

Prova 2

Nome: Jhoanat Helersen , Aluno de Sonda

Matrícula: 20200000880

* CAPÍTULO 3: 2, 4, 6, 8, 13, 16, 17, 19, 20,
22, 23, 27, 28 (13 QUESTÕES)

TOTAL: 13 DE 15

3.2) MODIFIQUE A REGRA TRAPEZOIDAL
DE FORMA QUE ELA ESTIME CORRETAMENTE
A INTEGRAL MESMA QUE comm-sz NÃO
dávida n unicamente. (Você ainda pode assumir
QUE $n \geq comm-sz$) 3. Localizar e AGREGAR de TADAT

PARA O cálculo do TRAPEZIO com
o valor $n \geq comm-sz$ o des. precisamos
que os processos aplique com m número
maior de TRAPEZIOS QUANDO o "comm-sz" não
for dividível por " n ". $n \geq comm-sz$ ditará
a DESTASONIA o PROCESSOS. INICIANDO DO 0.
NÉSSEROS OS PROCESSOS MAIS MAIS RECEBIDA 3 FISTAS.

delle Téc AGASTAMENTO.

3.9) MODIFIQUE O PROGRAMA QUE APENAS
IMPRIMA UMA LINHA DE SAÍDA DE CADA
PROCESSO (MTI - outent.c) Para que a saída
deja IMPRESSA NA orden de classificagões do
PROCESSO; a SAÍDA DO PROCESSO "0"^o PREMIRE,
depois o processo "1"^o e assim por diante

$n\%_{\text{com_sz}} =$

3. 6) Se $\text{com_sz} = 4$ e szpilha que é

é um vetor com $n = 1^{\text{a}}$ componentes. ~~(a)~~ como

os componentes de x seriam distribuídos

ENTRE OS PROCESSOS EM UM PROGRAMA QUE

USAVA UMA ~~DISTRIBUIÇÃO~~ EM BLOCOS? $\rightarrow: \text{III}$

NA DISTRIBUIÇÃO EM BLOCOS, O NÚMERO
" n " DE ELEMENTOS É DIVIDIDO

IGUALMENTE E FORA ENTRE OS
PROCESSOS, CASO OS RESULTADOS DA DIVISÃO

ENTRE n e com_sz não tenha resto ($n\%_{\text{com_sz}} = 0$).

SE TIVER RESTO, PODEMOS DISTRIBUI-LO EM

ordem ENTRE OS PROCESSOS.

CADA PROCESSO VAI RECEBER A

INTIMA de $n/\text{com_sz} = 14 = 3$ ($\text{int}(n/\text{com_sz})$).

O RESTO DA DIVISÃO = 2 ($n\%_{\text{com_sz}} = 2$)

LOGO O PROCESSO 0 e 1 vão receber

4 elementos e 2 e 3 vão ficar com 3:

rank	0	1	2	3
$X[0], X[1], X[2], X[3]$	$X[4], X[5] \times [6, 7] \times [8, 9]$	$X[10], X[11] \times [12, 13] \times [14, 15]$		

3.6) b) como os componentes de x seriam

distribuídos entre os processos em um programa que realiza uma DISTRIBUIÇÃO CÍCLICA?

TABELA 3.9
CGIZIO

Em uma DISTRIBUIÇÃO cíclica, o NÚMERO "n" de ELEMENTOS É DISTRIBUÍDO UM POR VZ EM CADA PROCESSO Em um DETERMINADO NÚMERO DE CICLOS.

$x[0]$ vai para o PROCESSO 0, e $x[1]$ vai para o PROCESSO 1, e $x[2]$ vai para o PROCESSO 2 e $x[3]$ vai para o PROCESSO 3, FECHANDO UM CICLO. O PRÓXIMO CICLO SEGUE A MESMA LÓGICA.

RANK	0	1	2	3
	$x[0], x[4]$	$x[1], x[5]$	$x[2], x[6]$	$x[3], x[7]$
	$x[8], x[2]$	$x[9], x[13]$	$x[10] \otimes$	$x[11]$
	:			

2.6) a) como os componentes de X

SERIAM DISTRIBUÍDOS ENTRE OS PROCESSOS

Em uma programa que usava uma distribuição
CÍCLICA DE BLOCOS com TAMANHO DE Bloco b=?

