

CONTENTS

ビジュアルヘルプ – Integrate2D	2
Integrate2D コマンドのヘルプ	2

Integrate2D コマンドのヘルプ

Integrate2D [/Q /Z=zFlag /OPTS=options] [keyword = value ...]

Integrate2D コマンドは、実数値のユーザー定義関数またはウェーブの2次元数値積分を計算します。コマンドの結果は変数 V_value に格納され、エラーがない場合は変数 V_Flag が 0 に設定されます。

このコマンドは、Igor Pro 7.0 で追加されました。

フラグ

/OPTS=op 積分オプションを設定します。
デフォルトでは、x 積分と y 積分は、いずれもアダプティブ台形法を使って実行されます。

op は、x および y の積分方法を選択するために設定するビット単位のパラメーターです。

x には 1 ビット、y には 1 ビットを設定してください：

Bit 0 : Y 軸方向に台形 (1)
Bit 1 : Y 軸方向にロンバーク積分 (2)
Bit 2 : Y 軸方向にガウス積分 (4)
Bit 3 : X 軸方向に台形 (8)
Bit 4 : X 軸方向にロンバーク積分 (16)
Bit 5 : X 軸方向にガウス積分 (32)

ビットの設定の詳細については、「ビットパラメーターの設定」を参照してください。

これらの定数を使うと、例えば、/OPTS=(2 | 32) を使って、Y 方向にロンバーク積分、X 方向にガウス求積法を使うように指定することができます。

/Q 履歴エリアへの出力を抑制します。

/Z=zFlag zFlag を 1 に設定すると、エラー報告を抑制します。

キーワード

epsilon=ep 収束パラメーターを指定します。
デフォルトは ep =1e-5 です。
値が小さいほど、積分結果はより正確になりますが、その代わりに計算時間が長くなります。

Integrand=uF 積分するユーザー定義関数を指定します。
詳細については、以下の「被積分関数」を参照してください。

innerLowerLimit=y1
この下限が固定されている場合（つまり、x の関数ではない場合）、内側の積分の下限を指定します。
この下限に関数を指定する必要がある場合は、innerLowerFunc キーワードを参照してください。

innerUpperLimit=y2
この上限が固定されている場合（つまり、x の関数ではない場合）、内側の積分の上限を指定します。

この上限に関数を指定する必要がある場合は、innerUpperFunc キーワードを参照してください。

innerLowerFunc=y1Func

内側の積分の下限にユーザー定義関数を指定します。
以下の「限界関数」を参照してください。

innerUpperFunc=y2Func

内側の積分の上限にユーザー定義関数を指定します。
以下の「限界関数」を参照してください。

outerLowerLimit=x1

外側の積分の下限を指定します。

outerUpperLimit=x2

外側の積分の上限を指定します。

paramWave=pWave

pWave パラメーターとして、さまざまなユーザー定義関数に渡すウェーブを指定します。
ウェーブには、積分式または積分限界を評価するために必要な任意の数の値を含めることができます。

paramWave を省略すると、関数への pWave パラメーターは NULL になります。

srcWave=mWave

2次元積分を行う必要がある場合、ユーザー定義関数で補間データを返す代わりに、データを直接指定することができます。

mWave は2次元ウェーブでなければなりません。

より高い次元のウェーブも使用できますが、積分にはウェーブの最初の層のみが使用されます。

被積分関数

Integrate2D は、integrand キーワードを使って指定した、ユーザー定義の積分関数の一般的な2次元積分を計算します。

被積分関数は次の形式です：

```
Function integrandFunc(pWave,inX,inY)
    Wave/Z pWave
    Variable inX,inY

    ... do something
    return result
End
```

この関数は任意の名前で指定できます。

integrandFunc は単なる例です。

この関数は、示されているパラメーターを受け取り、実数値の結果を返す必要があります。

NaN を返すと、積分は終了します。

pWave は、paramWave キーワードを使って指定するパラメーターウェーブです。

このコマンドは、積分関数の呼び出しごとにこのウェーブを渡します。

Integrate2D を呼び出す時に paramWave を省略すると、pWave は NULL になります。

限界関数

限界関数は、内側の積分が定数ではなく x の関数である場合、その積分の下限または上限を提供します。
 限界関数は、innerLowerFunc および/または innerUpperFunc キーワードを使って指定します。
 限界関数の形式は次のとおりです：

```
Function limitFunction(pWave,inX)
    Wave/Z pWave
    Variable inX

    ... do something
    return result
End
```

