{bmatrix} \tag{21}$$

If the block diagonal matrices in J' have much higher ranks than matrices $M_{\psi\delta}$ and $C_{\psi\alpha}$, it is economical to solve concurrently the following auxiliary problems,

$$J'.\,\Delta\mathbf{x}''=\mathbf{Z}_0'-\mathbf{g}(\mathbf{x}_0) \tag{22}$$

to obtain the uncoordinated subsystem solutions $\Delta\mathbf{x}''$

$$\Delta\mathbf{x}'' = J'^{-1}.(\,\mathbf{Z}_0' - \mathbf{g}(\mathbf{x}_0)\,) \tag{23}$$

and subsequently coordinating them by taking into account corrections to subnetwork reference pressures and inter-subnetwork flows. Equation (23) allows us to re-formulate (21) and to reduce the coordination problem to the following:

$$\begin{bmatrix} H + C_{\psi\alpha} J'^{-1} C_{\alpha\psi} & M_{\psi\delta} \\ -M_{\delta\psi} & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{f}_{\psi} \\ \Delta\mathbf{x}_{\delta} \end{bmatrix} = \begin{bmatrix} -C_{\psi\alpha}\Delta\mathbf{x}'' \\ \mathbf{Z}'_{\delta}{}' \end{bmatrix} \tag{24}$$

On solution to (24) the iterative update to the calculated pressure is found from equations (13) and (23) as,

$$\Delta\mathbf{x}' = \Delta\mathbf{x}'' + J'^{-1} C_{\alpha\psi} \Delta\mathbf{f}_{\psi} \tag{25}$$

and

$$\Delta\mathbf{x} = \Delta\mathbf{x}' + K_{\alpha\delta}\Delta\mathbf{x}_{\delta} \tag{26}$$

giving

$$\mathbf{x}^{k+1} = \mathbf{x}^{k} + \Delta\mathbf{x} \tag{27}$$

which is analogous to (8).

The nonlinear network tearing algorithm can be summarised as follows:

1. Read in the system data.
2. Form subsystem data packets and send them to individual solvers.
3. Calculate uncoordinated subsystem solutions (equation 23).
4. Coordinate subsystem solutions (equations 24 - 26).
5. If the coordinated corrections from Step 4 are less then a given threshold value then STOP otherwise repeat from Step 2.

3. THE IMPLEMENTATION

The algorithm has been implemented in 3L Parallel Fortran 77 and executed on a network of transputers. A general structure of the program is illustrated in Figure 2. Steps 1, 2, 4 and 5 of the algorithm are contained within ’Task-A Coordination’ and Step 3 is contained within ’Task-B Subsystem Solution’. Since the copies of task B are placed on separate transputers, the issue of minimisation of data traffic between Task-A and -B had to be carefully addressed.

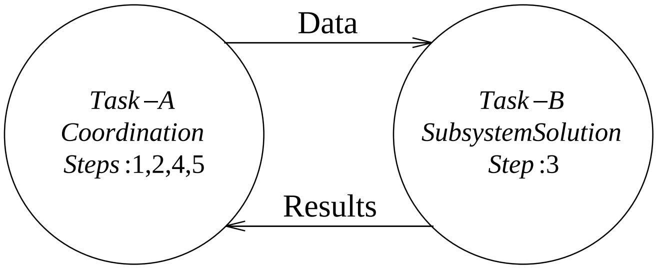

Figure 2 Program structure

Let’s consider a typical network of N nodes, with each node connected to 4 other nodes in the network. A transfer of nonzero elements of the Jacobian matrix from Task-A to Task-B would involve sending 5xN real values together with associated pointers, resulting in a transfer of at least 11xNx4 bytes of data (if a sparse storage structure such as ’pointer-column-index’ is used, or 15xNx4 bytes if it is not).

Instead, in our implementation, the Jacobian matrices are derived locally within Task-B from network topological data, which require sending of 6xNx4 bytes of data. Apart from an obvious reduction of a data traffic an additional benefit of concurrent evaluation of Jacobian matrices has been achieved.

The coordination process, equation (24), requires the knowledge of the inverse of Jacobians. If implemented directly, this would imply the need to send from Task-B to Task-A 3xNxNx4 bytes of data! This problem had been noted by other researchers [2] but no effective solution was proposed at the time.
Our implementation resolves the problem by stipulating Task-B to evaluate explicitly selected columns of inverse Jacobian $J'^{-1}C_{\alpha\psi}$ which is all that is needed by Task-A. By doing so, the required data transfer is reduced to 3xNxCx4 bytes; where C is a number of inter-subnetwork connections and is much smaller than N. The above modification resulted in a reduction of data traffic between Task-B and Task-A by a factor of 10 to 100 for a typical large scale system.

Additionally, optimisation of data transfers within Task-A was performed by using the facility of threads as discussed in [3].

Testing the program on real-life networks, it has been found that the time required by Task-B depends quasi-linearily on the number of nodes in the subnetwork, and the time required by Task-A depends quadratically on the number of inter-subnetwork connections. Since the number of inter-subnetwork connections does not depend on the network size but only on the number of subsystems and the average node connectivity, the algorithm offers a potential of significant concurrent processing gains for large scale systems.

With the reduction of the volume of transferred data, the time spent on communication constituted approximately 1% of total processing time for a 100 node system. As the volume of data transmitted between two tasks is linearily proportional to the network size, the data transfer time will continue to be negligible.

4. CONCLUSIONS

While the idea of decomposed solution of large scale linear systems has been well researched, the decomposition of nonlinear systems remains a relatively unexplored area. This paper presents an efficient algorithm for parallel solution of nonlinear networks. The algorithm generalises the idea of network tearing introduced for linear systems by Kron. The proposed algorithm overcomes the problem of large data transfers, usually associated with diakoptics.

REFERENCES

1. KRON G., -’Diakoptics: The piecewise solution to large scale systems’, Macdonald, 1963

2. BOWDEN K., - Kron’s Method of Tearing on Transputer Arrays, The Computer Journal, Vol. 33, No.5, 1990.

3. HOSSEINZAMAN A., BARGIELA A., - Parallel simulation of nonlinear networks using diakoptics’, Proceedings PACTA’92, Barcelona, Sept. 1992.