

Desenvolvimento de um programa executivo para simulação de circuitos de tratamento de minérios

Otávio Gil *
Marcelo Castier **
Krishnaswamy Rajagopal **

1. INTRODUÇÃO
2. PROGRAMAS EXISTENTES
3. CONSIDERAÇÕES PARA O DESENVOLVIMENTO DO PROGRAMA EXECUTIVO
4. DESCRIÇÃO DO PROGRAMA EXECUTIVO
5. RESULTADOS E CONCLUSÕES
6. BIBLIOGRAFIA
7. ANEXOS

* Aluno de mestrado da COPPE/UFRJ

** Professores da COPPE/UFRJ

Resumo

A realização do balanço de massa é uma etapa importante quando se deseja projetar, modificar ou determinar condições ótimas de operação em circuitos de tratamento de minérios. Essa tarefa envolve extensivos trabalhos de cálculo, principalmente quando se lida com circuitos complexos, com várias unidades, correntes e recírculos. Assim sendo, torna-se necessária a utilização de programas computacionais de simulação para que esses cálculos possam ser realizados de forma rápida e eficiente.

1. Introdução

Uma das questões mais importantes entre as que se colocam para quem trabalha na área de operação ou projeto de circuitos de tratamento de minérios é alcançar a eficiência máxima desses circuitos. Como os processos industriais operam com grandes quantidades de matéria, e apresentam elevados gastos energéticos, pequenas melhorias podem resultar grandes economias. Alterações nos valores das variáveis ou na estrutura do circuito podem ter impacto significativo sobre a eficiência. Isto justifica os esforços que vêm sendo desenvolvidos para abordar essa questão de maneira mais sistemática.

A busca da eficiência máxima de um circuito pode dar-se tanto ao nível de se procurar alcançar a melhor configuração possível (tipo, número de unidades e suas interligações), bem como ao nível de se procurar estabelecer os valores ótimos das variáveis que podem ser manipuladas. Para que se alcancem essas condições ótimas é preciso comparar diferentes alternativas, avaliando o desempenho de cada uma. Por esse motivo, torna-se imperativo que se desenvolvam métodos rápidos, eficazes e econômicos para realizar tais avaliações, o que hoje vem sendo conseguido através do emprego de computadores digitais. O emprego do computador, enquanto ferramenta auxiliar, torna-se sobretudo indispensável quando se trabalha com circuitos complexos, com grande número de recírculos, que exigem um volumoso trabalho de cálculo.

Neste trabalho é apresentado o desenvolvimento de um programa executivo, que a partir da descrição de uma determinada planta, gera um programa para sua simulação, bem como a estrutura de dados que o mesmo utilizará. Tanto o programa executivo quanto os programas gerados utilizam a linguagem Fortran.

A característica principal que se busca alcançar é a flexibilidade, ou seja, a possibilidade de se simular diferentes configurações de diferentes circuitos com o mesmo sistema computacional.

Uma das etapas necessárias para uma completa avaliação de um processo é a simulação do balanço de massa ao longo de vários circuitos alternativos. Para este fim foi desenvolvido um sistema computacional seqüencial modular na COPPE/UFRJ. O sistema é composto por um programa executivo, que a partir da descrição do circuito a ser simulado, gera um programa de simulação específico para esse circuito. A estrutura de dados também é gerada automaticamente, sendo assim, específica para a simulação de cada circuito.

Nesse artigo é feita uma descrição do programa executivo e da estrutura de dados gerada por ele. São apresentados exemplos da geração automática do programa de simulação e da estrutura de dados.

2. Programas existentes

O uso de simuladores na área de tratamento de minérios é relativamente recente se comparado ao emprego dos mesmos na indústria química (1). A razão disso reside no fato de que modelos matemáticos adequados para os equipamentos presentes na configuração de qualquer planta de concentração mineral não eram disponíveis. Ainda hoje, muito há para ser desenvolvido nesses modelos.

O desenvolvimento da simulação aplicada a processos químicos se inicia no meio da década de 1950, e a primeira publicação ocorreu em 1958, o "Flexible Flowsheet"(6).

Um primeiro estágio do uso das técnicas de simulação configurou-se pela criação de programas particulares para cada planta a ser analisada. Segundo Ravicz e Norman(3), a principal limitação desses programas é a falta de flexibilidade que apresentam diante das inevitáveis alterações na estrutura de um processo durante sua fase de projeto. Na busca da superação dessa limitação desenvolveram-se sistemas computacionais de simulação flexíveis e gerais, capazes de simular diferentes plantas sem que a estrutura de seus programas tivesse que ser alterada. Nesses sistemas os equipamentos e as correntes que os interligam são parte dos dados de entrada. Em revisões sobre o assunto encontradas na literatura (2,4,5) são discutidos e comparados diversos sistemas, todos específicos para processos químicos. Quase todos utilizam estrutura matricial de dados.

