MITSUBOSHI

第42回　FrontISTR研究会

自動時間増分調整・カットバック機能の活用事例

2018/3/16

三ツ星ベルト株式会社

徳田明彦

発表内容

1. 自動時間増分調整・カットバック機能の概要
2. 当社における活用事例と効果
3. 解析制御設定の詳細
4. その他

1. 自動時間増分調整・カットバック機能の概要

MITSUBOSHI

自動時間増分調整機能（時間増分の自動設定機能）

- FrontISTR 5.0α (2018年2月1日・GitHubにて公開）から実装。
- 非線形静解析（!SOLUTION, TYPE=NLSTATIC）におけるサブステップの時間増分（Δt）を、収束状況に応じて自動的に調整する。
  - 収束しにくい場合　→　Δtを小さくする。
  - 収束しやすい場合　→　Δtを大きくする。

- 「収束しやすさ」の判断には、「ニュートン・ラフソン法(NR)ループ」、および「接触ループ」の回数を用いる。
- 非線形静解析の解析進捗の管理を、（擬似的な）時間（time）とする。これは多くの商用ソフトと同じ考え方。

カットバック機能

- FrontISTR 5.0α (2018年2月1日・GitHubにて公開)から実装。
- 非線形静解析(!SOLUTION, TYPE=NLSTATIC)において、あるサブステップが収束しなかった場合、前のサブステップが完了した時点に状態を戻し、時間増分(Δt)を小さくしてそのサブステップをやり直す。
- 「カットバック発動(収束しない)」の判断には、「ニュートン・ラフソン法(NR)ループの上限回数」、「接触ループの上限回数」、「残差力の値」、を用いる。
  - 残差力の値: NRの上限回数に達していなくても、残差力がある値を超えたら発散傾向にあると判断して即時NRを打ち切り、効率を上げる。
- 自動増分の場合は常に有効になる。固定増分で収束に失敗した場合は有効にならず、その時点で終了する。
- 以下の場合にはエラー終了する
  - カットバックの連続した回数が設定した値に達した場合
  - Δtが設定した最小時間幅を下回った場合

