

# 基礎情報処理

Information Processing Basics  
プログラム基礎2

2004年11月11日

高等教育研究開発推進センター  
小山田耕二

# Outline

1. コンピュータとはなにか
2. デジタル情報の世界
3. 論理回路からコンピュータまで1
4. 論理回路からコンピュータまで2
5. プログラム基礎1
6. プログラム基礎2
7. データ構造とアルゴリズム1
8. データ構造とアルゴリズム2
9. コンピュータネットワーク
10. 情報倫理
11. さまざまな情報処理
12. コンピュータ科学の諸問題

# 6 . プログラム基礎 2

## 6.1 コンパイラとインタプリタ

### 6.1.1 コンパイラ

### 6.1.2 インタプリタ

## 6.2 応力解析と連立1次方程式

### 6.2.1 全体剛性マトリクス

### 6.2.2 拘束条件・荷重条件の処理

### 6.2.3 N元連立1次方程式を解くプログラミング

### 6.2.4 Gaussの消去法(1)

### 6.2.5 Gaussの消去法(2)

## 6.1 コンパイラとインタプリタ



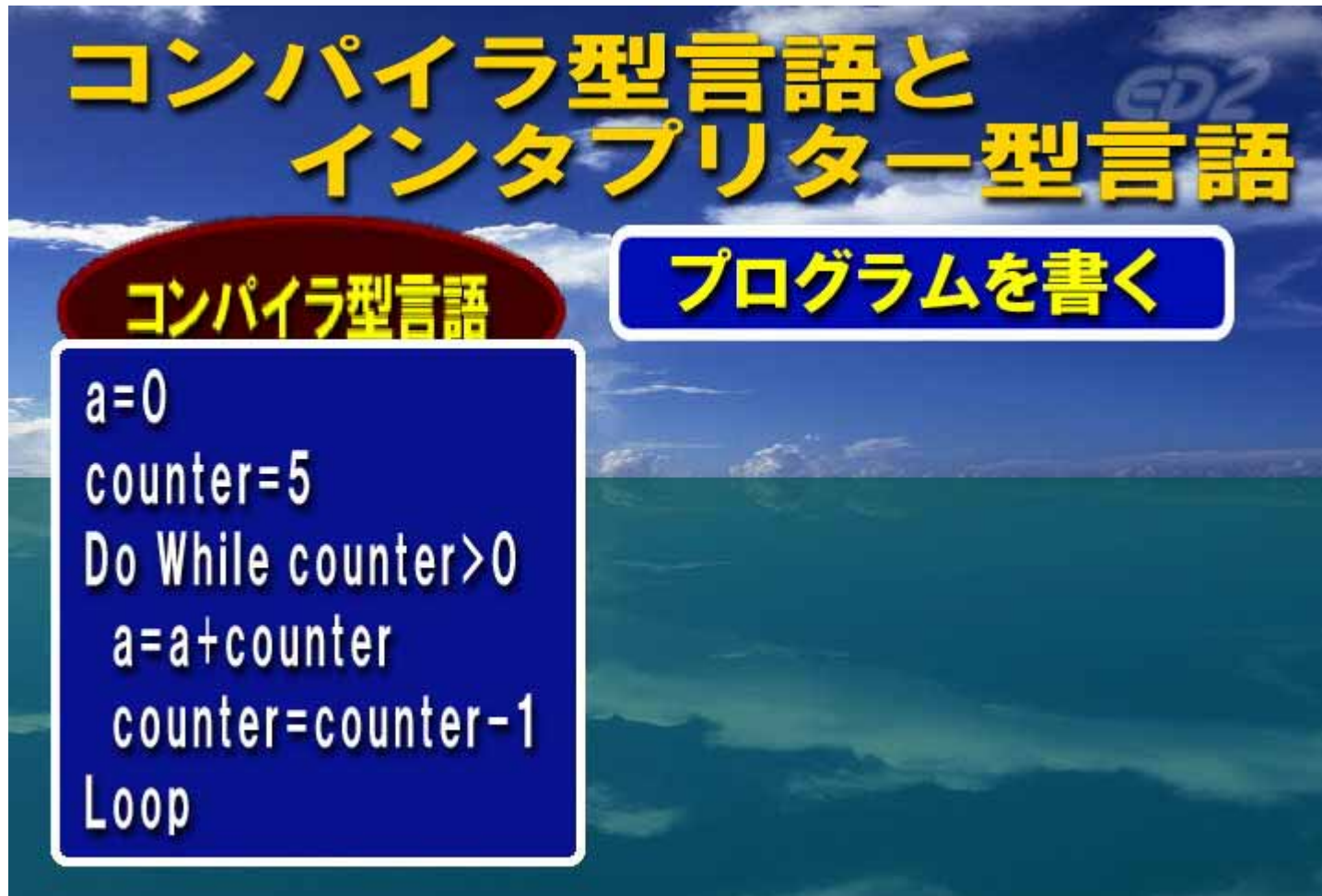
## 6.1 コンパイラとインタプリタ

コンピュータではこのプログラム言語をそのまま実行することは不可能

機械語というコンピュータ用の言語に翻訳する必要がある。

- インタプリタ方式
  - 翻訳を毎回行う方法
- コンパイラ方式
  - あらかじめすべて翻訳しておく方法

## 6.1.1 コンパイラ



コンパイラ型言語と  
インタプリター型言語

コンパイラ型言語

プログラムを書く

```
a=0  
counter=5  
Do While counter>0  
  a=a+counter  
  counter=counter-1  
Loop
```

## 6.1.1 コンパイラ

The slide features a blue sky background with clouds. At the top, the title 'コンパイラ型言語とインタプリター型言語' is written in large yellow characters. Below the title, there are two main columns. The left column is headed by a red oval containing the text 'コンパイラ型言語'. Below this header is a blue box containing the following code: `a=0`, `counter=5`, `Do While counter>0`, `a=a+counter`, `counter=counter-1`, and `Loop`. The right column is headed by a blue box containing the text 'コンパイルする'. Below this header is a blue box containing a 10x10 grid of binary digits (0s and 1s).

