

**ALGORITMO PARA REDUÇÃO DA BANDA DE
MATRIZES SIMÉTRICAS ESPARSAS**

José Luiz F. de Arruda Serra

Departamento de Estruturas, Faculdade de Engenharia Civil
Universidade Estadual de Campinas – UNICAMP
13084-100, Campinas – S.P. - Brasil

RESUMO

Um algoritmo simples porém eficiente para redução da banda de matrizes simétricas esparsas de conectividade tais como matrizes de rigidez e redes de distribuição de água é apresentado.

O procedimento toma uma seqüência arbitrária de entrada e cria uma nova seqüência para os sistemas complexos de redes, renumerando os índices dos nós procurando otimizar a banda. Assim, considerável economia de memória no armazenamento e tempo de computação é alcançado na solução dos sistemas envolvidos.

Vários exemplos são apresentados para demonstrar a confiabilidade e eficiência do processo proposto.

ABSTRACT

A simple but efficient automatic algorithm for reducing the bandwidth of symmetric sparse connectivity matrices like stiffness matrix and water distribution network is presented.

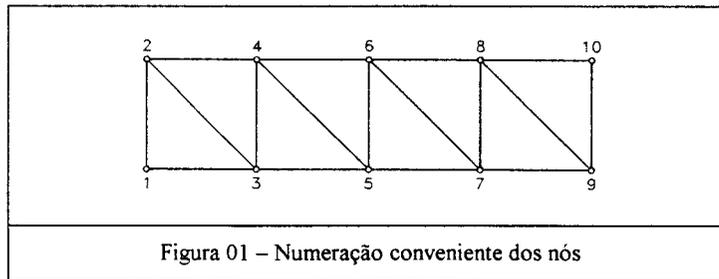
The procedure takes an arbitrary input sequence and creates a new sequence for complex network systems by renumbering the node labels looking for an optimal bandwidth. Thus, considerable improvement for computer storage and time is achieved in the solution of the systems

Several examples are presented to demonstrate the reliability and the efficiency of the method.

1. O ALGORITMO PARA REDUÇÃO DA BANDA

Para reduzir a banda de uma matriz esparsa, procurou-se elaborar, em benefício da simplicidade, um procedimento baseado em apenas dois critérios para obter uma nova numeração para os pontos nodais de uma rede de conectividade qualquer. Nesta fase da análise a largura da banda é considerada como a maior diferença que ocorre entre os numerais de nós conectados. Como será visto nos exemplos, os resultados obtidos pelo programa desenvolvido são altamente satisfatórios, raramente não alcançando os melhores resultados apresentados na bibliografia.

Para compreensão dos dois critérios usados pelo algoritmo proposto, vamos supor a treliça da figura abaixo, cujos nós foram numerados de modo a obter a menor banda possível.



A seguir está apresentada uma tabela com os nós, a matriz [mc] de suas conexões, a soma dos termos de uma linha da matriz [mc] dividida pelo número de termos (soma ponderada) e a soma dos valores extremos de cada linha.

Nó	Matriz de conexão [mc]					Soma ponderada	Soma dos extremos
1	1	2	3			2,0	4
2	2	1	3	4		2,5	5
3	3	1	2	4	5	3,0	6
4	4	2	3	5	6	4,0	8
5	5	3	4	6	7	5,0	10
6	6	4	5	7	8	6,0	12
7	7	5	6	8	9	7,0	14
8	8	6	7	9	10	8,0	16
9	9	7	8	10		8,5	17
10	10	8	9			9,0	18

Observa-se na tabela que quando a numeração dos nós é conveniente, a matriz de conexão [m] apresenta as colunas **soma ponderada** e **soma dos extremos** em ordem ascendente.

O algoritmo desenvolvido procura a partir de uma numeração original arbitrária dos nós, uma nova numeração que satisfaça na medida do possível esses dois critérios. Não há garantia de se obter o mínimo absoluto para a banda - nenhum procedimento garante o mínimo absoluto - mas uma banda próxima da ideal pode ser conseguida. Outra característica de todos os processos é que a numeração original afeta o resultado obtido.