Entre os sistemas de simulação para circuitos de Tratamento de Minérios podem ser citados os trabalhos de Riquelme e Villavueva (1), King (6), e o projeto 'SPOC' no Canadá (7). No primeiro trabalho é apresentado o desenvolvimento de um simulador que envolve subrotinas computacionais para Moinhos de Barras, Moinhos de Bolas, Hidrociclones e Bancas de Flotação. O simulador de King é aplicável somente para Bancas de Flotação. Nos três simuladores o armazenamento das informações é feito em matrizes.

3. Considerações para o desenvolvimento do programa executivo

3.1. A estrutura matricial de dados

Apesar de todo o trabalho já desenvolvido, ainda permanecem carências no ferramental computacional para a análise de muitos processos. Os problemas existentes podem ser notados em três características dos atuais programas (8,9): não são disponíveis ao público; só podem resolver um número limitado de processos; não são flexíveis quanto à sua partição.

A primeira característica negativa dos sistemas atuais é ainda mais marcante no Brasil, onde se tem pouco acesso aos mesmos. A ausência de um sistema disponível e de fácil utilização dificulta tanto o trabalho a nível da produção quanto a nível da educação, pois fica dificultada a formação de profissionais com experiência prática em simulação de processos (9).

A segunda característica negativa decorre da estrutura de dados desses sistemas. Na estrutura matricial, em cada posição da matriz é armazenado um tipo específico de informação. Em processos nos quais existem correntes com conjuntos diferentes de propriedades que as caracterizam, é necessário reservar espaço de memória para todos os tipos de informações em todas as correntes. Isto representa um desperdício de espaço de memória. Esta é a principal dificuldade para a criação de um sistema

capaz de simular plantas que envolvem tanto correntes líquido-vapor, quanto correntes sólido-vapor ou líquido-sólido. É importante notar que mesmo quando trabalhamos com circuitos que envolvam apenas operações de tratamento de minérios, correntes líquido-sólido, ocorre desperdício de memória devido a possíveis diferenças significativas quanto às faixas granulométricas das correntes (1).

Da estrutura matricial de dados também decorre a terceira característica negativa citada: falta de flexibilidade no sistema para que se possa extrair porções do mesmo, modificá-lo, ou mesmo adaptá-lo para aplicações específicas (9).

Toda a programação baseou-se em uma matriz construída a partir das variáveis de interesse conhecidas no momento da sua construção. Para a inclusão de novas variáveis, não previstas anteriormente, é preciso alterar a matriz e consequentemente os diferentes programas que nela se baseavam (10,11). Para trabalhar-se com parte do sistema, ou utiliza-se parte da matriz vazia, o que representa desperdício de memória, ou constrói-se nova matriz simplificada, o que mais uma vez implica em grande tarefa de programação. Sendo sistemas de difícil partição, não são próprios para o trabalho em microcomputadores (8).

Na busca da superação dessas limitações é que foi proposta (8,9,10) a criação de um novo sistema. Esse sistema deveria ser flexível: divisível em partes, facilmente modificável, adaptável para diferentes processos e diferentes computadores, bem integrado e de fácil comunicação com o usuário. Para tanto, tal sistema deveria ter algumas características que já se faziam presentes em sistemas tradicionais, como "linguagem orientada" para os dados de entrada, técnicas avançadas para solução de ciclos, estrutura modular e verificação de erros. Adicionalmente, o sistema deveria apresentar uma estrutura de dados flexível, que pudesse ser modificada de acordo com a natureza do processo a ser simulado. Essa abordagem se contrapõe aos sistemas massivos, que solucionam com a mesma estrutura de dados todos os casos para os quais foram previstos.

3.2. A estrutura "plex" de dados

O termo "plex" vem do grego e significa a combinação de partes ou elementos interligados numa estrutura. A estrutura "plex", também denominada estrutura de listas encadeadas, foi proposta inicialmente por Ross (1961) (12), e seu uso em projeto de processos foi sugerido por Porter (1969) (13).

Nesta estrutura, os dados são armazenados sob a forma de vetores de informação e posição. Os dados não se encontram em posições fixas, como no caso da forma matricial. Cada vetor é auto-informado, ou seja, contém nele próprio as informações sobre as posições dos dados. Um elemento do vetor pode conter tanto uma informação sobre o processo quanto sobre uma posição de dados. Um elemento que contém esse último tipo de informação é chamado

"ponteiro". É através desses ponteiros que o programa terá acesso aos diferentes dados (9,10,11).