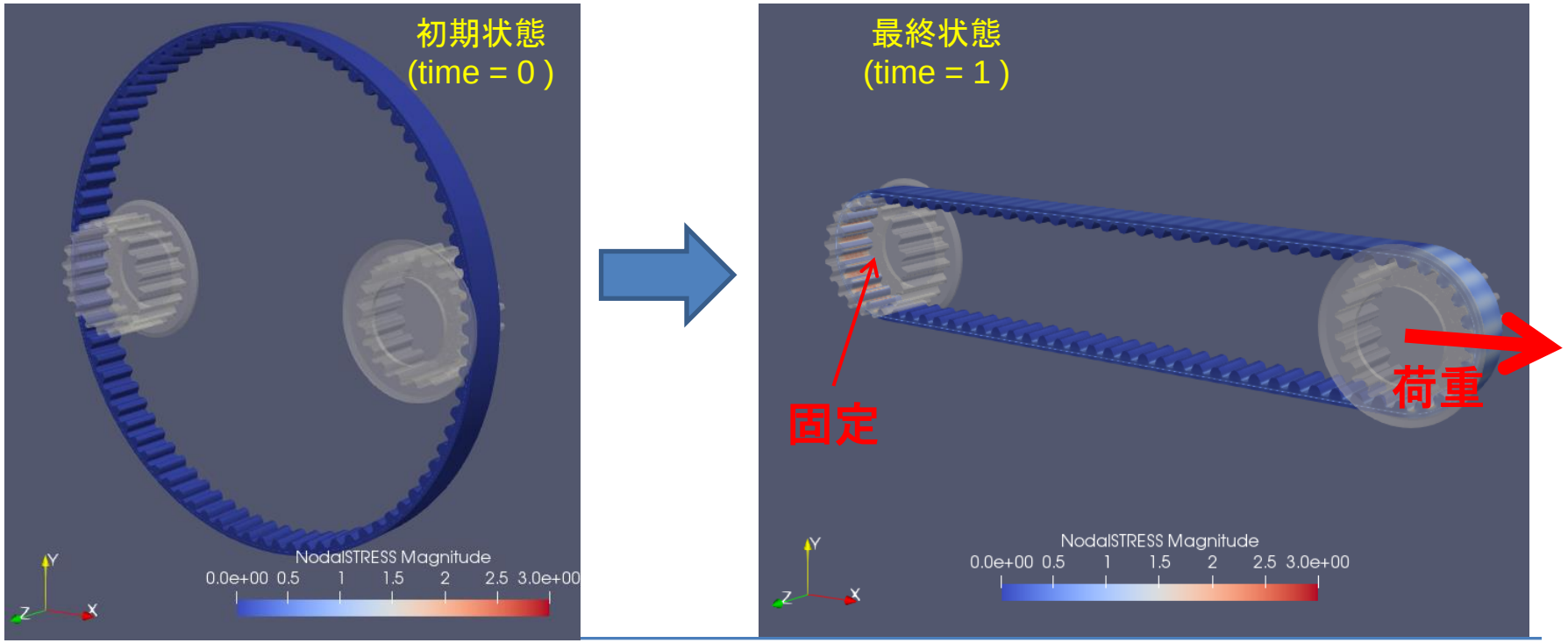
時刻指定結果出力機能

- FrontISTR 5.0α (2018年2月1日・GitHubにて公開)から実装。
- 自動時間増分調整機能やカットバック機能を用いると、サブステップ間隔が不均等になり、等間隔で結果を得たい場合に不便。
- そこで、指定した時刻では必ず立ち止まって計算を行い、結果を出力する機能を追加した(時刻点を定義する **!TIME_POINTS** カードを新設)。
  - 自動時間増分調整機能をONにしているときのみ有効。
- 指定した時刻点で必ず計算が行われるよう、増分が調整される
  - 時刻点に合わせるために増分が減少した場合、次の増分では元の増分に復帰する。
- 指定した時刻点では、(!WRITEカードの間隔指定によらず)
  必ず結果が出力される
  - !WRITEの出力指定時点に追加する形で、!TIME_POINTS指定時刻で出力が行われる。
  - !TIME_POINTSによる指定点でのみ結果がほしい場合は、!WRITE, FREQUENCYの値を十分大きく取る。

2. 当社における活用事例と効果

歯付ベルトの解析モデル

- 歯付ベルト（いわゆる“タイミングベルト”）は、ベルトの歯がプーリの歯と噛み合って動力を伝達する。
- ベルトを2つのプーリ間に掛け、片側のプーリを固定、もう一つのプーリに荷重を与えて引っ張り、ベルトを張るところまでの計算を実施。



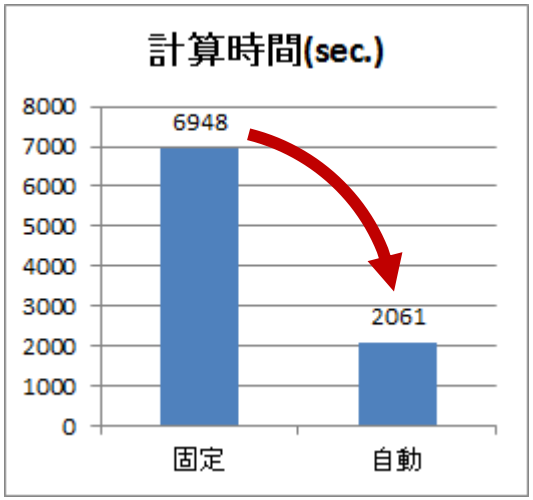
初期状態ではベルトとプーリが離れており、不安定（弱いバネで支持している）。

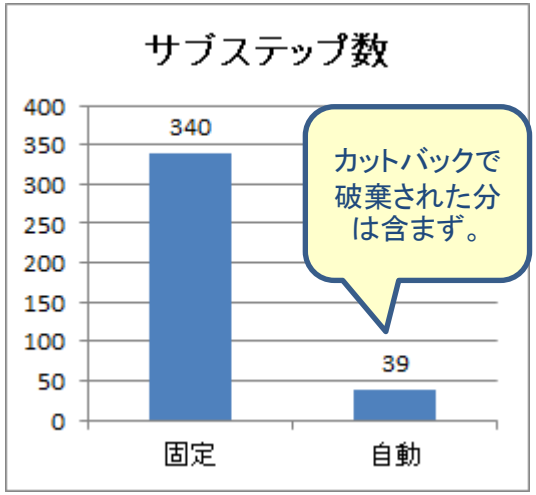
歯付ベルトの解析条件等

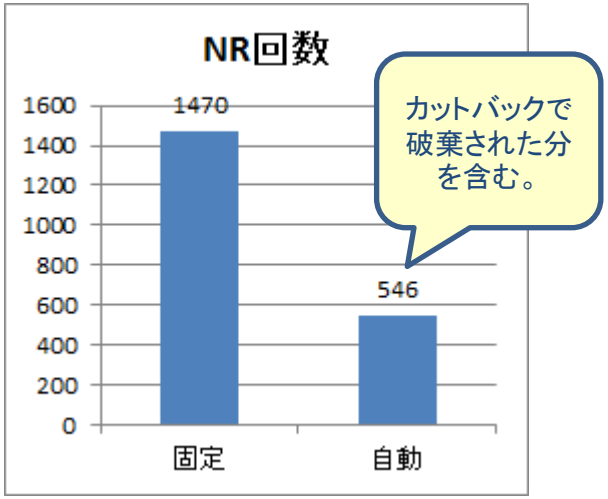
- メッシュ： 節点数：166,884 、要素数：136,720
  - 主体は六面体1次要素（361, Fbar要素）で、材料はゴム(Mooney-Rivlin)。
  - 心線はトラス要素で、線形材料。
- ソフト： FrontISTR 5.0α（2018/2/1公開版）
- 解析設定等：
  - 非線形解析(NLSTATIC)、接触有り（Slagrange, 有限滑り）・摩擦無し。
  - ソルバー： MKL pardiso（領域分割並列無し・スレッド並列のみ）
  - NRループ収束判定値： 1.0E-8
  - NRループ上限回数： 10
  - 接触ループ上限回数： 15
  - サブステップの設定
    - 固定増分： サブステップ分割数 = 340 , 結果出力間隔 = 34（10等分）
    - 自動増分： ステップ開始時刻 = 0, 終了時刻 = 1.0, 結果出力指定 = 0.1間隔
      時間増分初期値 = 0.01, 上限値 = 1.0, 下限値 = 0.00001
      ＊自動増分・カットバックの設定（!AUTOINC_PARAM）は後述