### コンパイラ型言語と インタプリター型言語

コンパイラ型言語	コンパイルする
<code>a=0</code>	1000110011001000
<code>counter=5</code>	1000111011011000
<code>Do While counter&gt;0</code>	1011000000000000
<code>  a=a+counter</code>	1011010000000101
<code>  counter=counter-1</code>	0000000011100000
<code>Loop</code>	1111111011001100
	0111010111111010
	1010001000010010
	0000000011001111

# 6.1.1 コンパイラ

コンパイラの実行は、以下の処理から構成される

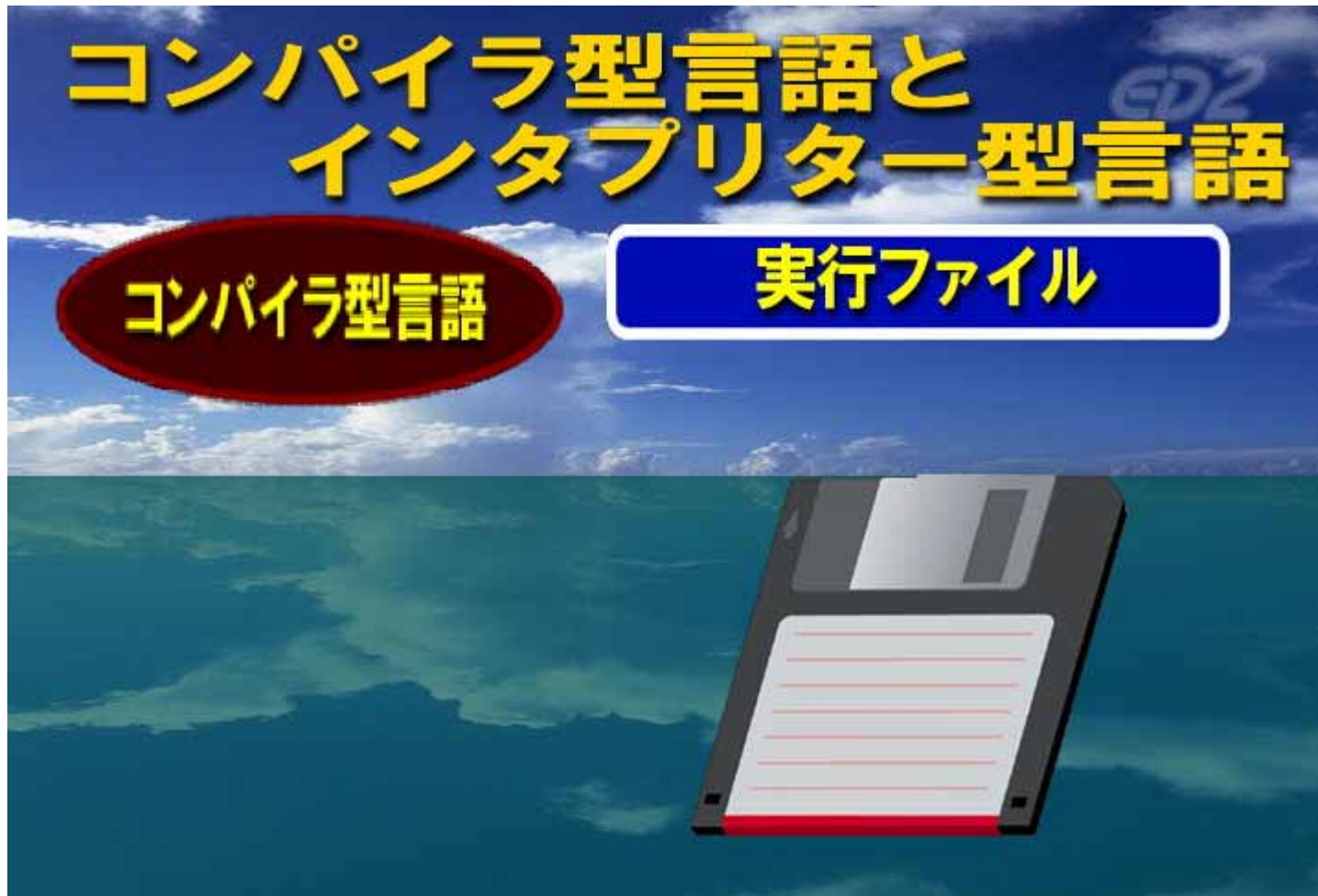
- ソースコードを読み込み、字句列に分解する字句解析部
  - 字句: ソースコードに出現する文字列の中で意味を持つ最小単位 (main, if, \$a などの関数名や変数名、{} () [] などの括弧、+-=\*などの演算記号、;,: などといった区切り文字、3.14259のような定数)
  - 字句解析: 連続した文字の並びを、意味における最小単位である字句に分割すること。
- 字句の列をもとにプログラミング言語の構文木を構築する構文解析部、
  - 構文解析: 字句解析によって得られた字句の並びについて、字句間の関係を解析すること
- 対象計算機のコードを生成するコード生成部
- コード生成の前段階で効率の高いコードに変換する最適化部



