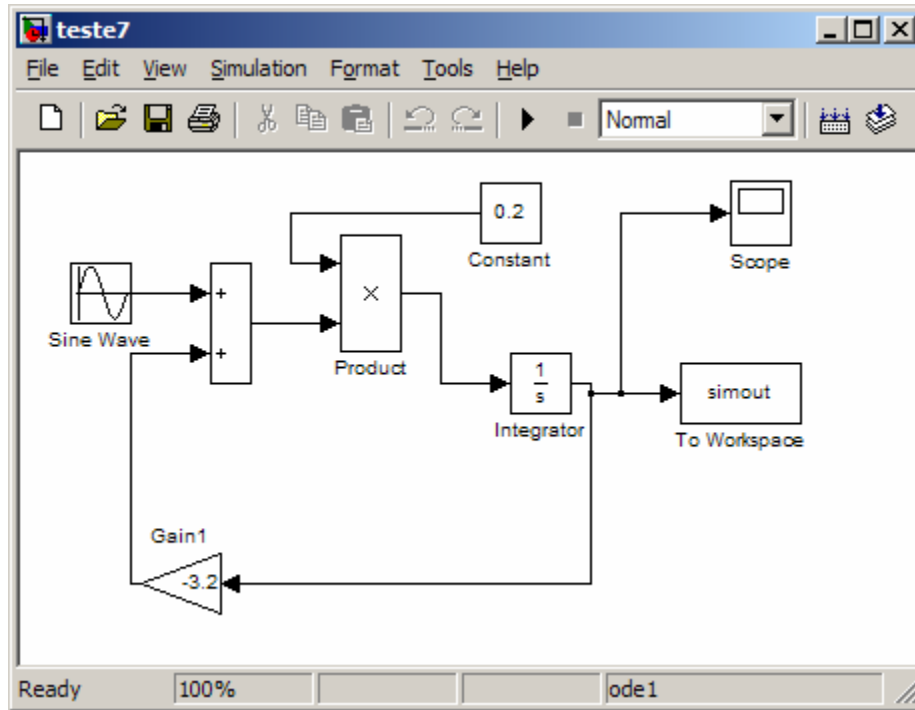


DEBUG gerado a partir da versão do dia 29/11/2004

→ Sistema 07 (modelo SIMULINK):



→ Sistema 07 (modelo BLOCKSIM):

```
5 0.0001
0 0 INPUT 480
0 1 SUM 2
5 1 SUM 2
2 2 CONST 0.2
1 3 MULTIPLY 2
2 3 MULTIPLY 2
3 4 INTEGRATOR 2
4 5 GAIN -3.2
```

→ DEBUG

Zerando matrizes e vetores...

Vetor nfactors[] recebe -1.

Matrizes typeelemmtx[][] e tobeprocessed[][] recebem -1.

Demais vetores e matrizes recebem 0.

Contadores auxiliares zerados.

Lendo linha 1 do arquivo.

Lendo parametro 1 da linha 1 do arquivo.

Lendo parametro 2 da linha 1 do arquivo.

Lendo parametro 3 da linha 1 do arquivo.

Lendo parametro 4 da linha 1 do arquivo.

Marcando a existencia do noh 0.

Somando mais um noh ao total: TOTAL 1 nohs

-> Gerando condicoes iniciais do sistema.

...Achada variavel de INPUT.

...Valor typeelemmtx[0][0] = TYPE_INPUT <- Marcando tipo de elemento.

...Valor valuemtx[0][0] = 480.000000 <- Condicao inicial de INPUT

...Valor inputs[0][0] = 480.000000 <- condicao inicial de INPUT no noh 0

```

...Valor inputparms[0][0] = 480.000000 <- guarda parametro de INPUT do
noh 0
...nodeinput[1] = 0 <- guarda noh de input
...ninputs++ = 1 <- contabilizando inputs
Lendo linha 2 do arquivo.
Lendo parametro 1 da linha 2 do arquivo.
Lendo parametro 2 da linha 2 do arquivo.
Lendo parametro 3 da linha 2 do arquivo.
Lendo parametro 4 da linha 2 do arquivo.
Marcando a existencia do noh 0.
Somando mais um noh ao total: TOTAL 2 nohs
-> Gerando condicoes iniciais do sistema.
...Achada funcao de SOMATORIA.
...Valor typeelemmtx[0][1] = TYPE_SUM <- Marcando tipo de elemento.
...Valor valuemtx[0][1] = 0 <- Condicao inicial de somatoria
...Valor tobeprocessed[0][1] = 1 <- SOMADOR pode ser processado
...Valor tobeprocessed[1][1] = -2 <- Subtraindo um ponto para cada fator
(input para somador) existente
...nlinks[0] = 1 <- contabiliza numero de links no noh 0
...nfactors[1] = -2 <- numero (negativado) de fatores do somador
(permissao da saida do somador)
Lendo linha 3 do arquivo.
Lendo parametro 1 da linha 3 do arquivo.
Lendo parametro 2 da linha 3 do arquivo.
Lendo parametro 3 da linha 3 do arquivo.
Lendo parametro 4 da linha 3 do arquivo.
Marcando a existencia do noh 5.
Somando mais um noh ao total: TOTAL 3 nohs
-> Gerando condicoes iniciais do sistema.
...Achada funcao de SOMATORIA.
...Valor typeelemmtx[5][1] = TYPE_SUM <- Marcando tipo de elemento.
...Valor valuemtx[5][1] = 0 <- Condicao inicial de somatoria
...Valor tobeprocessed[5][1] = 1 <- SOMADOR pode ser processado
...Valor tobeprocessed[1][1] = -3 <- Subtraindo um ponto para cada fator
(input para somador) existente
...nlinks[5] = 1 <- contabiliza numero de links no noh 5
...nfactors[1] = -3 <- numero (negativado) de fatores do somador
(permissao da saida do somador)
Lendo linha 4 do arquivo.
Lendo parametro 1 da linha 4 do arquivo.
Lendo parametro 2 da linha 4 do arquivo.
Lendo parametro 3 da linha 4 do arquivo.
Lendo parametro 4 da linha 4 do arquivo.
Marcando a existencia do noh 2.
Somando mais um noh ao total: TOTAL 4 nohs
-> Gerando condicoes iniciais do sistema.
...Achada CONSTANTE.
...Valor typeelemmtx[2][2] = TYPE_CONST <- Marcando tipo de elemento.
...nodeconsts[1] = 2 <- guarda noh de constante
...nconsts++ = 1 <- contabilizando constantes
Lendo linha 5 do arquivo.
Lendo parametro 1 da linha 5 do arquivo.
Lendo parametro 2 da linha 5 do arquivo.
Lendo parametro 3 da linha 5 do arquivo.
Lendo parametro 4 da linha 5 do arquivo.
Marcando a existencia do noh 1.
Somando mais um noh ao total: TOTAL 5 nohs
-> Gerando condicoes iniciais do sistema.
...Achada funcao de MULTIPLICACAO.
...Valor typeelemmtx[1][3] = TYPE_MULTIPLY <- Marcando tipo de elemento.
...Valor valuemtx[1][3] = 1 <- Condicao inicial de produtorio
...Valor tobeprocessed[1][3] = 1 <- MULTIPLICADOR pode ser processado
...Valor tobeprocessed[3][3] = -2 <- Subtraindo um ponto para cada fator
(input para produtorio) existente