UMA DISTRIBUIÇÃO BLOCO-CÍCLICA, AN-

UMA JUNGAIS DAS PROPRIEDADES DE BLOCOS

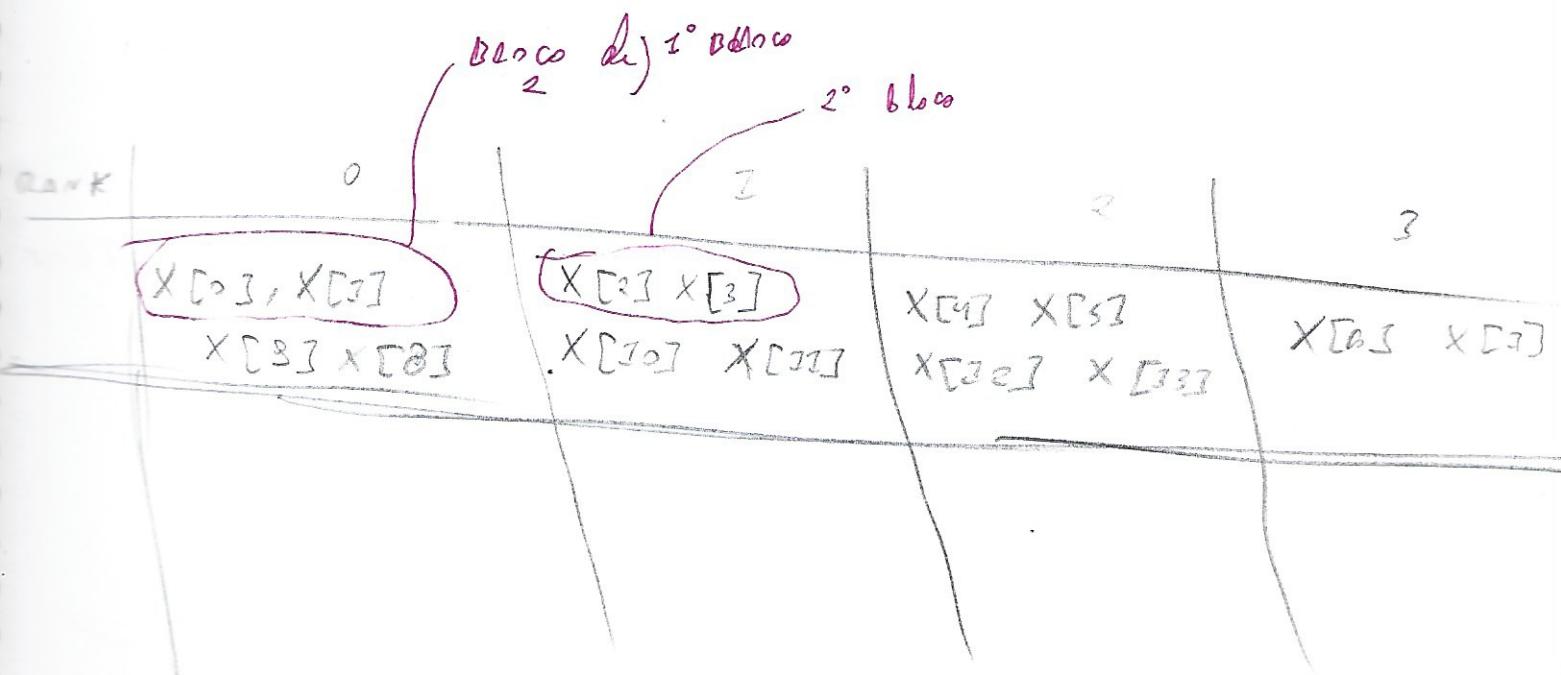
E CÍCLICA. PARA UM DETERMINADO "b"

o vetor "X" DE "n" ELEMENTOS SERIA DIVIDIDO

EM BLOCOS. O NÚMERO DE BLOCOS SERIA

$n/b = 19/2 = 7$. CASO A DIVISÃO

INTEIRA TIUVESSO RESTO, ELA SERIA DISTRIBUÍDA
ENTRE OS BLOCOS SIGUINHOS A MESMA ORDEM.



Q3.9) ESCREVA UM PROGRAMA MPI QUE
IMPLEMENTE A MULTIPLICAÇÃO DE UM
VETOR POR UM PRODUTO ESCALAR A
ESCALAR. O USUÁRIO DEVE INSERIR DOIS
VETORES E UM ESCALAR, TUDO LIDOS PELO
PROCESSO "0" E DISTRIBUIDOS ENTRE OS
PROCESSOS. OS RESULTADOS SÃO CALCULADOS
E COLETADOS NO PROCESSO "0" QUE OS
IMPRIME. VOCÊ PODE ASSUMIR QUE N
É ORDEM DOS VETORES, E IGUALMENTE DIVISÍVEL
POR COMM-SZ.

