# GET: Generalised Epithelial Transport

The aim of GET is provide a tool which lets users access existing epithelial transport models (from the model repository), assemble new models from collections of existing CellML models, and make use of standard experiment protocol descriptions to test their models against existing data. It’s the first step toward the whole-nephron modelling tool.

The goal of this document is for Andre to keep track of what he is doing in a format that is hopefully useful for others to follow along.

Here is the EB 2013 abstract submitted on this tool:

Generalized Epithelial Transport Modeling: Computational Physiology

David Phillip Nickerson1, Kirk Lee Hamilton2, Jonna Terkildsen1,

Peter John Hunter1. 1Auckland Bioengineering Institute, The

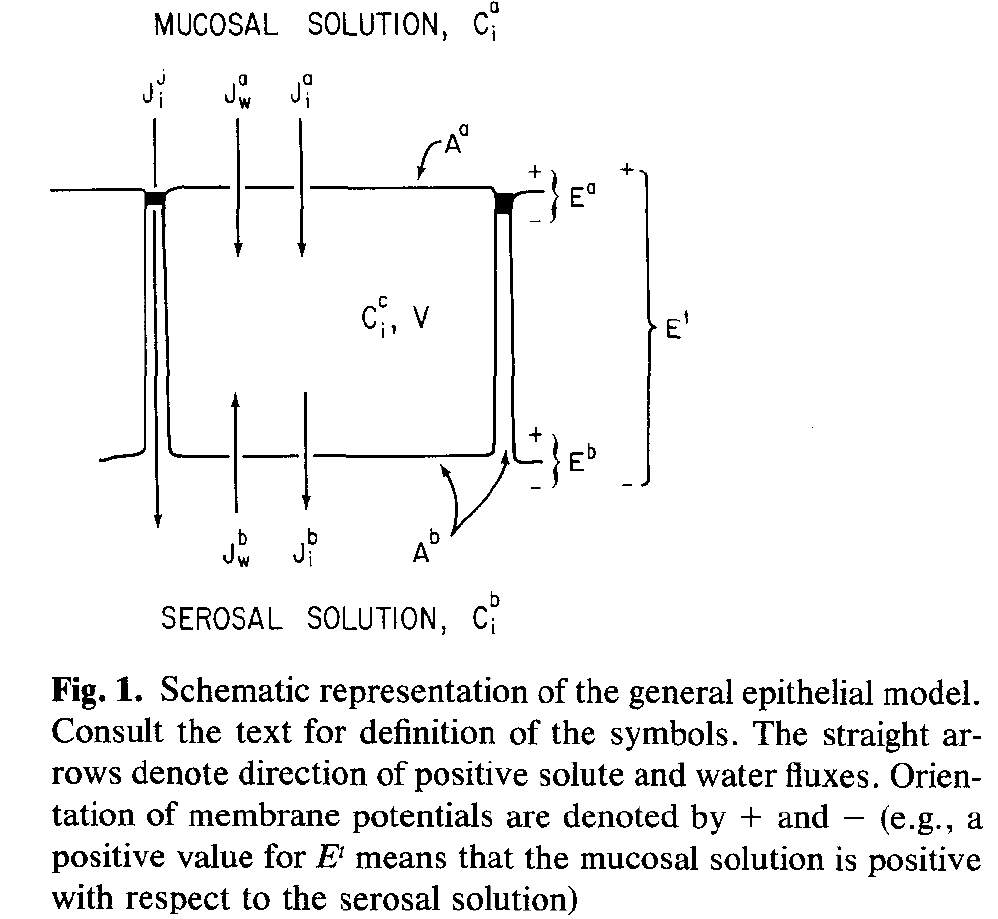
University of Auckland, Auckland, New Zealand, 2Department of