Em linhas gerais o programa desenvolvido é através de ciclos, procurando convergir para uma numeração apropriada, segundo os passos abaixo descritos:

1. Leitura dos pontos nodais de cada elemento da estrutura (ou rede).
2. montagem da matriz [mc] de conexão dos nós e do vetor {ncolm} que define o número de elementos de cada linha da matriz [mc].
3. Cálculo do vetor da soma ponderada {ps} e sua ordenação em ordem crescente, obtendo o vetor {ordem} que relaciona as alterações da ordenação de {ps} (procedimentos *PondSum* e *Ordena* do programa).
4. Baseado no vetor {ordem}, altera-se as linhas e colunas da matriz [mc], procurando satisfazer este critério da soma ponderada. O ajuste das colunas segue uma técnica especial: a matriz após a troca simples das linhas é percorrida em sua totalidade e troca-se todos os números relacionados, isto é, se por exemplo a linha 3 foi trocada com a 7, cada vez que o número 3 for encontrado, ele é substituído pelo 7 e vice versa. Desta forma preserva-se as conexões corretamente.
5. Determina-se um vetor {ordem final} que relaciona a numeração original com a atual. Este vetor é o seu valor anterior, alterado segundo o vetor {ordem}. Notar que o vetor {ordem} reproduz apenas a última alteração (corrente), enquanto o vetor {ordem final} vai acumulando todas as alterações para manter o elo com a numeração original.
6. Determina-se a largura da banda. Se esta for menor que a banda mínima obtida nas fases anteriores, guarda-se seu valor *bmin* e o correspondente {ordem final}_{mínimo}.
7. Repete-se os passos 3, 4, 5 e 6 para o critério da soma dos extremos (vetor {es} - procedimento *ExtreSum*).
8. Repete-se o processo completo, isto é, ordenação e conseqüente alterações de [mc], até que os dois critérios estejam ordenados ou a banda não tenha sofrido alterações a um certo tempo ou ainda atingiu-se um número máximo prescrito para os ciclos, evitando eventuais ciclos infinitos.

Os passos 4, 5 e 6, estão agrupados no procedimento *Change* do programa.

No final do processo o vetor {ordem final}_{mínimo} é o que relaciona a numeração original com a primeira numeração obtida que conduz à banda mínima conseguida pelo algoritmo. Observou-se nos exemplos processados, que o valor mínimo da banda não é estacionário; algumas vezes é obtido nos primeiros ciclos, voltando a crescer - mesmo que ligeiramente - e não retornando ao valor mínimo apesar de muitos ciclos completos serem processados posteriormente.

Nos inúmeros testes realizados em dezenas de "redes", observou-se o seguinte:

- As alterações sofridas na largura da banda são mais acentuadas nos primeiros ciclos;
- Para uma mesma malha e mesma numeração original, se houver uma inversão na ordem dos critérios, isto é, iniciar-se os ciclos pelo critério da soma dos extremos em lugar do critério da soma ponderada, o resultado final para o valor da banda mínima pode ser diferente.
- Ao sair de um critério não há garantia que ele esteja ordenado, pois a fase que altera as colunas troca números na linha, modificando os valores da soma ponderada ou da soma dos extremos, podendo portanto afetar sua ordenação.

A observação destes fatos e o desejo de aplicar "perturbações" - mesmo que pequenas - na matriz de incidência conduziu para uma alteração no procedimento que ordena os vetores da soma ponderada e soma dos extremos; passou-se a trocar de linhas também valores iguais (classicamente só se efetua troca de linhas em uma ordenação caso os valores sejam diferentes, pois normalmente não há porque ordenar valores iguais). Isto afeta a fase seguinte do

acerto das colunas e favorece encontrar outros arranjos que podem melhorar a banda. Notou-se também que com esta alteração deixou de influir nos resultados a inversão da ordem dos critérios dentro de um ciclo. É certo que esta alteração prejudica o tempo de processamento. Assim, a critério do usuário, basta alterar no procedimento que ordena itens o sinal da expressão que compara os itens; o sinal \leq ou apenas $<$ é a única alteração necessária para privilegiar ou o tempo (sinal \leq) ou a banda mínima (sinal $<$), lembrando que a diferença nos valores obtidos para a banda mínima é pequeno, assim como no tempo de processamento, principalmente com os computadores atuais, cada vez mais velozes.

Baseado nestas observações e nos resultados obtidos nos testes, outra alteração no procedimento "linear" acima sugerido conduziu a resultados melhores: antes de se trocar de critério, repete-se algumas vezes a ordenação para o mesmo critério antes de se passar para o outro, isto é, para cada ciclo completo (externo), efetua-se várias vezes a ordenação - se necessária - segundo cada um dos critérios (ciclos internos).

