Bone Compression Project

Su Yang, Jing Liu, Jiayuan Shi, Qianyun Zou

2016/2/18

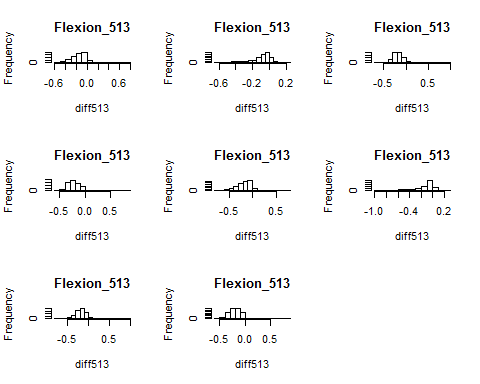
# Visualization

We were using specimen 513 from Flexion group and specimen 581 from Union group as examples to study on the distribution of differences between experimental results and simulation results.

Flexion\_513 <- read.csv("Flexion\_513.csv",header=T)  
Union\_581 <- read.csv("Union\_581.csv",header=T)

We created a matrix to record the differences between experimental results and eight simulation results, so it contained 8 columns and 2493 rows. We then used the data to plot histograms to see the distributions.

# Flexion\_513  
par(mfrow=c(3,3))  
matrix.diff513 <- matrix(rep(0,19944), nrow=2493)  
i <- 1  
k <- 0  
while(i < 24){  
 diff513 <- Flexion\_513[,i+1] - Flexion\_513[,i+2]  
 k <- k+1  
 matrix.diff513[,k] <- diff513  
 hist(diff513, main = "Flexion\_513")  
 i <- i+3  
}

 According to the histograms, the second and sixth simulations were highly skewed, thus we used the median difference for these two and mean difference for the rest.

df513 <- data.frame(matrix.diff513)  
mean(df513[,1])

## [1] -0.1465736

median(df513[,2]) # the 2nd is closest to 0 one

## [1] -0.0678

mean(df513[,3])

## [1] -0.1681694

mean(df513[,4])

## [1] -0.208455

mean(df513[,5])

## [1] -0.1771904

median(df513[,6])

## [1] -0.077

mean(df513[,7])

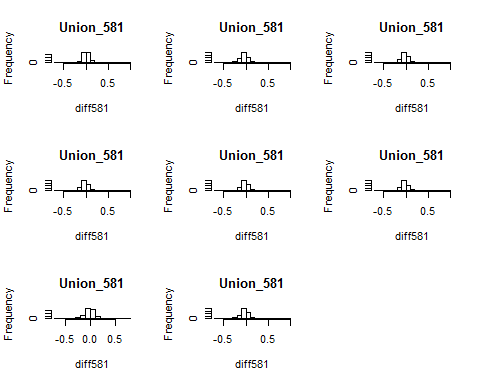
## [1] -0.168908

mean(df513[,8])

## [1] -0.186472

We repeated the same loop on the Union group and plotted histogram to see their distributions.

# Union\_581  
par(mfrow=c(3,3))  
matrix.diff581 <- matrix(rep(0,15512), nrow=1939)  
i <- 1  
k <- 0  
while(i < 24){  
 diff581 <- Union\_581[,i+1] - Union\_581[,i+2]  
 k <- k+1  
 matrix.diff581[,k] <- diff581  
 hist(diff581, main = "Union\_581")  
 i <- i+3  
}

 All the differences were normally distributed, thus we used mean difference for all of them.

df581 <- data.frame(matrix.diff581)  
mean(df581[,1])

## [1] 0.01820387

mean(df581[,2])

## [1] -0.02498484

mean(df581[,3])

## [1] -0.01184528

mean(df581[,4])

## [1] -0.01739428

mean(df581[,5])

## [1] -0.01694198

mean(df581[,6])

## [1] -0.01833476

mean(df581[,7])

## [1] -0.01128592

mean(df581[,8])

## [1] -0.01891991

# Two-Way ANOVA

We conducted two-way analysis of variance (ANOVA) for both Flexion group and Union group to examine whether the types of tissue and methods of simulation have influence on mean difference and squared error between the experimental and simulation results.