効果検証

- 計算機： デスクトップワークステーション
  - Xeon E5-2687 v3 (3.1GHz, 10コア) デュアル構成(計20コア)
  - メモリ: 256GB、 OS: RedHat Linux 7.、コンパイラ:インテルコンパイラ2017
- 計算コスト比較







自動時間増分調整・カットバック機能の適用により、計算時間が固定(従来)の約30%に短縮された。

固定増分の場合、最も効率的な分割数は解析ケース毎に異なる(今回は340)。それを突き止めるために数回の試行が必要なので、それを含めた実際の時間短縮効果は更に大きい。

解析進捗状況の出力

- 「FSTR.sta」ファイルに解析計算の進捗状況が出力されるようになった。

STEP	SUB STEP	STAT	# of CONT ITER	MAX # NEWTON ITER	TOT # NEWTON ITER	START TIME	TIME INC	END TIME	MESSAGE
1	1	1F	2	10	11	0.0000E+00	1.0000E-02	0.0000E+00	Failed to converge due to MAXITER.
1	1	2F	2	10	11	0.0000E+00	5.0000E-03	0.0000E+00	Failed to converge due to MAXITER.
1	1	S	7	5	25	0.0000E+00	2.5000E-03	2.5000E-03	
1	2	S	7	3	16	2.5000E-03	2.5000E-03	5.0000E-03	
1	3	S	3	2	4	5.0000E-03	2.5000E-03	7.5000E-03	
1	4	S	3	2	4	7.5000E-03	2.5000E-03	1.0000E-02	
1	5	S	2	2	3	1.0000E-02	3.7500E-03	1.3750E-02	
1	6	S	2	2	3	1.3750E-02	5.6250E-03	1.9375E-02	
1	7	S	3	2	5	1.9375E-02	8.4375E-03	2.7813E-02	
1	8	S	3	2	5	2.7813E-02	1.2656E-02	4.0469E-02	
1	9	S	10	5	28	4.0469E-02	1.8984E-02	5.9453E-02	
1	10	S	2	3	5	5.9453E-02	1.8984E-02	7.8437E-02	
1	11	S	5	3	10	7.8437E-02	1.8984E-02	9.7422E-02	
1	12	S	2	2	3	9.7422E-02	2.5781E-03	1.0000E-01	
1	13	S	7	8	21	1.0000E-01	4.2715E-02	1.4271E-01	
1	14	S	4	6	10	1.4271E-01	4.2715E-02	1.8543E-01	
1	15	S	3	3	5	1.8543E-01	1.4570E-02	2.0000E-01	
1	16	1F	1	10	10	2.0000E-01	4.2715E-02	2.0000E-01	Failed to converge due to MAXITER.
1	16	S	4	4	8	2.0000E-01	2.1357E-02	2.2136E-01	
1	17	S	3	4	6	2.2136E-01	2.1357E-02	2.4271E-01	
1	18	S	4	6	10	2.4271E-01	3.2036E-02	2.7475E-01	
1	19	S	8	6	21	2.7475E-01	2.5249E-02	3.0000E-01	
1	20	1F	15	8	51	3.0000E-01	3.2036E-02	3.0000E-01	Failed to converge due to MAXCONTITER.
1	20	S	6	4	14	3.0000E-01	1.6018E-02	3.1602E-01	