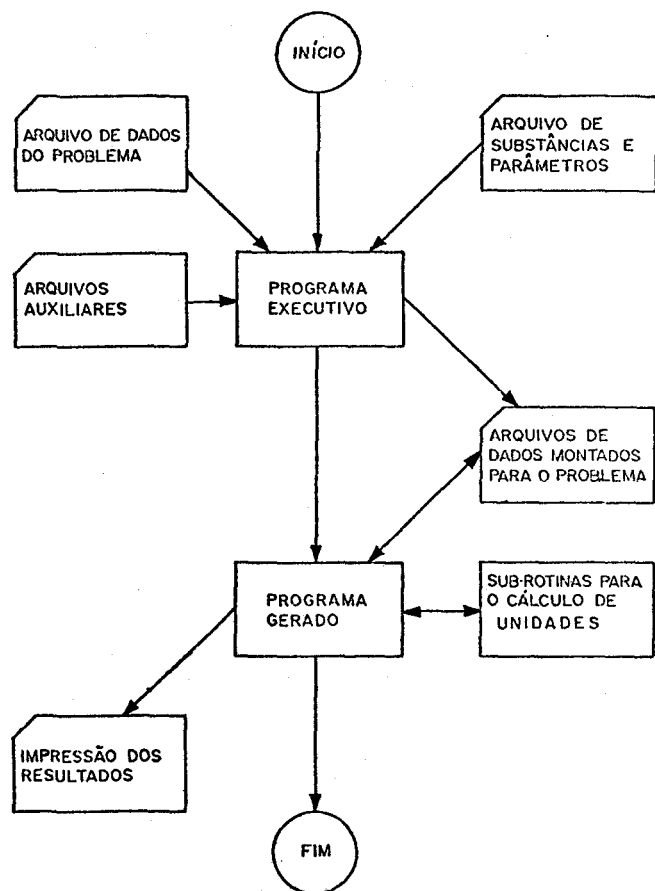
Todas as vantagens dessa estrutura de dados advêm da geração de estruturas de dados específicas para cada planta a ser simulada. A não existência de uma única estrutura, massiva e fixa para todos os casos, confere ao sistema maior flexibilidade.

Em 1976 foi implementado o sistema protótipo "Plexys" (14), que serviu de base para o desenvolvimento do ASPEN (9,15,16,17), que pode ser considerado o primeiro sistema com estrutura "plex" de dados. Outra referência à utilização dessa estrutura para operações com sólidos pode ser encontrada no Trabalho de Neville e Seider (18). Nesse caso foi utilizada uma combinação dos simuladores Flowtran (19) e Plexys (14).

4. Descrição do programa executivo

A linguagem escolhida para a implementação do programa executivo foi o FORTRAN IV, que por ser a linguagem computacional para fins científicos mais difundida, facilita a adaptação do programa para diferentes computadores. O programa aqui descrito, o programa gerado por ele, e os arquivos de dados usados por ambos os programas, integram um sistema computacional, cujo esquema geral de funcionamento pode ser mais facilmente explicado com auxílio da Figura 1.

Figura 1 - Esquema geral do funcionamento do sistema de simulação



4.1. Descrição geral do funcionamento

O programa executivo lê os dados de entrada em um arquivo específico para cada problema. Os dados são revistos em dois aspectos: sua correção sintática e sua coerência. Uma sintaxe para a entrada de dados foi definida e o programa executivo verifica a sua obediência, enviando advertências ou parando a execução nos casos em que é violada. A verificação da coerência é feita para que informações inconsistentes não sejam aceitas pelo programa.

Se o programa executivo termina a análise dos dados de entrada sem encontrar erros graves que parem a execução, os dados, organizados, são gravados em arquivos em disco e um programa FORTRAN de simulação é gerado automaticamente. Este segundo programa é específico para executar os cálculos de cada processo distinto, da mesma forma em que também são específicos os arquivos em disco que guardam a estrutura de dados.

O programa gerado é compilado e, em seguida, executado. Durante a execução, os dados arquivados são lidos, os cálculos efetuados, e os novos dados obtidos gravados nos arquivos. Um relatório final é emitido, e se o programa identifica algum erro na execução, um diagnóstico é impresso.

A geração de um segundo programa possibilita um dimensionamento exato das variáveis e uma redução na alocação de memória para o código compilado, uma vez que as subrotinas dos módulos não existentes no processo não são anexadas.

4.2. A estrutura de dados

Pelas vantagens apresentadas no item 3 optou-se pela estrutura de listas encadeadas de dados. As informações são armazenadas em três vetores. No primeiro vetor são armazenadas informações de caráter geral, os componentes presentes no processo e as classes de correntes definidas. No segundo vetor ficam as correntes, enquanto no terceiro, as informações das unidades.