# read in the full data set that contains all one-dimension data points  
flexion\_data <- read.csv("Flexion.csv",header = T)  
union\_data <- read.csv("uni.csv",header = T)  
# put difference of simulation and experiment results into matrix for flexion group  
flexion.difference <- matrix(rep(0,279216),nrow=2493)  
i <- 1  
j <- 1  
# use while loop to calculate the mean difference and mean squared error for flexion group  
flexion.mean <- NULL  
flexion.se <- NULL  
while (i < 225) {  
 flexion.difference[,j] <- flexion\_data[,i]-flexion\_data[,i+1]  
 flexion.mean[j] <- mean(flexion.difference[,j],na.rm = T)  
 flexion.se[j] <- sum((flexion.difference[,j])^2,na.rm = T)  
 i <- i+2  
 j <- j+1  
}   
flexion.mean # mean difference for Flexion group

## [1] -0.146573566 -0.104138708 -0.168169354 -0.208455034 -0.177190413  
## [6] -0.147868512 -0.168907982 -0.186471961 -0.118523414 -0.016403971  
## [11] -0.170008974 -0.167416658 -0.179466478 -0.081040743 -0.181319495  
## [16] -0.180374420 -0.079854393 -0.033042991 -0.094269819 -0.107792555  
## [21] -0.118836284 -0.057400738 -0.124377599 -0.137192287 -0.044131127  
## [26] -0.016434256 -0.058038187 -0.056425431 -0.066182992 -0.031108624  
## [31] -0.071898195 -0.070503730 -0.055510710 -0.019734978 -0.097357481  
## [36] -0.093479543 -0.085546450 -0.056732050 -0.098493141 -0.095939791  
## [41] -0.095773486 0.009999037 -0.133301444 -0.132667188 -0.159029603  
## [46] -0.037956037 -0.182380465 -0.179794986 0.027498814 -0.107361527  
## [51] 0.020889273 0.019962352 -0.238825064 -0.202681949 0.013284012  
## [56] 0.013036256 -0.018422834 0.014805144 -0.018473060 -0.018478745  
## [61] -0.018458980 0.015600000 -0.018596255 -0.018600587 -0.050080786  
## [66] 0.005467028 -0.064332611 -0.079728039 -0.103735860 -0.038618331  
## [71] -0.096699599 -0.110820939 -0.026543260 -0.002174313 -0.035063649  
## [76] -0.034028638 -0.045325352 -0.021103957 -0.050269484 -0.049200671  
## [81] -0.117555282 -0.013443602 -0.089700361 -0.099985279 -0.145080064  
## [86] -0.060756077 -0.124647413 -0.133578981 -0.101431540 -0.019300261  
## [91] -0.152610287 -0.144140679 -0.278489295 -0.183200000 -0.286283342  
## [96] -0.277947624 -0.054856197 -0.026956839 -0.082321460 -0.079789049  
## [101] -0.085229563 -0.067926274 -0.106534978 -0.103451304 -0.188971600  
## [106] -0.136961974 -0.218324549 -0.215146651 -0.322614882 -0.298329122  
## [111] -0.338388528 -0.330938428

flexion.se # mean squared error for Flexion group

## [1] 108.43825 69.56601 125.58687 171.00974 153.90989 146.39255 131.61861  
## [8] 146.86428 95.78464 18.28714 132.90615 130.96279 160.36935 72.81694  
## [15] 146.15040 145.82172 38.07186 24.07120 47.64185 56.18610 54.73701  
## [22] 37.39852 58.58412 68.75955 13.21415 8.27395 21.43628 20.81231  
## [29] 19.76348 10.31805 24.94921 23.92563 41.61722 13.92168 65.23368  
## [36] 60.86214 62.63556 41.79841 65.62887 62.07031 113.66871 32.24005  
## [43] 135.21089 136.58673 151.02747 56.31942 170.41175 169.00875 259.16469  
## [50] 185.31080 274.68154 274.42013 359.65832 378.42585 277.56040 276.97924  
## [57] 16.01300 13.58482 16.17574 16.14137 16.00930 13.64339 16.18614  
## [64] 16.14946 41.10382 13.38657 49.19363 61.05656 75.72020 32.67772  
## [71] 62.70528 78.82927 15.84695 14.02875 17.92633 17.79986 18.50561  
## [78] 14.40881 19.94705 19.66132 68.39308 18.50396 60.91276 72.77235  
## [85] 107.24781 53.41309 83.09181 96.04908 68.54327 21.35631 120.73608  
## [92] 113.37108 282.25177 255.52637 269.68731 254.60528 53.19779 35.88088  
## [99] 71.93139 70.29777 73.04265 79.36414 87.63518 84.97729 288.88096  
## [106] 234.03464 300.04574 291.22752 542.19701 642.19232 512.97650 491.96052

# put difference of simulation and experiment results into matrix for union group  
union.difference <- matrix(rep(0,320096),nrow=2858)  
i <- 1  
j <- 1  
# use while loop to calculate the mean difference and mean squared error for union group  
union.mean <- NULL  
union.se <- NULL  
while (i < 225) {  
 union.difference[,j] <- union\_data[,i]-union\_data[,i+1]  
 union.mean[j] <- mean(union.difference[,j],na.rm = T)  
 union.se[j] <- sum((union.difference[,j])^2,na.rm = T)  
 i <- i+2  
 j <- j+1  
}   
union.mean # mean difference for Union group