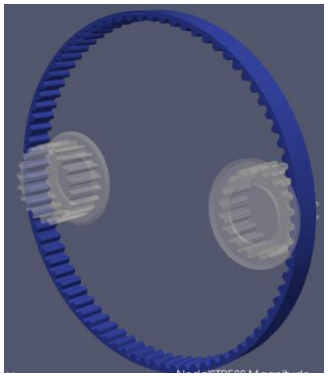
NR回数上限でカットバック

出力指定時刻に立ち止まるため、一時的に時間増分が短くなる。

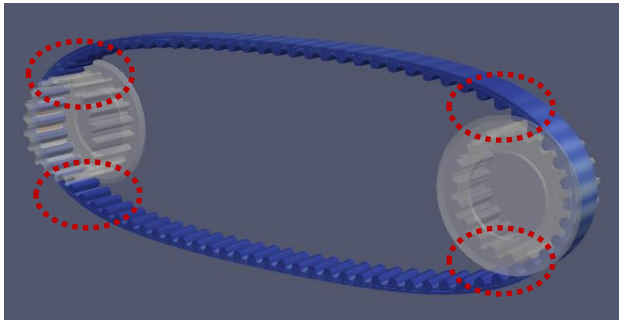
接触ループ回数上限でカットバック

時間増分の推移（自動の場合）

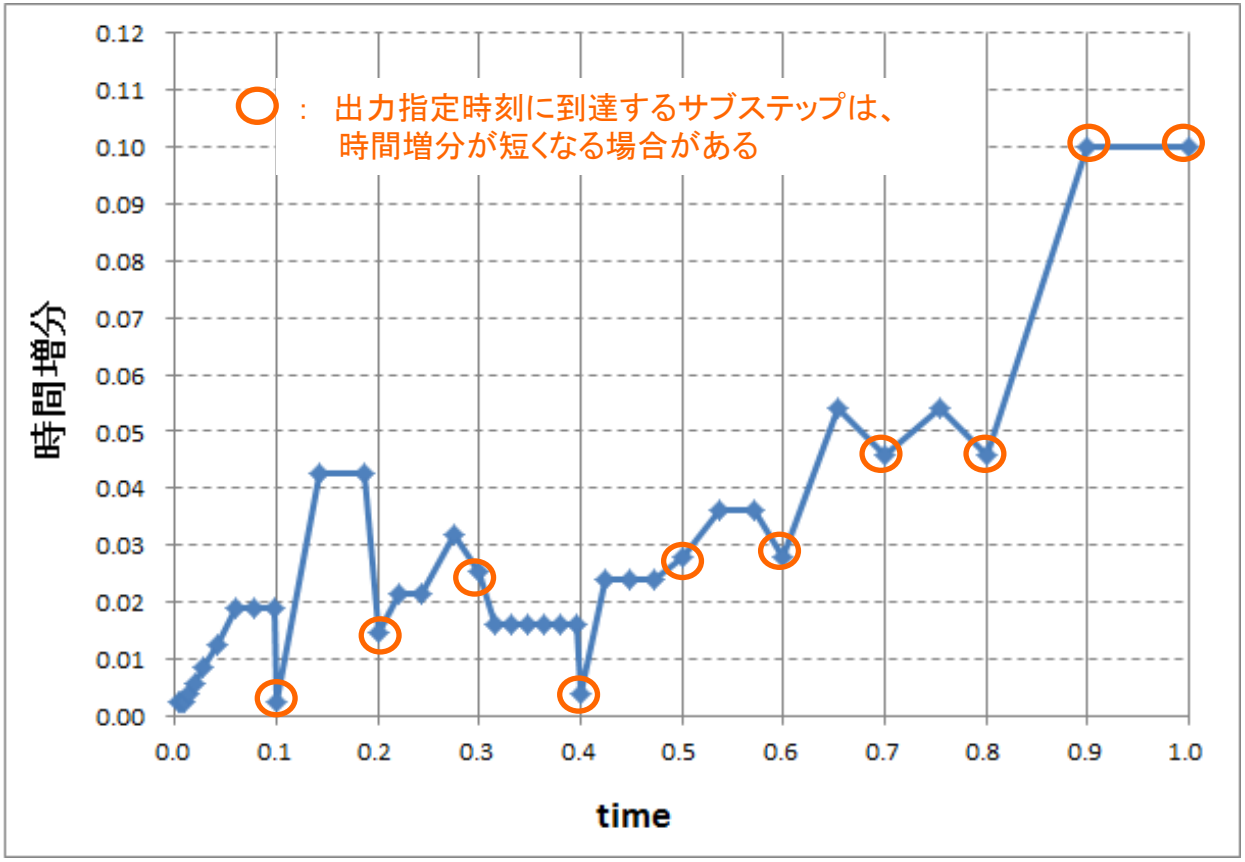
- 難所は、time=0（開始直後）とtime=0.3～0.4付近の2個所。



time=0: ベルトとプーリが離れていて不安定。



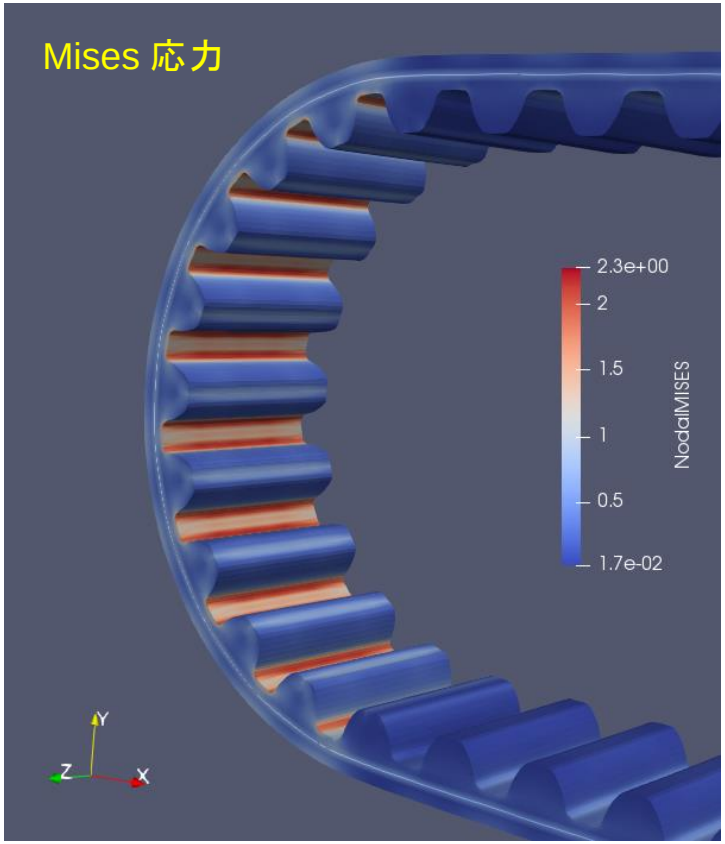
time=0.3: ここからtime=0.4にかけて接触節点数が急速に増える。



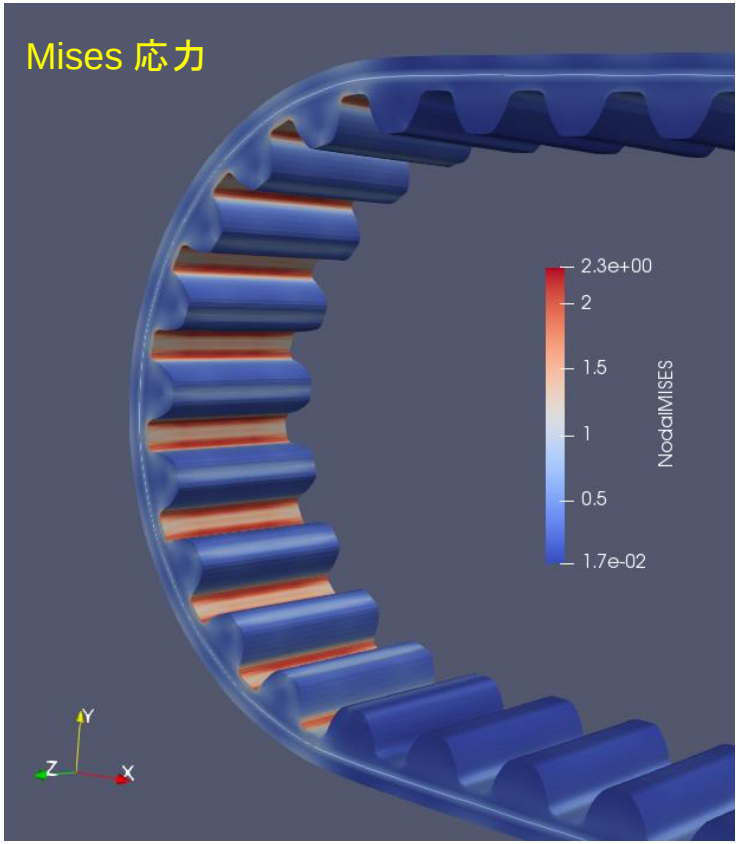
- 横軸のtime値は、各サブステップの終了時刻を基準とした。
- カットバックで破棄されたサブステップは除外している。

解析結果（最終状態）比較

- 解析結果に差異はない。



固定増分



自動増分

3. 解析制御設定の詳細

自動時間増分調整・カットバックの設定ヘッダ（!AUTOINC_PARAM）（1）

これを使用する!STEPより前に記述しなければならない。

!AUTOINC_PARAM, NAME=AP1 ← このパラメータの名前（!STEPで指定する）

0.67, 9999, 9999, 9999, 2 ← Δt 減少条件：**倍率, NR最大, NR合計, 接触反復, 必要連続回数**

1.5, 4, 9999, 5, 2 ← Δt 増加条件：**倍率, NR最大, NR合計, 接触反復, 必要連続回数**

0.50, 10 ← カットバック：**倍率, 連続回数上限**

- 倍率：自動増分調整、またはカットバック発動時、これを前回のΔtに乗じて今回のΔtとする。
  - デフォルト　減少：0.25、　増加：1.25
- NR最大：前回のサブステップにおけるNRの最大回数（*1）
  - Δt減少条件：NR最大回数がこの値を上回る（デフォルト：10）。
  - Δt増加条件：NR最大回数がこの値以下である（デフォルト：1）。
- NR合計：前回のサブステップにおけるNRの合計回数（*1）
  - Δt減少条件：NR合計回数がこの値を上回る（デフォルト：50）。
  - Δt増加条件：NR合計回数がこの値以下である（デフォルト：1）。

