Visualising the problems with balancing lithium-sulfur batteries by "mapping" internal resistance

M. J. Lacey, K. Edström, D. Brandell

Department of Chemistry – Ångström Laboratory, Uppsala University

Supporting Information

Experimental information

Electrode preparation: Lithium bis(trifluoromethyl)sulfonimide (LiTFSI, Novolyte) and lithium nitrate (LiNO$_3$) were dried at 120 °C overnight prior to use. All other materials were used as received. To prepare the cathodes, sulfur powder (Aldrich) and Ketjen Black (KB, EC-600 JD, Akzo Nobel) were first mixed manually in a 65:21 w/w ratio and heated to 155 °C in order to melt S into the pores of the carbon. The composite was then dispersed into 5% EtOH in H$_2$O, with conductive additives and binder, to give a slurry with a final composition of solids of 65% S, 21% KB, 3.5% Super C65 (Imerys), 3.5% carbon nanofibers (CNF, 20-200 nm x 100 μm, Aldrich) and 7% binder. The binder was a 4:1 mixture of poly(ethylene oxide) (PEO, M_w = 4,000,000, Aldrich) and poly(2-ethyloxazoline) (POZ, M_w ~500,000, Aldrich). The slurry was mixed by planetary ball-milling for 2 hours and cast onto Al foil by doctor blading to a loading of 2.0 – 2.2 mg$_S$ cm^{-2}. The electrodes were dried in air, cut into 13 mm discs, transferred to an Ar-filled glove box and dried under vacuum at 55 °C overnight.

Cell preparation: CR2025 coin cells were prepared comprising the above-described cathodes, a porous polyethylene separator (SOLUPOR, Lydall Performance Materials) and lithium metal foil as the anode. The Li foil was either 125 μm ("thick Li", Cyprus Foote Mineral) or 30 μm ("thin Li", Rockwood Lithium) in thickness. The electrolyte was 6 μL mg$_S^{-1}$ of 1 M LiTFSI, 0.25 M LiNO$_3$ in a 1:1 v/v mixture of 1,2-dimethoxyethane (DME, Novolyte) and 1,3-dioxolane (DOL, anhydrous, Aldrich).

Electrochemical testing:

Electrochemical measurements were run using a VMP2 (Bio-Logic). Cells were cycled at a constant curent of C/10 (167.2 mA g$_S^{-1}$). The current was interrupted every 5 minutes for 0.5 seconds. In order to program correct cycling of the cell, the voltage was limited to between 1.8 V and 2.6 V on current steps, and the next charge/discharge step was triggered if the voltage exceeded the limits of 1.9 V or 2.5 V respectively. Resistance was

