Hi-Mg Carbonate XRD data analysis-0-100% calcite range

John M. Jaeger

May 23, 2017

## code description

This code is designed to import and analyze X-ray diffraction data collected at the University of Florida on a Rigaku Ultima IV diffractometer. The data are from carbonate mineral standards created to calibrate the diffractometer to provide quantitative measures of high-Mg calcite. The general approach follows: Sepulcre, S., Durand, N., and Bard, E., 2009, Global and Planetary Change: Global And Planetary Change, v. 66, p. 1–9, doi: 10.1016/j.gloplacha.2008.07.008.

The aragonite source was a bahamiam coral courtesy of Roger Portell, Florida Museum of Natural history. The hi-Mg source were Florida sand dollars collected by Karen Vyverberg, University of Florida.

This code analyzes a full range of calcite/aragonite mixtures, with assumed values used for 100% hi-mg calcite and 100% aragonite mixtures. These assumed values are used to better constrain polynomial fit.

Code generated by: John M. Jaeger Associate Professor 241 Williamson Hall P.O. Box 112120 Dept. of Geological Sciences University of Florida Gainesville FL 32611-2120, USA (352) 846-1381 ORCID ID# orcid.org/0000-0003-0248-489X <http://people.clas.ufl.edu/jmjaeger/>

### Load packages

# if necessary uncomment and install packages.  
# install.packages("dplyr")  
# install.packages("knitr")  
library(dplyr)  
require(knitr)  
library(ezknitr)  
library(ezknitr)  
library(ggplot2)  
library(splines)  
#library(propagate)

Import the data:

# load the End-member Dataset and give the dataset a shorter name  
  
XRD.data <- read.table("../raw\_data/UF\_HMC\_data\_20170515.csv",header=T,sep=",")   
XRDdata <- tbl\_df(XRD.data)  
  
#testdata<-select(XRDdata, Sample, AreaR, HMC)  
#area.ratio <- testdata$AreaR  
#HMC.conc <- testdata$HMC

Calculate peak area ratios:

area.ratio <- XRDdata$HMC\_area/(XRDdata$HMC\_area+XRDdata$Aragonite\_area)  
HMC.conc <- XRDdata$HMCconc  
HMC.data <- cbind(area.ratio,HMC.conc)

Develop polynomial calibration model for low-Mg calcite:

model.fit3<- lm(HMC.conc~ poly(area.ratio,3))  
summary(model.fit3)

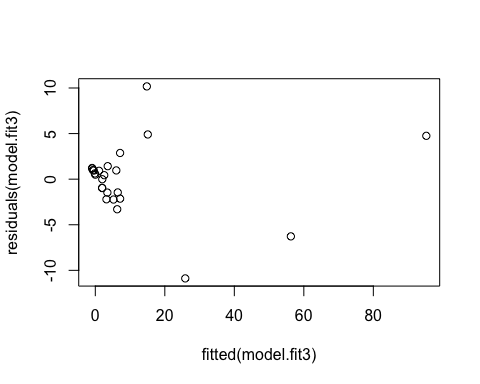
##   
## Call:  
## lm(formula = HMC.conc ~ poly(area.ratio, 3))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -10.884 -1.471 0.495 1.167 10.174   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 10.4960 0.8247 12.727 2.44e-11 \*\*\*  
## poly(area.ratio, 3)1 90.7798 4.1235 22.015 5.45e-16 \*\*\*  
## poly(area.ratio, 3)2 43.7872 4.1235 10.619 6.70e-10 \*\*\*  
## poly(area.ratio, 3)3 27.3409 4.1235 6.631 1.45e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.123 on 21 degrees of freedom  
## Multiple R-squared: 0.9683, Adjusted R-squared: 0.9638   
## F-statistic: 213.8 on 3 and 21 DF, p-value: 6.811e-16

# Confidence intervales for model parameters  
confint(model.fit3, level=0.95)

## 2.5 % 97.5 %  
## (Intercept) 8.78096 12.21104  
## poly(area.ratio, 3)1 82.20458 99.35498  
## poly(area.ratio, 3)2 35.21197 52.36236  
## poly(area.ratio, 3)3 18.76568 35.91608

Plot data and model:

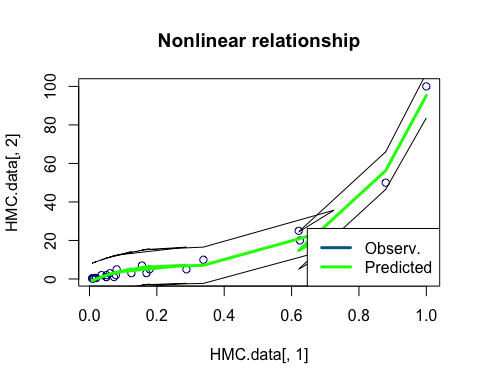
# Plot of fitted vs residuals  
# No clear pattern should show in the residual plot if the model is a good fit  
plot(fitted(model.fit3),residuals(model.fit3))



# Predicted values and confidence intervals  
polypredicted.intervals <- predict(model.fit3,data.frame(x=area.ratio),interval='prediction',  
 level=0.95)  
  
pp.int <- cbind(area.ratio,polypredicted.intervals)  
pp.int[ order(pp.int[,1]), ]