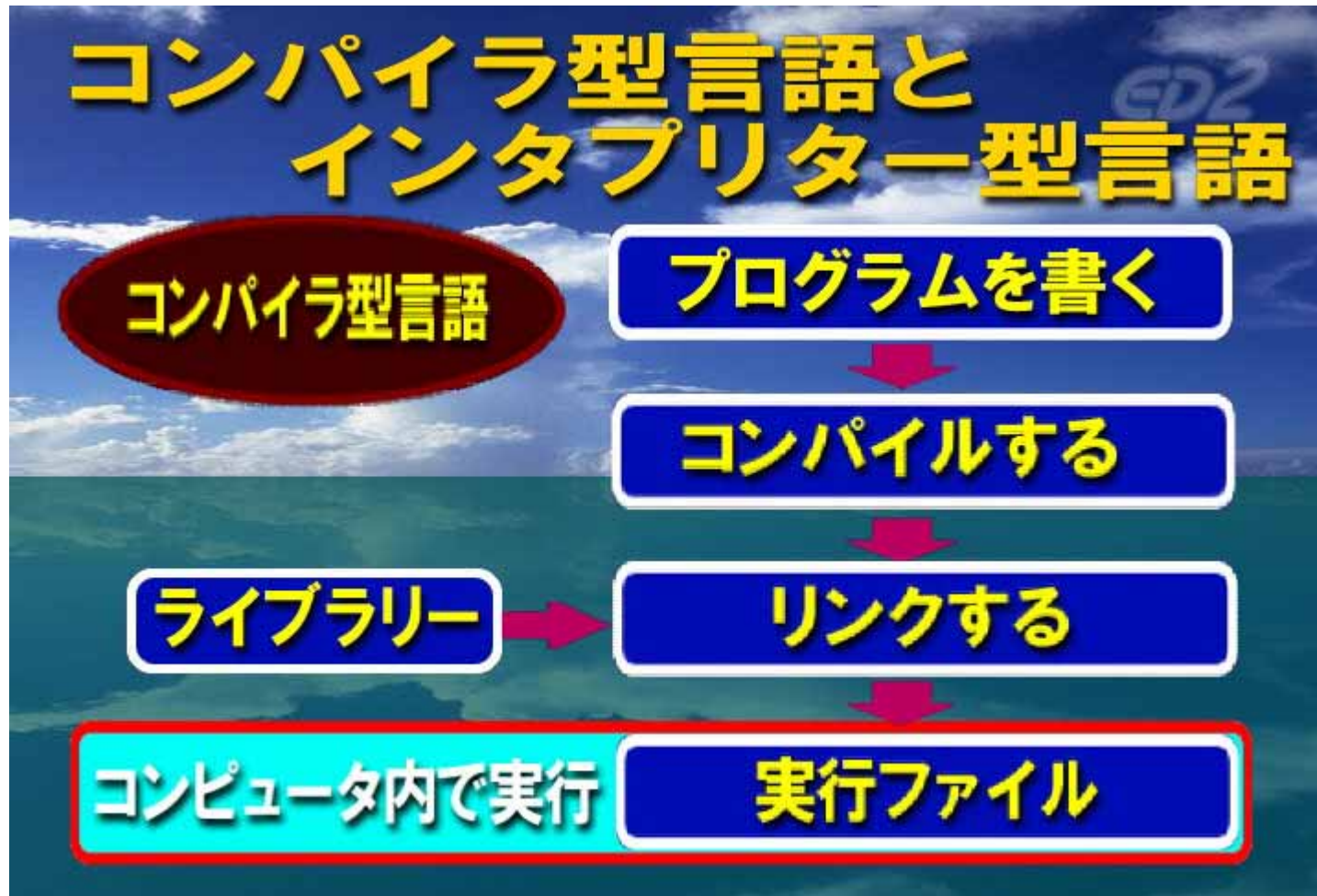
# 構文解析

- 構文解析はコンパイラにおけるもっとも中心的な過程のひとつ
- 形式言語の構文解析は曖昧でない言語を対象としている。
  - ある言語が曖昧であるとは、ひとつの文にたいして2通り以上の構文解析が可能
- インターネット上の Web ページやメールアドレスをあらわす URL は、次のような構文をもつ：
  - `http://サーバ名/パス名` (web ページをあらわす場合, `http://ja.wikipedia.org/` など)
  - `mailto:ユーザ名@ドメイン名` (電子メールアドレスをあらわす場合, `mailto:god@heaven.mil` など)
- プログラミング言語において次のような数式を計算する場合に必要となる。
  - $2 + (a + b \times c) \div (d - e)$ 
    - かっこの中を先に計算
    - 演算子の優先順位を考慮
  - 数値:2 + 式:( 式:( 変数:a + 式:( 変数:b × 変数:c ) ) ÷ 式:( 変数:d - 変数:e ) )

