



**Biomechanics  
Laboratory**  
School of Human Kinetics  
University of Ottawa

**Graph User's Manual**

Walter O. Martindale, M.P.E.  
School of Physical Education and Recreation  
University of British Columbia

and

D. Gordon E. Robertson, Ph.D.  
School of Human Kinetics  
University of Ottawa

**Last revision: 29 June 2003**  
© Copyright, D.G.E.Robertson, 1994-2003

# How to Prepare a Graphics Control File (.GRA) for the Biomech Program, GRAPH.

This outline describes how to prepare a graphics control (.GRA) file for the program, GRAPH, a plotting program used with the Biomech software package. GRAPH was programmed by D.G.E. Robertson, Ph.D., for plotting data generated by programs from the Biomech package.

The Biomech programs are a powerful set of software designed for computing the kinematics and kinetics of human motion from input data derived from digitized film and digitally recorded force data. This software is described in another document entitled, Biomech User's Manual.

The format of this manual is similar to that of the Biomech User's Manual in that the graphing control file will be described line-by-line, with examples, and the FORTRAN format statement describing the necessary input format of the data. Each line in the .GRA file will have an associated number to indicate its relative position in the file and each line will be referred to as a LINE. **Note, not all LINES need to be included in the .GRA file.**

---

**LINE 1** Graph title, GTITLE, a descriptive title for the plot control file. Appears only in the graphic "scratch" output file. (A80)

Example:

---

```
Joint angular velocity, moment of force and power during running.
```

---

GTITLE should contain information describing what the graphics control file will plot. The information contained in the .GRA control file is often difficult to interpret or to edit if the user cannot recall exactly what the file was designed to plot.

**LINE 2**, a description of the graph borders (sizes), number of plots to produce, and other information. (16G5) Variable names are positioned below so that the last letter in the variable name aligns vertically with the end of the 15 field of the variable it describes.

Example:

```

NRECS      $NEWX      $EFILE      $RESC      YSIZE      LCODE      IXDEC      IVER
  NGRAPH    $NEWY      NYTICS      XSIZE      ICODE      $CFILE     IYDEC      FT
0    3    F    T    T    10  T T  5.5  9.0    9  10    F    1   -1  0  pj
                X )- .) Y

```

NRECS is the number of unformatted machine language records in the input data file (maximum is 200). Each record corresponds to a single biomechanical quantity, such as, the angular velocity of a marker, the potential energy of a segment or the net moment of force at a joint. A record includes all the data from frames 1 to NPOS. A zero, 0, tells GRAPH to read all records until an end-of-file mark. If NRECS is an integer greater than zero, GRAPH only reads that many records or to an end-of-file whichever comes first. The ordering of the records in each type of file (.KS, .RA, .FO, etc.) varies with the number of segments, markers and joints for each trial. Each program arranges these records in a specific manner. The appendix describes the order by which each program writes records into the data files.

The output record order from the various programs in the Biomech package are in the appendices of this manual. Users of this package may wish to create their own plotting data files containing output records from several different files.

NGRAPH states how many separate plots are to be produced by the GRAPH program, and how many repetitions of LINES 11 to 18 are to be included in your plotting control file.

NOTE: When plotting, if your ordinal axes on different plots are to be different, you must create separate control files for each of the different axes, as GRAPH creates only one set of ordinate axis specifications each time the program is run.

\$NEWX is a logical (T or F, for True or False) variable which lets one supply a new X-axis label.

T lets you include a new label in the graphic control file, F is the default, which labels the abscissa as "TIME".

\$NEWY is a logical variable which affects the presence or absence of a label on the ordinate axis.

T indicates that you will include the axis label in the file, F indicates that the ordinate will not be labelled.

\$EFILE is a logical variable indicating the location of event codes in the plotting run. If \$EFILE is T the program reads event codes from your Biomech control file. If \$EFILE is F, GRAPH reads event codes (frame number and code name) from the terminal.

NYTICS is the number of tics that will be drawn in the ordinate axis of your plot. With NYTICS set, you must include a line containing the NYTICS labels.

\$RESCX is a logical variable enabling or disabling rescaling of your X-axis dimension. If blank or F, the program will plot with automatic scaling in seconds. If \$RESCX is T, GRAPH will you must include on LINE #6 the minimum, maximum and interval for the X- axis tic marks.

\$RESCY is a logical variable enabling or disabling rescaling of your data. If \$RESCY is F, the program will automatically plot all of your data on the same axis. If \$RESCY is T, GRAPH will read scaling information later in the graphic control file, enabling different sets of data to be plotted over different regions of the ordinate axis.