## [1] 1.820387e-02 -2.498484e-02 -1.184528e-02 -1.739428e-02 -1.694198e-02  
## [6] -1.833476e-02 -1.128592e-02 -1.891991e-02 9.543081e-03 -9.365921e-02  
## [11] -7.674132e-02 -7.222651e-02 -1.686603e-01 -1.652553e-01 -7.785624e-02  
## [16] -1.844397e-01 -2.290037e-05 -9.523579e-02 -7.693949e-02 -8.044105e-02  
## [21] -9.168289e-02 -9.441408e-02 -2.537533e-02 -9.876275e-02 -1.050185e-02  
## [26] -1.335314e-01 -9.762740e-02 -1.043004e-01 -1.337649e-01 -1.398198e-01  
## [31] -5.279142e-02 -1.524444e-01 4.691471e-02 1.386600e-01 1.285016e-01  
## [36] 1.263566e-01 1.228149e-01 1.223633e-01 7.631800e-02 1.243254e-01  
## [41] 2.506402e-02 1.567684e-01 1.047324e-01 1.201016e-01 1.283957e-01  
## [46] 1.354144e-01 8.269771e-02 1.462168e-01 -1.270375e-02 -1.405289e-01  
## [51] -1.032625e-01 -1.108355e-01 -1.447512e-01 -1.502409e-01 -7.885810e-02  
## [56] -1.658458e-01 1.170256e-02 -2.284291e-01 -1.443037e-01 -1.396014e-01  
## [61] -1.757283e-01 -1.733341e-01 -6.012150e-02 -1.965193e-01 5.396759e-02  
## [66] 3.085589e-01 2.561301e-01 2.462396e-01 2.877649e-01 2.827113e-01  
## [71] 1.977569e-01 3.196125e-01 -1.136768e-02 -1.282465e-01 -1.141813e-01  
## [76] -1.138910e-01 -1.322437e-01 -1.313254e-01 -6.085300e-02 -1.358201e-01  
## [81] -5.468621e-02 -4.511957e-01 -3.936149e-01 -3.936460e-01 -3.577057e-01  
## [86] -3.544575e-01 -1.626323e-01 -3.871902e-01 -8.959723e-02 -1.862631e-01  
## [91] -1.761372e-01 -1.751574e-01 -2.486635e-01 -2.471287e-01 -1.635912e-01  
## [96] -2.507047e-01 6.654982e-02 3.523938e-01 2.915648e-01 3.084099e-01  
## [101] 3.619355e-01 3.765020e-01 2.361054e-01 3.888654e-01 1.866945e-02  
## [106] -1.228588e-01 -8.500899e-02 -8.358214e-02 -9.719430e-02 -9.690438e-02  
## [111] -1.692846e-02 -1.011225e-01

union.se # mean squared error for Union group

## [1] 34.682506 42.822910 41.023274 41.482149 40.652777 40.984330  
## [7] 27.491128 41.950922 37.628122 133.780540 122.146804 115.550235  
## [13] 194.450995 186.284622 128.632267 210.862493 8.683980 38.901566  
## [19] 32.481108 32.843102 38.429902 38.592316 14.113974 40.632516  
## [25] 35.652128 117.981071 86.899405 93.422306 110.743329 118.548034  
## [31] 56.758305 132.221725 29.055917 70.535133 64.155332 62.508161  
## [37] 51.764687 51.345928 44.264041 51.807247 36.172016 146.424342  
## [43] 102.830434 111.982827 105.185200 108.485985 63.538099 121.424008  
## [49] 19.130106 74.761851 61.060404 60.032498 79.991378 78.466873  
## [55] 34.875369 86.993325 29.119735 179.222760 104.261231 100.018769  
## [61] 131.810394 128.402870 65.609223 151.078440 48.389967 373.291935  
## [67] 293.240669 279.278182 234.687586 228.256455 310.002364 277.452177  
## [73] 9.747391 74.695723 65.121086 66.109481 75.026522 75.409769  
## [79] 46.364921 78.884079 109.999450 986.665245 813.282225 833.891216  
## [85] 714.770802 723.095060 321.114590 773.849438 60.877700 164.253918  
## [91] 148.783409 152.649667 212.913080 215.539335 147.904996 216.641292  
## [97] 80.332582 565.994535 424.604911 465.913070 485.017739 520.023488  
## [103] 379.826193 543.765267 32.845343 79.303765 53.909259 53.769655  
## [109] 60.526424 61.122220 16.869765 65.135292