```

```

...nlinks[1] = 1 <- contabiliza numero de links no noh 1
...nfactors[3] = -2 <- numero (negativado) de fatores do produtorio
(permissao da saida do produtorio)
Lendo linha 6 do arquivo.
Lendo parametro 1 da linha 6 do arquivo.
Lendo parametro 2 da linha 6 do arquivo.
Lendo parametro 3 da linha 6 do arquivo.
Lendo parametro 4 da linha 6 do arquivo.
-> Gerando condicoes iniciais do sistema.
...Achada funcao de MULTIPLICACAO.
...Valor typeelemmtx[2][3] = TYPE_MULTIPLY <- Marcando tipo de elemento.
...Valor valuemtx[2][3] = 1 <- Condicao inicial de produtorio
...Valor tobeprocessed[2][3] = 1 <- MULTIPLICADOR pode ser processado
...Valor tobeprocessed[3][3] = -3 <- Subtraindo um ponto para cada fator
(input para produtorio) existente
...nlinks[2] = 1 <- contabiliza numero de links no noh 2
...nfactors[3] = -3 <- numero (negativado) de fatores do produtorio
(permissao da saida do produtorio)
Lendo linha 7 do arquivo.
Lendo parametro 1 da linha 7 do arquivo.
Lendo parametro 2 da linha 7 do arquivo.
Lendo parametro 3 da linha 7 do arquivo.
Lendo parametro 4 da linha 7 do arquivo.
Marcando a existencia do noh 3.
Somando mais um noh ao total: TOTAL 6 nohs
-> Gerando condicoes iniciais do sistema.
...Achada funcao de INTEGRADOR.
...Valor typeelemmtx[3][4] = TYPE_INTEGRATOR <- Marcando tipo de
elemento.
...Valor integtype[3][4] = 2 <- Tipo do metodo de integracao a ser
utilizado
...Valor tobeprocessed[3][4] = 1 <- INTEGRADOR pode ser processado
...nlinks[3] = 1 <- contabiliza numero de links no noh 3
Lendo linha 8 do arquivo.
Lendo parametro 1 da linha 8 do arquivo.
Lendo parametro 2 da linha 8 do arquivo.
Lendo parametro 3 da linha 8 do arquivo.
Lendo parametro 4 da linha 8 do arquivo.
-> Gerando condicoes iniciais do sistema.
...Achada funcao de GANHO.
...Valor typeelemmtx[4][5] = TYPE_GAIN <- Marcando tipo de elemento.
...Valor valuemtx[4][5] = -3.200000 <- Valor do GANHO
...Valor tobeprocessed[4][5] = 1 <- GANHO pode ser processado
...nlinks[4]++ = 1 <- contabiliza numero de links no noh 4
***** Inicializando t... t = 0 *****
Buffer -> stepbuffer = 1 <- a condicao iniciada foi gravada em
stepbuffer = ZERO
>>> valuemtx:
0480.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
-01 001 -01 -01 -01 -01
-01 -03 -01 001 -01 -01
-01 -01 -01 001 -01 -01
-01 -01 -01 -03 001 -01
-01 -01 -01 -01 -01 001
-01 001 -01 -01 -01 -01
Passo t = 0.000000 [s]
Flag de processo = 0 <- inicio do passo de simulacao. Nada pode ser
processado por enquanto.