calculated from the voltage drop during each current according to Ohm's law (i.e., R = dE/dI) over the full interruption time of 0.5 s. The data was processed and plotted using a simple computer program written in-house in the programming language R. An example script is given below.
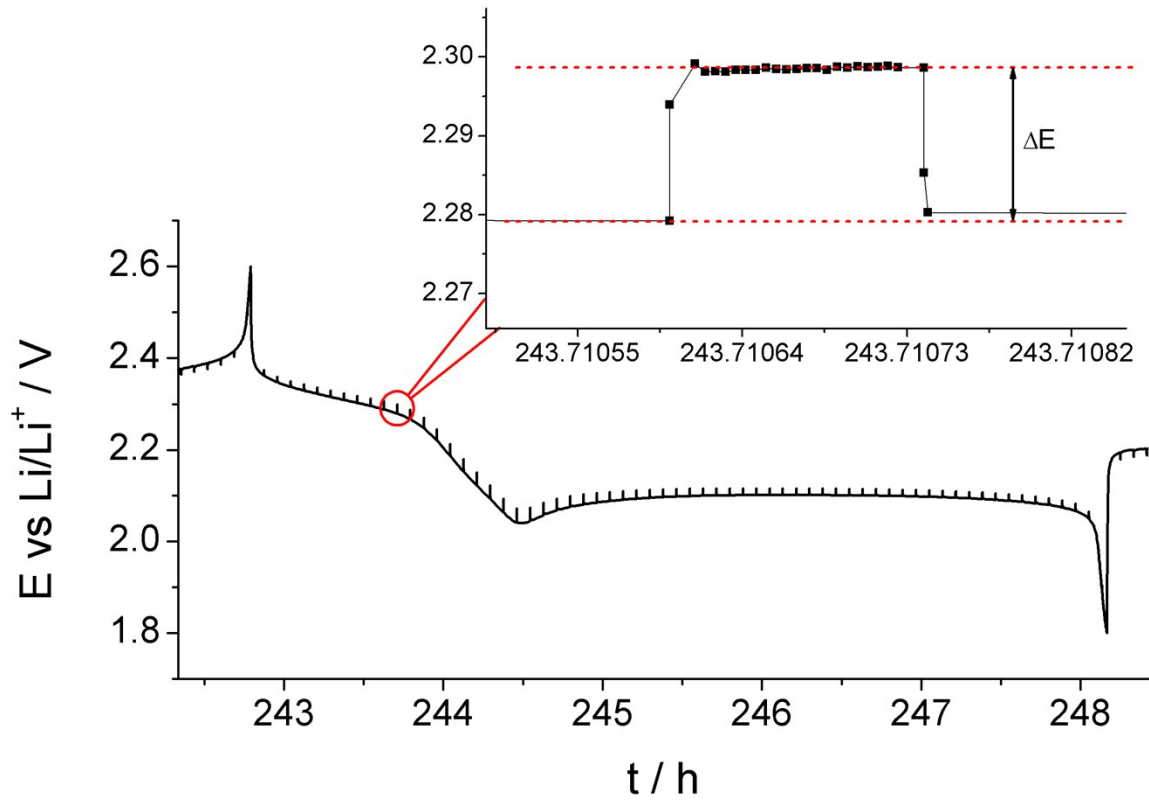
Typical voltage profile

Figure S1: Typical discharge voltage profile showing repeated current interruptions. Inset is a zoomed view of a single current interruption, indicating the voltage difference over which the resistance is determined.

Technique validation

Current-interrupt measurements were validated by comparison with other conventional measurements, specifically impedance spectroscopy (both at OCV and under load) and current pulse. A cell of similar specification was employed for this comparison: the positive electrode differs from those discussed in the main manuscript only in that it uses a binder system containing poly(vinylpyrrolidone) (PVP) instead of poly(2-ethyloxazoline) (POZ), and is coated to approx. 50% higher loading on carbon-coated aluminium. The cells show broadly similar reversible capacities and coulombic efficiencies.

The cell was cycled at a constant rate of C/10, and every five cycles was stopped exactly one hour after the start of the charge cycle, to ensure measurements were taken at almost the same state of charge. Resistance determination was then made in the following order with according to the following procedure:

- Impedance spectroscopy at OCV: the cell was allowed to relax at OCV for 5 minutes. The impedance spectrum was collected over the range 200 kHz to 200 mHz with a peak-to-peak amplitude of 10 mV.
- Current interruption: as for the mapping approach, a charging current of C/10 was applied for 5 minutes, and then interrupted, and measured over 0.5 seconds.
- Impedance spectroscopy under load: a charging current of C/10 was applied for 5 minutes. The impedance spectrum was collected by oscillating the current by ±100 μA (approx. 20% of the C/10 current) over the range 200 kHz to 200 mHz.
- Current pulse: the cell was allowed to relax at OCV for 5 minutes. Currents of ±C/10 were applied as alternating current pulses of 1 second duration (10 pulses in total). The resistance was determined from the average voltage drop.