For Flexion group, we firstly constructed two matrices. They are, with three columns: type, method and mean difference;  
 with three columns: type, method and squared error.

flexion.type <- c(rep(c(rep("CF",4),rep("vM",4)), 14))  
flexion.method <- c(rep(c("Idealized","Experimental","Specific","Generic"),28))  
# Flexion matrices  
flexion.data.mean <- cbind(flexion.type, flexion.method, flexion.mean )  
flexion.data.se <- cbind(flexion.type, flexion.method, flexion.se)

Next, we used two-way ANOVA procedure to conduct a hypothesis tests to check if the Flexion type, method and their interactions have significant influence on the mean difference or squared error.

# Influence of Flexion types and methods on mean difference  
flexion.mean.fit <- aov(flexion.mean ~ flexion.type\*flexion.method)  
summary(flexion.mean.fit)

## Df Sum Sq Mean Sq F value Pr(>F)   
## flexion.type 1 0.0639 0.06391 10.684 0.00147 \*\*  
## flexion.method 3 0.0573 0.01912 3.196 0.02657 \*   
## flexion.type:flexion.method 3 0.0067 0.00223 0.373 0.77284   
## Residuals 104 0.6221 0.00598   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The p-value of is 0.00147, which indicates a significant influence of Flexion type on the mean difference. Similarly, the p-value of is 0.02657, which indicates a significant influence of Flexion method on the mean difference.

# Influence of Flexion types and methods on squared error  
flexion.se.fit <- aov(flexion.se ~ flexion.type\*flexion.method)  
summary(flexion.se.fit)

## Df Sum Sq Mean Sq F value Pr(>F)   
## flexion.type 1 75975 75975 5.108 0.0259 \*  
## flexion.method 3 18659 6220 0.418 0.7403   
## flexion.type:flexion.method 3 11397 3799 0.255 0.8573   
## Residuals 104 1546770 14873   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The p-value of is 0.0259, which indicates a significant influence of Flexion type on the squared error.

For Union group, we also firstly constructed two matrices. They are,  
 with three columns: type, method and mean difference;  
 with three columns: type, method and squared error.

type.union <- c(rep(c(rep("CF",4),rep("vM",4)), 14))  
union.method <- c(rep(c("Idealized","Experimental","Specific","Generic"),28))  
# Union matrices  
union.data.mean <- cbind(type.union, union.method, union.mean )  
union.data.se <- cbind(type.union, union.method, union.se )

Next, we used two-way ANOVA procedure to conduct a hypothesis tests to check if the Union type, method and their interactions have significant influence on the mean difference or squared error.

# Influence of Union types and methods on mean difference  
union.mean.fit <- aov(union.mean ~ type.union\*union.method)  
summary(union.mean.fit)

## Df Sum Sq Mean Sq F value Pr(>F)  
## type.union 1 0.003 0.00306 0.094 0.760  
## union.method 3 0.015 0.00505 0.156 0.926  
## type.union:union.method 3 0.023 0.00779 0.240 0.868  
## Residuals 104 3.378 0.03248

All the p-values are larger than 0.05, which indicates non-significant influence of any Union factors on the mean difference.

# Influence of Union types and methods on squared error  
union.se.fit <- aov(union.se ~ type.union\*union.method)  
summary(union.se.fit)

## Df Sum Sq Mean Sq F value Pr(>F)  
## type.union 1 9964 9964 0.263 0.609  
## union.method 3 141505 47168 1.243 0.298  
## type.union:union.method 3 159942 53314 1.405 0.245  
## Residuals 104 3945671 37939

All the p-values are larger than 0.05, which indicates non-significant influence of any Union factors on the mean difference.

# Tukey HSD Test

We next conducted Tukey HSD Test to determine which methods in the group differ significantly from each other.

# Tukey test on mean difference for methods at 95% confidence level  
flexion.mean.Tukey <- TukeyHSD(flexion.mean.fit, conf.level=0.95)  
flexion.mean.Tukey