XSIZE is a variable sets (in inches) the length of the X-axis of each of the NGRAPH plots described by the graphics control file.

YSIZE sets (in inches) the length of the Y-axis of your plots.

ICODE is the event code number of the start of your data for plotting. Valid codes, currently in the U.Ottawa version of GRAPH are:

- 1: HSR
- 2: TOR
- 3: HSL
- 4: TOL
- 5: CAT
- 6: FIN
- 7: HIT
- 8: OFF
- 9: BEG
- 10: END
- 11: IFS
- 12: ITO
- 13: CFS
- 14: CTO

Other codes for this may be programmed a version where the FORTRAN source is available (CODES is a 14 element declared INTEGER array). If ICODE is a positive integer, usually 9, indicating array element number 9, "BEG" for the beginning of a trial, the program looks in your control file (LINE 18, Biomech User's Manual) for a line containing "BEG", with an associated frame number. GRAPH starts plotting from the frame number specified as "BEG". If ICODE is specified as a negative integer, GRAPH starts the plot from the frame before that specified with "BEG". If ICODE is left blank or is specified as "0", GRAPH plots from the first frame of data. Another example of the use

of ICODE could be, if one specified ICODE=-1, GRAPH would scan the Biomech control file for the word "HSR" (see the list above) and would start plotting data from one frame before that specified for the frame identified as HSR (Heel-Strike Right).

LCODE is the event code number indicating to GRAPH what to search for in the Biomech control file as a finishing code. In this sample LCODE is specified as 10, or "END". As with ICODE, a negative integer causes GRAPH to stop plotting one frame before the frame specified with LCODE, and a "0" indicates that GRAPH will plot to the last frame of data.

\$CFILE left blank or declared F (False) tells that there is NO control file and that your graphics control file must contain the parameters normally found in your Biomech control file. If \$CFILE is declared T (True) a control file will be read and must contain various data. If you use a control file the following information will be collected from it:

Trial code  
NPOS, INCR  
DELTAT, TIME0, TBMASS  
The various event codes

Note: If you are plotting data from a Biomech package program and \$CFILE is T you will NOT need to include the information in LINES 3, 4, 5 or 6 listed below.

When you have a control file GRAPH reads your file from FORTRAN unit 8. Your next line in the graphics file should be LINE 7. If there is no control file to read, GRAPH reads starting codes and associated frame numbers from the terminal. You are prompted by the computer for codes and frame numbers. Here you respond with film frame number and associate code number. You type data as two integers separated by a comma. The program continues to read pairs until you enter the numbers: 0,0.

IXDEC, IYDEC are the number of decimal digits to be used in the printing of X and Y numbers, respectively. Maximum is 3, 0 means print only a decimal, and -1 means no decimal part. (2G5)

IVER identifies the version number of the .GRA file. A blank area or the numbers 0 or 1 indicates version 1 format. A 2 indicates version 2 format. Version 2 format combines LINES 14 through 17 into one line. See LINE 14 below.

FT identifies which Biomech filetype is to be used for input. Suitable codes include: AV, EN, FR, FO, JP, KM, KS, MM, MU, NS, PJ, PS and RA (see appendices).

**LINE 3**, assuming no control file, (\$CFILE=F) is the title . CTITLE as in the control file for Biomech (80A1). This title appears in the printed output for your plot, and does not plot. CTITLE is usually in the form: Trial code, description, researcher's initials or name.

Example:

```
Running study, trial 5, Jane Doe, 3/25/86, your initials.
```

This level of explanatory detail may seem unnecessary when preparing your 20th data file, but 6 months later, when you or someone else has to look at your data to reassess conclusions this detail will be valuable.

**LINE 4**, assuming no control file (\$CFILE=F), is the TRIAL CODE, which will appear in the legend. The program reads 8 characters for your trial code (variable name ALEG(J), J=8,15), and plots this with the graph legend, after all of your curves have plotted.

**LINE 5**, assuming no control file (\$CFILE=F), contains the variables NPOS and INCR, the number of elements in each input array, and the frame increment, respectively. (Format 215).

Example:

```
44      1
```

meaning 44 frames of input data to plot, with the data plotted for each frame. These variables are described in Biomech User's Manual.

**LINE 6**, assuming no control file (\$CFILE=F), contains the variables DELTAT, TIMEO and IXDEC, the time interval between data elements, the starting time for your trial (usually 0.0), and the number of digits to be drawn following the decimal point in the abscissa's time labels. (The format is 3G5, elements should be as F5.2, F5.2, and I5, respectively).

Example:

```
0.02  0.0  1
```

indicating a time interval between data elements of .02 sec, a starting time of 0.0 sec, and that the label should be plotted with 1 decimal digit following the decimal point on the abscissa.