## 6.1.1 コンパイラ



## 6.1.1 コンパイラ



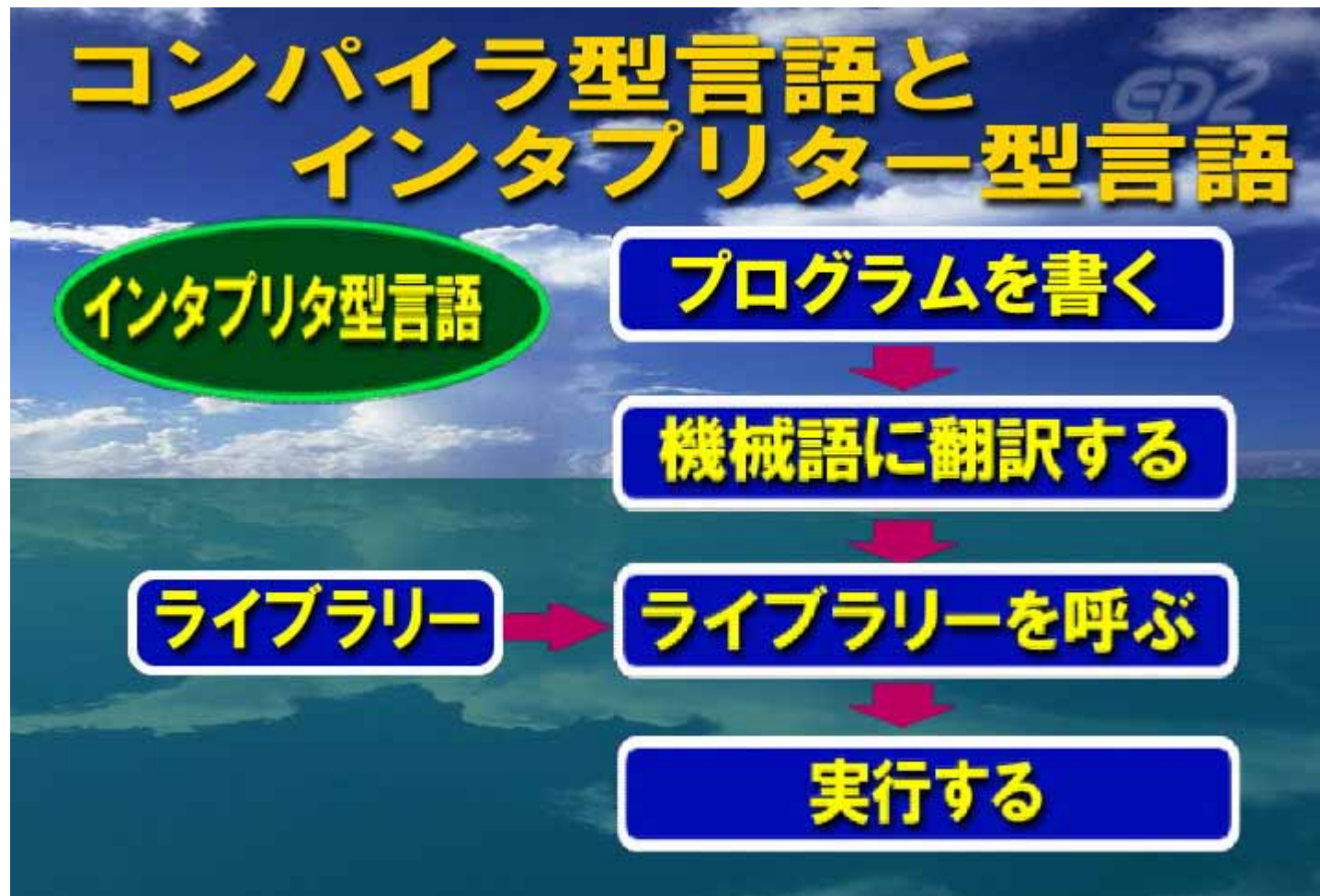
# リンク

- 機械語のプログラムの断片を結合し実行可能なプログラムを作成するプログラム
  - ソースファイルをコンパイルするとオブジェクトファイルが生成される。
  - 他のオブジェクトファイルやライブラリを結合して1つのプログラムを作成
- 静的リンク
  - 他のオブジェクトやライブラリをすべて1つのファイルに結合
  - 単独で動作可能・組み込み向けプログラムなどで使われる。
  - 出来上がったプログラムのサイズが大きい
- 動的リンク
  - ライブラリの参照を、名前のみで解決
  - 出来上がったプログラムにはライブラリの部分のプログラムは含まれないので出来上がったプログラムのサイズが小さい
  - 実行時に、ライブラリ空間上の実際のプログラムと結合(ローダと呼ばれるプログラムで実現)

# いろいろなコンパイラ

- 開発環境とは別の環境で実行できるコードを生成するクロスコンパイラ
  - 新しいコンピュータが開発されるとき、BIOSやOSなどもっとも基本となるプログラムについて、既存のものがそのままでは実行できない場合に利用。
  - 組み込みシステムやPDAなど、それ自体が開発環境を動作させるだけの性能を持たない場合に利用。
- 直接CPUで解釈実行可能なコードを生成せずに、中間コードを生成し、別のインタプリタによって実行するバイトコードコンパイラ
- プログラミング言語の文法と出力形式を定義してコンパイラを作るためのコンパイラをコンパイラコンパイラ

## 6.1.2 インタプリタ



## 6.1.2 インタプリタ

- インタプリタとは、プログラム言語で書かれたソースコードを逐次解釈しつつ実行するコンピュータ用のプログラム
- インタプリタで実行されるプログラミング言語をスクリプト言語と呼ぶこともある
- この方式は毎回翻訳を行うため処理速度に問題がある。
- コンパイラと比較して
  - 会話的な応答性に優れる。(プログラムを作成/変更してから実行するまでの手間がない)。
  - 単純な実装では実行に時間がかかりがち。しかし動的に最適化を施すことができるので一概にコンパイラより遅いとは言えない。
  - 実行には言語に対応するインタプリタ(ランタイム)環境が必要