```

```

Calculando INPUTS baseados no parametro de amplitude fornecido
-> INPUT 0 @ t = 0.000000 : amplitude*sen(2*pi*f*t) = 0.000000
Este valor eh inserido na variavel inputs[nodeinput[i]][stepbuffer].
nodeinput[i] fornece o noh onde INPUT eh aplicado.
Inicia laco FOR para insercao de INPUTS e CONSTS. Sao estes valores que
permitem o inicio da execucao dos demais processamentos.
>>> valuemtx:
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 -1 +1 -1 -1
-1 -1 -1 -3 +1 -1
-1 -1 -1 -1 -1 +1
-1 +1 -1 -1 -1 -1
Atualiza matriz valuemtx[0][0] = valor de input no noh 0 = 0.000000
Atualiza matriz tobeprocessed[0][0] = nlinks[0] = 1 <- o noh jah possui
um valor. A matriz tobeprocessed recebe o numero de links a serem
processados.
Flag de processo = 1 <- pelo menos um noh tem valor. Talvez seja
possivel executar uma operacao com este valor.
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 -1 +1 -1 -1
-1 -1 -1 -3 +1 -1
-1 -1 -1 -1 -1 +1
-1 +1 -1 -1 -1 -1
SOMADOR:
Atualiza matriz tobeprocessed[0][1] = 1
Atualiza matriz tobeprocessed[1][1] = nfactors[resultnode] = -3 <- noh
resultante do somador possui n-fatores a serem utilizados
>>> valuemtx:
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 -1 +1 -1 -1
-1 -1 -1 -3 +1 -1
-1 -1 -1 -1 -1 +1
-1 +1 -1 -1 -1 -1
MULTIPLICADOR:
Atualiza matriz tobeprocessed[1][3] = 1
Atualiza matriz tobeprocessed[3][3] = nfactors[resultnode] = -3 <- noh
resultante do multiplicador possui n-fatores a serem utilizados
>>> valuemtx:
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1

```

```

-1 -3 -1 +1 -1 -1
-1 -1 +1 +1 -1 -1
-1 -1 -1 -3 +1 -1
-1 -1 -1 -1 -1 +1
-1 +1 -1 -1 -1 -1
Atualiza matriz valuemtx[2][2] = valor de constante no noh 2 = 0.000000
Atualiza matriz tobeprocessed[2][2] = nlinks[2] = 1 <- o noh jah possui
um valor. A matriz tobeprocessed recebe o numero de links a serem
processados.
Flag de processo = 1 <- pelo menos um noh tem valor. Talvez seja
possivel executar uma operacao com este valor.
>>> valuemtx:
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 +1 +1 -1 -1
-1 -1 -1 -3 +1 -1
-1 -1 -1 -1 -1 +1
-1 +1 -1 -1 -1 -1
MULTIPLICADOR:
Atualiza matriz tobeprocessed[2][3] = 1
Atualiza matriz tobeprocessed[3][3] = nfactors[resultnode] = -3 <- noh
resultante do multiplicador possui n-fatores a serem utilizados
INTEGRADOR:
Atualiza matriz tobeprocessed[3][4] = 1
Flag de processo = 1 <- INTEGRADOR gera valor usando condicao inicial
ou valores historicos (buffer). Talvez seja possivel executar uma
operacao com este valor.
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 +1 +1 -1 -1
-1 -1 -1 -3 +1 -1
-1 -1 -1 -1 -1 +1
-1 +1 -1 -1 -1 -1
SOMADOR:
Atualiza matriz tobeprocessed[5][1] = 1
Atualiza matriz tobeprocessed[1][1] = nfactors[resultnode] = -3 <- noh
resultante do somador possui n-fatores a serem utilizados
Inicio do processamento para o passo de integracao atual.
Flag de processamento deve ser 1 para iniciar o processo.
Dentro do WHILE. Executar enquanto houver necessidade de processamento.
Flag de processamento eh ZERADO.
Processando elemento SOMATORIA em [0][1]
// Calcula-se o valor do resultado parcial da somatória
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]+valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// o valor de processos atingir -1.
if (tobeprocessed[resultnode][resultnode]==-1) {

```

```

        // Este nó resultado possui X processamentos a dever
        tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
    }
>>> valuemtx:
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +1 +1 -1 -1
-1 -1 -1 -3 +1 -1
-1 -1 -1 -1 -1 +1
-1 +1 -1 -1 -1 -1
Processando elemento PRODUTORIO em [2][3]
// Calcula-se o valor do resultado parcial da multiplicação
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]*valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// atingir o valor -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +1 -1
-1 -1 -1 -1 -1 +1
-1 +1 -1 -1 -1 -1
Processando elemento INTEGRADOR em [3][4]
// Se o bloco pode ser processado
// (INTEGRADOR não precisa ter o nó de entrada liberado
// uma vez que lida com valores históricos)
if (tobeprocessed[actualnode][resultnode] > 0) {
    // O nó resultado é liberado para processamento.
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
    // O bloco foi processado.
    tobeprocessed[actualnode][resultnode] = 0;
>>> valuemtx:
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000

```

```

0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +0 -1
-1 -1 -1 -1 +1 +1
-1 +1 -1 -1 -1 -1
Processando elemento GANHO em [4][5]
// O nó resultado é liberado para processamento.
// Este nó resultado possui X processamentos a dever
tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Calcula-se o valor do resultado
valuemtx[resultnode][resultnode] =
valuemtx[actualnode][resultnode]*valuemtx[actualnode][actualnode];
// Reativa-se o flag de processamento
processflag = 1;
>>> valuemtx:
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +0 -1
-1 -1 -1 -1 +0 +0
-1 +1 -1 -1 -1 +1
Processando elemento SOMATORIA em [5][1]
// Calcula-se o valor do resultado parcial da somatória
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]+valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// o valor de processos atingir -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 +1 -1 +1 -1 -1