Nos vetores, um número limitado de informações ocupa posições fixas, que servem para orientar a busca dos dados que ocupam posições diferentes em cada processo simulado. A procura das informações é feita com o auxílio de ponteiros. Os ponteiros fornecem a posição onde um determinado dado é encontrado no vetor, ou seja, o ponteiro é o índice da posição onde o dado se encontra.

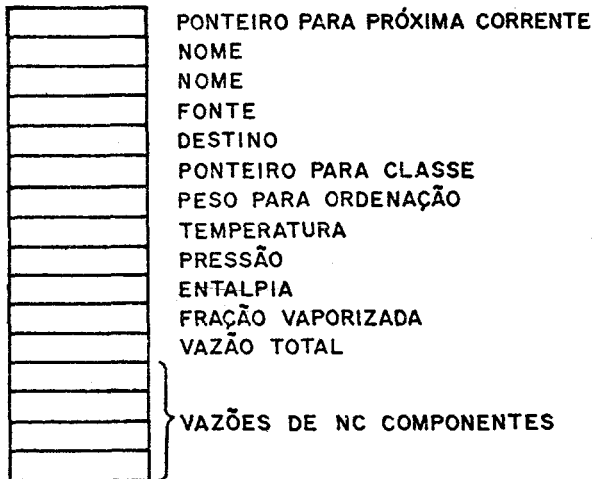
4.2.1. Tipos e Classes de correntes

Na estrutura "Plex" as correntes não são obrigatoriamente representadas por perfis idênticos; são reservadas posições de memória para componentes, faixas granulométricas e propriedades específicas para cada corrente. Essa multiplicidade de perfis dificulta a busca de informações, que só pode ser levada a efeito mediante o conhecimento do tipo e da classe de cada corrente.

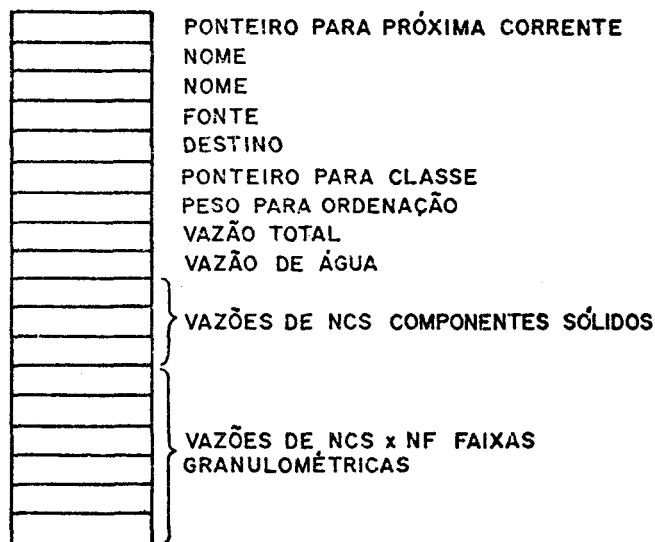
No atual estado do programa as correntes podem ser classificadas em dois tipos: "VL1", para correntes líquido-vapor, e "SL1", para correntes sólido-líquido. As primeiras são típicas dos processos químicos, enquanto as outras dos processos de tratamento de minérios. Na Figura 2 estão representados os perfis de correntes característicos dos dois tipos de correntes.

Figura 2 - Perfis das correntes dos tipos VL1 e SL1

TIPO VL1



TIPO SL1



NC — NÚMERO DE COMPONENTES
 NCS — NÚMERO DE COMPONENTES SÓLIDOS
 NF — NÚMERO DE FAIXAS GRANULOMÉTRICAS

Enquanto a inclusão de um novo tipo de classe de correntes exige prévia adaptação do sistema, uma classe de corrente é definida pelo próprio usuário. Duas correntes da mesma classe apresentam o mesmo perfil, ou seja, são do mesmo tipo, apresentam os mesmos componentes, e as mesmas faixas granulométricas.

No vetor das unidades são armazenadas informações sobre os parâmetros das mesmas e sobre as interligações do processo, ou seja, correntes de entrada e saída de cada unidade.

4.3. A leitura dos dados de entrada e a geração automática

Foi desenvolvida uma pseudolinguagem de entrada dos dados com o objetivo de permitir uma fácil utilização do programa, e ao mesmo tempo restringir as possibilidades de uso inadequado. Nesse sentido, duas providências foram tomadas:

- . Redução a necessidade de formatação dos dados de entrada;
- . Realização de verificações nos dados de entrada para evitar que informações erradas ou incompletas sejam aceitas como verdadeiras pelo programa.