## 6.1.2 インタプリタ

コンパイラ型言語と  
インタプリター型言語

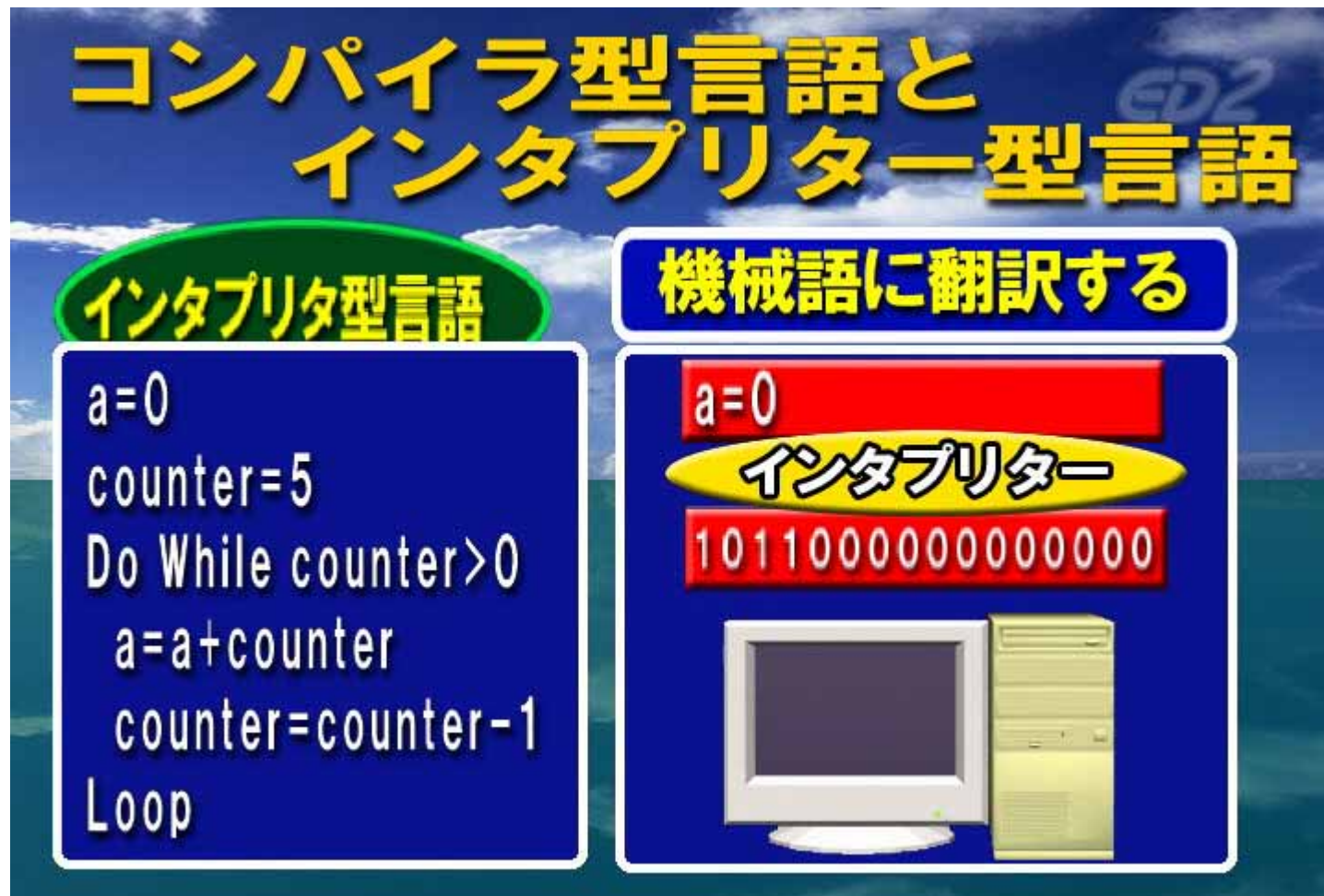
インタプリタ型言語

```
a=0  
counter=5  
Do While counter>0  
  a=a+counter  
  counter=counter-1  
Loop
```

プログラムを書く



## 6.1.2 インタプリタ




## 6.1.2 インタプリタ

コンパイラ型言語と  
インタプリタ型言語

インタプリタ型言語

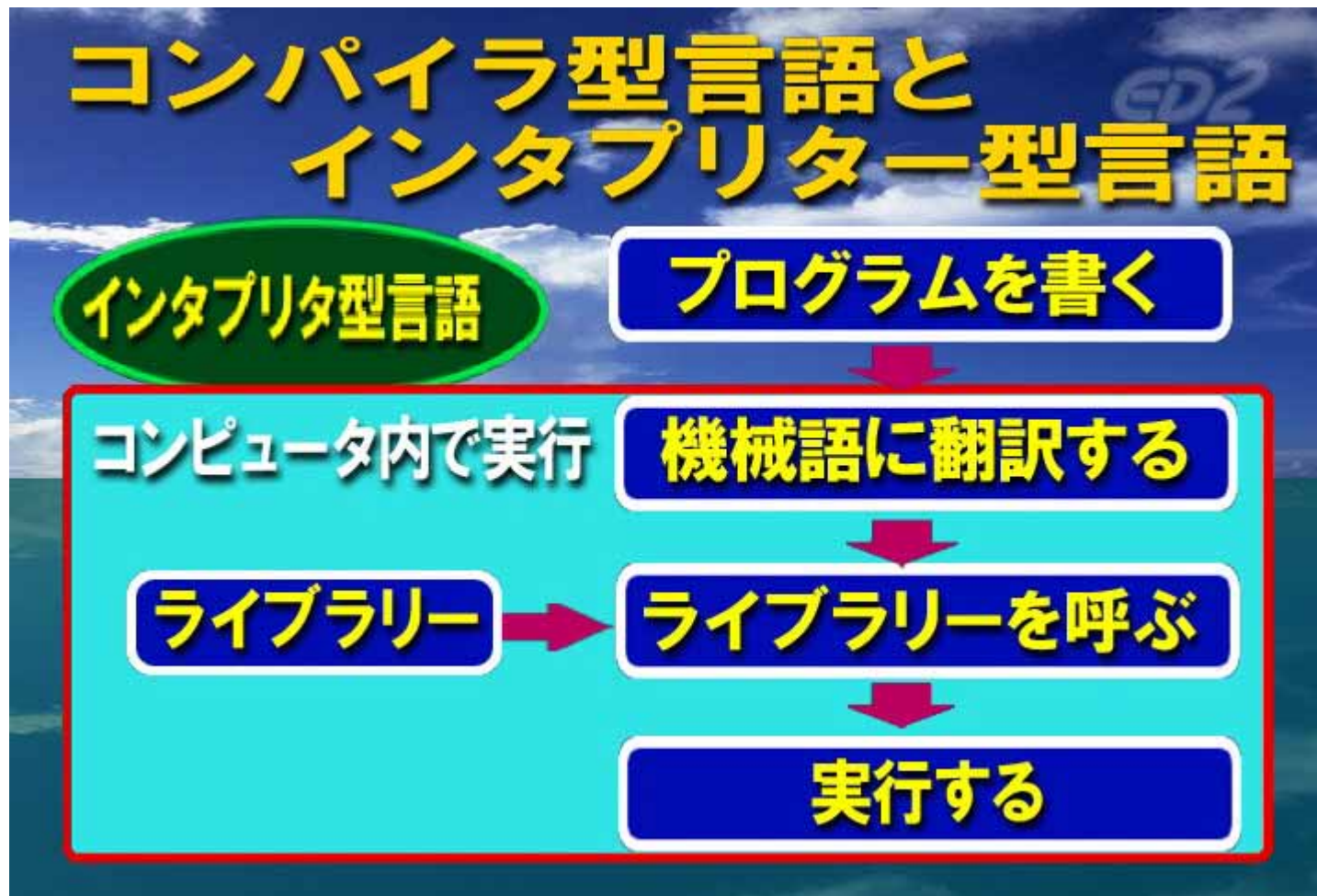
```
a=0  
counter=5  
Do While counter>0  
  a=a+counter  
  counter=counter-1  
Loop
```