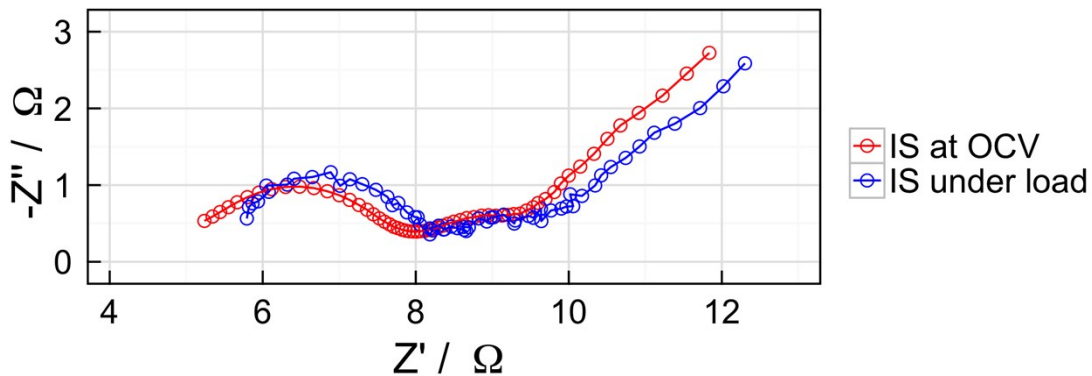Typical impedance spectra acquired at OCV and under load are presented in Fig. S2.

Figure S2: Nyquist plots of the impedance spectra acquired at the 10th cycle at OCV conditions and under C/10 load.

In general, the impedance measured under load was shifted to higher values in the "real" axis by ~0.5 Ω. Impedance spectra were fitted using ZView according to the equivalent circuit shown in Figure S3. Despite the heavily depressed semicircles the circuit fits the impedance spectra obtained at OCV relatively well while the cell shows stable, reversible cycling. The equivalent circuit may be broadly interpreted as a series combination of a combined ionic and electronic resistance (R1), charge transfer at the negative electrode (R2-CPE1) and at the positive electrode (R3-CPE2), with a constant phase element CPE3 with a fixed phase of 45° to approximate diffusion. The internal resistance was taken as the sum of the 'real' resistances, R1 + R2 + R3. A typical fit for this circuit is given in Figure S4.
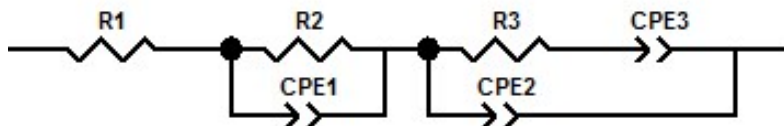

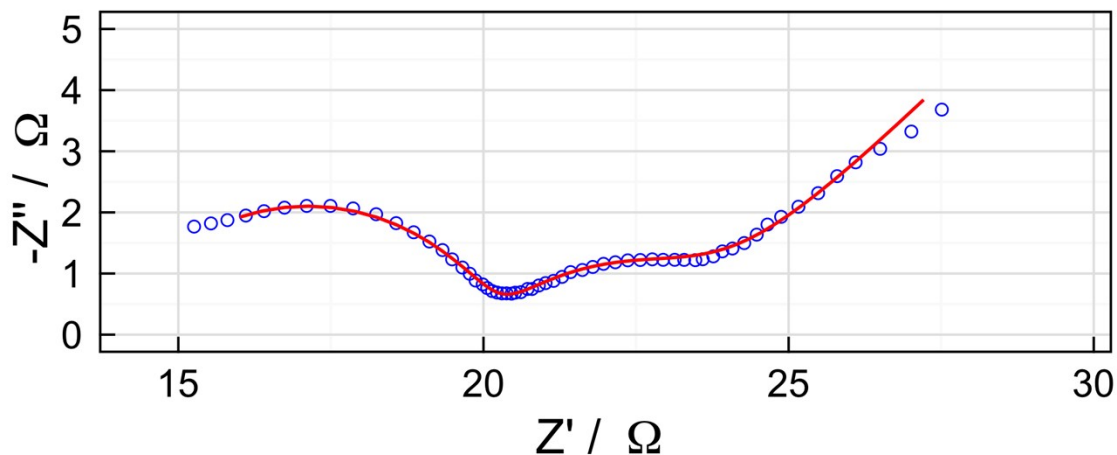Figure S3: Equivalent circuit used to fit impedance spectra.