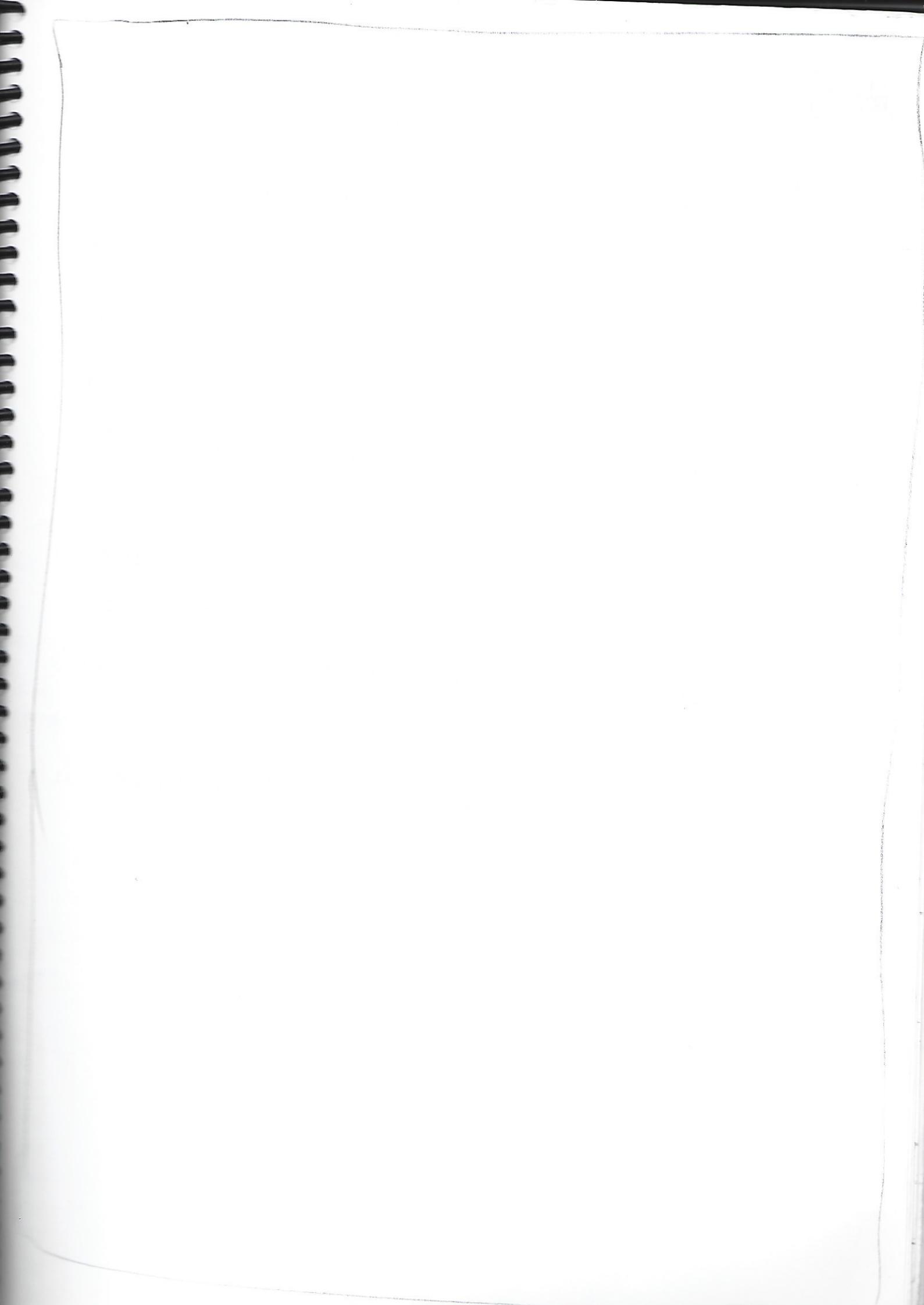
VAMOS MULTIPLICAR VETOR-1 PELO ESCALAR
E O RESULTADO FAZENDO O PRODUTO COM

VETOR 2. O VETOR 1 E VETOR 2 SÃO

DIVISÍVEIS COM MPI - SCATTER ENTRE OS
PROCESSOS E CADA (PEGA O DADO DO VETOR E
DISSEMINA ENTRE OS PROCESSOS.)

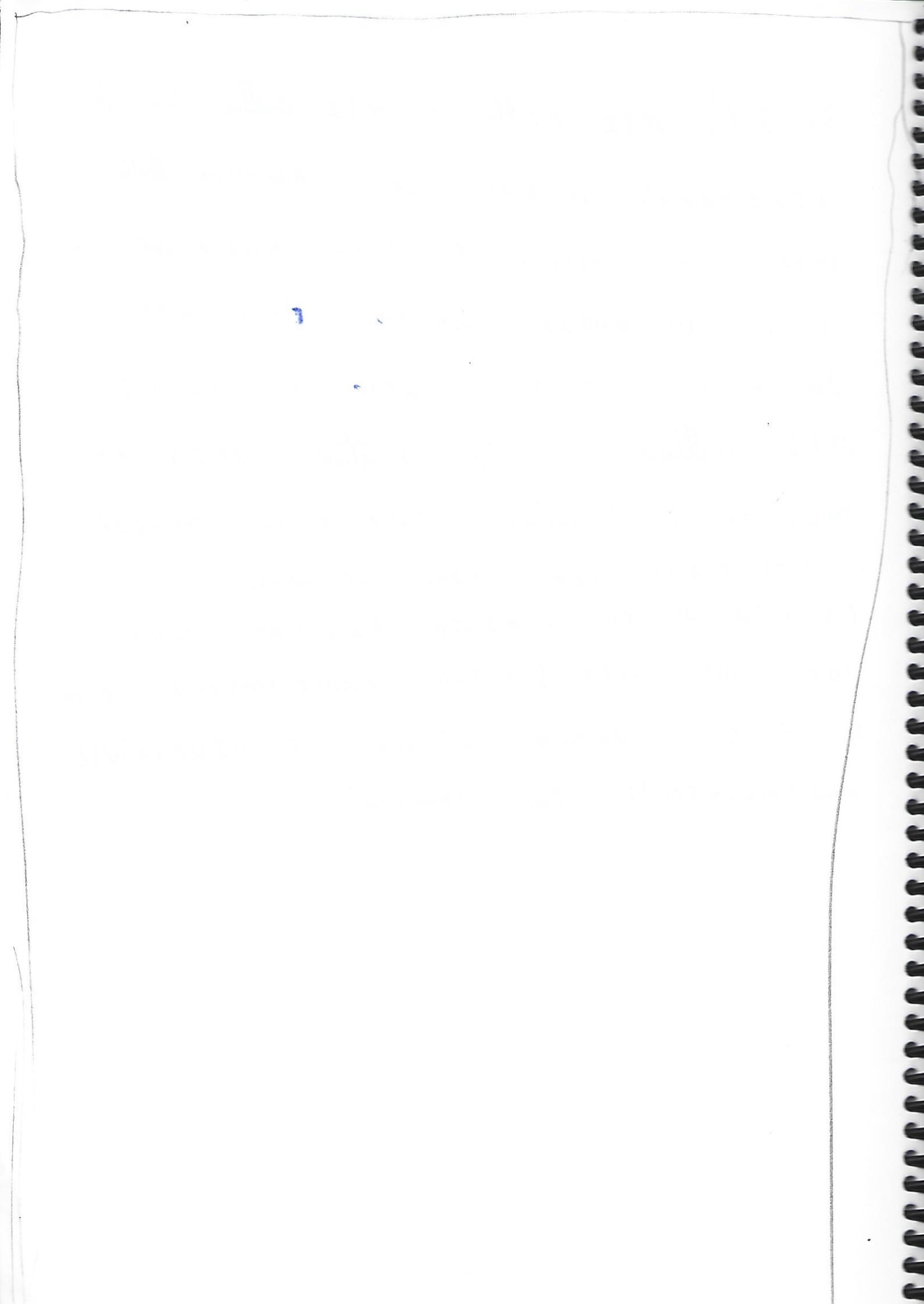
VOCAL-VETOR-1 E LOCAL-VETOR-2 SÃO ATÉ BEM
PARA TER OS RESULTADOS SOMADO PELO
MPI - REDUCE.