実行する

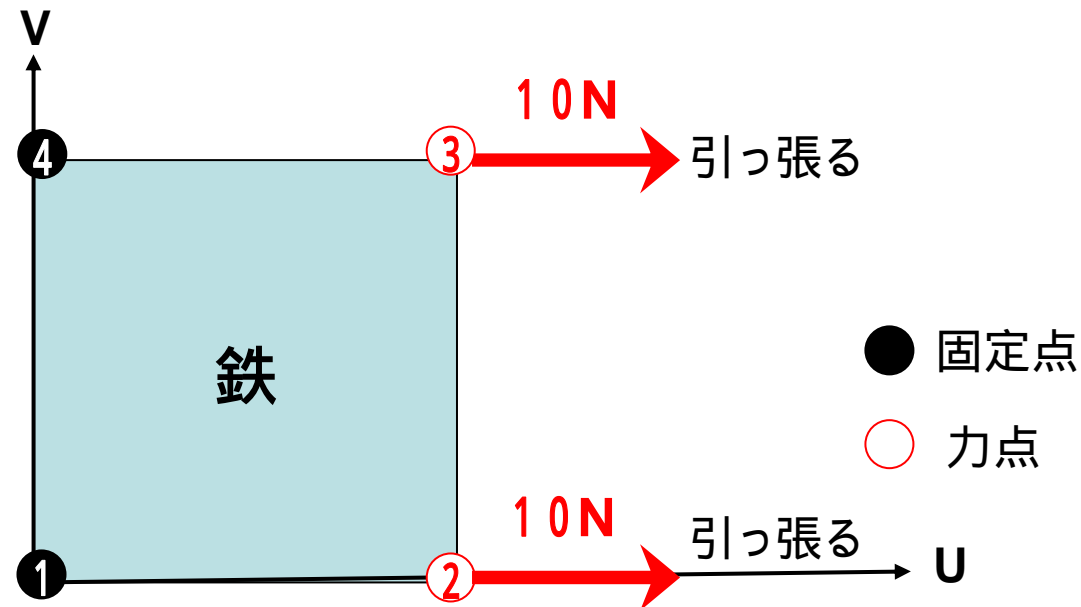


1011000000000000

## 6.1.2 インタプリタ



## 6.2 応力解析と連立1次方程式



### 問題

一辺の長さが1である鉄製(ヤング率; 21000、ポアソン比; 0.333)の正方形(厚さ1)の2つの頂点を完全拘束し、残りの頂点を10Nで辺に平行な方向に引っ張ったとき、頂点の変位量を計算せよ。

## 6.2.1 全体剛性マトリクス

全体	1	1	2	2	3	3	4	4	×	変位	外力
1	15748	0	-11810	3933	0	-7871	-3938	3938		U1	FX1
1	0	15748	3938	-3938	-7871	0	3933	-11810		V1	FY1
2	-11810	3938	15748	-7871	-3938	3933	0	0		U2	FX2
2	3933	-3938	-7871	15748	3938	-11810	0	0		V2	FY2
3	0	-7871	-3938	3938	15748	0	-11810	3933		U3	FX3
3	-7871	0	3933	-11810	0	15748	3938	-3938		V3	FY3
4	-3938	3933	0	0	-11810	3938	15748	-7871		U4	FX4
4	3938	-11810	0	0	3933	-3938	-7871	15748		V4	FY4

第5回講義資料参照のこと

## 6.2.2 拘束条件・荷重条件の処理

全体	1	1	2	2	3	3	4	4	変位	外力
1	1	0	0	0	0	0	0	0	U1	0
1	0	1	0	0	0	0	0	0	V1	0
2	0	0	15748	-7871	-3938	3933	0	0	U2	10
2	0	0	-7871	15748	3938	-11810	0	0	V2	0
3	0	0	-3938	3938	15748	0	0	0	U3	10
3	0	0	3933	-11810	0	15748	0	0	V3	0
4	0	0	0	0	0	0	1	0	U4	0
4	0	0	0	0	0	0	0	1	V4	0

× =

接点2と3についてのみ考える

# 4 × 4 行列式

$$\begin{pmatrix} \begin{array}{|c|c|c|c|} \hline 15748 & -7871 & -3938 & 3933 \\ \hline -7871 & 15748 & 3938 & -11810 \\ \hline -3938 & 3938 & 15748 & 0 \\ \hline 3933 & -11810 & 0 & 15748 \\ \hline \end{array} \\ \times \begin{pmatrix} U2 \\ V2 \\ U3 \\ V3 \end{pmatrix} = \begin{pmatrix} 10 \\ 0 \\ 10 \\ 0 \end{pmatrix}$$