## area.ratio fit lwr upr  
## 2 0.007796116 -0.92579692 -10.007726 8.156132  
## 3 0.008547936 -0.86653504 -9.940243 8.207173  
## 1 0.010761625 -0.69372215 -9.744100 8.356656  
## 5 0.014023452 -0.44362600 -9.461954 8.574702  
## 6 0.018260848 -0.12672202 -9.107417 8.853973  
## 4 0.020057720 0.00496222 -8.961093 8.971018  
## 10 0.035671456 1.08317642 -7.786071 9.952424  
## 7 0.049643807 1.95121083 -6.871797 10.774218  
## 8 0.050056343 1.97549899 -6.846645 10.797643  
## 12 0.050579790 2.00620838 -6.814878 10.827295  
## 14 0.060459325 2.56332969 -6.245370 11.372029  
## 9 0.072937839 3.20773707 -5.603346 12.018820  
## 11 0.078421426 3.47075087 -5.346711 12.288213  
## 17 0.080624187 3.57303186 -5.247789 12.393852  
## 13 0.124442053 5.23482827 -3.716237 14.185894  
## 19 0.155887198 6.04067232 -3.035112 15.116456  
## 15 0.169234678 6.29971930 -2.827914 15.427352  
## 16 0.178588504 6.45496258 -2.707378 15.617303  
## 18 0.287860082 7.13751538 -2.264519 16.539550  
## 20 0.338251568 7.12719666 -2.293114 16.547507  
## 23 0.621150753 14.82572678 5.137027 24.514427  
## 22 0.624651535 15.09706935 5.402275 24.791864  
## 21 0.725445426 25.88382190 16.096804 35.670840  
## 24 0.879437690 56.27370479 46.509757 66.037652  
## 25 1.000000000 95.24527845 83.644283 106.846273

# Add lines to the existing plot  
plot(HMC.data[,1],HMC.data[,2],type='p',col='navy',main='Nonlinear relationship',lwd=1)  
  
lines(pp.int[,1],pp.int[,2],col='green',lwd=3)  
lines(pp.int[,1],pp.int[,3],col='black',lwd=1)  
lines(pp.int[,1],pp.int[,4],col='black',lwd=1)  
  
# Add a legend  
legend("bottomright",c("Observ.","Predicted"),   
 col=c("deepskyblue4","green"), lwd=3)



The next step is to develop uncertainty in % HMC for a given area ratio, third-order polynomial.

# Predicted values and confidence intervals  
testvalue = data.frame(area.ratio=0.1)  
predicted.value <- predict(model.fit3,testvalue,interval='prediction', level=0.95)  
predicted.value

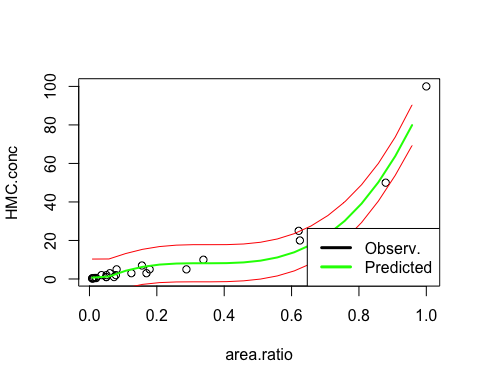
## fit lwr upr  
## 1 4.392582 -4.473789 13.25895

Now a spline model: # <https://www.r-bloggers.com/splines-opening-the-black-box/> # <http://stackoverflow.com/questions/15837763/b-spline-confusion> # <https://www.rdocumentation.org/packages/splines2/versions/0.2.4/topics/predict>

library(splines)  
spline1.pred <- lm(HMC.conc ~ bs(area.ratio, df=4))#df=4 means knots at end points and quantiles of X  
summary(spline1.pred)

##   
## Call:  
## lm(formula = HMC.conc ~ bs(area.ratio, df = 4))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -10.4006 -1.2533 0.1532 1.5550 10.2295   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.6933 2.1143 0.328 0.74638   
## bs(area.ratio, df = 4)1 -1.4999 3.2636 -0.460 0.65077   
## bs(area.ratio, df = 4)2 25.3434 7.2775 3.482 0.00235 \*\*   
## bs(area.ratio, df = 4)3 -27.8224 8.0709 -3.447 0.00255 \*\*   
## bs(area.ratio, df = 4)4 95.0198 4.2819 22.191 1.47e-15 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.115 on 20 degrees of freedom  
## Multiple R-squared: 0.9699, Adjusted R-squared: 0.9639   
## F-statistic: 161.3 on 4 and 20 DF, p-value: 6.461e-15

plot(area.ratio,HMC.conc)  
u=seq(min(area.ratio),max(area.ratio),by=.05)  
B=data.frame(area.ratio=u)  
splinepredicted.interval=predict(spline1.pred,newdata=B,interval ="prediction",level=0.95)  
lines(u,splinepredicted.interval[,1],lwd=2,col="green")  
lines(u,splinepredicted.interval[,2],col='red',lwd=1)  
lines(u,splinepredicted.interval[,3],col='red',lwd=1)  
  
# Add a legend  
legend("bottomright",c("Observ.","Predicted"),   
 col=c("black","green"), lwd=3)

 The last step is to develop uncertainty in % HMC for a given area ratio, spline model.

# Predicted values and confidence intervals  
predicted.splinevalue <- predict(spline1.pred,testvalue,interval='prediction', level=0.95)  
predicted.splinevalue

## fit lwr upr  
## 1 3.878848 -5.055359 12.81305

Export fits and prediction intervals:

polyout <- cbind(area.ratio,polypredicted.intervals)  
splineout <- cbind(u,splinepredicted.interval)  
write.csv(polyout, file="../reports/UF\_HMC\_polyfit.csv")  
write.csv(splineout, file="../reports/UF\_HMC\_splinefit.csv")