Figure S4: typical equivalent circuit fit (red line) for an impedance spectrum acquired at OCV (blue points), fitted to the circuit shown in Fig. S3. Shown here: 60th cycle.

Despite a relatively large peak-to-peak amplitude, the noise level for the impedance spectra acquired under load was sufficiently high that ZView could not fit the spectra to the equivalent circuit. Subsequently, for this comparison, resistances were only determined from fitting of spectra acquired at OCV. It should be noted that due to the high complexity of the system, it is very difficult to fit an appropriate equivalent circuit to this data, and for most of the measured spectra we believe that the fitting slightly underestimates the true resistance.
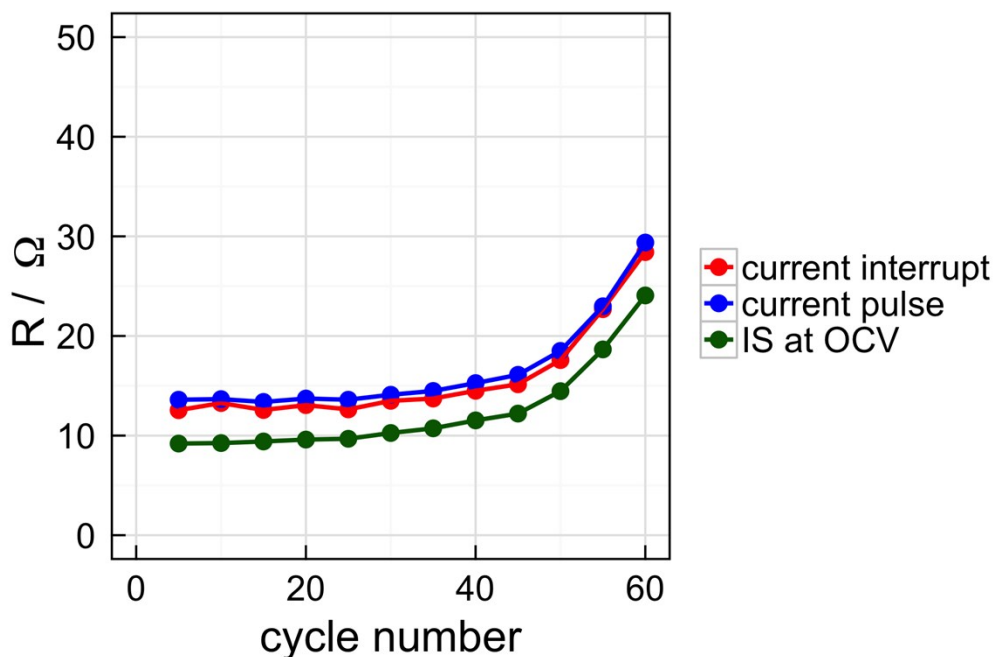

Figure S5: internal resistance for a Li-S cell measured at approximately the same state of charge over 60 cycles by three different techniques.

The determined internal resistances for typical Li-S cell using the current-interrupt, current pulse and impedance techniques is given in Figure S5. In general, the current-interrupt and current pulse methods give very similar values, whereas the values obtained from impedance are lower by a few Ohms. Although values from impedance under load could not be determined from equivalent circuit fitting, we estimate that they would be ~0.5 – 1 Ω higher than those measured at OCV, for each measurement.

In general, the DC methods (current-interrupt and current pulse) give slightly higher values for the internal resistance than the impedance method, reflecting additional contributions to the measured resistance. Crucially, however; by following the resistance by each method over 60 cycles, as shown in Fig. S5, demonstrates that each method allows the changes and trends in internal resistance to be followed accurately. For the purposes of the analysis in this manuscript, the current-interrupt method is therefore sufficiently valid.