Os resultados apresentados neste trabalho foram obtidos com a definição dos valores 15 para número máximo de ciclos completos (externos) e 5 para máximo dos dois ciclos internos. Isto significa que em casos extremos, o número máximo de mudanças na matriz de incidência é 150. Nas páginas após as conclusões finais e referências bibliográficas estão incluídos os fluxogramas do algoritmo e seus procedimentos (subrotinas).

2. COMPARAÇÃO DE RESULTADOS E EXEMPLOS

A tabela 1 compara os valores obtidos para seis exemplos amplamente divulgados e analisados por diversos autores. Estes exemplos são apresentados por Grooms [1] e estão numerados na mesma ordem do artigo para facilitar eventuais consultas, uma vez que as respectivas figuras não puderam ser reproduzidas por economia de espaço.

Exemplo	Banda original	Autores									Presente trabalho
		[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	
1	15	3	3	2	2	3	5	5	3	2	3
2	36	7	5	5	5	5	12	7	7	5	5
3	18	5	4	4	4	4	5	5	5	4	4
4	37	7	9	9	8	9	9	9	8	9	7
5	52	19	19	–	16	–	26	28	–	21	16
6	16	9	7	–	6	7	9	9	7	8	7

A tabela 2 compara os resultados obtidos por Akhras & Dhatt [2] para os exemplos apresentados em seu artigo com os nomes WHELL, WAT, CAV e CIR. Também pelo motivo acima mencionado, as figuras não estão reproduzidas. Os 10 exemplos incluem figuras planas e tridimensionais formadas por barras e portanto com dois nós por elemento e figuras formadas por superfícies, discretizadas em elementos triangulares com três ou seis pontos nodais, elementos quadriláteros com oito pontos nodais e para as não planas, elementos hexaédricos com oito pontos nodais, contemplando portanto a maioria de discretizações possíveis para estruturas e redes de distribuição.

TABELA 2 – Resultados obtidos para bandas				
Exemplo	WHELL	WAT	CAV	CIR
Banda original	67	48	36	39
Akhras & Dhatt [2]	27	10	18	25
Presente trabalho	27	8	17	25