*1) 接触解析の場合、接触ループとNRの2重ループになっているので、各接触ループにおけるNR回数の最大値と、全接触ループのNR回数合計値の2通りで判断する。接触無しの場合、最大回数=合計回数。

自動時間増分調整・カットバックの設定ヘッダ（!AUTOINC_PARAM）（2）

これを使用する!STEPより前に記述しなければならない。

!AUTOINC_PARAM, NAME=AP1 ← このパラメータの名前（!STEPで指定）

0.67, 9999, 9999, 9999, 2 ← Δt 減少条件：**倍率, NR最大, NR合計, 接触反復, 必要連続回数**

1.5, 4, 9999, 5, 2 ← Δt 増加条件：**倍率, NR最大, NR合計, 接触反復, 必要連続回数**

0.50, 10 ← カットバック：**倍率, 連続回数上限**

- 接触反復：前回のサブステップにおける、接触ループの回数
  - Δt減少条件：接触ループ回数がこの値を上回る（デフォルト：10）。
  - Δt増加条件：接触ループ回数がこの値以下である（デフォルト：1）。
- 各条件の組み合わせ
  - Δt減少条件：NR最大、NR合計、接触反復のいずれかが満たされると条件成立（OR）。
  - Δt増加条件：NR最大、NR合計、接触反復の全てが満たされると条件成立(AND)。
- 必要連続回数：
  - 減少・増加条件が成立するサブステップが、「必要連続回数」回連続すると、実際に自動増分調整が発動する（デフォルト　減少：1、　増加：2）。

＊接触無しの場合、
接触ループ回数=0
として考える。

自動時間増分調整・カットバックの設定ヘッダ（!AUTOINC_PARAM）（3）

これを使用する!STEPより前に記述しなければならない。

!AUTOINC_PARAM, NAME=AP1 ← このパラメータの名前（!STEPで指定）

0.67, 9999, 9999, 9999, 2 ← Δt 減少条件：**倍率, NR最大, NR合計, 接触反復，必要連続回数**

1.5, 4, 9999, 5, 2 ← Δt 増加条件：**倍率, NR最大, NR合計, 接触反復，必要連続回数**

0.50, 10 ← カットバック：**倍率, 連続回数上限**

- カットバック条件
  - 倍率：　前回失敗したサブステップを、時間増分を「倍率」倍にしてやり直す（デフォルト：0.25）。
  - 連続回数上限：　カットバックの連続発動回数がこの値を超えると、JOB終了（デフォルト：5）。