```

```

-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +0 -1
-1 -1 -1 -1 +0 +0
-1 +0 -1 -1 -1 +0
Dentro do WHILE. Executar enquanto houver necessidade de processamento.
Flag de processamento eh ZERADO.
Processando elemento PRODUTORIO em [1][3]
// Calcula-se o valor do resultado parcial da multiplicação
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]*valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// atingir o valor -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 +0 -1 +0 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 +1 +0 -1
-1 -1 -1 -1 +0 +0
-1 +0 -1 -1 -1 +0
Dentro do WHILE. Executar enquanto houver necessidade de processamento.
Flag de processamento eh ZERADO.
Passo t = 0.000100 [s]
Flag de processo = 0 <- inicio do passo de simulacao. Nada pode ser
processado por enquanto.
Calculando INPUTS baseados no parametro de amplitude fornecido
-> INPUT 0 @ t = 0.000100 : amplitude*sen(2*pi*f*t) = 18.091288
Este valor eh inserido na variavel inputs[nodeinput[i]][stepbuffer].
nodeinput[i] fornece o nohonde INPUT eh aplicado.
Inicia laco FOR para insercao de INPUTS e CONSTS. Sao estes valores que
permitem o inicio da execucao dos demais processamentos.
>>> valuemtx:
0018.0913 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+1 -1 -1 -1 -1 -1
-1 -3 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
Atualiza matriz valuemtx[0][0] = valor de input no noh 0 = 18.091288

```


Atualiza matriz tobeprocessed[0][0] = nlinks[0] = 1 <- o noh jah possui um valor. A matriz tobeprocessed recebe o numero de links a serem processados.

Flag de processo = 1 <- pelo menos um noh tem valor. Talvez seja possivel executar uma operacao com este valor.

>>> tobeprocessed:

```
+1 +1 -1 -1 -1 -1
-1 -3 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
```

SOMADOR:

Atualiza matriz tobeprocessed[0][1] = 1

Atualiza matriz tobeprocessed[1][1] = nfactors[resultnode] = -3 <- noh resultante do somador possui n-fatores a serem utilizados

>>> valuemtx:

```
0018.0913 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
```

>>> tobeprocessed:

```
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
```

MULTIPLICADOR:

Atualiza matriz tobeprocessed[1][3] = 1

Atualiza matriz tobeprocessed[3][3] = nfactors[resultnode] = -3 <- noh resultante do multiplicador possui n-fatores a serem utilizados

>>> valuemtx:

```
0018.0913 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
```

>>> tobeprocessed:

```
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 +1 -1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
```

Atualiza matriz valuemtx[2][2] = valor de constante no noh 2 = 0.000000

Atualiza matriz tobeprocessed[2][2] = nlinks[2] = 1 <- o noh jah possui um valor. A matriz tobeprocessed recebe o numero de links a serem processados.

Flag de processo = 1 <- pelo menos um noh tem valor. Talvez seja possivel executar uma operacao com este valor.

>>> valuemtx:

```
0018.0913 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
```

>>> tobeprocessed:

```
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
```

```

-1 -1 +1 +1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
MULTIPLICADOR:
Atualiza matriz tobeprocessed[2][3] = 1
Atualiza matriz tobeprocessed[3][3] = nfactors[resultnode] = -3 <- noh
resultante do multiplicador possui n-fatores a serem utilizados
INTEGRADOR:
Atualiza matriz tobeprocessed[3][4] = 1
Flag de processo = 1 <- INTEGRADOR gera valor usando condicao inicial
ou valores historicos (buffer). Talvez seja possivel executar uma
operacao com este valor.
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 +1 +1 -1 -1
-1 -1 -1 -3 +1 -1
-1 -1 -1 -1 -1 -1
-1 +1 -1 -1 -1 -1
SOMADOR:
Atualiza matriz tobeprocessed[5][1] = 1
Atualiza matriz tobeprocessed[1][1] = nfactors[resultnode] = -3 <- noh
resultante do somador possui n-fatores a serem utilizados
Inicio do processamento para o passo de integracao atual.
Flag de processamento deve ser 1 para iniciar o processo.
Dentro do WHILE. Executar enquanto houver necessidade de processamento.
Flag de processamento eh ZERADO.
Processando elemento SOMATORIA em [0][1]
// Calcula-se o valor do resultado parcial da somatória
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]+valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// o valor de processos atingir -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0018.0913 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0018.0913 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +1 +1 -1 -1
-1 -1 -1 -3 +1 -1
-1 -1 -1 -1 -1 -1
-1 +1 -1 -1 -1 -1
Processando elemento PRODUTORIO em [2][3]
// Calcula-se o valor do resultado parcial da multiplicação
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]*valuemtx[actualnode][actualnode];

```

```

// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// atingir o valor -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0018.0913 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0018.0913 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +1 -1
-1 -1 -1 -1 -1 -1
-1 +1 -1 -1 -1 -1
Processando elemento INTEGRADOR em [3][4]
// Se o bloco pode ser processado
// (INTEGRATOR não precisa ter o nó de entrada liberado
// uma vez que lida com valores históricos)
if (tobeprocessed[actualnode][resultnode] > 0) {
    // O nó resultado é liberado para processamento.
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
    // O bloco foi processado.
    tobeprocessed[actualnode][resultnode] = 0;
}
>>> valuemtx:
0018.0913 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0018.0913 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +0 -1
-1 -1 -1 -1 +1 -1
-1 +1 -1 -1 -1 -1
Processando elemento GANHOS em [4][5]
// O nó resultado é liberado para processamento.
// Este nó resultado possui X processamentos a dever
tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Calcula-se o valor do resultado
valuemtx[resultnode][resultnode] =
valuemtx[actualnode][resultnode]*valuemtx[actualnode][actualnode];
// Reativa-se o flag de processamento

```

```

processflag = 1;
>>> valuemtx:
0018.0913 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0018.0913 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +0 -1
-1 -1 -1 -1 +0 +0
-1 +1 -1 -1 -1 +1
Processando elemento SOMATORIA em [5][1]
// Calcula-se o valor do resultado parcial da somatória
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]+valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// o valor de processos atingir -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0018.0913 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0018.0913 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 +1 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +0 -1
-1 -1 -1 -1 +0 +0
-1 +0 -1 -1 -1 +0
Dentro do WHILE. Executar enquanto houver necessidade de processamento.
Flag de processamento eh ZERADO.
Processando elemento PRODUTORIO em [1][3]
// Calcula-se o valor do resultado parcial da multiplicação
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]*valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// atingir o valor -1.
if (tobeprocessed[resultnode][resultnode]==-1) {