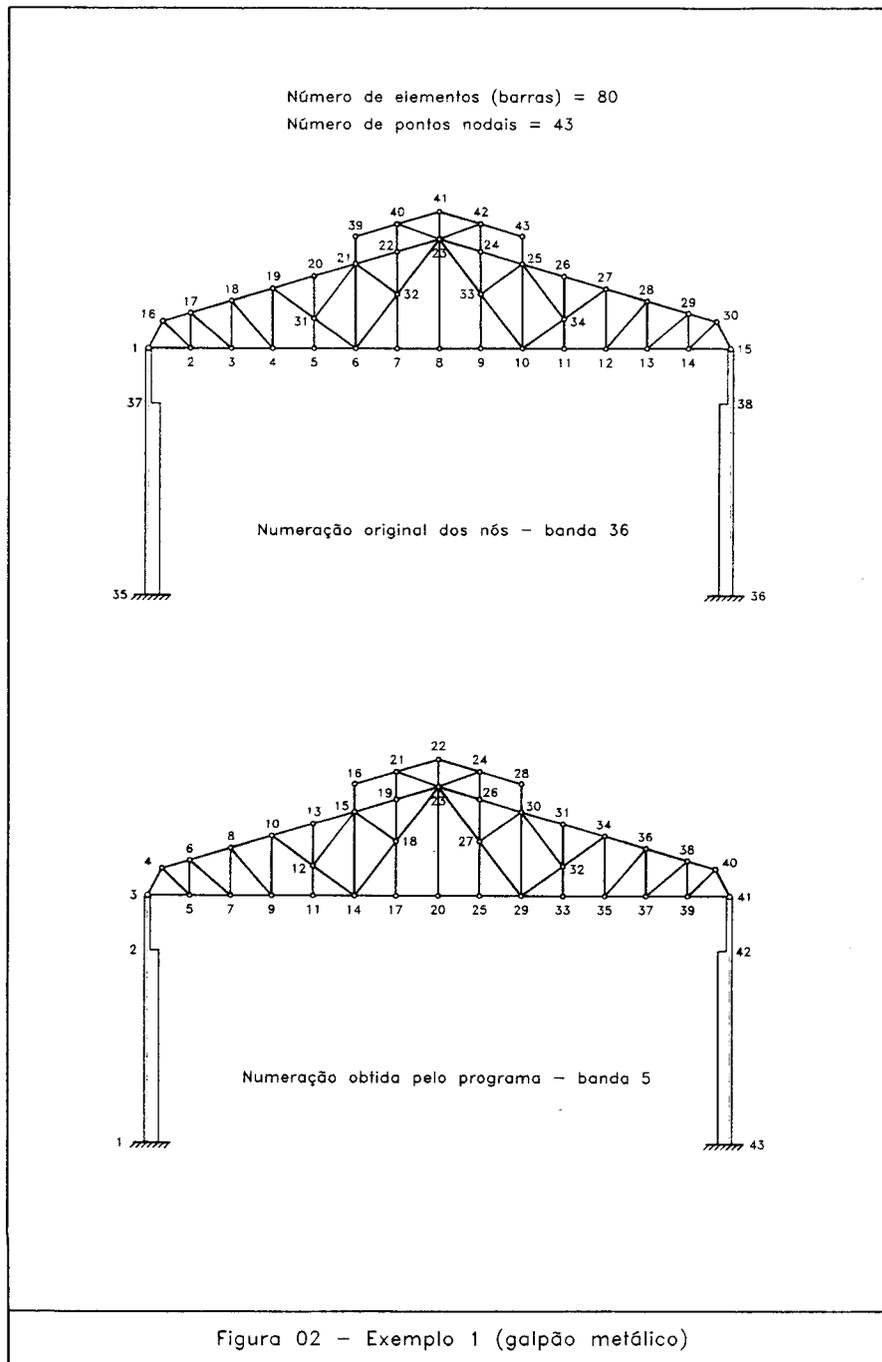
Foram incluídos mais 2 exemplos de estruturas comuns na área de Engenharia Civil e analisados pelo autor. O primeiro é um galpão industrial de estrutura metálica com cobertura em duas águas e lanternim (figura 02) e o segundo é um pórtico de edifício de concreto armado (figura 03). As respectivas figuras apresentam os resultados obtidos pelo programa.