詳細

この演算は、次の形の一般的な 2 次元積分を計算します。

$$I = \int_{x1}^{x2} dx \int_{y1(x)}^{y2(x)} f(x,y) dy$$

ここで、y1 と y2 は一般に x の関数ですが、単純な定数である可能性もあります。
 また、f(x,y) は実数値関数です。
 この積分は、「外側の」積分を考慮して評価されます。

$$I = \int_{x1}^{x2} G(x) dx$$

ここで、G(x) は「内側の」積分を表します。

$$G(x) = \int_{y1(x)}^{y2(x)} f(x,y) dy$$

このコマンドでは、内側の積分と外側の積分に対して異なる積分アルゴリズムを指定できます。
 最もシンプルな積分アルゴリズムは台形法です。
 ロムベルグ積分を使うことで、計算の精度を向上させることができます。
 一方、ガウス積分の性能は、被積分関数の性質に大きく依存します。

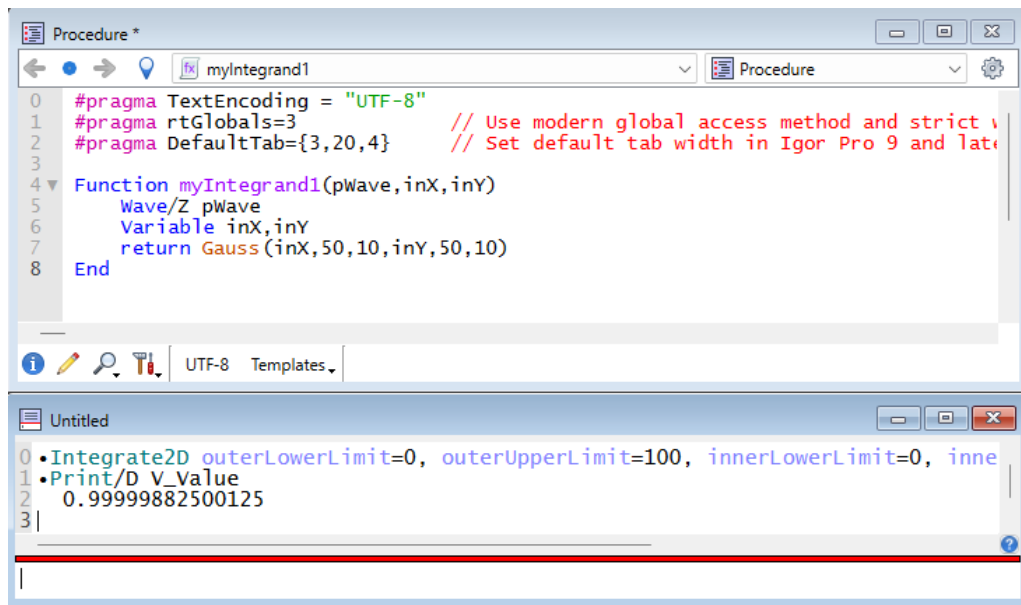
例 1：2 次元関数を固定された範囲で積分する

仮に、組み込みの 2 次元ガウス関数の正規化を確認したい場合を考えます。
 ユーザー定義関数は次のように定義されます：

```
Function myIntegrand1(pWave,inX,inY)
    Wave/Z pWave
    Variable inX,inY
    return Gauss(inX,50,10,inY,50,10)
End
```

積分を実行するには、次のコマンドを実行してください：

```
Integrate2D outerLowerLimit=0, outerUpperLimit=100, innerLowerLimit=0,
    innerUpperLimit=100, integrand=myIntegrand1
Print/D V_Value
```