- 上記パラメータの意味
  - Δt 減少：「NR最大＞9999 .or. NR合計＞9999 .or. 接触反復＞9999」が成り立つサブステップが、2回連続したとき、今回のΔtは前回の0.67倍になる。
  - Δt 増加：「NR最大≦4 .and. NR合計≦9999 .and. 接触反復≦5」が成り立つサブステップが、2回連続したとき、今回のΔtは前回の1.5倍になる。
  - カットバック：　前回失敗したサブステップを、時間増分を0.5倍してやりなおす。ただし、カットバック発動が連続して10回を超えると（または時間増分の最小値を下回ると）JOB終了する。

＊本事例のパラメータでは、時間増分の減少はカットバックのみで行い、自動時間増分調整はカットバックで減少した時間増分を回復させるためだけに用いている。

結果出力時刻リストの定義（!TIME_POINTS）（1）

これを使用する!STEPより前に記述しなければならない。

このパラメータの名前（!STEPで指定）

時刻記述方式：GENERATE式（開始, 終了, 間隔で指定）

!TIME_POINTS, NAME=TP1, GENERATE, TIME=TOTAL

複数ステップの場合、時刻を通算する。

0.0, 1.0, 0.1 ← 時刻記述：**開始, 終了, 間隔**

- GENERATE　有り：　結果出力時刻を、「開始, 終了, 間隔」で指定する。
  　　無し：　結果出力時刻を個々に記述する。
- TIME　　STEP（デフォルト）：　複数ステップの場合、各ステップ開始時点からの時刻とする。
  　　TOTAL　　：　　〃　　、第1ステップからの通算時刻とする。
  　　＊1JOBに単一ステップしかない場合、TOTALとSTEPは同じ。
- 時刻記述
  - GENERATE有り：「開始, 終了, 間隔」を1行にカンマ区切りで横に並べる。
  - GENERATE無し：　出力時刻を、1行に1件のみ記述し、必要な行数縦に並べる。

結果出力時刻リストの定義（!TIME_POINTS）（2）

これを使用する!STEPより前に記述しなければならない。

```
!TIME_POINTS, NAME=TP1, GENERATE, TIME=TOTAL
0.0, 1.0, 0.1
```

GENERATE有り

この2つは同じ意味

```
!TIME_POINTS, NAME=TP1,  TIME=TOTAL
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
```

GENERATE無し

＊時刻点の入力は昇順に行わなければならない

ステップの定義（!STEP）（1）

既存 !STEPの拡張

!STEP, SUBSTEPS=1000000, CONVERG=1.0e-8, MAXITER=10, MAXCONTITER=15, MAXRES=1E+2, INC_TYPE=AUTO, TIMEPOINTS=TP1, AUTOINCPARAM=AP1

0.01, 1.0, 0.00001, 1.0

- 新規追加パラメータ
  - INC_TYPE： FIXED（固定増分／デフォルト），AUTO（自動時間増分調整）
  - TIMEPOINTS： 出力時刻リスト名（!TIME_POINTSで指定）。
  - AUTOINCPARAM： 自動時間増分調整パラメータ名（!AUTOINC_PARAMで指定）。
  - MAXCONTITER： 接触ループの回数上限（デフォルト：10）
  - MAXRES： 残差力の上限。これを超えると、NR回数上限に達していなくてもカットバックする。（デフォルト：1.0E+10）
- 修正
  - SUBSTEPS： INC_TYPE=AUTOの場合はサブステップ数の上限とする。FIXEDの場合は従来通りサブステップ分割数。

ステップの定義（!STEP）（2）

既存 !STEPの拡張

!STEP, SUBSTEPS=1000000, CONVERG=1.0e-8, MAXITER=10, MAXCONTITER=15, MAXRES=1E+2, INC_TYPE=AUTO, TIMEPOINTS=TP1, AUTOINCPARAM=AP1

0.01, 1.0, 0.00001, 1.0

INC_TYPE=AUTOの場合：

初期時間増分, ステップの時間幅，サブステップ時間増分の下限，サブステップ時間増分の上限

（INC_TYPE=FIXEDの場合は従来通り（STATICの場合省略可）：**サブステップの時間増分, ステップの時間幅**

- サブステップ時間増分の下限：
  - 自動時間増分調整によって、時間増分がこの値を下回らない。
  - カットバックによって時間増分がこの値を下回った場合、JOBは終了する。
- サブステップ時間増分の上限：
  - 自動時間増分調整によって、時間増分がこの値を上回らない。
- ステップの時間幅： この!STEPの経過時間

4. その他

F-bar要素（361：六面体1次要素）の使用 （1）

- 非線形静解析での361要素のデフォルトはB-bar要素だが、「大ひずみ．かつ．大回転」に対応していない（2016/11/28・第32回FrontISTR研究会「大ひずみ・大回転解析の注意点」参照）。
- FrontISTR 5.0αにて、「大ひずみ．かつ．大回転」でも正しく計算できる「F-bar」要素が追加されたので、今回の事例ではそれを用いた。