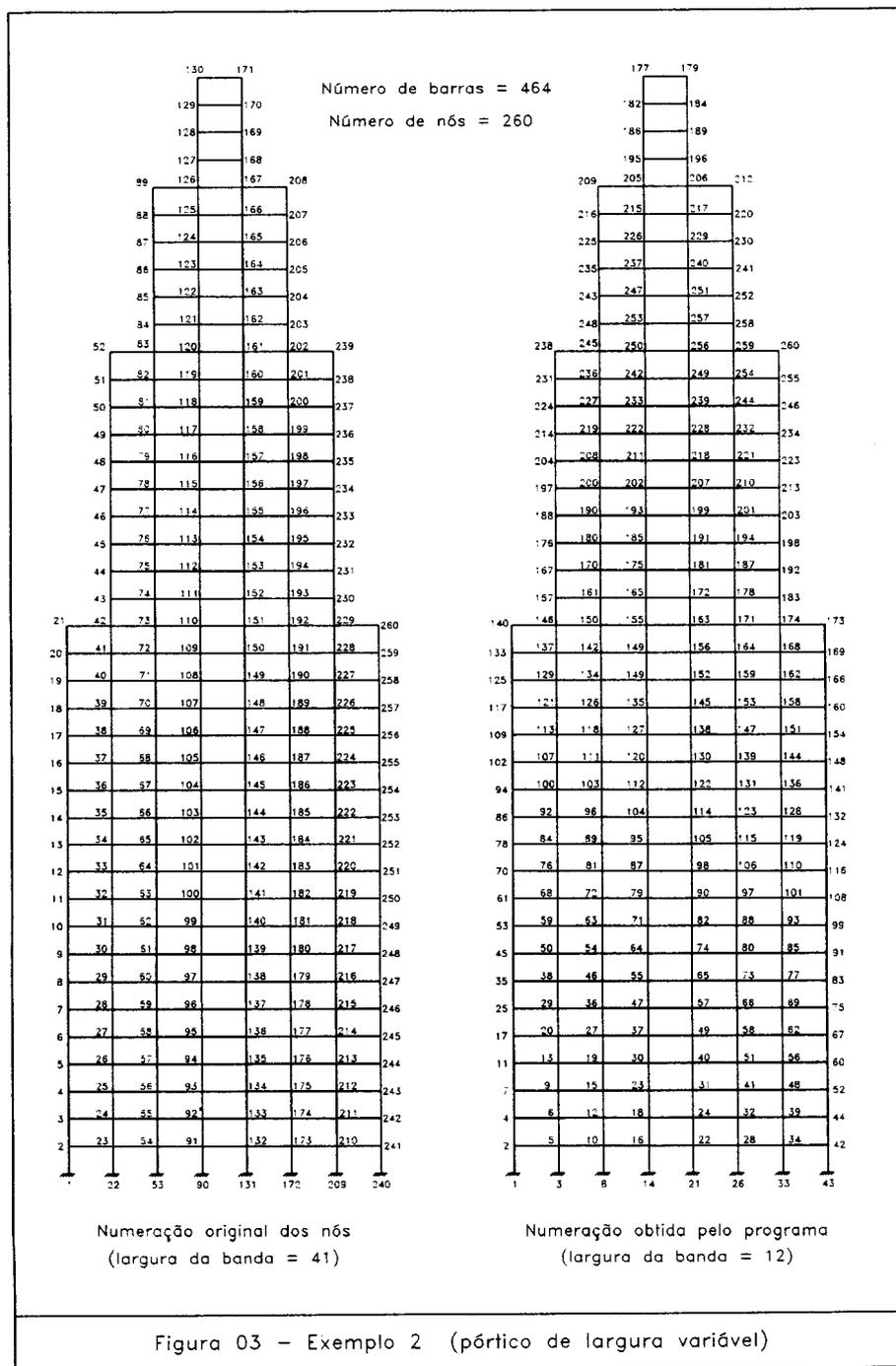
3. CONCLUSÃO

À vista da simplicidade do algoritmo proposto e resultados obtidos, julga-se que o trabalho foi concluído com sucesso. A implementação deste procedimento nos programas de estruturas conduz a uma indubitável melhora de eficiência. Além da economia de memória, tempo de computação e precisão dos resultados, a incorporação do algoritmo permite que a numeração dos pontos nodais dos elementos seja liberada para facilitar eventual geração automática dos nós e elementos, mesmo que prejudicando "a priori" a largura da banda.