**LINE 7** is a list of variables describing the marking of the axes of your plot(s) (16G5). The listing is displayed as is LINE 2:

Example:

```
AYMIN    AYGRID    AXMAX    GRHT      Colours
  AYMAX    AXMIN    AXGRID
  0.    9.    1.    0.    0. 0.10 0.20      e777f623
```

AYMIN is the position of the bottom of the graph, in inches, based upon the system plot routine's position for the bottom of your plot. With AYMIN=0, GRAPH positions the bottom border of the plot about 1.5 inches above the bottom of the paper, to allow space for labels, an abscissa title, and a border. AYMAX is the highest position of the ordinate axis. In the example AYMAX=9 indicates that the graph will be AYMAX-AYMIN=9-0=9 inches in height.

AYGRID is the spacing for the ordinate axis. AYGRID=1 indicates that the ordinate will be divided every inch. AYGRID=.5 would indicate that the ordinate would be divided every .5 inch, and GRAPH would require that NYTICS be 19, to accommodate the 10 marks for 0 to 9 and the 9 marks for the .5 intervals at 0.5, 1.5, 2.5, etc.

AXMIN is the value of the abscissa array used to set the smallest value plotted on the abscissa. If AXMIN is specified as zero, GRAPH uses the TIME array's first element as AXMIN is scaling the axis.

AXMAX is the value of the abscissa array used to set the largest value plotted on the abscissa. If AXMAX is specified as zero, GRAPH uses the TIME array's last element as AXMAX, and uses this in scaling the axis.

If AXMIN and AXMAX are to be sections of a trial, you may specify these values based upon IPOS and LPOS, starting and ending times or film frames (LINE 12). AXGRID is

the value used to specify the marking of the time graduations on the abscissa. In the example AXGRID is .10, telling GRAPH to put a tic and a label every .10 second on the axis.

GRHT is the height of the labelling. If left blank the default is 0.13 inches.

Colours are hexadecimal codes for setting the colours of various graphics objects. The eight codes are ordered: Title (CLRTTL), Time axis (CLRTIM), Border (CLRBOR), Axes (CLRAXS), Legend (CLRLEG), Captions (CLRCAP), Events (CLREVT), and Default for curves (CLRCRV). Codes range from 0 to 9, and a to f, where a=10, ..., f=15).

**LINE 8** contains YNUMS, the numbers for labelling the marks on the ordinate axis. LINE 8 is used if NYTICS is not equal to zero. In the example (LINE 2) above, NYTICS=10. GRAPH looks for NYTICS values in LINE 8 (20A4), and plots them in the order listed, from the bottom of the ordinate to the top.

Example:

```
-50  -25   0   25  -5   0   5  -10   0   10
```

This example is used in an instance when \$RESCY is TRUE, meaning that the ordinate is divided into segments with different scaling. In this sample, the bottom four sections of the ordinate are labelled for data scaled between -50 and +25. LINE 16 explains how to cause GRAPH to scale an array to fit such a label.

**LINE 9** contains XLABEL (20A4), and is used only if \$NEWX is True (LINE 2). XLABEL is used to include your own label for the abscissa. If \$NEWX = F, the default label TIME (SECONDS) is printed, centred below the abscissa. If \$NEWX = T, your own XLABEL appears, centred below the abscissa.

Example:

```
Time (milliseconds)
```

If you are NOT changing the abscissa label, and \$NEWX=F, do NOT include LINE 9.



**LINE 10** contains YLABEL (20A4), and is used only if \$NEWY is True (LINE 2). YLABEL is normally used to label the ordinate axis with appropriate data-set names and units. YLABEL is normally TRUE, since most applications of GRAPH require different labelling of the ordinate.

Example (based upon LINE 8, above):

```
Ang. velocity (rad/s)           Moment (N.m)           Power (W)
```

If \$RESCY=F (LINE 2), the ordinate is scaled from top to bottom with one scale factor, and thus has only one set of units and one label.

Example (if \$RESCY=F):

```
Angular acceleration of the hip (m/s2)
```

YLABEL is printed centred on the length of the ordinate axis. If you are not putting a label on the ordinate axis, do NOT include LINE 10. Note, though, that this is rare.

**LINE 11** contains GTITLE (20A4), which plots across the top of the graph. This is the first LINE of the graphic control file which is specific to the current plot. In LINE 2, if NGRAPH is greater than 1, all of the previous lines of information and all of the LINES in your graphic control file are general to all plots appearing when you run the GRAPH program. GTITLE is the first LINE of your control file within a loop of graph specific data. Thus, if your ordinate axis for each plot is to be the same, NGRAPH can be > 1, and your file can loop starting with LINE 11. If your graphs are to have different ordinate labelling and scaling, you must prepare a separate control file with the preceding LINES specific to the different data sets.