剛性マトリクス

変位

外力



# 4元1次方程式

$$\left\{ \begin{array}{l} 15748 U_2 + -7871 V_2 + -3938 U_3 + 3933 V_3 = 10 \\ -7871 U_2 + 15748 V_2 + 3938 U_3 + -11810 V_3 = 0 \\ -3938 U_2 + 3938 V_2 + 15748 U_3 + 0 V_3 = 10 \\ 3933 U_2 + -11810 V_2 + 0 U_3 + 15748 V_3 = 0 \end{array} \right.$$



## 6.2.3 N元連立1次方程式を解く プログラミング

N元連立1次方程式

$$A \cdot x = b$$

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & \cdots & a_{0,n-1} \\ a_{10} & a_{11} & a_{12} & \cdots & a_{1,n-1} \\ a_{20} & a_{21} & a_{22} & \cdots & a_{2,n-1} \\ a_{30} & a_{31} & a_{32} & \cdots & a_{3,n-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n-1,0} & a_{n-1,1} & a_{n-1,2} & \cdots & a_{n-1,n-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \cdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ \cdots \\ b_{n-1} \end{bmatrix}$$

## 6.2.4 Gaussの消去法(1)

### 前進消去(1)

k 番目の式の両辺を  $a_{kk}$  で割る。  
(第 k 番目の式を  $x_k$  について解くことと同じ)

$k = 0, 1, 2, \dots, n-2$  の順に

$a_{kj} = a_{kj} / a_{kk} \quad (j = k+1 \text{ から } n-1 \text{ まで})$

### 前進消去(2)

第 k 行の  $a_{ik}$  倍を第 i 行から引く。(第 i 式から  $x_k$  を消去している)  
その結果、 $a_{ik}$  は 0 になる。  
これを繰り返すと  $x_0, x_1, \dots, x_{n-3}, x_{n-2}$  と順に消去される。

$i = k+1, k+2, \dots, n-1$  の順に

$a_{ij} = a_{ij} - a_{ik} \times a_{kj} \quad (j = k+1 \text{ から } n-1 \text{ まで})$

## 6.2.5 Gaussの消去法(2)

### 後退代入

最後は $x_{n-1}$ だけの式になる。  
それを解いて「後退代入」を繰り返して $x_{n-2}$ 、 $x_{n-3}$ 、 $\dots$   
 $x_1$ 、 $x_0$ を順に求める。

$k = 0, 1, 2, \dots, n-2$ の順に

$$x_k = b_k - \sum_{j=k+1}^{n-1} a_{kj} x_j$$

# 4元1次方程式を解くプログラム

$$\left\{ \begin{array}{l} 15748 U_2 + -7871 V_2 + -3938 U_3 + 3933 V_3 = 10 \\ -7871 U_2 + 15748 V_2 + 3938 U_3 + -11810 V_3 = 0 \\ -3938 U_2 + 3938 V_2 + 15748 U_3 + 0 V_3 = 10 \\ 3933 U_2 + -11810 V_2 + 0 U_3 + 15748 V_3 = 0 \end{array} \right.$$

# Gaussの消去法：変数宣言

言語：Visual Basic

2 3	C	D	E	F
2 4	15748	-7871	-3938	3933
2 5	-7871	15748	3938	-11810
2 6	-3938	3938	15748	0
2 7	3933	-11810	0	15748

Excel上の配置

```
Dim NN As Integer  
Dim A() As Single  
Dim B() As Single  
Dim X() As Single
```

NN = 4      ……行列数

```
ReDim A(NN, NN)  
ReDim B(NN)  
ReDim X(NN)
```

```
For i = 1 To NN  
  B(i) = Cells(23 + i, 13)  
  For j = 1 To NN  
    A(i, j) = Cells(23 + i, 3 + j)  
  Next j  
Next i
```

Sub Solve4()

変数宣言

1.000E+01

0

1.000E+01

0

マトリクス

外力

変位

# Gaussの消去法：前進消去 1

$L=1, NN=4$

15748	-7871	-3938	3933
-7871	15748	3938	-11810
-3938	3938	15748	0
3933	-11810	0	15748

割り算に使うものを選択 (ピボット)

‘前進消去

For  $L = 1$  To  $NN - 1$

$P = A(L, L)$

For  $j = L + 1$  To  $NN$

$A(L, j) = A(L, j) / P$

Next  $j$

$B(L) = B(L) / P$

} ピボット設定

} ピボットで割り算

## 割り算結果

1.000	-0.500	-0.250	0.250
-7871	15748	3938	-11810
-3938	3938	15748	0
3933	-11810	0	15748

A

6.350E-04
0
1.000E+01
0

B

# Gaussの消去法：前進消去 2

$L=1, NN=4$

	1	2	3	4
1	1	-0.500	-0.250	0.250
2	-7871	15748	3938	-11810
3	-3938	3938	15748	0
4	3933	-11810	0	15748

```
For i = L + 1 To NN
```

```
  Q = A(i, L)
```

```
  For j = L + 1 To NN
```

```
    A(i, j) = A(i, j) - Q * A(L, j)
```

```
  Next j
```

```
  B(i) = B(i) - Q * B(L)
```

```
Next i
```

```
Next L
```

Qを掛けて  
引く

(具体的な操作)

2行1列目を1行目に掛け、2行目から引いたものを2行目に代入

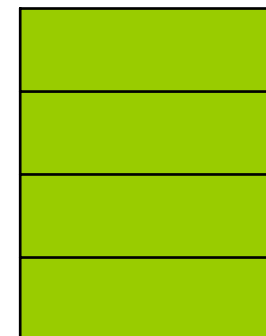
3行1列目を1行目に掛け、3行目から引いたものを3行目に代入

4行1列目を1行目に掛け、4行目から引いたものを4行目に代入

(結果)

1.000	-0.500	-0.250	0.250
0.000	11813.999	1969.750	-9844.249
0.000	1969.750	14763.250	983.500
0.000	-9844.249	983.500	14765.749