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = flexion.mean ~ flexion.type \* flexion.method)  
##   
## $flexion.type  
## diff lwr upr p adj  
## vM-CF -0.0477743 -0.07675817 -0.01879043 0.0014652  
##   
## $flexion.method  
## diff lwr upr p adj  
## Generic-Experimental -0.055162356 -0.10913295 -0.001191766 0.0431667  
## Idealized-Experimental -0.048569144 -0.10253973 0.005401445 0.0935717  
## Specific-Experimental -0.052206779 -0.10617737 0.001763811 0.0617960  
## Idealized-Generic 0.006593212 -0.04737738 0.060563801 0.9887094  
## Specific-Generic 0.002955577 -0.05101501 0.056926167 0.9989507  
## Specific-Idealized -0.003637634 -0.05760822 0.050332955 0.9980516  
##   
## $`flexion.type:flexion.method`  
## diff lwr upr  
## vM:Experimental-CF:Experimental -0.057388586 -0.14781847 0.033041300  
## CF:Generic-CF:Experimental -0.067992066 -0.15842195 0.022437820  
## vM:Generic-CF:Experimental -0.099721232 -0.19015112 -0.009291345  
## CF:Idealized-CF:Experimental -0.043217655 -0.13364754 0.047212231  
## vM:Idealized-CF:Experimental -0.111309219 -0.20173911 -0.020879333  
## CF:Specific-CF:Experimental -0.063957125 -0.15438701 0.026472761  
## vM:Specific-CF:Experimental -0.097845018 -0.18827490 -0.007415132  
## CF:Generic-vM:Experimental -0.010603480 -0.10103337 0.079826406  
## vM:Generic-vM:Experimental -0.042332646 -0.13276253 0.048097240  
## CF:Idealized-vM:Experimental 0.014170931 -0.07625896 0.104600817  
## vM:Idealized-vM:Experimental -0.053920633 -0.14435052 0.036509253  
## CF:Specific-vM:Experimental -0.006568539 -0.09699843 0.083861347  
## vM:Specific-vM:Experimental -0.040456432 -0.13088632 0.049973454  
## vM:Generic-CF:Generic -0.031729165 -0.12215905 0.058700721  
## CF:Idealized-CF:Generic 0.024774411 -0.06565547 0.115204297  
## vM:Idealized-CF:Generic -0.043317153 -0.13374704 0.047112733  
## CF:Specific-CF:Generic 0.004034941 -0.08639495 0.094464827  
## vM:Specific-CF:Generic -0.029852952 -0.12028284 0.060576934  
## CF:Idealized-vM:Generic 0.056503576 -0.03392631 0.146933463  
## vM:Idealized-vM:Generic -0.011587988 -0.10201787 0.078841898  
## CF:Specific-vM:Generic 0.035764106 -0.05466578 0.126193993  
## vM:Specific-vM:Generic 0.001876214 -0.08855367 0.092306100  
## vM:Idealized-CF:Idealized -0.068091564 -0.15852145 0.022338322  
## CF:Specific-CF:Idealized -0.020739470 -0.11116936 0.069690416  
## vM:Specific-CF:Idealized -0.054627363 -0.14505725 0.035802523  
## CF:Specific-vM:Idealized 0.047352094 -0.04307779 0.137781980  
## vM:Specific-vM:Idealized 0.013464201 -0.07696568 0.103894087  
## vM:Specific-CF:Specific -0.033887893 -0.12431778 0.056541993  
## p adj  
## vM:Experimental-CF:Experimental 0.5114761  
## CF:Generic-CF:Experimental 0.2898092  
## vM:Generic-CF:Experimental 0.0200175  
## CF:Idealized-CF:Experimental 0.8170995  
## vM:Idealized-CF:Experimental 0.0056229  
## CF:Specific-CF:Experimental 0.3675786  
## vM:Specific-CF:Experimental 0.0242702  
## CF:Generic-vM:Experimental 0.9999583  
## vM:Generic-vM:Experimental 0.8324120  
## CF:Idealized-vM:Experimental 0.9997065  
## vM:Idealized-vM:Experimental 0.5915329  
## CF:Specific-vM:Experimental 0.9999984  
## vM:Specific-vM:Experimental 0.8625421  
## vM:Generic-CF:Generic 0.9585241  
## CF:Idealized-CF:Generic 0.9897919  
## vM:Idealized-CF:Generic 0.8153357  
## CF:Specific-CF:Generic 0.9999999  
## vM:Specific-CF:Generic 0.9702110  
## CF:Idealized-vM:Generic 0.5318333  
## vM:Idealized-vM:Generic 0.9999238  
## CF:Specific-vM:Generic 0.9231505  
## vM:Specific-vM:Generic 1.0000000  
## vM:Idealized-CF:Idealized 0.2880216  
## CF:Specific-CF:Idealized 0.9965529  
## vM:Specific-CF:Idealized 0.5752027  
## CF:Specific-vM:Idealized 0.7373184  
## vM:Specific-vM:Idealized 0.9997913  
## vM:Specific-CF:Specific 0.9414405

# plot to make it clear  
plot(flexion.mean.Tukey)