例 2 : 関数限界を使って 2 次元関数を積分する

この例では、半径 1 の球の体積を計算します。

これを行うには、対称性を利用して、1 つの八分円のみに体積を積分します。

積分の範囲は、x 方向が $[0,1]$ 、y 方向が $[0,\sqrt{1-x^2}]$ です。

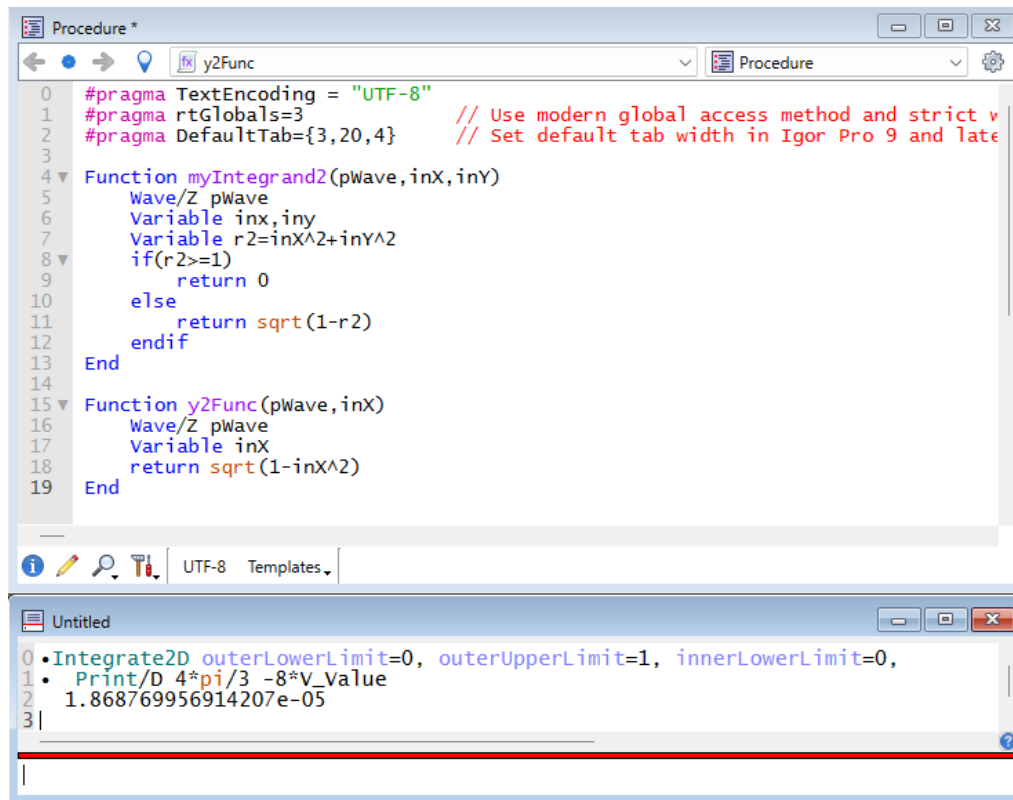
この場合、2 つのユーザー定義関数を定義する必要があります：1 つは被積分関数用、もう 1 つは内側の積分の上限用です：

```
Function myIntegrand2(pWave,inX,inY)
    Wave/Z pWave
    Variable inx,iny
    Variable r2=inX^2+inY^2
    if(r2>=1)
        return 0
    else
        return sqrt(1-r2)
    endif
End
```

```
Function y2Func(pWave,inX)
    Wave/Z pWave
    Variable inX
    return sqrt(1-inX^2)
End
```

積分を実行するには、次のコマンドを実行してください：

```
Integrate2D outerLowerLimit=0, outerUpperLimit=1, innerLowerLimit=0,
            innerUpperFunc=y2Func, integrand=myIntegrand2
Print/D 4*pi/3 -8*V_Value // 計算エラー
```

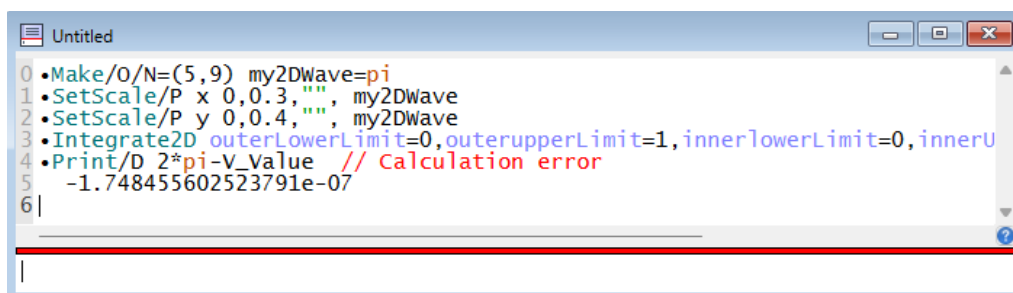


被積分関数は、 r^2 が 1 を超えないことを検査することに注意してください。
 これは、`sqrt` 関数が $r^2 > 1$ の場合に NaN を返すためです。
 これは浮動小数点数の丸め誤差により発生する可能性があります。
 NaN を返すと、積分が終了します。

例 3 : 固定限界を使って 2D ウェーブを積分する

次の例では、ウェーブ `my2DWave` で定義される表面と平面 $z=0$ の間の体積を、 x 方向の積分限界 $[0,1]$ 、 y 方向の積分限界 $[0,2]$ で計算します。
 簡単にするため、ウェーブの値は定数 (n) に設定しています。

```
Make/O/N=(5,9) my2DWave=pi
SetScale/P x 0,0.3,"", my2DWave
SetScale/P y 0,0.4,"", my2DWave
Integrate2D
    outerLowerLimit=0,outerUpperLimit=1,innerLowerLimit=0,innerUpperLimit=2,s
rcWave=
    my2DWave
Print/D 2*pi-V_Value // 計算エラー
```



参照

Integrate、Differentiate、Integrate1D、area、areaXY