- F-bar要素の適用手順
  - STEP1： メッシュファイル(*.msh）で、F-bar要素を適用する!SETIONが何番目に登場するか、調べてメモしておく。

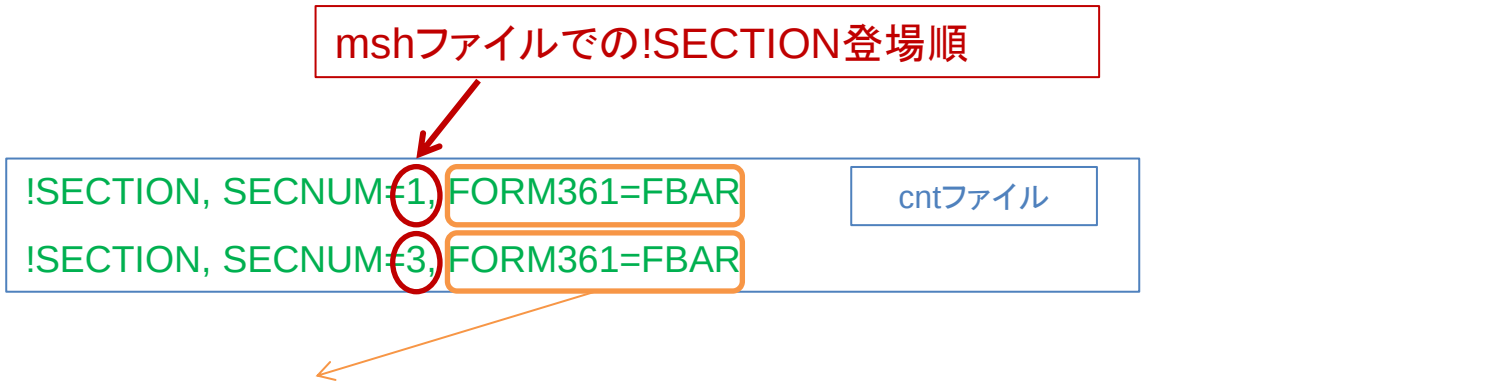
```
!SECTION, TYPE=SOLID, EGRP=M2FMAT_RUBBER, MATERIAL=RUBBER
 1.0000000000000E+00
!SECTION, TYPE=SOLID, EGRP=M2FMAT_TRUSS, MATERIAL=TRUSS
 2.5000000000000E-01
!SECTION, TYPE=SOLID, EGRP=M2FMAT_STEEL, MATERIAL=STEEL
 1.0000000000000E+00
!SECTION, TYPE=SOLID, EGRP=M2F_TRUSS_GEO1, MATERIAL=TRUSS
 5.0000000000000E-01
```

mshファイル

1番目と3番目の!SECTIONに適用したい

F-bar要素（361：六面体1次要素）の使用 （2）

- F-bar要素の適用手順
  - STEP2： 解析制御ファイル(*.cnt)に、!SECTIONを追加する。
    - SECNUM= に、mshファイルで検索した、適用する!SECTIONの順番を記述する。
    - 複数の!SECTIONが有る場合、複数行記述する。



自動時間増分調整機能のカスタマイズ

- 自動時間増分調整機能の条件を変更・追加を行う場合
- 変更個所：
  - **src/analysis/static/fstr_Ctrl_TimeInc.f90**
    - subroutine fstr_TimeInc_SetTimeIncrement　の内容を変更する。

```
!decrease condition
to_be_decreased = .false.
if( NRstatI(knstMAXIT) > pAinc%NRbound_s(knstMAXIT) ) to_be_decreased = .true.
if( NRstatI(knstSUMIT) > pAinc%NRbound_s(knstSUMIT) ) to_be_decreased = .true.
if( NRstatI(knstCITER) > pAinc%NRbound_s(knstCITER) ) to_be_decreased = .true.

!increase condition
to_be_increased = .true.
if( NRstatI(knstMAXIT) > pAinc%NRbound_l(knstMAXIT) ) to_be_increased = .false.
if( NRstatI(knstSUMIT) > pAinc%NRbound_l(knstSUMIT) ) to_be_increased = .false.
if( NRstatI(knstCITER) > pAinc%NRbound_l(knstCITER) ) to_be_increased = .false.
```

NR最大
NR合計
接触反復

「to_be_decreased が true になるサブステップ」 が、pAinc%NRtimes_s（繰返し数）回連続すると、時間増分が減少。

「to_be_increased　が true になるサブステップ」 が、pAinc%NRtimes_l（繰返し数）回連続すると、時間増分が増加。

カットバック機能のカスタマイズ （1）

- カットバック条件を変更・追加する場合。
- 変更個所（1）：カットバックを発動させる判断をするところ
  - **src/analysis/static/fstr_solve_NonLinear.f90** （非線形静解析のループを回すところ）
    - 現状では、「接触無し（fstr_Newton）」、「拡張Lagrange接触（fstr_Newton_contactALag）」、「標準Lagrange接触（fstr_Newton_contactSLag）」がそれぞれ別のサブルーチンに分かれているので、必要なものを改変する。
  - 例：標準Lagrange（fstr_Newton_contactSLag）の場合。