CÓDIGO PRÓXIMA
PÁGINA.



3. 13) MPI scatter & MPI Gather tem a
LIMITAÇÃO DE QUE CADA PROCESSO DEVE
ENVIAR EM RECEBER O MESMO NÚMERO DE
ITENS DE DADOS. QUANDO ESTE NÃO
for o caso, devemos usar as Funções
MPI Gather & MPI scatter. VEJA AS
PÁGINAS DO MANUAL PARA ESSAS FUNÇÕES
E MODELE AQUELE PROGRAMA DE PRODUTO ESCALAR PARA
QUE QUE POSSA LIDAR CORRETAMENTE com
o caso quando "n" não é DIVISÍVEL
UNIFORMEMENTE por "num_sz".

CÓDIGO NA PRÓXIMA
PÁGINA



3. 16) SUPONHA QUE COMO $SZ = 8 \times 8$

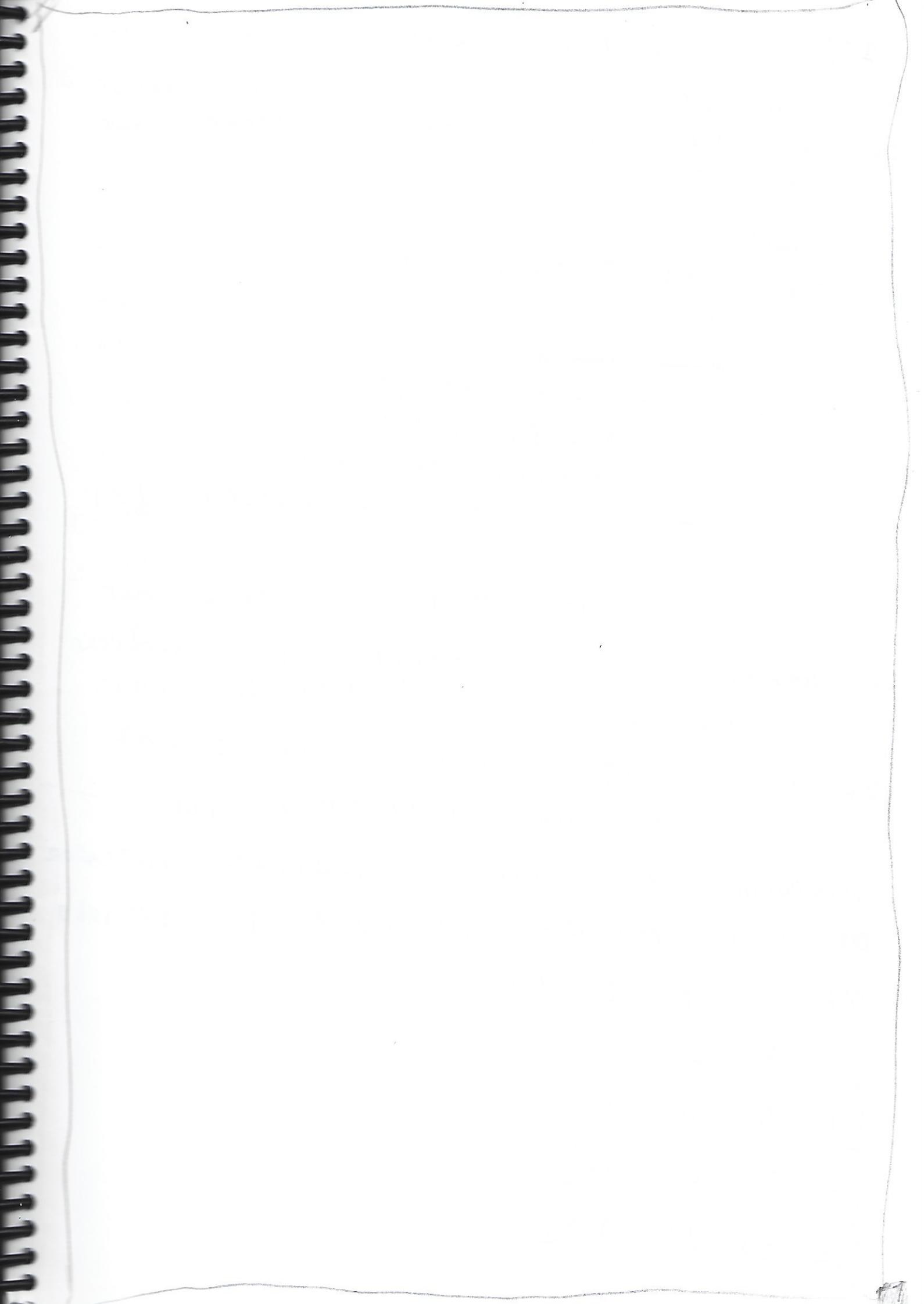
Vetor $x = (0, 1, 2, \dots, 75)$ foi distribuído ENTRE OS PROCESSOS USANDO UMA DISTRIBUIÇÃO DE BLOCOS. DESENHE UM DIAGRAMA ILUSTRANDO AS ETAPAS EM UMA IMPLEMENTAÇÃO BORBOLETA DE ALL GATHER DE EX.



3. IF) O TÍPO MPI CONTEÚDO Pode
SER USADO PARA CONSTUIR UM TÍPO
DE DADOS DERIVADO DE UMA
COLEÇÃO DE ELEMENTOS CONSECUTIVOS
EM MATRIZ. Sua SINTAXE É

```
int MPI_Type_contiguous(  
    int count /* in */,  
    MPI_Datatype old_type /* in */  
    MPI_Datatype * new_type /* out */)
```

MODELAR AS FUNÇÕES READ SECTION
É BEM VECTORS PARA QUE SEjam USADAS EM TIPOS
DE DADOS MPI CRIADOS POR UMA
CHAMADA PARA MPI_TYPE CONSEGUNTE E
UM ARGUMENTO DE CONTAGEM DE I
NAS CHAMADAS PARA MPI_SCATTER E
MPI_GATHER.