O programa executivo utiliza arquivos auxiliares, onde são guardados nomes de usuários, palavras-chaves que podem estar presentes no arquivo de entrada de dados, mensagens de erros, tipos de correntes e unidades disponíveis, e parâmetros de substâncias para cálculo de propriedades termodinâmicas.

A leitura do arquivo de dados é iniciada pelo nome do usuário e a verificação se ele está autorizado a utilizar o programa. A seguir, opcionalmente, são lidos os sistemas de unidades para a entrada dos dados e para o relatório de saída. Se omitidos, o programa admite o Sistema Internacional de Unidades (S.I.). Internamente todos os cálculos são feitos no S.I. e as transformações de unidades são feitas utilizando um programa desenvolvido por Medeiros, Escola de Química-UFRJ.

Em seguida, deve ser definido se o problema a ser simulado é novo ou modificação de outro simulado imediatamente antes. Na versão atual do programa, todos os problemas devem ser especificados como problemas novos.

Na etapa seguinte, é lido e armazenado o título do problema. O próximo passo é a entrada dos componentes presentes no problema. Para cada componente definido, são buscados os parâmetros para o cálculo de propriedades termodinâmicas. Caso queira evitar esta busca, como ocorre, em geral, nos balanços de massa de circuitos de tratamento de minérios, coloca-se o símbolo "*" antes do nome do componente.

O bloco a seguir é para a leitura das classes de correntes definidas pelo usuário. Devem ser especificados, para cada classe, o nome, o tipo e os componentes. Para as classes de correntes cujo tipo é SL1, é necessário, ainda, especificar quais os componentes líquidos, e as faixas granulométricas.

A leitura das correntes presentes é feita classe por classe. Especifica-se a classe no arquivo de entrada e a seguir as correntes que pertencem a essa classe. A reserva de espaços de memória para cada corrente dependerá da classe à qual pertença.

Em seguida, são feitas especificações para correntes e passa-se ao bloco para leitura das unidades. São lidos o nome, o tipo e as entradas e saídas de cada unidade. A leitura dos parâmetros das unidades é feita por subrotinas específicas para cada uma.

Terminada a leitura do arquivo de entrada e organizadas as informações, o executivo inicia a geração do programa de simulação.

A seqüência em que as unidades serão calculadas pode ser estabelecida pelo usuário, ou determinada automaticamente pelo próprio executivo, através de subrotinas específicas para esse fim. Tanto o programa de simulação quanto a estrutura de dados são gravados em arquivos específicos.

5. Resultados e conclusões

Dois exemplos de geração automática de programas de simulação e estruturas de dados são apresentados a seguir.

No primeiro, cujo fluxograma está na Figura 3, são apresentados o arquivo de entrada de dados, o programa gerado e os vetores "plex" montados (Anexo 1). No segundo exemplo, fluxograma da Figura 4, apresenta-se o arquivo de entrada de dados e o programa gerado (Anexo 2) para um circuito com recírculos. Foram escolhidas as correntes 5 e 6 como correntes de abertura, introduzindo-se nesses pontos unidades de convergência.

Figura 3 - Fluxograma do circuito um

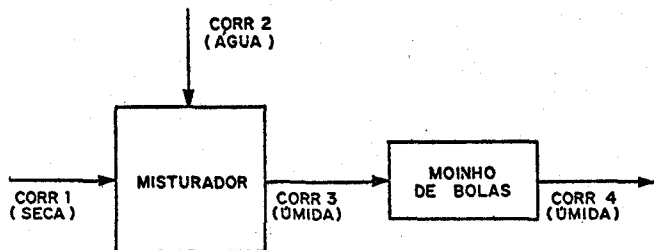
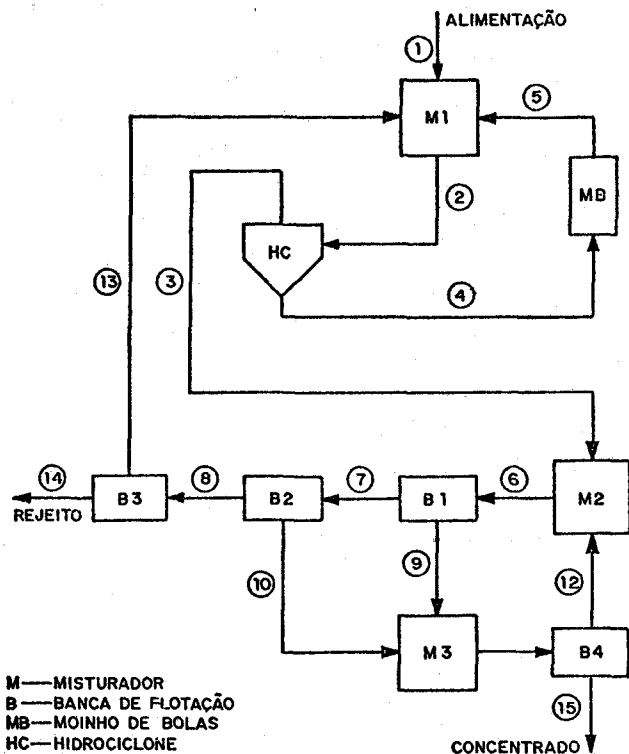