①NR反復回数上限、および残差力上限によるカットバック（NRループの内側・ループの最後）

カットバック発動の判断if文

```
622       ! ----- check divergence
623       if( iter == fstrSOLID%step_ctrl(cstep)%max_iter .or. res > fstrSOLID%step_ctrl(cstep)%maxres ) then
624         if( hecMESH%my_rank == 0) then
625           write(   *,'(a,i5,a,i5)') '     ### Fail to Converge  : at total_step=', cstep, '  sub_step=', sub_step
626         end if
627         fstrSOLID%NRstat_i(knstMAXIT) = max(fstrSOLID%NRstat_i(knstMAXIT),iter) ! logging newton iteration(maxtier)
628         fstrSOLID%NRstat_i(knstSUMIT) = fstrSOLID%NRstat_i(knstSUMIT) + iter    ! logging newton iteration(sumofiter)
629         fstrSOLID%NRstat_i(knstCITER) = count_step                              ! logging contact iteration
630         fstrSOLID%CutBack_stat = fstrSOLID%CutBack_stat + 1   連続発動回数の加算
631         if( iter == fstrSOLID%step_ctrl(cstep)%max_iter ) fstrSOLID%NRstat_i(knstDRESN) = 1
632         if( res > fstrSOLID%step_ctrl(cstep)%maxres     ) fstrSOLID%NRstat_i(knstDRESN) = 2
633         return
634       end if
```

カットバック理由コードの設定

- カットバック発動条件を追加する場合。
  - カットバック発動判断のif文に、新たな発動条件を「OR」で追加する。
  - カットバック理由コードの設定を追加する。

iter: NRループ回数、res: 残差力

fstrSOLID%step_ctrl(cstep)%xxxxxxxx :

!STEPから入力した各パラメータ値

カットバック機能のカスタマイズ （2）

②接触反復回数によるカットバック（接触ループの内側・ループの最後 ）

カットバック発動の判断if文

```
674       ! ----- check divergence
675       if( count_step >= fstrSOLID%step_ctrl(cstep)%max_contiter ) then
676         if( hecMESH%my_rank == 0) then
677           write(   *,'(a,i5,a,i5)') '     ### Contact failed to Converge  : at total_step=', cstep, '  sub_step=', sub_step
678         end if
679         fstrSOLID%NRstat_i(knstCITER) = count_step                    ! logging contact iteration
680         fstrSOLID%CutBack_stat = fstrSOLID%CutBack_stat + 1   連続発動回数の加算
681         fstrSOLID%NRstat_i(knstDRESN) = 3
682         return
683       end if
684
685     enddo loopFORcontactAnalysis
```

カットバック理由コードの設定

count_step: 接触ループ反復回数

fstrSOLID%step_ctrl(cstep)%xxxxxxxx :

!STEPから入力した各パラメータ値

③ソルバーのエラーによるカットバック（NRループの内側・ソルバーの実行直後 ）

```
542         call solve_LINEQ_contact(hecMESH, hecMAT, fstrMAT, istat)  ソルバー （ステータス: istat）
543       endif
544       call fstr_recover_initial_config_to_mesh(hecMESH,fstrSOLID,coord)
545       ! ----- check matrix solver error
546       if( istat /= 0 ) then   カットバック発動の判断if文
547         if( hecMESH%my_rank == 0) then
548           write(   *,'(a,i5,a,i5)') '     ### Fail to Converge  : at total_step=', cstep, '  sub_step=', sub_step
549         end if
550         fstrSOLID%NRstat_i(knstDRESN) = 4
551         fstrSOLID%CutBack_stat = fstrSOLID%CutBack_stat + 1   連続発動回数の加算
552         return
553       end if
```

カットバック理由コードの設定

カットバック機能のカスタマイズ （3）

- 変更個所（2）：カットバック理由コードにて、FSTR.staにメッセージ出力するところ
  - **fstr_Ctrl_TimeInc.f90** （subroutine fstr_TimeInc_PrintSTATUS）

```
90    if( Cutback_stat > 0 ) then
91      write(etime,'(1pE12.4)') current_time
92      write(cstate,'(I4,A)') Cutback_stat,'F'
93      if( NRstatI(knstDRESN) == 1 ) write(message,'(A)') 'Failed to converge due to MAXITER.'
94      if( NRstatI(knstDRESN) == 2 ) write(message,'(A)') 'Failed to converge due to MAXRES.'
95      if( NRstatI(knstDRESN) == 3 ) write(message,'(A)') 'Failed to converge due to MAXCONTITER.'
96      if( NRstatI(knstDRESN) == 4 ) write(message,'(A)') 'Failed to converge due to MatSolveError.'
97      if( Cutback_stat == pAinc%CBbound ) write(message,'(A)') '# of successive cutback reached max.'
98    else
99      write(etime,'(1pE12.4)') current_time+time_inc
100     write(cstate,'(A5)') 'S'
101     write(message,'(A)') ''
102   endif
```

追加したカットバック理由コードに対応するメッセージのWRITE文を追加する。

まとめ

- 自動時間増分・カットバック機能の活用により、伝動ベルトの引っ張り解析の計算時間が従来（固定増分）の約30%に短縮された。

- 固定増分では途中で収束せずJOB停止するケースが多発し、何度もやり直す必要が生じるため、実際の作業時間としては1/10以下になった。

- ソースコードのカスタマイズにより、必要に応じて増分・カットバックの条件を追加設定する事が出来るようになった。