3.28) MPI - TYPE - INDEXED Pode ser usada
PARA CONSTRUIR UM TIPO DE DADOS DERIVADO
DE ELEMENTOS DE ARRAY ARBITRÁRIO. SUA
SINTAXE É

Int MPI - TYPE - indexed {

int count /* in */;

int ~~count~~ *

int array - of - block lengths []

int array - of - block cements []

MPI - DATA TYPE dd MPI - t

MPI - DATA TYPE NEN - MPI - T - P /* entry */;

Ao CONTRARIO DE MPI - TYPE - CREATE - STRUCT,
OS DESCENDIMENTOS SÃO MEDIDOS EM UNIDADES
DO ANTES MPI T - NÃO EN BYTES.
USE O TIPO MPI INDEXADO PARA CRIAR
UM TIPO DE DADOS DERIVADOS QUE
CORRESPONDE À PARTE TRIANGULAR SUPERIOR
DE UMA MATRIZ QUADRADA, POR EXEMPLO
NA MATRIZ 4×4 :

$$\begin{bmatrix} 0 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 4 & 5 & 6 & 7 \end{bmatrix}$$

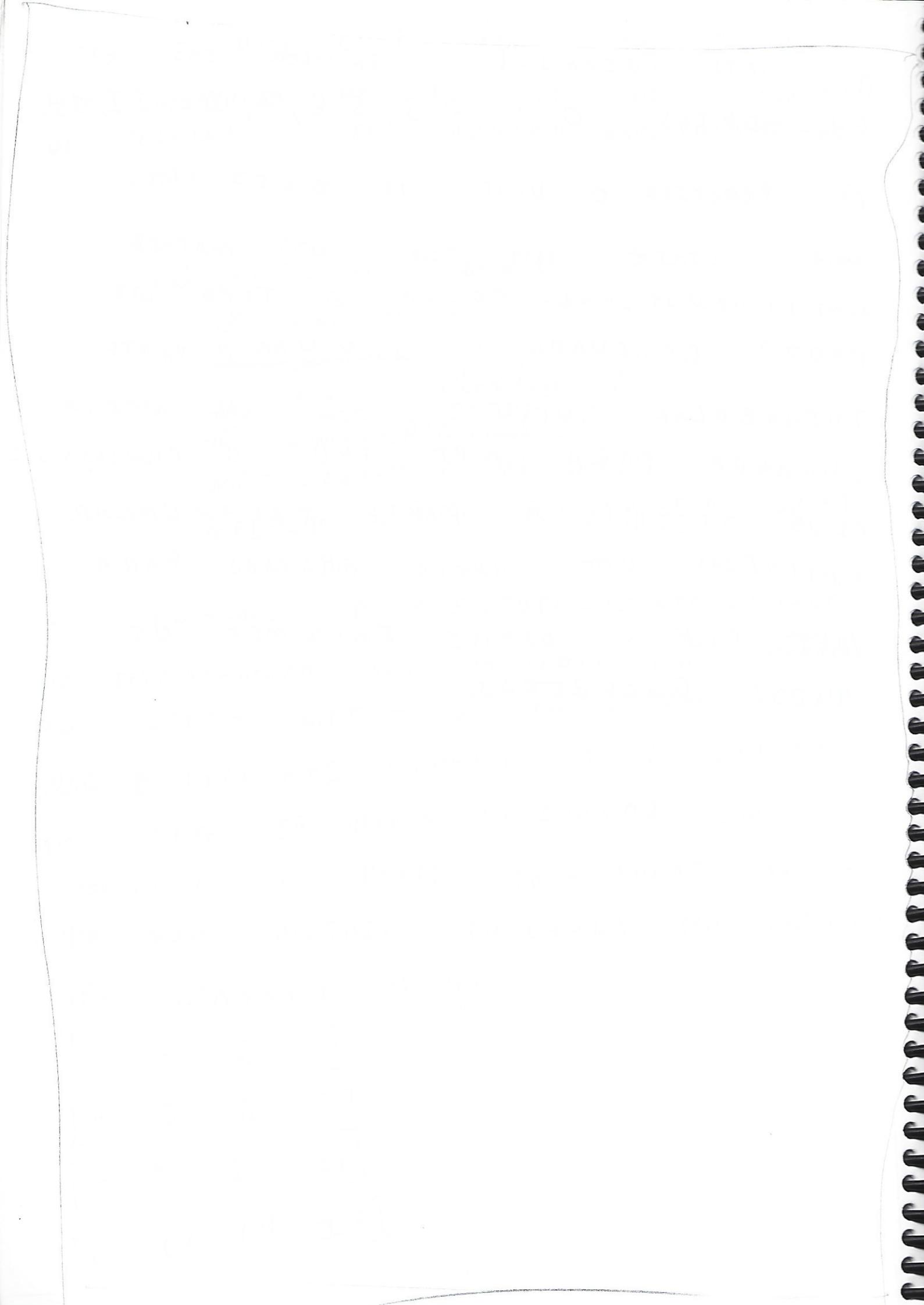
$$\begin{bmatrix} 8 & 9 & 10 & 11 \end{bmatrix}$$

$$\begin{bmatrix} 12 & 13 & 14 & 15 \end{bmatrix}$$

A PARTE TRIANGULAR SUPERIOR SÃO OS ELEMENTOS 0, 1, 2, 3, 5, 6, 7, 10, 11, 15.