Example:

```
Hip angular velocity, moment and power.
```

You may, however, use LINE 13 to alter the height of your axes in each plot, but the variables NYTICS and YNUMS are held throughout the full running of GRAPH.

**LINE 12** contains information specific to the current graph, including what data of the file to plot, how many curves to plot, etc:

Example:

```

IPOS      NCURVE   $LEG      $LONEV   NCAPNS
  LPOS      $TTL    $EVT      $TIM
  0      0      5      T F T      T      F      T      2
                .) $LEGBORD

```

IPOS is the data element (usually film frame number) to use as the starting point for the plot. IPOS=0 causes GRAPH to read from the first data element of the array being plotted. IPOS...0 causes the program to start plotting from frame number IPOS.

LPOS is the data element (usually film frame number) to use as the finishing point for the plot. LPOS=0 causes GRAPH to plot to the last data element in the data array. LPOS...0 causes the program to finish plotting at frame number LPOS.

NCURVE is the number of curves to plot on this particular graph. GRAPH reads NCURVE arrays from your data file, (input logical unit #1) as specified in LINE 14. (Maximum is 16.)

\$TTL is a logical variable stating whether or not the plot is to have a title plotted. Here you can specify T (true) to plot GTITLE or F (false) to skip plotting the GTITLE (shown in LINE 11). LINE 11 may have a title even if \$TTL=F. This will help you keep better record of what your plot control file contains.

\$LEGBORD if T (true) includes a border around the legend. If F (false) or blank no border will surround the legend.

\$LEG is a logical variable stating whether or not GRAPH will plot a legend (as specified in LINE 17) describing each of the NCURVE curves in the plot.

\$EVT is a logical variable for plotting the event marks on the graph. If \$EVT=T, event vertical lines will be drawn on the plot at the times (frame numbers) associated with the event codes specified in your Biomech control file or earlier in this outline (LINE 4). If \$EVT=F, the lines will not be drawn.

\$LONEV is a logical variable for changing the level being plotted in conjunction with the event code and line. If \$LONEV=T the event codes will be full word descriptions, rotated 90 degrees. If \$LONEV=F, event codes will be as listed above (LINE 2: ICODE). The long event labels ALLOW only of eight elements, in order:

- 1: HEEL-STRIKE RIGHT
- 2: TOE-OFF RIGHT
- 3: HEEL-STRIKE LEFT
- 4: TOE-OFF LEFT
- 5: HEEL-OFF RIGHT
- 6: HEEL-OFF LEFT
- 7: FOOT-FLAT RIGHT
- 8: FOOT-FLAT LEFT.

\$TIM is a logical variable used to include or exclude a plot of the time axis (line with tics, but no labels) wherever a "0" appears on the ordinate axis. \$TIM is used in plotting several curves on the same graph, scaled differently on different regions of the ordinate axis.

NCAPNS is an integer describing how many lines of captions will be plotted on the graph. In this example, 2 captions will be plotted. These captions are read as LINE 18, near the end of the program.

---

**LINE 13** lets you alter the size of the plot ordinate, the position of the legend, and the length of the line drawn when the event marks are plotted. LINE 13 contains (16G5):

Example:

---

```
YMIN YMAX YGRID XLEG YLEG EVLEN
0. 0. 0. 2.6 2.0 0.
```

---

YMIN lets you respecify the position of the bottom of the ordinate for the specific graph that you are plotting. If YMIN=0. the graph will plot using AYMIN (LINE 7).

YMAX lets you respecify the top border of the plot to YMAX inches. If YMAX=0. the plot uses AYMAX (LINE 7).

YGRID lets you respecify the spacing of the NYTICS marks on the ordinate. YGRID=0. allows the AYGRID to plot as in LINE 7. It is possible to alter the dimensions of each graph using these three parameters. This is usually not done.

XLEG and

YLEG are used to reposition the legend. The default position of the legend is the top left corner of the graph. XLEG moves the legend XLEG inches to the right of this position. YLEG moves the legend YLEG inches down from the top of the graph. These are useful parameters for moving the legend to a region of the graph where it will not plot over any of the data curves. In the sample above, the legend starts 2.6 inches to the right and 2.0 inches down from the top left corner.

EVLEN lets you specify the length of the vertical lines drawn with your event codes. If EVLEN=0, or if EVLEN is greater than the height of the axis, these lines are drawn the full length of the ordinate.