```

```

        // Este nó resultado possui X processamentos a dever
        tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
    }
>>> valuemtx:
0018.0913  0000.0000  0000.0000  0000.0000  0000.0000  0000.0000
0000.0000  0018.0913  0000.0000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.2000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0003.6183  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0000.0000  0000.0000  -003.2000
0000.0000  0000.0000  0000.0000  0000.0000  0000.0000  0000.0000
>>> tobeprocessed:
+0  +0  -1  -1  -1  -1
-1  +0  -1  +0  -1  -1
-1  -1  +0  +0  -1  -1
-1  -1  -1  +1  +0  -1
-1  -1  -1  -1  +0  +0
-1  +0  -1  -1  -1  +0
Dentro do WHILE. Executar enquanto houver necessidade de processamento.
Flag de processamento eh ZERADO.
Passo t = 0.000200 [s]
Flag de processo = 0 <- inicio do passo de simulacao. Nada pode ser
processado por enquanto.
Calculando INPUTS baseados no parametro de amplitude fornecido
-> INPUT 0 @ t = 0.000200 : amplitude*sen(2*pi*f*t) = 36.156867
Este valor eh inserido na variavel inputs[nodeinput[i]][stepbuffer].
nodeinput[i] fornece o noh onde INPUT eh aplicado.
Inicia laco FOR para insercao de INPUTS e CONSTS. Sao estes valores que
permitem o inicio da execucao dos demais processamentos.
>>> valuemtx:
0036.1569  0000.0000  0000.0000  0000.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0000.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0000.0000  0000.0000  -003.2000
0000.0000  0000.0000  0000.0000  0000.0000  0000.0000  0000.0000
>>> tobeprocessed:
+1  -1  -1  -1  -1  -1
-1  -3  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -3  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
Atualiza matriz valuemtx[0][0] = valor de input no noh 0 = 36.156867
Atualiza matriz tobeprocessed[0][0] = nlinks[0] = 1 <- o noh jah possui
um valor. A matriz tobeprocessed recebe o numero de links a serem
processados.
Flag de processo = 1 <- pelo menos um noh tem valor. Talvez seja
possivel executar uma operacao com este valor.
>>> tobeprocessed:
+1  +1  -1  -1  -1  -1
-1  -3  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -3  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
SOMADOR:
Atualiza matriz tobeprocessed[0][1] = 1
Atualiza matriz tobeprocessed[1][1] = nfactors[resultnode] = -3 <- noh
resultante do somador possui n-fatores a serem utilizados
>>> valuemtx:
0036.1569  0000.0000  0000.0000  0000.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0001.0000  0000.0000  0000.0000

```

```

0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
MULTIPLICADOR:
Atualiza matriz tobeprocessed[1][3] = 1
Atualiza matriz tobeprocessed[3][3] = nfactors[resultnode] = -3 <- noh
resultante do multiplicador possui n-fatores a serem utilizados
>>> valuemtx:
0036.1569 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 +1 -1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
Atualiza matriz valuemtx[2][2] = valor de constante no noh 2 = 0.000000
Atualiza matriz tobeprocessed[2][2] = nlinks[2] = 1 <- o noh jah possui
um valor. A matriz tobeprocessed recebe o numero de links a serem
processados.
Flag de processo = 1 <- pelo menos um noh tem valor. Talvez seja
possivel executar uma operacao com este valor.
>>> valuemtx:
0036.1569 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 +1 +1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
MULTIPLICADOR:
Atualiza matriz tobeprocessed[2][3] = 1
Atualiza matriz tobeprocessed[3][3] = nfactors[resultnode] = -3 <- noh
resultante do multiplicador possui n-fatores a serem utilizados
INTEGRADOR:
Atualiza matriz tobeprocessed[3][4] = 1
Flag de processo = 1 <- INTEGRADOR gera valor usando condicao inicial
ou valores historicos (buffer). Talvez seja possivel executar uma
operacao com este valor.
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 +1 +1 -1 -1
-1 -1 -1 -3 +1 -1
-1 -1 -1 -1 -1 -1
-1 +1 -1 -1 -1 -1
SOMADOR:

```

```

Atualiza matriz tobeprocessed[5][1] = 1
Atualiza matriz tobeprocessed[1][1] = nfactors[resultnode] = -3 <- noh
resultante do somador possui n-fatores a serem utilizados
Inicio do processamento para o passo de integracao atual.
Flag de processamento deve ser 1 para iniciar o processo.
Dentro do WHILE. Executar enquanto houver necessidade de processamento.
Flag de processamento eh ZERADO.
Processando elemento SOMATORIA em [0][1]
// Calcula-se o valor do resultado parcial da somatória
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]+valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// o valor de processos atingir -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0036.1569  0000.0000  0000.0000  0000.0000  0000.0000  0000.0000
0000.0000  0036.1569  0000.0000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.2000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0000.0000  0000.0000  -003.2000
0000.0000  0000.0000  0000.0000  0000.0000  0000.0000  0000.0000
>>> tobeprocessed:
+0  +0  -1  -1  -1  -1
-1  -2  -1  +1  -1  -1
-1  -1  +1  +1  -1  -1
-1  -1  -1  -3  +1  -1
-1  -1  -1  -1  -1  -1
-1  +1  -1  -1  -1  -1
Processando elemento PRODUTORIO em [2][3]
// Calcula-se o valor do resultado parcial da multiplicação
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]*valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// atingir o valor -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0036.1569  0000.0000  0000.0000  0000.0000  0000.0000  0000.0000
0000.0000  0036.1569  0000.0000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.2000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0000.2000  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0000.0000  0000.0000  -003.2000
0000.0000  0000.0000  0000.0000  0000.0000  0000.0000  0000.0000