O PROCESSO 0 DEVE SER LEDO EM UMA MATRIZ nn COMOS UMA MATRIZ UNIDIMENSIONAL, CRIAR O TIPO DE DADOS DERIVADO E ENVIAR A PARTE TRIANGULAR SUPERIOR COM UMA CHAMADA PARA MPI SEND. O PROCESSO 1 DEVE RECEBER A PARTE TRIANGULAR SUPERIOR COM UMA CHAMADA PARA MPI REC e DEPOIS IMPRIMIR OS DADOS RECEBIDOS.

CÓDIGO ABAIXO !



3.20) AS FUNÇÕES MPI PACK E MPI UNPACK

FORNCEM UNA ALTERNATIVA AOS TIPOS DE DADOS DERIVADOS PARA AGRUPAR DADOS.

O MPI PACK COPIA OS DADOS E SEREM ENVIADA OS, UM BLOCO POR VEZ, EM UM buffer FORNECIDO PELA USUÁRIO. O BUFFER PODE ENTÃO SER ENVIADO E RECEBIDO. DEPOIS QUE OS DADOS SÃO RECEBIDOS, MPI UNPACK PODE SER USADO PARA DESCOMPACTA-LOS DO BUFFER DE RECEPÇÃO. A SINTaxe É:

ENT MPI PACKS

```
void * in buf /* in */,  
int In Count /* in */,  
MPI - DATATYPE datatype /* in */,  
Void * Pack buf /* out */,  
int Pack buf Sz /* in */,  
Int * Position /* in/out */,  
MPI Comm /* in */;
```

PODEMOS, PORTANTO, FAZER PARTE OS DADOS DE ENTRADA PARA O PROGRAMA DE REGRAS TRAPEZOIDAL, COM O SEGUINTE:

MPI Pack (f_a , 1, MPI_DOUBLE, Pack buf, 500, &written, com)

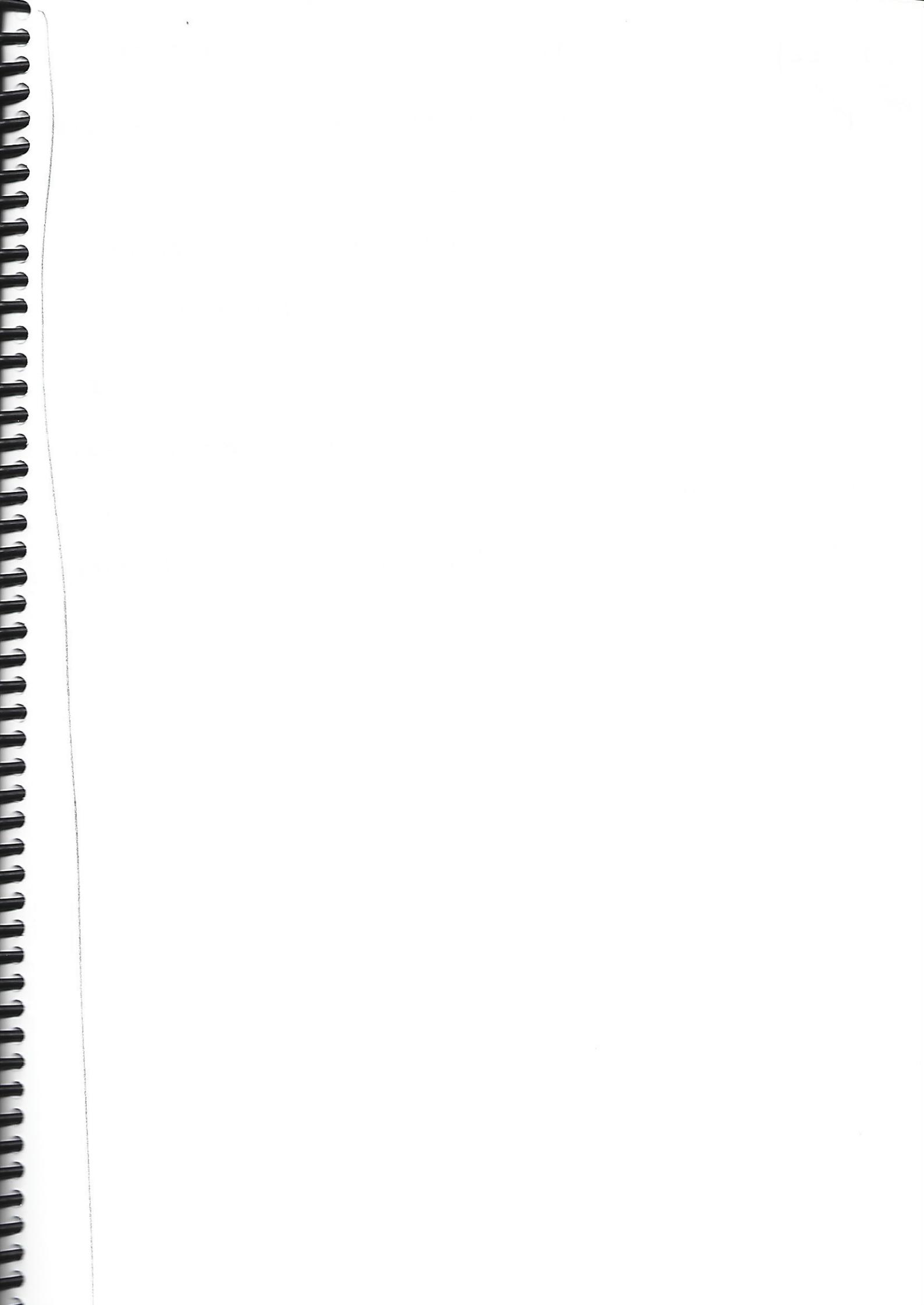
MPI Pack (f_b , 1, MPI_DOUBLE, Pack buf, 500, &written, com)

MPI Pack (f_n , 1, MPI_DOUBLE, Pack buf, 500, &written, com)

ESCREVA OUTRA FUNÇÃO GET Input para
o PROGRAMA DE REGRA TRAPEZOIDAL. ESTE
DEVE USAR MPI-PACK NO PROCESSO

E MPI - UNPACK nos outros processos
Porque é melhor?