Physiology, University of Otago, Dunedin, New Zealand

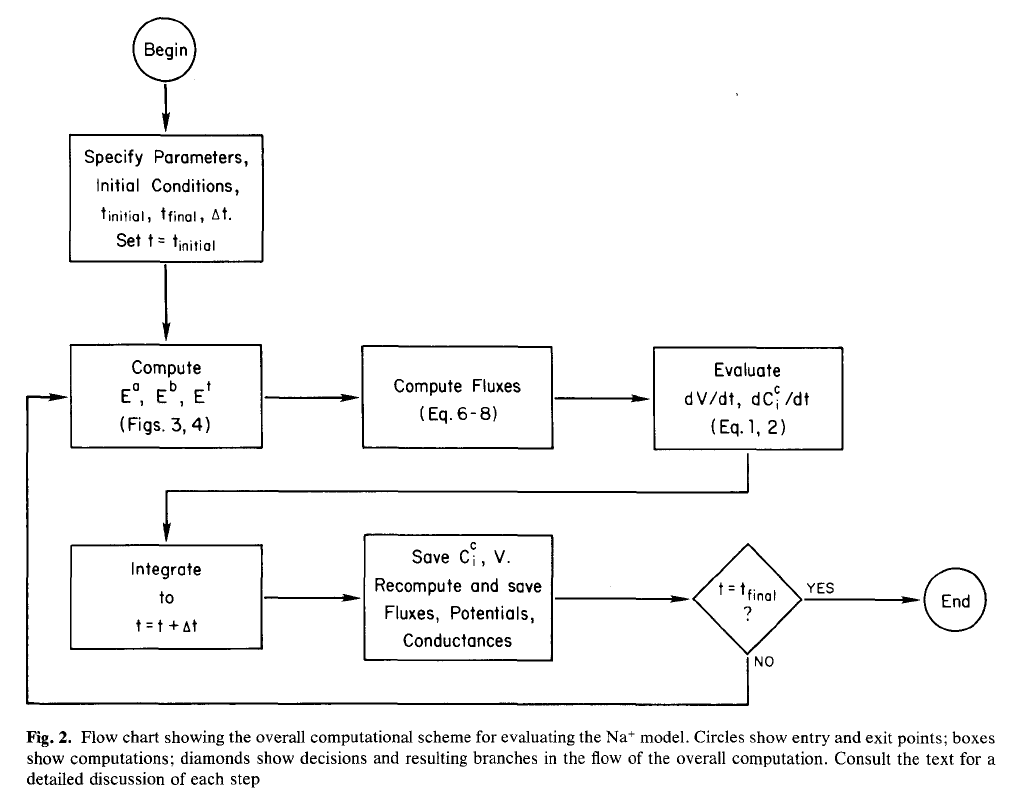
We present an epithelial modeling tool facilitating the development and validation of models of individual transport processes and whole cell transport. Making use of established standards in computational physiology, our tool enables the user to assemble epithelial cell models from a library of reusable transport process descriptions, starting from either a blank canvas or pre-existing templates and models from the library. We are developing a library of reusable renal epithelial transport models, ranging from models of individual transporters to whole cell models. Our tool makes use of biological information associated with these models, such as location within a cell membrane or identification of the ions being transported, to guide the user in the assembly of new models. This, combined with other associated information in our library, enables the user to determine a suitable starting point for their work. Such as adding a novel model of the SGLT2 to an existing proximal tubule cell model; or a generic epithelial cell ready for specific transporter models. We additionally associate experimental protocols with the mathematical models, which allow the user to compare their work against existing data. Users are able to contribute new models, protocols, and data to the library for easy dissemination to, and reuse by, the community. Supported by the Virtual Physiological Rat Project, NIH grant [P50-GM094503].

## 15 November 2012: Status

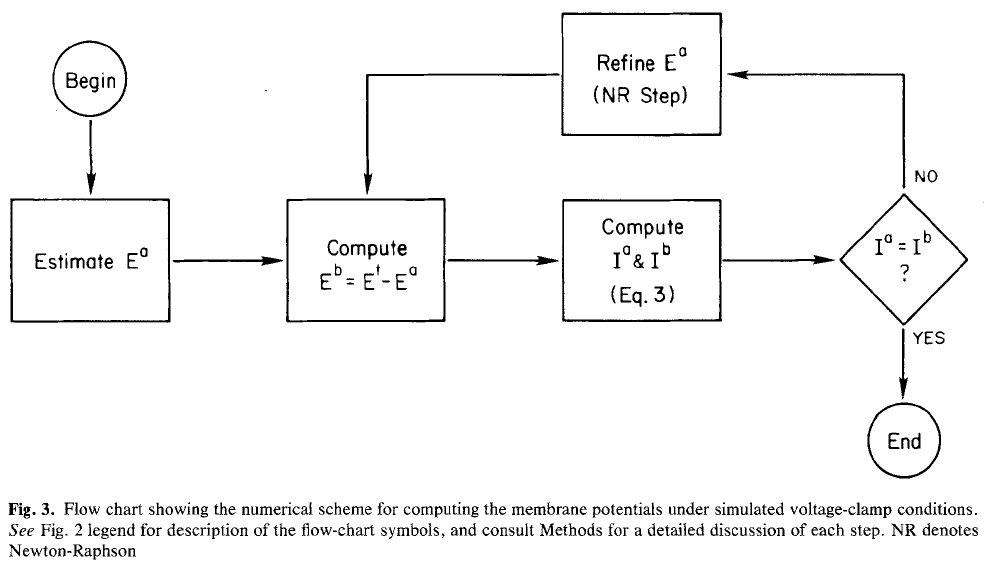
Following the methods described in the article: *General Method for the Derivation and Numerical Solution of Epithelial Transport Models* by Richard Latta, Chris Clausen, and Leon C. Moore (*J. Membrane Biol.* **82**, 67-82 (1984)), I have coded up the first version of GET which hard-codes their test ‘Na+ model’ and computational scheme for evaluating it. Their general epithelial model is shown below.