---

**LINE 14** is the list of arrays that you want to plot (1615). GRAPH reads all of the arrays in your input data file and stores them in memory. You select which of the arrays you wish to plot based upon an understanding of the output order of the program which generated your data. For instance, the segment kinematics output file contains (in two's compliment binary) arrays as briefly outlined in the NRECS section of LINE 2. A summary of the output ordering from each of the programs in Biomech is appended.

If you selected version 2 format on LINE 2, the format of this card and the next 3 will be combined. For this new format see the section, LINE 14 Version 2, below.

Example:

---

1	3	6	12	28
---	---	---	----	----

---

instructs GRAPH to plot arrays from the input file (attached to unit 1) as listed. If you wish to plot the same array on two pictures in the same run of GRAPH you must be careful to NOT rescale the array the second time it plots, as the array is written over in memory by the scaled data. Thus, if you are to plot the same data on two plots in the same run of GRAPH, you must scale the data on the first plot, then specify scaling information carefully to plot the array as rescaled the second time it is plotted.

If you make the array number negative the curve will be plotted with symbols only, otherwise a line and symbols will be plotted. \*\*\* **New** \*\*\* Adding 1000 will normalize the data to body mass if a control file is present.

**LINE 14 Version 2** tells which curve to plot and how each curve should appear in the figure. It combines the information from LINES 14 through 17 in a single line for each curve.

Example:

CURVE	COLOUR		YLOW	YTMIN		LABEL	
	ISYMBL	LINETYPE	YHIGH	YTMAX			
10	3	5	4	0	5	-100 400	Moment of force

CURVE is the record number of the curve to be plotted. See LINE 14 above for details.

ISYMBL is the plotting symbol to be used if symbols are to be drawn for this curve. Use numbers 0 to 16 only. Use -1 through -5 to produce dashed lines of increasing length.

COLOUR is the pen or screen colour for this curve. The number used depends on the plotting device chosen. Use 1 to 6 for HP 7475 type plotters or 0 to 15 for EGA/VGA screens.

LINETYPE selects the type of line. Use 0 to plot a line only. Use positive numbers to plot lines with symbols. E.g., 5 means plot a line with a symbol every 5th point. Negative numbers mean plot symbols only. E.g., -2 means plot a symbol for every second datum.

YLOW, YHIGH, YTMIN and YTMAX are described below under LINE 16. They are only necessary if you selected rescaling of the data on LINE 2. These numbers can be floating point values if a decimal point is included.

LABEL is the text label of this data for the legend. Leave blank if you don't want it included in the legend. You must select the legend option on LINE 12 to have a legend included on your figure.

**LINE 15** is a list of the ISYMBL numbers for the plotting subroutine to use for identifying the different curves. The program plots a symbol with every 4th data element, and matches the symbols in this LINE with the arrays in LINE 14 in order. The Calcomp plotter manual contains a list of available symbols and their codes for use in this LINE.

Example:

```
2   4   5   6   9
```

when matched with the information in LINE 14 permits you to see which line is which, as the legend contains a line segment with the appropriate symbol at each end. Combining LINES 14 and 15 gives data record 1 plotting with symbol number 2 appearing with every fourth data element. Data record 28 plots with symbol 9.

**LINE 16** is used only if \$RESCY = T (LINE 2). LINE 16 is repeated once for each curve specified by NCURVE (LINE 12) (G5 format, 4 fields).

Example:

YLOW	YHIGH	YTMIN	YTMAX
7	9	-10	10
4	6	-5	5
4	6	-5	5
0	3	-50	25
0	3	-50	25

The above example agrees with NYTICS, in that there are 10 elements between the lowest YLOW and the highest YHIGH (0 to 9, inclusive). LINE 8, YNUMS shows the 10 labels going from -50 to 25 by 25s, -5 to 5 by 5s, and -10 to 10 by 10s. Combining the information in LINES 14, 15, and 16 renders the following plot (when \$RESCY=T):

Array 1 will plot with symbol 2. The array will plot between ordinate points 7 and 9, with a "0" time axis drawn at point 8; scaling will be between -10 and 10. Array 3 and array 6 will plot with symbols 4 and 5, respectively. Both arrays will plot between ordinate points 4 and 6, with a "0" time axis drawn at point 5, scaling will be between -5 and +5. Array 12 and array 28 will plot with symbols 6 and 9, respectively. Both arrays will plot between ordinate points 0 and 3, with a "0" time axis drawn at point 2, scaling will be between -50 and +25.

The ordering of the array scaling information is not that important as long as the order of the array list agrees with the ordering of the information in the lines in LINE 16. It is a good idea to start these listings at the top and work to the bottom, or vice-versa, to avoid confusion for yourself later or for others attempting to decipher your listings.