Figura 4 - Fluxograma do circuito dois



M—MISTURADOR
B—BANCA DE FLOTAÇÃO
MB—MOINHO DE BOLAS
HC—HIDROCICLONE

Pode-se observar que o único argumento passado para cada subrotina de simulação de unidades é a posição inicial da unidade no vetor onde a mesma está armazenada. As demais informações são acessadas pela lógica interna dos ponteiros. Verifica-se em ambos os casos, que a presença de um programa executivo torna a simulação mais simples para o usuário, na medida em que exige menor esforço de programação.

6. Bibliografia

- (01) RIQUELME, G. D. & VILLANUEVA, R. A. Simulacion de plantas de concentracion de minerales. In: ENCONTRO DO HEMISFÉRIO SUL SOBRE TECNOLOGIA MINERAL, 1. Rio de Janeiro, Dezembro/82.
- (02) MOTARD, R. L.; SHACHAM, M.; ROSEN, E. M. Steady state chemical process simulation. AICHE Journal, (3):417, 1975.
- (03) RAVICZ, A. E. & NORMAN, R. L. Heat and mass balancing on a digital computer. Chem. Eng. Prog., 60(5):71, 1964.
- (04) EVANS, L. B.; STEWARD, D. G.; SPRAGUE, C. R. Computer-aided chemical process design. Chem. Eng. Prog., 64(4):39, 1969.
- (05) FLOWER, J. R. & WAITE HEAD, B. O. Computer-aided design: A survey of flowsheeting programs - Part II. The Chemical Engineer, 273:271-277, 1973.
- (06) KING, R. P. Simulation of flotation plants. AIME Transactions, 258: 286-293, 1975.
- (07) LAGNITTON, D. & HODOUIN, D. SPOC manual supplement 3-1: Material balance computation for process evaluation and modeling - Course notes for the SPOC workshops. CANMET, Energy, Mines and Resources Canada, Division Report MRP/MSL, 82-104 (IR), 1982.
- (08) EVANS, L. B. & SEIDER, W. D. The requirements of an advanced computing system. Chem. Eng. Prog., 72(6):80-83, 1976.
- (09) EVANS, L. B. & BABU JOSEPH. The plan to develop on advanced computing system for process engineering. In: CONFERENCE ON MATHEMATICAL MODELING OF COAL CONVERSION PROCESSES. Washington, Nov. 1976.
- (10) EVANS, L. B.; JOSEPH, B.; SEIDER, W. D. System structures for process simulation. AICHE Journal, 23(5): 658-666, 1977.

- (11) KAIJAUOTO, K. Experiences of the use of plex data structure in flowsheeting simulation. Comp. Chem. Eng., 3(1-4):307, 1979.
- (12) ROSS, D. T. Generalized technique for symbolic manipulation and numerical calculation. Comm. A.C.M., 4(3): 147-150, 1961.
- (13) PORTER, J. H. Computer-aided process design: An experience in dynamic man-machine communication. In: ANNUAL MEETING OF THE AIChE. Los Angeles, 1969.
- (14) EVANS, L. B. et alii. Plexsys II - Plex data management and plex building program user's manual, second revision. Chemical Engineering Department and Energy Laboratory, MIT, A-26, 1977.
- (15) GALLIER, P. W., et alii. ASPEN: advanced capabilities for modeling and simulation of industrial processes. ACS Symposium Series, 124: 293-308, 1980.
- (16) EVANS, L. B., et alii. ASPEN: an advanced system for process engineering. Comp. Chem. Eng., 3: 319-327, 1979.
- (17) BRITT, H. I., et alii. The ASPEN input language for computer - aided process design. In: INSTRUMENT SOCIETY OF AMERICA (ISA) - FALL CONFERENCE. Anaheim, Oct. 1981.
- (18) NEVILLE, J. M. & SEIDER, W. D. Coal pretreatment-extensions of flowtran to model solids-handling equipment. Comp. Chem. Eng., 4:49-61, 1980.
- (19) ROSEN, E. M. & PAULS, A. C. Computer aided chemical process design: The flowtran system. Comp. Chem. Eng., 1(1):11-21, 1977.