```

```

>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +1 -1
-1 -1 -1 -1 -1 -1
-1 +1 -1 -1 -1 -1
Processando elemento INTEGRADOR em [3][4]
// Se o bloco pode ser processado
// (INTEGRATOR não precisa ter o nó de entrada liberado
// uma vez que lida com valores históricos)
if (tobeprocessed[actualnode][resultnode] > 0) {
// O nó resultado é liberado para processamento.
// Este nó resultado possui X processamentos a dever
tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
>>> valuemtx:
0036.1569 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0036.1569 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0004 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +0 -1
-1 -1 -1 -1 +1 -1
-1 +1 -1 -1 -1 -1
Processando elemento GANHOS em [4][5]
// O nó resultado é liberado para processamento.
// Este nó resultado possui X processamentos a dever
tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Calcula-se o valor do resultado
valuemtx[resultnode][resultnode] =
valuemtx[actualnode][resultnode]*valuemtx[actualnode][actualnode];
// Reativa-se o flag de processamento
processflag = 1;
>>> valuemtx:
0036.1569 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0036.1569 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0004 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -000.0012
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +0 -1
-1 -1 -1 -1 +0 +0
-1 +1 -1 -1 -1 +1
Processando elemento SOMATORIA em [5][1]
// Calcula-se o valor do resultado parcial da somatória
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]+valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;

```



```

// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// o valor de processos atingir -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0036.1569  0000.0000  0000.0000  0000.0000  0000.0000  0000.0000
0000.0000  0036.1557  0000.0000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.2000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0000.2000  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0000.0000  0000.0004  -003.2000
0000.0000  0000.0000  0000.0000  0000.0000  0000.0000  -000.0012
>>> tobeprocessed:
+0  +0  -1  -1  -1  -1
-1  +1  -1  +1  -1  -1
-1  -1  +0  +0  -1  -1
-1  -1  -1  -2  +0  -1
-1  -1  -1  -1  +0  +0
-1  +0  -1  -1  -1  +0
Dentro do WHILE. Executar enquanto houver necessidade de processamento.
Flag de processamento eh ZERADO.
Processando elemento PRODUTORIO em [1][3]
// Calcula-se o valor do resultado parcial da multiplicação
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]*valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// atingir o valor -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0036.1569  0000.0000  0000.0000  0000.0000  0000.0000  0000.0000
0000.0000  0036.1557  0000.0000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.2000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0007.2311  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0000.0000  0000.0004  -003.2000
0000.0000  0000.0000  0000.0000  0000.0000  0000.0000  -000.0012
>>> tobeprocessed:
+0  +0  -1  -1  -1  -1
-1  +0  -1  +0  -1  -1
-1  -1  +0  +0  -1  -1
-1  -1  -1  +1  +0  -1
-1  -1  -1  -1  +0  +0
-1  +0  -1  -1  -1  +0
Dentro do WHILE. Executar enquanto houver necessidade de processamento.
Flag de processamento eh ZERADO.
Passo t = 0.000300 [s]

```

```

Flag de processo = 0 <- inicio do passo de simulacao. Nada pode ser
processado por enquanto.
Calculando INPUTS baseados no parametro de amplitude fornecido
-> INPUT 0 @ t = 0.000300 : amplitude*sen(2*pi*f*t) = 54.171065
Este valor eh inserido na variavel inputs[nodeinput[i]][stepbuffer].
nodeinput[i] fornece o noh onde INPUT eh aplicado.
Inicia laco FOR para insercao de INPUTS e CONSTS. Sao estes valores que
permitem o inicio da execucao dos demais processamentos.
>>> valuemtx:
0054.1711 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+1 -1 -1 -1 -1 -1
-1 -3 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
Atualiza matriz valuemtx[0][0] = valor de input no noh 0 = 54.171065
Atualiza matriz tobeprocessed[0][0] = nlinks[0] = 1 <- o noh jah possui
um valor. A matriz tobeprocessed recebe o numero de links a serem
processados.
Flag de processo = 1 <- pelo menos um noh tem valor. Talvez seja
possivel executar uma operacao com este valor.
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
SOMADOR:
Atualiza matriz tobeprocessed[0][1] = 1
Atualiza matriz tobeprocessed[1][1] = nfactors[resultnode] = -3 <- noh
resultante do somador possui n-fatores a serem utilizados
>>> valuemtx:
0054.1711 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
MULTIPLICADOR:
Atualiza matriz tobeprocessed[1][3] = 1
Atualiza matriz tobeprocessed[3][3] = nfactors[resultnode] = -3 <- noh
resultante do multiplicador possui n-fatores a serem utilizados
>>> valuemtx:
0054.1711 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000

```

```

>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 +1 -1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
Atualiza matriz valuemtx[2][2] = valor de constante no noh 2 = 0.000000
Atualiza matriz tobeprocessed[2][2] = nlinks[2] = 1 <- o noh jah possui
um valor. A matriz tobeprocessed recebe o numero de links a serem
processados.
Flag de processo = 1 <- pelo menos um noh tem valor. Talvez seja
possivel executar uma operacao com este valor.
>>> valuemtx:
0054.1711 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 +1 +1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
MULTIPLICADOR:
Atualiza matriz tobeprocessed[2][3] = 1
Atualiza matriz tobeprocessed[3][3] = nfactors[resultnode] = -3 <- noh
resultante do multiplicador possui n-fatores a serem utilizados
INTEGRADOR:
Atualiza matriz tobeprocessed[3][4] = 1
Flag de processo = 1 <- INTEGRADOR gera valor usando condicao inicial
ou valores historicos (buffer). Talvez seja possivel executar uma
operacao com este valor.
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 +1 +1 -1 -1
-1 -1 -1 -3 +1 -1
-1 -1 -1 -1 -1 -1
-1 +1 -1 -1 -1 -1
SOMADOR:
Atualiza matriz tobeprocessed[5][1] = 1
Atualiza matriz tobeprocessed[1][1] = nfactors[resultnode] = -3 <- noh
resultante do somador possui n-fatores a serem utilizados
Inicio do processamento para o passo de integracao atual.
Flag de processamento deve ser 1 para iniciar o processo.
Dentro do WHILE. Executar enquanto houver necessidade de processamento.
Flag de processamento eh ZERADO.
Processando elemento SOMATORIA em [0][1]
// Calcula-se o valor do resultado parcial da somatória
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]+valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se