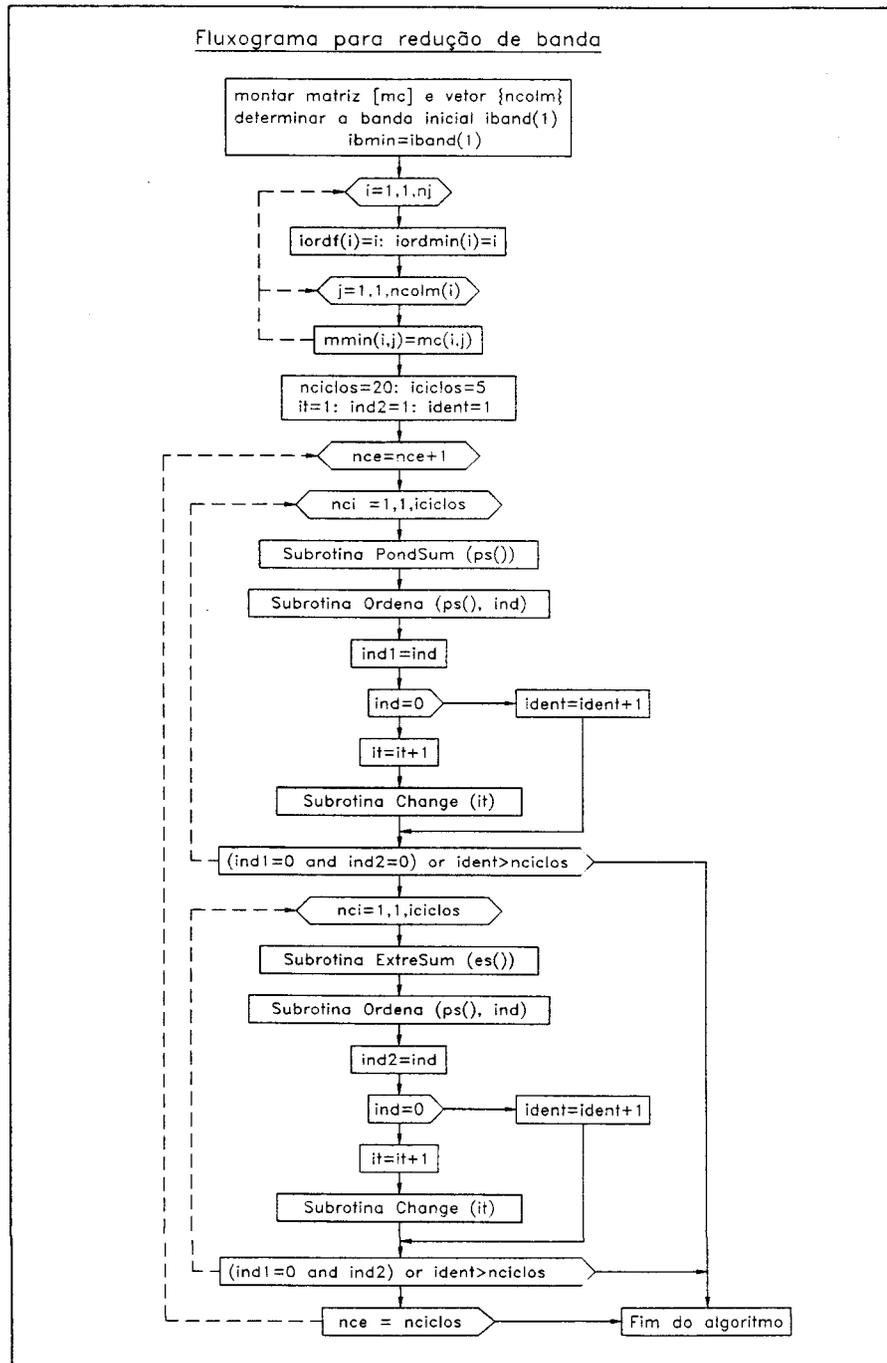
4. REFERÊNCIAS

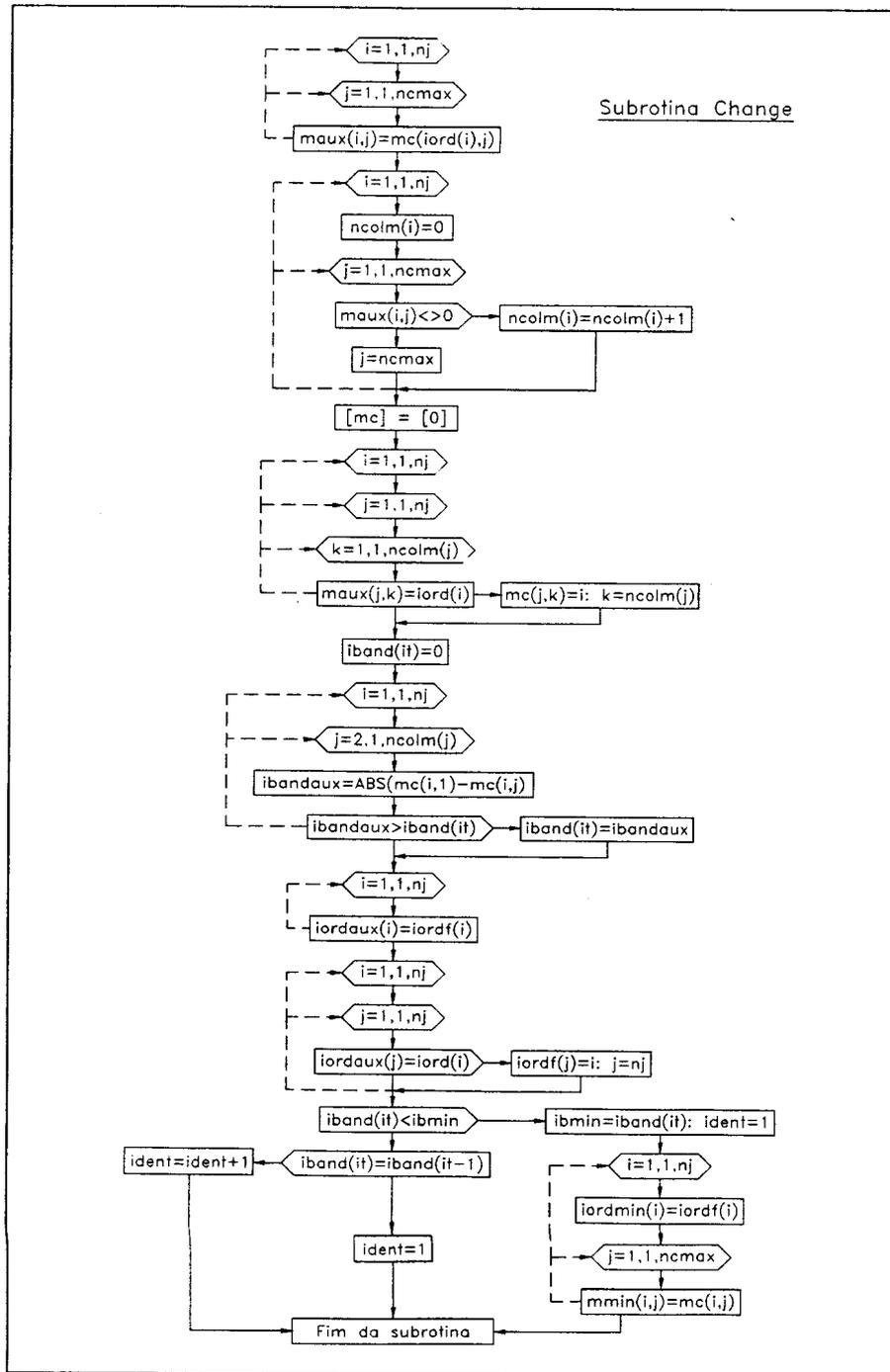
- [1] Grooms, H. R., Algorithm for Matrix Bandwidth Reduction, Journal of the Structural Division, ASCE (ST 1), pp 203-214, Jan., 1972.
- [2] Akhras, G & Dhatt, G., An Automatic Node Relabelling Scheme for Minimizing a Matrix or Network Bandwidth, International Journal for Numerical Methods in Engineering, vol. 10, pp 787-797, 1976.
- [3] Cuthil, E & Mckee, J., Reducing the Bandwidth of Sparse Symetric Matrix, Proceedings ACM National Conference, New York, 1969.
- [4] Collins, R.J., Bandwidth Reducing By Automatic Renumbering, Internatinal Journal for Numerical Methods in Engineering, vol 6, pp 345-356, 1973.
- [5] George, A., Computer Implementation of Finite Element Method. Tech. Rep. STAN – CS – 71 – 208, Computer Science Dept. Stanford Univ., 1971.