Agradecimentos

Agradecimentos a José Luiz Medeiros por sua colaboração, ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e ao Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro.

7. Anexos

7.1. Anexo 1

```

C
C
C
                                     ARQUIVO DE ENTRADA DE DADOS
C
C
USUARIO=GIL OTAVIO
$
$
NOVO
$
$
TITULO
TESTE DO MISTURADOR E DO MOINHO DE BOLA
$
$
COMPONENTES
*AGUA , *MINERIO (MIN)}
$
$
CLASSES DE CORRENTES
$
TIPO=SL1 , NOME=AGUA
COMPONENTES LIQUIDOS
AGUA;
$
TIPO=SL1 , NOME=SECA
COMPONENTES SOLIDOS
MINERIO;
FAIXAS
FA1,FA2,FA3,FA4,FA5;
$
TIPO=SL1 , NOME=UMIDA
COMPONENTES LIQUIDOS
AGUA;
COMPONENTES SOLIDOS
MINERIO;
FAIXAS
FA1,FA2,FA3,FA4,FA5;
$
$
CORRENTES
$
CLASSE=AGUA
CORR2;
$
CLASSE=SECA
CORR1;
$
CLASSE=UMIDA
CORR3,CORR4;
$
$
UNIDADES
$
TIPO=BOLA , NOME=MOINHO1
ENTRADAS= CORR3;
SAIDAS= CORR4;
VF = 100
S = 1.0,0.8,0.7,0.6;
B = 0.4,0.3,0.2,0.1,
$
NOME=MSTR1 , TIPO=MSTR
ENTRADAS= CORR1,CORR2;
SAIDAS= CORR3;
$
$
ESPECIFICACOES
$
CORRENTE = CORR1
FTOTAL = 100
F = 40,28,20,10,2;

```


*
 CORRENTE * CORR2
 FLIQ = 50
 \$
 \$
 SEQUENCIA
 MSTR1,MOINHO1

PROGRAMA GERADO

C
 C
 SSET AUTOBIND
 SBIND=FROM OBJECT/PSPW/=
 SRESET FREE
 SRESET LIST
 C
 FILE 65(KIND=PACK,PACKNAME="PACK.",TITLE="SIS/CHVSAI.",
 1 FILETYPE=7)
 FILE 66(KIND=PACK,PACKNAME="PACK.",TITLE="SIS/SAIMAL.",
 1 FILETYPE=7)
 FILE 76(KIND=PRINTER,MAXRFSIZE=22)
 FILE 79(KIND=PACK,PACKNAME="PACK.",TITLE="ARQ/PLEX1.",
 1 MAXRECSIZE=1,BLOCKSIZE=30,AREASIZE=100)
 FILE 80(KIND=PACK,PACKNAME="PACK.",TITLE="ARQ/PLEX2.",
 1 MAXRECSIZE=1,BLOCKSIZE=30,AREASIZE=100)
 FILE 81(KIND=PACK,PACKNAME="PACK.",TITLE="ARQ/PLEX3.",
 1 MAXRECSIZE=1,BLOCKSIZE=30,AREASIZE=100)
 FILE 56(KIND=PACK,PACKNAME="PACK.",TITLE="SIS/TIPCOR.",
 1 FILETYPE=7)
 FILE 57(KIND=PACK,PACKNAME="PACK.",TITLE="SIS/TIPUNI.",
 1 FILETYPE=7)

C
 DIMENSION V(104), C(52), D(40)
 DIMENSION AREA1(1), AREA2(4)
 COMMON /PLEX1/ V
 COMMON /PLEX2/ C
 COMMON /PLEX3/ D
 COMMON /AREA1/ AREA1
 COMMON /AREA2/ AREA2

C
 CALL CAUECA
 CALL INICIO (0)
 2 CALL MSTR (26)
 1 CALL BOLA (4)
 CALL RELATO
 STOP
 END

VETOR PLEX1

*		52	
1	104.0	53	SL1
2	105.0	54	0.
3	1.000	56	5.000
4	9.000	57	1.000
5	20.00	58	0.
6	38.00	59	6.000
7	-1.000	60	1.000
8	-1.000	61	2.000
9	10.00	62	FA1
10	TEST	63	
11	E DO	64	
12	MOI	65	FA2
13	TURA	66	
14	DOR	67	
15	E DO	68	FA3
16	MOI	69	
17	NHO	70	
18	DE B	71	FA4
19	OLA	72	
20	29.00	73	
21		74	FA5
22	AGUA	75	
23		76	
24		77	-1.000
25		78	UMID
26		79	A
27		80	SL1
28	1.000	81	
29	-1.000	82	1.000
30	MIN	83	5.000
31	MINE	84	1.000
32	RIO	85	0.