```

```

// o valor de processos atingir -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0054.1711 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0054.1711 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +1 +1 -1 -1
-1 -1 -1 -3 +1 -1
-1 -1 -1 -1 -1 -1
-1 +1 -1 -1 -1 -1
Processando elemento PRODUTORIO em [2][3]
// Calcula-se o valor do resultado parcial da multiplicação
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]*valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// atingir o valor -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0054.1711 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0054.1711 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +1 -1
-1 -1 -1 -1 -1 -1
-1 +1 -1 -1 -1 -1
Processando elemento INTEGRADOR em [3][4]
// Se o bloco pode ser processado
// (INTEGRADOR não precisa ter o nó de entrada liberado
// uma vez que lida com valores históricos)
if (tobeprocessed[actualnode][resultnode] > 0) {
    // O nó resultado é liberado para processamento.
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
    // O bloco foi processado.
    tobeprocessed[actualnode][resultnode] = 0;
>>> valuemtx:
0054.1711 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0054.1711 0000.0000 0001.0000 0000.0000 0000.0000

```

```

0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0011 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +0 -1
-1 -1 -1 -1 +1 -1
-1 +1 -1 -1 -1 -1
Processando elemento GANHO em [4][5]
// O nó resultado é liberado para processamento.
// Este nó resultado possui X processamentos a dever
tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Calcula-se o valor do resultado
valuemtx[resultnode][resultnode] =
valuemtx[actualnode][resultnode]*valuemtx[actualnode][actualnode];
// Reativa-se o flag de processamento
processflag = 1;
>>> valuemtx:
0054.1711 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0054.1711 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0011 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -000.0035
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +0 -1
-1 -1 -1 -1 +0 +0
-1 +1 -1 -1 -1 +1
Processando elemento SOMATORIA em [5][1]
// Calcula-se o valor do resultado parcial da somatória
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]+valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// o valor de processos atingir -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0054.1711 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0054.1676 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0011 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -000.0035
>>> tobeprocessed:

```

```

+0 +0 -1 -1 -1 -1
-1 +1 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +0 -1
-1 -1 -1 -1 +0 +0
-1 +0 -1 -1 -1 +0
Dentro do WHILE. Executar enquanto houver necessidade de processamento.
Flag de processamento eh ZERADO.
Processando elemento PRODUTORIO em [1][3]
// Calcula-se o valor do resultado parcial da multiplicação
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]*valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// atingir o valor -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0054.1711 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0054.1676 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0010.8335 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0011 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -000.0035
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 +0 -1 +0 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 +1 +0 -1
-1 -1 -1 -1 +0 +0
-1 +0 -1 -1 -1 +0
Dentro do WHILE. Executar enquanto houver necessidade de processamento.
Flag de processamento eh ZERADO.
Passo t = 0.000400 [s]
Flag de processo = 0 <- inicio do passo de simulacao. Nada pode ser
processado por enquanto.
Calculando INPUTS baseados no parametro de amplitude fornecido
-> INPUT 0 @ t = 0.000400 : amplitude*sen(2*pi*f*t) = 72.108283
Este valor eh inserido na variavel inputs[nodeinput[i]][stepbuffer].
nodeinput[i] fornece o noh onde INPUT eh aplicado.
Inicia laco FOR para insercao de INPUTS e CONSTS. Sao estes valores que
permitem o inicio da execucao dos demais processamentos.
>>> valuemtx:
0072.1083 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+1 -1 -1 -1 -1 -1
-1 -3 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1

```

```

-1 -1 -1 -1 -1 -1
Atualiza matriz valuemtx[0][0] = valor de input no noh 0 = 72.108283
Atualiza matriz tobeprocessed[0][0] = nlinks[0] = 1 <- o noh jah possui
um valor. A matriz tobeprocessed recebe o numero de links a serem
processados.
Flag de processo = 1 <- pelo menos um noh tem valor. Talvez seja
possivel executar uma operacao com este valor.
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
SOMADOR:
Atualiza matriz tobeprocessed[0][1] = 1
Atualiza matriz tobeprocessed[1][1] = nfactors[resultnode] = -3 <- noh
resultante do somador possui n-fatores a serem utilizados
>>> valuemtx:
0072.1083 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
MULTIPLICADOR:
Atualiza matriz tobeprocessed[1][3] = 1
Atualiza matriz tobeprocessed[3][3] = nfactors[resultnode] = -3 <- noh
resultante do multiplicador possui n-fatores a serem utilizados
>>> valuemtx:
0072.1083 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 +1 -1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
Atualiza matriz valuemtx[2][2] = valor de constante no noh 2 = 0.000000
Atualiza matriz tobeprocessed[2][2] = nlinks[2] = 1 <- o noh jah possui
um valor. A matriz tobeprocessed recebe o numero de links a serem
processados.
Flag de processo = 1 <- pelo menos um noh tem valor. Talvez seja
possivel executar uma operacao com este valor.
>>> valuemtx:
0072.1083 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:

```

```

+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 +1 +1 -1 -1
-1 -1 -1 -3 -1 -1
-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
MULTIPLICADOR:
Atualiza matriz tobeprocessed[2][3] = 1
Atualiza matriz tobeprocessed[3][3] = nfactors[resultnode] = -3 <- noh
resultante do multiplicador possui n-fatores a serem utilizados
INTEGRADOR:
Atualiza matriz tobeprocessed[3][4] = 1
Flag de processo = 1 <- INTEGRADOR gera valor usando condicao inicial
ou valores historicos (buffer). Talvez seja possivel executar uma
operacao com este valor.
>>> tobeprocessed:
+1 +1 -1 -1 -1 -1
-1 -3 -1 +1 -1 -1
-1 -1 +1 +1 -1 -1
-1 -1 -1 -3 +1 -1
-1 -1 -1 -1 -1 -1
-1 +1 -1 -1 -1 -1
SOMADOR:
Atualiza matriz tobeprocessed[5][1] = 1
Atualiza matriz tobeprocessed[1][1] = nfactors[resultnode] = -3 <- noh
resultante do somador possui n-fatores a serem utilizados
Inicio do processamento para o passo de integracao atual.
Flag de processamento deve ser 1 para iniciar o processo.
Dentro do WHILE. Executar enquanto houver necessidade de processamento.
Flag de processamento eh ZERADO.
Processando elemento SOMATORIA em [0][1]
// Calcula-se o valor do resultado parcial da somatória
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]+valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// o valor de processos atingir -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0072.1083 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0072.1083 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +1 +1 -1 -1
-1 -1 -1 -3 +1 -1
-1 -1 -1 -1 -1 -1
-1 +1 -1 -1 -1 -1
Processando elemento PRODUTORIO em [2][3]
// Calcula-se o valor do resultado parcial da multiplicação

```



```

valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]*valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// atingir o valor -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0072.1083 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0072.1083 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +1 -1
-1 -1 -1 -1 -1 -1
-1 +1 -1 -1 -1 -1
Processando elemento INTEGRADOR em [3][4]
// Se o bloco pode ser processado
// (INTEGRATOR não precisa ter o nó de entrada liberado
// uma vez que lida com valores históricos)
if (tobeprocessed[actualnode][resultnode] > 0) {
// O nó resultado é liberado para processamento.
// Este nó resultado possui X processamentos a dever
tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
>>> valuemtx:
0072.1083 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0072.1083 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0022 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +0 -1
-1 -1 -1 -1 +1 -1
-1 +1 -1 -1 -1 -1
Processando elemento GANHOS em [4][5]
// O nó resultado é liberado para processamento.
// Este nó resultado possui X processamentos a dever
tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Calcula-se o valor do resultado

```

```

valuemtx[resultnode][resultnode] =
valuemtx[actualnode][resultnode]*valuemtx[actualnode][actualnode];
// Reativa-se o flag de processamento
processflag = 1;
>>> valuemtx:
0072.1083 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0072.1083 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0022 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -000.0069
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 -2 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +0 -1
-1 -1 -1 -1 +0 +0
-1 +1 -1 -1 -1 +1
Processando elemento SOMATORIA em [5][1]
// Calcula-se o valor do resultado parcial da somatória
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]+valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;
// O nó resultado é liberado para processamento somente se
// o valor de processos atingir -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0072.1083 0000.0000 0000.0000 0000.0000 0000.0000 0000.0000
0000.0000 0072.1013 0000.0000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.2000 0001.0000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.2000 0000.0000 0000.0000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0022 -003.2000
0000.0000 0000.0000 0000.0000 0000.0000 0000.0000 -000.0069
>>> tobeprocessed:
+0 +0 -1 -1 -1 -1
-1 +1 -1 +1 -1 -1
-1 -1 +0 +0 -1 -1
-1 -1 -1 -2 +0 -1
-1 -1 -1 -1 +0 +0
-1 +0 -1 -1 -1 +0
Dentro do WHILE. Executar enquanto houver necessidade de processamento.
Flag de processamento eh ZERADO.
Processando elemento PRODUTORIO em [1][3]
// Calcula-se o valor do resultado parcial da multiplicação
valuemtx[resultnode][resultnode] =
valuemtx[resultnode][resultnode]*valuemtx[actualnode][actualnode];
// Subtrai-se um processo a dever do nó de entrada
tobeprocessed[actualnode][actualnode]--;
// O bloco foi processado.
tobeprocessed[actualnode][resultnode] = 0;
// Soma-se um processo realizado para o nó resultado.
tobeprocessed[resultnode][resultnode]++;
// Reativa-se o flag de processamento
processflag = 1;

```

```

// O nó resultado é liberado para processamento somente se
// atingir o valor -1.
if (tobeprocessed[resultnode][resultnode]==-1) {
    // Este nó resultado possui X processamentos a dever
    tobeprocessed[resultnode][resultnode] = nlinks[resultnode];
}
>>> valuemtx:
0072.1083  0000.0000  0000.0000  0000.0000  0000.0000  0000.0000
0000.0000  0072.1013  0000.0000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.2000  0001.0000  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0014.4203  0000.0000  0000.0000
0000.0000  0000.0000  0000.0000  0000.0000  0000.0022  -003.2000
0000.0000  0000.0000  0000.0000  0000.0000  0000.0000  -000.0069
>>> tobeprocessed:
+0  +0  -1  -1  -1  -1
-1  +0  -1  +0  -1  -1
-1  -1  +0  +0  -1  -1
-1  -1  -1  +1  +0  -1
-1  -1  -1  -1  +0  +0
-1  +0  -1  -1  -1  +0
Dentro do WHILE. Executar enquanto houver necessidade de processamento.
Flag de processamento eh ZERADO.

```