**LINE 17** contains the listing of LABELS for each of the curves. The labels are plotted in the legend with a line segment and appropriate symbol beside the label. These labels must be in the same order as the array listing in LINE 14.

Example:

```
Angular velocity
Moment of force
Moment power
```

This will plot with the upper left corner of the legend's area positioned at (XLEG, YLEG) inches to the right of and down from the top left corner of the graph.

**LINE 18** is a listing of the captions for the plot (2F5.2,10A4). GRAPH reads NCAPNS (LINE 12) lines of data here, and plots these captions where specified. In the example of LINE 12, above, NCAPNS is "2".

Example:

```
0.20 1.00Extending
0.20 2.80Flexing
```

which would plot the word EXTENDING with the bottom corner of the first "E" starting at the (x,y) coordinate (0.2,1.0) with (0,0) considered to be the bottom left corner of the figure. GRAPH reads as many lines of captions here as specified by NCAPNS. Based on the reading format, the captions may contain up to 40 characters.

## APPENDICES

### Order of Records from the Biomech Programs.

All of the programs in the Biomech package of software have optional disk file output for the results of their computations. These output options are selected when preparing the Biomech control file as outlined in Biomech User's Manual. In that manual, output and program option selection is done using CARD# 5, which contains "switches" for telling the various programs to compute and output results according to your needs. These "switches" are called NOPTNS(I), where the "I" refers to the column number in CARD# 5. The following outlines the optional output NOPTNS specific to each of the programs, and lists the order of the output records from each file. The programs write the data in unformatted binary records, with all of the data from time=0 to time=end contained on each record. GRAPH reads all of the data in the selected file and plots the record that you select in CARD# 14 of the GRAPH control file. These records may be copied from a variety of output files from the same original input data, and should all be of the same length. This is simple when plotting data from the same trial. If plotting data from different trials, the records will be of varying lengths, and the program AVERAGE must be used to fill or reduce the records to the same length.

The following is a list of the files which GRAPH can read and are produced by the Biomech programs. Note, all these files are in binary (machine language) format and are unreadable by most conventional software.

File Extension	Program	Description
.AV	AVERAGE	ensemble averages and confidence intervals
.EN	ENERGY	total body & segmental energies
.FO	FORCE	joint forces and moments of force
.FR	FOURIER	Fourier reconstructed coordinates
.JP	FORCE	joint moments and powers
.KM	KINEMATIC	marker kinematics
.KS	KINEMATIC	segmental & total body kinematics
.MM	MOMENTUM	segmental & total body momenta
.MU	MUSCLE	muscle length kinematics
.NS	NOISE	noise analysis (same format as .FR above)
.PJ	POWERS	joint powers etc.
.PS	POWERS	segmental powers etc.
.RA	KINEMATIC	relative angular kinematics

The following output files are NOT suitable as input to the GRAPH program:

.DG	Imager	unscaled digitizer coordinates
.CN	CINADATA	unfiltered marker coordinates
.DR	KINEMATIC	changes in relative angles
.RF	BioProc2	reaction force data from force plate
.SN	SIMULATE	simulation marker coordinates



## APPENDIX A

### AVERAGE PROGRAM

#### **.AV file**

The ordering of records created by the Average program depend upon which files and which records were selected for averaging. They are not predefined. For each record that was averaged four new records are created: one is the mean, the next is the standard deviation, and the last two are the mean minus and plus the confidence interval selected during the averaging process (95<sup>th</sup>, 99<sup>th</sup> or 1 or 2 standard deviations, etc.) Usually there are 101 elements in each curve (i.e., 0 to 100%). The file begins with two records that may be ignored. The second can be used as the data for the abscissa.

RECORD	DATA
1	Information about number of records.
2	Abscissa in percentages (%)
3	Curve 1, mean
4	Curve 1, standard deviation
5	Curve 1, mean minus confidence interval
6	Curve 1, mean plus confidence interval
7-10	Curve 2, repeated as above
11-14	Curve 3, repeated as above
etc.	

## APPENDIX B

### ENERGY PROGRAM

#### **.EN file**

NOPTNS(31)=1 means ENERGY will write segment mechanical energies to a disk file with extension, .EN, in the following order:

RECORD	DATA
1	Total body mechanical energy (J)
2	Segment 1, translational kinetic energy (J)
3	Segment 1, rotational kinetic energy (J)
4	Segment 1, potential energy (J)
5	Segment 1, total segment energy (J)
6-9	Segment 2, repeated as above
10-13	Segment 3, repeated as above
etc.	