CÓDIGO sob ABNIXO



Q3. 22)

CRONOMETRAR NOSSA IMPLEMENTAÇÃO

DA REGRA TRAPEZOIDAL QUE USA MPI - PESQUISA
Como VOCÊ ESCOLHEU Δt E OS NÚMEROS DE TRAPÉZIO?

Como OS TEMPOS MÍNIMOS SE COMPARAM
AOS TEMPOS MÉDIOS E MÉDIA NOS?

QUAIS SÃO OS ACELERAMENTOS? QUAIS

SÃO AS EFICIÊNCIAS? com BASE NOS

DADOS QUE VOCÊ COLETOU, VOCÊ DIRIA
QUE A REGRA TRAPEZOIDAL É ESCALAR?

MANDO O CÓDIGO

ABAIXO TEMOS

For wave MPI - WTime();

$n = 2$

TEMPO	TEMPO SERIAL	TEMPO PARALELO
MÍNIMO	$3,96 \times 10^{-3} [\text{ms}]$	$1,95 \times 10^{-3} [\text{ms}]$
MÁXIMO	$3,96 \times 10^{-3} [\text{ms}]$	$1,96 \times 10^{-3} [\text{ms}]$
MÉDIA	$3,96 \times 10^{-3} [\text{ms}]$	$1,964 \times 10^{-3} [\text{ms}]$
MEDIANA	$3,96 \times 10^{-3} [\text{ms}]$	$1,964 \times 10^{-3} [\text{ms}]$
RESULTADO	$3,33 \times 10^{-4}$	$3,33 \times 10^{-4}$

* Com. esso agora podemos

obter SPEEDUP e EFFICIENCIA

PELA DEFINIÇÃO

$$S = \text{SPEEDUP} = \frac{T_{\text{SERIAL}}}{T_{\text{PARALELO}}}$$

$$\text{EFEICIENCIA} = \frac{S}{P} = \frac{T_{\text{SERIAL}}}{P \cdot T_{\text{PARALELO}}}$$

ASSIM FAZEMOS A SIGUIENTE
TABELA!

$$a = 2; b = 100000$$

n	SPEEDUP	EFICIENCIA	RESULTADO
10	$S = \frac{3,35 \times 10^0}{3,42 \times 10^{-1}} = 1,93$	E = 0,78	$3,35 \times 10^{24}$
100	$S = \frac{5,34 \times 10^0}{2,91 \times 10^{-1}} = 1,84$	E = 0,87	$3,33 \times 10^{14}$
1000	$S = \frac{4,08 \times 10^{-5}}{2,022 \times 10^{-5}} = 2,02$	E = 1,01	$3,33 \times 10^{39}$
100000	$S = \frac{3,36 \times 10^{-3}}{2,00 \times 10^{-3}} = 2,00$	E = 0,93	$3,33 \times 10^{59}$

A ESCALA BIGLIADA

WEI TWPAD - SUPERCOMPUTADOR - MFR

9 PROCESSOS	120 TRABALHOS	EFICIENCIA
8 PROCESSOS	200 TRABALHOS	
16 PROCESSOS	400 TRABALHOS	
32 PROCESSOS	800 TRABALHOS	

PERCEBEU SE QUE COM 100 TRAPEZOS
TIIVEMOS A PRECISÃO DE TRES CASAS DECIMAS
possentes. com isso → SPEEDUP E
EFICIÊNCIA CRESCERAM A Partir
DESSA QUANTIDADE DE TRAPEZOS.

FOI REALIZADO com VALOR MÍNIMO
ESSAS MEDIDAS ASSEGURAR MELHOR CASO.

O TRAPEZIO NÃO DEMOSTRAU SER
ESCALÁVEL DEVIDO A EFICIÊNCIA
NÃO PERMANECER CONSTANTE AUMENTANDO
A ENTRADA E QUANTIDADE DE PROCESSOS
PELO MÉTODO EM Comum!

Q3. 23) Embora não saibamos os detalhes internos da implementação do MPI REDUCE, podemos supor que ele usa uma estrutura semelhante à árvore binária que discutimos. Se for esse o caso, esperaríamos que seu tempo de execução crescesse aproximadamente à taxa de \log_2^P , uma vez que existem cerca de $\log_2(P)$ níveis na árvore. (Aqui $P = \text{commsz}$)

Uma vez que o tempo de execução da regra trapezoidal serial é aproximadamente proporcional a " n ", o número de trapézios, e a regra trapezoidal paralela simplesmente aplica a trapezios

n/P em cada processo, com nossa suposição sobre MPI REDUCE, obtemos uma fórmula para o tempo de execução geral da regra trapezoidal paralela que se parece com algumas constantes a e b .

a) USE A FÓRMULA 3.22 E SEU PROGRAMA MATEMÁTICOS (POR EXEMPLO, MATLAB) PARA OBTIR UMA ESTIMATIVA DE MÍNIMOS QUADRADOS DOS VALORES DE " a " E " b "?

n° de trapézios	TEMPO ($P=2$)
10	$5,904 \times 10^{-8}$
100	$7,204 \times 10^{-6}$
1.000	$2,99 \times 10^{-5}$
10.000	$2,20 \times 10^{-9}$
100.000	$2,02 \times 10^{-3}$

A Tabela é que a coluna da MATRIZ, é " n/p " (SEGUNDOS COLUNA $\log_2(P)$, com os tempos)