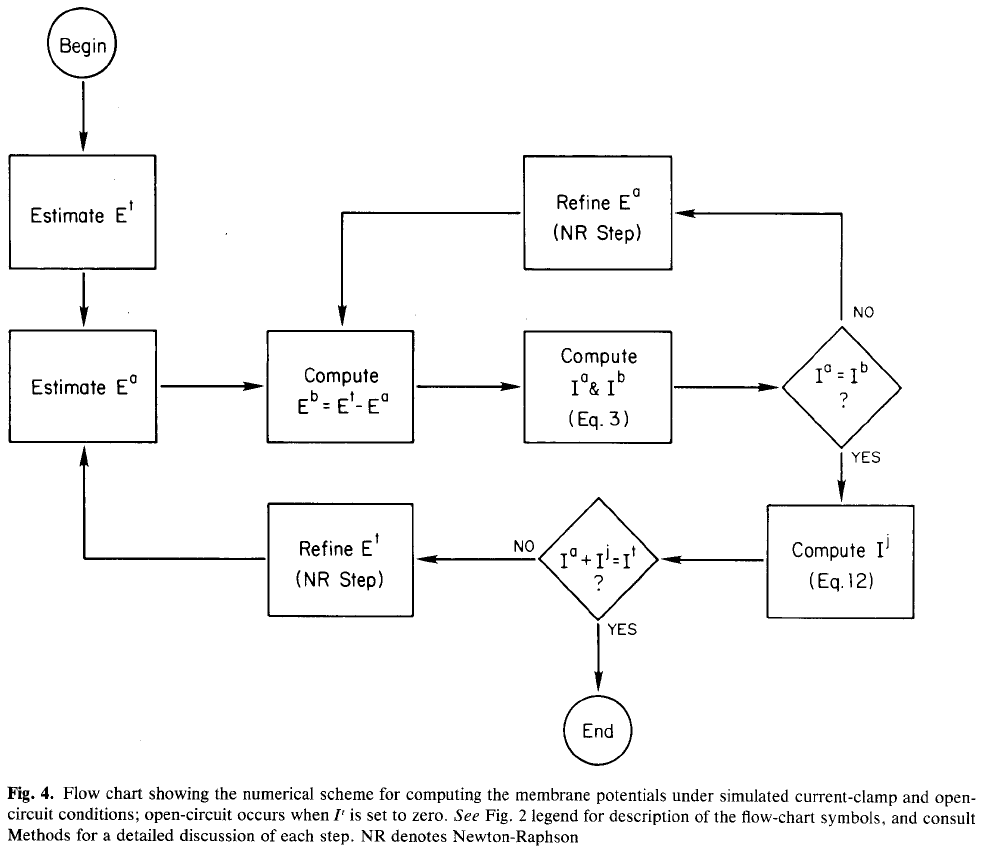
The actual ‘Na+ model’ includes five intracellular solutes, namely Na+, K+, Cl-, X1-, and X2. The overall computational scheme for evaluating the model is shown below:



The step computing Ea, Eb, and Et takes into account the need to maintain electroneutrality, and the strategy Latta et al uses to compute the potentials depends on whether the computation is simulating a preparation that is voltage-clamped transepithelially (Et=constant; Et=0 short-circuit current It=Isc), or one that is under current-clamp (It=constant) or open-circuit conditions (It=0). For the voltage-clamp case the following numerical scheme is used to compute Ea and Eb:



For the current-clamp case, the following scheme is used to compute Ea and Eb:



As shown in Fig. 4, the current-clamp case consists of a series of voltage-clamp iterations from Fig. 3. GET makes use of KINSOL in place of the Newton-Raphson minimisation in the above figures, and I had to tweak the various KINSOL parameters/configurations in order to reliably get the entire Latta protocol to compute. CVODES is used as the integrator.

### Results

As mentioned above, GET currently consists of the Latta et al Na+-model hard-coded along with the algorithm described in Fig 2-4 above which lets me reproduce the results shown in the Latta et al paper. [Note for Andre: this is revision 3401 in my subversion repository.]

**Steady state values:**

|  |  |
| --- | --- |
| C\_c[Na] = 5.07327E+00  C\_c[K] = 7.16444E+01  C\_c[Cl] = 1.16925E+01  E\_t = -7.60549E+01  E\_a = -1.81879E+01  E\_b = -5.78669E+01  V/V(t=0) = 9.68858E-01 |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## 21 November 2012: planning

These are the next steps to implement as I currently see it:

1. Modularise the hard-coded Na-model to make sure the single “biophysical mechanism” (?) – the sodium pump – is distinct from the base passive flow across the membranes. Check that the same results for the Latta protocols persist.
2. Replace the hard-coded sodium pump with a CellML description of the same pump. This will still have the hard-coded connections to the appropriate compartments/membrane. Check that the same results for the Latta protocols persist.
3. Add in the annotations and logic to allow the sodium pump to be automatically connected into the cell model. Initially, just use GET-specific annotations to get things working, but maybe look into FMA or SBO to see about compartment and membrane identifiers. Check with Mike and Sarala regarding properties for describing transport between compartments/across a membrane. Check that the same results for the Latta protocols persist.
4. Start playing with some of the other cell models, particularly the ones Jonna has already coded in CellML.