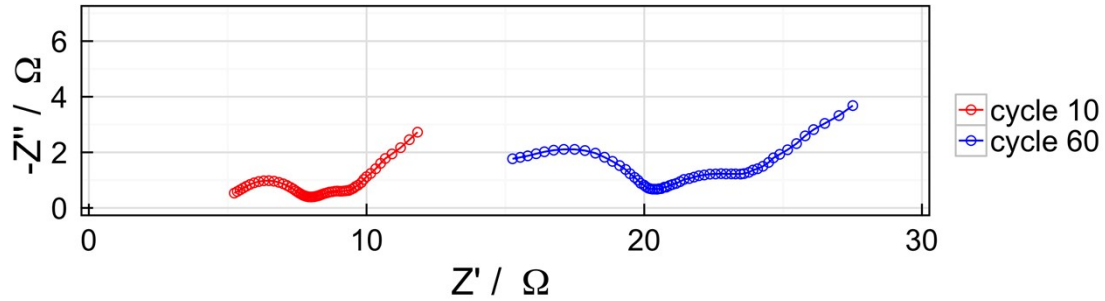**Impedance spectroscopy measurements: before and after**

Figure S6: Nyquist plots of impedance spectra measured at OCV for cycles 10 and 60 for the Li-S cell used in the technique validation (Figs. S2 – S5).

Figure S6 shows impedance spectra measured after 10 and 60 cycles for a typical Li-S cell as used in this work. In Fig. 5, the internal resistance is determined to rise from approx. 9.2 Ω on the 10th cycle to 24.1 Ω on the 60th cycle (the resistance measured by current interruption, by comparison, is determined to be 12.5 Ω at the 10th cycle and 28.4 Ω at the 60th cycle). From Figure S6 it can be seen that, for this case, the high-frequency resistance (R1, in the Fig. S3 equivalent circuit, corresponding largely to the electrolyte resistance and other electronic resistances) is the largest contributor to the increasing internal resistance (from ~5 Ω to ~15 Ω from cycles 10-60).

Data analysis with R

This is an example script written in R for analysing data exported from an Arbin battery cycling instrument. Although the experimental work presented here was performed on a Bio-Logic VMP, the script used is extremely similar; they differ only in the procedure required to import the data exported for the instrument into R (which is simpler in the case given here). This script uses the `readxl` add-on package for importing data stored in Microsoft's Excel spreadsheet format, and the `plyr` and `dplyr` packages for manipulating the imported data.

```r
setwd("/path/to/data/file") # Set the working directory to the
directory containing the data file to be analysed.
library(readxl)
library(plyr)
library(dplyr)
```

The electrode active material mass `S.mass` is defined beforehand for later unit conversion. The name of the data file is also defined here.

```r
S.mass = 2.85 # mass of the active material, in mg.
filename = "filename.xlsx"
```

Data files from this experiment are relatively large. Arbin's MITS Pro software creates a new sheet for the raw data points once the sheet reaches a maximum size, in which case multiple sheets need to be re-joined. The custom function below reads in the file, discards the first sheet (which only contains basic information) and the last sheet (summary information) and initially stores a list of the imported sheets in the variable `l`. These sheets are recombined into a single table with `rbind.fill()`, and the function ultimately returns a table with only selected columns needed for analysis.

```r
import.arbxl <- function(file, mass = 1) {

l <- lapply(excel_sheets(file)[-c(1,length(excel_sheets(file)))],
            read_excel, path = file,
            skip = 1,
            col_names=c("pt", "t", "date.t", "step.t", "stepn",
"cycn", "i", "E", "cQ", "dQ", "cEn", "dEn", "dVdt", "IR", "x", "y",
"z"))

l <- rbind.fill(l)

x <- data.frame(t = l$t, # time (s)
                cycn = l$cycn, # cycle number
                I = l$i, # current (A)
```

```
                E = l$E, # voltage (E)
                cQ = l$cQ * (1E6/mass), # charge capacity (Ah)
                dQ = l$dQ * (1E6/mass) # discharge capacity (Ah)
    )
    return(x)
}
```