## APPENDIX C

### FORCE PROGRAM

#### **.FO file**

NOPTNS(53)=1 means FORCE will write net forces and moments of force to a disk file with extension, .FO, in the following order. FORCE computes joint forces in the order specified in your Biomech control file. Note, if a support moment is computed it is the tenth record after the three joints of the lower extremity

RECORD	DATA
1	Segment 1, proximal joint, X force (N)
2	Segment 1, proximal joint, Y force (N)
3	Segment 1, proximal joint moment of force (N.m)
4-6	Segment 2, repeated as above
7-9	Segment 3, repeated as above
10	Possibly the support moment
10-12	Segment 4, repeated as for segment 3
etc.	

#### **.JP file**

NOPTNS(71)=1 means FORCE will write the joint angular velocities, net moments of force and the power produced by the moments of force to a disk file with extension, .JP, in the following order:

RECORD	DATA
1	Joint 1, angular velocity (rad/s)
2	Joint 1, net moment of force (N.m)
3	Joint 1, power of net moment of force (W)
4-6	Joint 2, repeated as above
7-9	Joint 3, repeated as above
etc.	

## APPENDIX D

### FOURIER and NOISE PROGRAMS

#### **.FR or .NS file**

NOPTNS(2) (column # 2=0) means FOURIER or NOISE will write reconstructed data to a disk file with extension, .FR or .NS, for plotting purposes. FOURIER and NOISE essentially reconstruct the digitized data for the Imager or FIGURE programs, with the data smoothed by reconstruction with the harmonic frequencies defined by NOPTNS(1) in the Biomech control file. The number of body markers processed depends on the markers selected in the control file. See Biomech User's Manual for details.

NOPTNS column number and selected output order for FOURIER or NOISE follow:

RECORD	DATA
1	Fourier marker 1, X data adjusted (slope removed, cm)
2	" " 1, X data reconstructed from Fourier series
3	" " 1, Y data adjusted (slope removed, cm)
4	" " 1, Y data reconstructed from Fourier series
5-8	Fourier marker 2, repeated as above
etc.	

## APPENDIX E

### KINEMATIC PROGRAM

#### **.KM file**

NOPTNS(12)=1 means KINEMATIC will write marker kinematics to a disk file with extension, .KM, in the following order:

RECORD	DATA
1	Marker 1, position, X-coordinate
2	Marker 1, position, Y-coordinate
3	Marker 1, velocity, X
4	Marker 1, velocity, Y
5	Marker 1, acceleration X
6	Marker 1, acceleration Y
7-12	Marker 2, repeated as above
13-18	Marker 3, repeated as above
etc.	

Marker 2 data begins at record 7, Marker 3 at record 13, etc.

#### **.KS file**

NOPTNS(11)=1 means KINEMATIC will write records of segment kinematics for those specified in your control file to a disk file with extension, .KS, in the following order:

RECORD	DATA
1	Time array, based upon sampling rate and initial time
2	Total body centre of gravity, X-coordinate
3	Total body centre of gravity, Y-coordinate
4	Total body c. of g. velocity, X direction
5	Total body c. of g. velocity, Y direction
6	Total body c. of g. acceleration, X direction
7	Total body c. of g. acceleration, Y direction
8	Total body c. of g. mech. energy
9	Percent of time elapsed
10	Repeat of (9)

The order of the following records repeats for each defined segment.

11	Segment 1, position of C of G, X (m)
12	Segment 1, position of C of G, Y (m)
13	Segment 1, absolute angle (rad)
14	Segment 1, velocity of C of G, X (m/s)
15	Segment 1, velocity of C of G, Y (m/s)
16	Segment 1, angular velocity (rad/s)
17	Segment 1, acceleration of C of G, X (m/s <sup>2</sup> )

18	Segment 1, acceleration of C of G, Y (m/s <sup>2</sup> )
19	Segment 1, angular acceleration (rad/s <sup>2</sup> )
20-28	Segment 2, repeated as above
29-37	Segment 3, repeated as above
etc.	

Segment 2 begins at record 20. A good way to track down which record you wish to plot out of this or any file, is to create a table, i.e.,

Segment	Foot	Shank	Thigh
CGX	11	20	29
CGY	12	21	30
ANGLE	13	22	31
VX	14	23	32
VY	15	24	33
OMEGA	16	25	34
AX	17	26	35
AY	18	27	36
ALPHA	19	28	37

The length of any of these files depends upon the number of segments, as the Biomech package programs with this type of output loop through the “unformatted writes” once for each segment under scrutiny.