33		86	7.000
34		87	1.000
35		88	1.000
36		89	2.000
37	2.000	90	FA1
38	50.00	91	
39	AGUA	92	
40		93	FA2
41	SL1	94	
42		95	
43	1.000	96	FA3
44	0.	97	
45	0.	98	
46	0.	99	FA4
47	1.000	100	
48	1.000	101	
49	1.000	102	FA5
50	77.00	103	
51	SECA	104	

VETOR PLEX2

*	
1	52.00
2	53.00
3	4.000
4	12.00
5	CORR
6	2
7	0.
8	26.00
9	38.00
10	1.000
11	50.00
12	25.00
13	CORR
14	1
15	0.
16	26.00
17	50.00
18	6.000
19	100.0
20	40.00
21	28.00
22	20.00
23	10.00
24	2.000
25	39.00
26	CORR
27	3
28	26.00
29	4.000
30	77.00
31	7.000
32	0.
33	0.
34	0.
35	0.
36	0.
37	0.
38	0.
39	-1.000
40	CORR
41	4
42	4.000
43	0.
44	77.00
45	7.000
46	0.
47	0.
48	0.
49	0.
50	0.
51	0.
52	0.

VETOR PLEX3

*	
1	40.00
2	41.00
3	4.000
4	26.00
5	MOIN
6	HO1
7	BOLA
8	
9	1.000
10	13.00
11	15.00
12	17.00
13	1.000
14	25.00
15	1.000
16	39.00
17	100.0
18	1.000
19	0.8000
20	0.7000
21	0.6000
22	0.4000
23	0.3000
24	0.2000
25	0.1000
26	-1.000
27	MSTR
28	1
29	MSTR
30	
31	2.000
32	35.00
33	38.00
34	40.00
35	2.000
36	12.00
37	4.000
38	1.000
39	25.00
40	0.

7 2. Anexo 2

```
C
C
C
      ARQUIVO DE ENTRADA DE DADOS
C
C
USUARIO=GIL OTAVIO
$
NOVO
$
$
TITULO
TESTE DA SEQUENCIA DE CALCULO E DO PROGRAMA GERADO
$
$
COMPONENTES
*AGUA,*MINERAL1,*MINERAL2;
$
$
CLASSES DE CORRENTES
$
TIPO=SL1 , NOME=UMIDA2
COMPONENTES LIQUIDOS
AGUA;
COMPONENTES SOLIDOS
MINERAL1,MINERAL2;
FAIXAS
FA1,FA2,FA3,FA4,FA5;
$
$
CORRENTES
$
CLASSE=UMIDA2
CORR1,CORR2,CORR3,CORR4,CORR5,CORR5A,
CORR6,CORR6A,CORR7,CORR8,CORR9,CORR10,
CORR11,CORR12,CORR13,CORR14,CORR15;
$
$
UNIDADES
$
TIPO=MSTR , NOME=MSTR1
ENTRADAS= CORR1,CORR5A,CORR13;
SAIDAS= CORR2;
$
TIPO=HCICL , NOME=HCICL1
ENTRADAS= CORR2;
SAIDAS=CORR3,CORR4;
$
TIPO=BOLA , NOME=MOINHO1
ENTRADAS= CORR4;
SAIDAS= CORR5;
$
TIPO=MSTR , NOME=MSTR2
ENTRADAS= CORR3,CORR12;
SAIDAS= CORR6;
$
TIPO=BANCA , NOME=BANCA1
ENTRADAS= CORR6A;
SAIDAS=CORR7,CORR9;
$
TIPO=BANCA , NOME=BANCA2
ENTRADAS= CORR7;
SAIDAS= CORR8,CORR10;
$
TIPO=BANCA , NOME=BANCA3
ENTRADAS= CORR8;
SAIDAS= CORR13,CORR14;
$
TIPO=BANCA , NOME=BANCA4
ENTRADAS= CORR11;
SAIDAS= CORR12,CORR15;
$
TIPO=SUBSUC , NOME=SUBSUC1
ENTRADAS=CORR5;
SAIDAS=CORR5A;
$
TIPO=SUBSUC , NOME=SUBSUC2
ENTRADAS=CORR6;
SAIDAS=CORR6A;
```

```
$
TIPO=MSTR , NOME=MSTR3
ENTRADAS= CORR9,CORR10;
SAIDAS= CORR11;
$
$
SEQUENCIA
BANCA1,BANCA2,MSTR3,BANCA4,BANCA3,MSTR1,
HCICL1,MOINHO1,SUBSUC1,MSTR2,SUBSUC2;
```