A



B

# Gaussの消去法: 前進消去 3

$L=2, NN=4$

ピボットで割り算

1.000	-0.500	-0.250	0.250
0.000	11813.999	1969.750	-9844.249
0.000	1969.750	14763.250	983.500
0.000	-9844.249	983.500	14765.749

2行2列目(ピボット)で2行目を割る

```
‘前進消去
For L = 1 To NN - 1
  P = A(L, L)
  For j = L + 1 To NN
    A(L, j) = A(L, j) / P
  Next j
  B(L) = B(L) / P
```

割り算結果

1.000	-0.500	-0.250	0.250
0.000	1.000	0.167	-0.833
0.000	1969.750	14763.250	983.500
0.000	-9844.249	983.500	14765.749

A

6.350E-04
4.231E-04
1.250E+01
-2.497

B



# Gaussの消去法：前進消去4

$L=2, NN=4$  Qを掛けて引く

	1	2	3	4
1	1.000	-0.500	-0.250	0.250
2	0.000	1.000	0.167	-0.833
3	0.000	1969.750	14763.250	983.500
4	0.000	-9844.249	983.500	14765.749

```

For i = L + 1 To NN
  Q = A(i, L)
  For j = L + 1 To NN
    A(i, j) = A(i, j) - Q * A(L, j)
  Next j
  B(i) = B(i) - Q * B(L)
Next i
Next L
    
```

(具体的な操作)

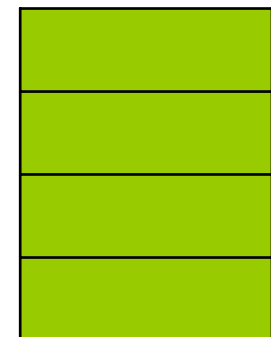
3行2列目を2行目に掛け、3行目から引いたものを3行目に代入

4行2列目を2行目に掛け、4行目から引いたものを4行目に代入

(結果)

1.000	-0.500	-0.250	0.250
0.000	1.000	0.167	-0.833
0.000	0.000	14434.833	2624.833
0.000	0.000	2624.833	6562.833

A



B

# Gaussの消去法：前進消去5

$L=3, NN=4$

ピボットで割り算

1.000	-0.500	-0.250	0.250
0.000	1.000	0.167	-0.833
0.000	0.000	14434.833	2624.833
0.000	0.000	2624.833	6562.833

3行3列目(ピボット)で3行目を割る

```
‘前進消去
For L = 1 To NN - 1
  P = A(L, L)
  For j = L + 1 To NN
    A(L, j) = A(L, j) / P
  Next j
  B(L) = B(L) / P
```

割り算結果

1.000	-0.500	-0.250	0.250
0.000	1.000	0.167	-0.833
0.000	0.000	1.000	0.182
0.000	0.000	2624.833	6562.833

A

6.350E-04
4.231E-04
8.082E-04
1.668

B

# Gaussの消去法：前進消去6

$L=3, NN=4$  Qを掛けて引く

	1	2	3	4
1	1.000	-0.500	-0.250	0.250
2	0.000	1.000	0.167	-0.833
3	0.000	0.000	1.000	0.182
4	0.000	0.000	2624.833	6562.833

```
For i = L + 1 To NN
  Q = A(i, L)
  For j = L + 1 To NN
    A(i, j) = A(i, j) - Q * A(L, j)
  Next j
  B(i) = B(i) - Q * B(L)
Next i
Next L
```

(具体的な操作)

4行3列目を3行目に掛け、4行目から引いたものを4行目に代入

(結果)

1.000	-0.500	-0.250	0.250
0.000	1.000	0.167	-0.833
0.000	0.000	1.000	0.182
0.000	0.000	0.000	6085.533

前進消去終了



B

# Gaussの消去法：後退代入1

ピボットで割り算

1.000	-0.500	-0.250	0.250
0.000	1.000	0.167	-0.833
0.000	0.000	1.000	0.182
0.000	0.000	0.000	6085.533

4行4列目(ピボット)で4行目を割る

後退代入  
 $X(NN) = B(NN) / A(NN, NN)$

割り算結果

1.000	-0.500	-0.250	0.250
0.000	1.000	0.167	-0.833
0.000	0.000	1.000	0.182
0.000	0.000	0.000	1.000

A

6.350E-04
4.231E-04
8.082E-04
-7.452E-05

B

# Gaussの消去法：後退代入2

1.000	-0.500	-0.250	0.250	<b>U2</b>	6.350E-04
0.000	1.000	0.167	-0.833	<b>V2</b>	4.231E-04
0.000	0.000	1.000	0.182	<b>U3</b>	8.082E-04
0.000	0.000	0.000	1.000	<b>V3</b>	-7.452E-05

**後退代入**

$$\begin{cases}
 V3 = -7.452E-05 \\
 U3 = 8.082E-04 - 0.182 * V3 \\
 V2 = 4.231E-04 - 0.167 * U3 + 0.833 * V3 \\
 U2 = 6.350E-04 + 0.500 * V2 + 0.250 * U3 - 0.250 * V3
 \end{cases}$$

(解)

U2	V2	U3	V3
9.71E-04	2.24E-04	8.22E-04	-7.45E-05

```
For L = NN - 1 To 1 Step -1
```

```
  S = B(L)
```

```
  For j = L + 1 To NN
```

```
    S = S - A(L, j) * X(j)
```

```
  Next j
```

```
  X(L) = S
```

```
Next L
```

'計算結果

```
For i = 1 To NN
```

```
  Cells(32, 3 + i).Select
```

```
  ActiveCell.FormulaR1C1 = X(i)
```

```
Next i
```

```
End Sub
```

# 小テスト(氏名: )

(1) 前進消去2, 3, 5におけるベクタBの内容を計算せよ



前進消去2



前進消去4



前進消去6

(2) 講義に関する感想等を述べよ。