### **.RA file**

NOPTNS(10) (column 10=0) means KINEMATIC will write the angular kinematics for the joints specified in your control file to a disk file with extension, .RA, in the following order:

RECORD	DATA
1	Joint 1, relative angle (rad)
2	Joint 1, relative angular velocity (rad/s)
3	Joint 1, relative angular acceleration (rad/s <sup>2</sup> )
4-6	Joint 2, repeated as above
7-9	Joint 3, repeated as above
etc.	

## APPENDIX F

### MOMENTUM PROGRAM

#### **.MM file**

NOPTNS(41)=1 means MOMENTUM will write segmental and total body momenta to a disk file with extension, .MM, in the following order:

RECORD	DATA
1	Segment 1, momentum, X (kg.m/s)
2	Segment 1, momentum, Y (kg.m/s)
3	Segment 1, total angular momentum (kg.m <sup>2</sup> /s)
4-6	Segment 2, repeated as above
7-9	Segment 3, repeated as above
etc.	

Data records for Segment 2 begin at Record 4. The last four records in the MOMENTUM output disk file are for total body computations:

RECORD	DATA
n+1	Total body momentum, X (kg.m/s)
n+2	Total body momentum, Y (kg.m/s)
n+3	Total body angular momentum (kg.m <sup>2</sup> /s)
n+4	Total body moment of inertia (kg.m <sup>2</sup> )

## APPENDIX G

### MUSCLE PROGRAM

#### **.MU file**

MUSCLE will write muscle length kinematics to a disk file with extension, .MU, in the following order:

RECORD	DATA
1	Gastrocnemius, length (cm)
2	Gastrocnemius, velocity of shortening (cm/s)
3	Gastrocnemius, acceleration (cm/s <sup>2</sup> )
4-6	Soleus, repeated as above
7-9	Semimembranosus, repeated as above
10-12	Vastus lateralis, repeated as above
13-15	Rectus femoris, repeated as above
16-18	Tibialis anterior, repeated as above
19-21	Gluteus maximus, repeated as above
21-24	Semitendinosus, repeated as above
25-27	Biceps femoris, repeated as above



## APPENDIX H

### POWERS PROGRAM

#### **.PJ file**

NOPTNS(65)=1 means POWERS will write joint power records to a disk file with extension, .PJ, in the following order:

RECORD	DATA
1	Proximal joint of seg. 1, joint angle (rad)
2	Proximal joint of seg. 1, joint angular vel. (rad/s)
3	Proximal joint of seg. 1, X linear vel. (m/s)
4	Proximal joint of seg. 1, Y linear vel. (m/s)
5	Proximal joint of seg. 1, X net force (N)
6	Proximal joint of seg. 1, Y net force (N)
7	Proximal joint of seg. 1, net moment (N.m)
8	Proximal joint of seg. 1, power transferred through joint (W)
9	Proximal joint of seg. 1, power transferred by moment(W)
10	Proximal joint of seg. 1, power of net moment (W)
11-20	Proximal joint of seg. 2, repeated as above
21-30	Proximal joint of seg. 3, repeated as above
etc.	

If the last segment is the trunk, there will be no associated records since it is connected to one or more of the other segments. For further clarification of this record listing, run POWERS with NOPTNS(63)=1, and examine the joint power listings.

#### **.PS file**

NOPTNS(64)=1 means POWERS will write segment powers to a disk file with extension, .PS, in the following order:

RECORD	DATA
1	Segment 1, distal joint power (W)
2	Segment 1, muscle power at distal joint (W)
3	Segment 1, joint power at proximal joint (W)
4	Segment 1, muscle power at proximal joint (W)
5	Segment 1, total power (W)
6	Segment 1, mech. energy, rate of change (W)
7	Segment 1, difference between (6) and (5) (W)
7-14	Segment 2, repeated as above
15-21	Segment 3, repeated as above
etc.	

NOPTNS(64)=1 and NOPTNS(66)=1 means POWERS will write trunk powers to the end of the segmental power records, in the same order. Thus, if you wish to plot data from the trunk

segment, you must count records through the segments to the trunk. Note, if NOPTNS(66)=0 POWERS will neither compute nor write the trunk powers. If NOPTNS(64) is not equal to 1, segmental powers will not be stored on a disk, POWERS only writes segmental powers to disk if you analyze every frame.

NOPTNS(62)=1 means POWERS will calculate and write to disk file total body power records. These will write on the end of the same disk file as the segmental and trunk power records. You must be aware of what output options you have specified to keep track of the record listing. The total body records will write at the end of the segmental listing if NOPTNS(66)=1, and at the end of the trunk records if NOPTNS(66)=1. Total body power records:

RECORD	DATA
n+1	Total body power (W)
n+2	Total body rate of change of energy (W)
n+3	Total body difference (W)
n+4	Total body "ideal difference" (W)