The function is called as below using the previously defined variables for the filename and active material mass, with the data imported into the object raw.

```
raw <- import.arbxl(filename, mass = S.mass)
```

Data management

To assist the analysis each data point is assigned as at rest "R", discharging "D" or charging "C" according to the current at that point, and a new column state is created in the raw table.

```
raw$state <- sapply(c(1:nrow(raw)), function(i) {
  if(raw$I[i] == 0) {
    return("R")
  } else {
    if(raw$I[i] < 0) return("D")
    if(raw$I[i] > 0) return("C")
  }
})
```

Each individual current interruption is then assigned a unique number so that the resistances can be calculated. A counting variable is first defined:

```
count.rest <- 0
```

The unique identifier is then determined using the sapply() function which works row by row through the entire data table. If any given point is marked as "R" and the previous data point is not "R", the counter variable is incremented with the assign() function.

```
raw$rests <- sapply(c(1:nrow(raw)), function(i) {
    if(i == 1) {
    } else {
      if(raw$state[i] == "R" & raw$state[i-1] != "R") {
        assign("count.rest", count.rest + 1, envir = .GlobalEnv)
      }
    }
    return(count.rest)
})
```

To simplify data plotting later, a single value for the capacity is calculated from the individual discharge and charge capacities.

```
raw$adjQ <- raw$dQ - raw$cQ
```

A function, `f.R` is then defined. This function calculates dE/dI using the last data points from the current and OCV steps respectively for each unique current interruption, i.e., the voltage drop is measured over the full duration of the current interruption (0.5 seconds, in this case).

```
f.R <- function(i) {
(tail(raw$E[raw$rests == i & raw$state == "R"], 1) -
    tail(raw$E[raw$rests == i-1], 1)) / (-1 * tail(raw$I[raw$rests ==
i-1], 1))
  }
```

The resistance is calculated for each individual current interruption using the `f.R` function and `sapply()`. A new table of processed data, `proc` is created which includes other information, e.g., capacity, voltage, etc. as measured at the end of the current step, as needed for plotting or further analysis.

```
proc <- data.frame(
   rest = seq(1:max(raw$rests)),
   state = sapply(c(1:max(raw$rests)), function(i)
tail(raw$state[raw$rests == i-1], 1)),
   cycn = sapply(c(1:max(raw$rests)), function(i)
tail(raw$cycn[raw$rests == i-1], 1)),
   Q = sapply(c(1:max(raw$rests)), function(i) tail(raw$adjQ[raw$rests
== i-1], 1)),
   E = sapply(c(1:max(raw$rests)), function(i) tail(raw$E[raw$rests ==
i-1], 1)),
   R = sapply(c(1:max(raw$rests)), f.R)
)
```

A final (but generally not necessary) step is to remove any data points with infinite values in case of artefacts in the data collection by converting to NA – which will be ignored when plotted.

```
proc$R[!is.finite(proc$R)] <- NA
```

Plotting

A typical "resistance map" plot can be created using `ggplot2`:

```
library(ggplot2)
library(scales)
```

```r
library(grid)
library(RColorBrewer)

custom.theme <- theme_bw() +
  theme(text = element_text(size=21)) +
  theme(axis.title.x = element_text(vjust=-0.3)) +
  theme(axis.title.y = element_text(vjust=0.8)) +
  theme(panel.border = element_rect(size=1,color = "black")) +
  theme(axis.ticks.length=unit(-0.25, "cm"),
axis.ticks.margin=unit(0.5, "cm")) +
  theme(panel.grid.major = element_line(size=0.5))

palette <- colorRampPalette(rev(brewer.pal(11, "Spectral")),
space="Lab")

ggplot(proc) +
  geom_tile(aes(x=cycn, y=Q, fill=R), width=1, height=14) +
  facet_grid( ~ state) +
  scale_fill_gradientn("R / "~Omega, colours=palette(100),
limits=c(min(proc$R),max(proc$R))) +
  xlab("cycle number") +
  ylab("capacity / mAh g"^-1~"") +
  custom.theme
```