Lembrando

$$\begin{bmatrix} n/p & \log_2(P) \\ n/p & \log_2(P) \\ n/p & \log_2(P) \\ n/p & \log_2(P) \\ n/p & \log_2(P) \end{bmatrix}, \{5,904 \cdot 10^{-8}, 7,204 \cdot 10^{-6}, 2,99 \cdot 10^{-5}, 2,20 \cdot 10^{-9}, 2,02 \cdot 10^{-3}\}$$

~~USEI~~ USEI A WOLFRAM ALPH



1998

12
25 Dec

Q3.23) b) COMENTE SOBRE A QUALIDADE
DOS TEMPOS DE EXECUÇÃO PREVISTO
USANDO A FÓRMULA E OS VALORES
PARA "a" e "b" CALCULADOS NA Parte (a)

OS RESULTADOS SÃO MUITO BONS, DEVIDO
A MPI = REDUCE S. J. A IMPLEMENTAÇÃO
COM FAVOR DE BINÁRIA CONSEGUE MUITO

3.27) ENCONTRÉ AS ACCELERAÇÕES E
EFICIÊNCIAS DO TIPO PARALELO IMPAR-
PAR. O PROGRAMA obtém acelerações livres?
É ESCALÁVEL? É ALTAMENTE ESCALONÁVEL?
É FRACAMENTE ESCALONÁVEL?

PROGRAMA ADAEX0!

Con ISSO MONTAMOS A
SIGUIENTE TABLA

n	SIRFAL		PARALELO
	1 PROCESADOR → T0J	2 PROCESADORES T0J	
100	$14,73 \times 10^{-6}$		
1.000	$135 \times 97 \times 10^{-6}$		
10.000	$1375,08 \times 10^{-6}$		$665,18 \times 10^{-5}$
100.000	$13037,73 \times 10^{-6}$		$1923,05 \times 10^{-5}$
			23630,86 s

PERCEBE - SE QUE OS TIPOS ANNUNCIAM CON
2 PROCESOS, UMA POSSIVEL EXPLICAÇÃO
DEVIDO A TRAÇA DE MENSAGEM
ENTRE OS NÚCLEOS DE USAMOS
USAMOS MAIS PROCESSADORES

* AGORA VAMOS CALCULAR SPPROS E
EFICIÊNCIA.

n	SPPROS →	EFICIÊNCIA
100	0,45	0,23
1.000	0,27	0,14
10.000	0,91	0,135
100.000	0,80	0,44

PERCEBER QUE O SPEEDUP
O PODEROSA
NÃO É LINEAR DEVIDO ELE
NÃO CRESCER LEMPARMENTE COM
O PROBLEMA

Agora VAMOS TESTAR A ESCALA DE UNI
DE PARA ISSO FOI TESTADO NA
NPAD

4 PROCESSOS
100 ELEMENTOS

$$E =$$

8 PROCESSOS
200 ELEMENTOS

$$E =$$

16 PROCESSOS
400 ELEMENTOS

$$E =$$

32 PROCESSOS
800 ELEMENTOS

$$E =$$

NÃO É ESCALAVEL PORQUE A EFICIÊNCIA
NÃO SE MANTÉVE CONSTANTE.

O PROGRAMA MANTÉM APROXIMADAMENTE
CONSTANTE A EFICIÊNCIA QUANDO O
AUMENTAMOS O PROBLEMA E
MANTEMOS IGUAL O NÚMERO DE
CÓDIGO DIFERENTES. P. T. C. E. F. R. C. I. K.

DESSA FORMA NÃO É FORTEMENTE
ESCALÁVEL PORQUE A EFICIÊNCIA
NAS PERMANECE CONSTANTE COM
AUMENTO DO PROCESSADORES.

Q16 EXTRA: QUANDO O COMPILADOR
NÃO É CAPAZ DE VETORIZAR
AUTOMATICAMENTE, OU VETORIZAR DE
FORMA INEFICIENTE, O OpenMP PROVÉ
A diretiva "omp simd", com a qual
o PROGRAMA PODE INDICAR UM LAGE EXPLÍCI-
TAMENTE PARA A COMPILAÇÃO VETORIZAR.

NO CÓDIGO ABASTECE A INCLUSÃO
DA CLÁUSULA "reduction" FUNCIONA
DE FORMA SEMELHANTE A FLAG
"-ffast-math", INDICANDO QUE A
REDUÇÃO NA VARIAVEL sera FEITA
E DEVE SER FEITA

#pragma omp simd reduction(+:soma)

for (i=0; i < n; i++) {

 x = (i + 0.5) * h

 soma += q.0 / (1.0 + x*x)

}

PODE SER NÃO É NECESSÁRIO USAR A CLÁUSULA
PRIVATE(x) NESTE CASO MAS SERIA ENSO

DEVERIA OMP SIMD POSSER SER COMBINADA
COM A diretiva OMP PARALLEL FOR?

O "SIMD" REALIZA multiplos ITERAÇÕES DO LOOP SEJA EXECUTADA CONCURRENTEMENTE USANDO A INSTRUÇÃO "SIMD"

OU SEJA SINGLE INSTRUCTION MULTIPLE DATA, ASSIM REALIZA VETORIZAÇÃO DE BLOCOS DE ITERAÇÕES DE LOOPS QUE PERTENCEM A MESMA THREADS.

ASSIM, CASO SEJA ADICIONADO "PARALLEL FOR SIMD" STREAM GERA DIFERENTES THREADS, QUE PROCESSA AS OPERAÇÕES VETORIALMENTE EM TEMPOS DINTINTOS, TORNANDO A ATRIBUIÇÃO DE "x" POSSÍVEL DE CONDECAS ARRISCADA. DEVE SE USAR PRIVATE(X) NOSSO CONTEXTO "SIMD", "PARALLEL SIMD" (SEM PRIVATE(X)).