- [6] Rosen, R., Matrix Bandwidth Minimization. Proceedings of the 23rd ACM National Conference, Brandon Systems Press, Princeton, N.J., 1968.
- [7] Akyuz, F.A. & UTKU, S., An Automatic Node-relabelling SHEME for Bandwidth Minimization of Stiffness Matrix, Journal of the American Institute of Aeronautics and Astronautics. 6(4), pp 728-730, 1968.
- [8] Melosh, R.J. & Bandford, R.M., Efficient Solution of Load-Deflection Equations, Journal of Structural Division, ASCE, 95, pp 661-676, apr., 1969.
- [9] Teixeira, F.G. & Groehs, A.G., Sistema de Reordenação Nodal Para Soluções do Tipo Banda, XXV Jornadas Sul-Americanas de Engenharia Estrutural, Porto Alegre, Brasil, Anais vol.1, pp 57-68, 1991.



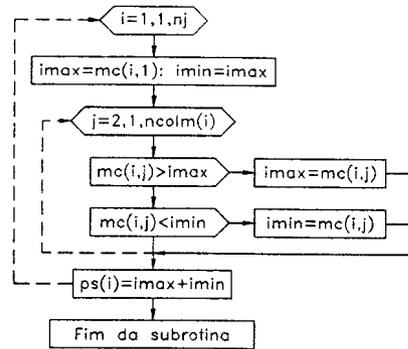


Fluxograma para redução de banda

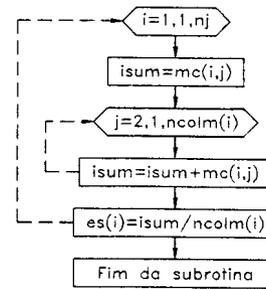




Subrotina PondSum (ps())



Subrotina ExtreSum (es())



Subrotina Ordena (list(), ind)

