18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009
http://mssanz.org.au/modsim09

An algorithm for simulation of electrochemical systems with surface-bulk coupling strategies

Buoni, M., Petzold, L.

University of California, Santa Barbara, Department of Computer Science
Email: petzold@engineering.ucsb.edu

Abstract: In this work we have developed a general and fully three-dimensional numerical strategy for simulating electrochemical systems on irregular domains with moving boundaries. This involves solving the governing partial differential equations with algebraic constraints in the bulk electrolyte which are stiffly coupled to an active surface where chemical reactions and other physical processes occur. Our method makes only a few assumptions about the active surface, namely that it is driven by the bulk chemical species concentrations and applied potential and that it produces a flux of each species back into solution. Otherwise, the details of the surface model are of little consequence. The particular application we study here is copper electrodeposition as applied to filling trench/via interconnects in computer processors.

To summarize, our computational method for the bulk electrolyte region splits the three distinct physical phenomena that occur into fractional time steps. The homogeneous reactions and diffusion are handled with backward Euler discretization to treat stiffness, while the electrical migration is treated with a projection step which satisfies the charge neutrality constraint exactly. Spatial discretization is performed using the finite volume method, which conserves species mass exactly and retains second order spatial accuracy, even near the irregular boundary. A nonuniform grid region above the active surface is used to resolve the diffusion boundary layer that is about two orders of magnitude thicker than the trench dimensions. The level set method is adopted to move the interface, but modified to prevent the degradation of accuracy that can result from the first order accurate fast marching method. The closest point algorithm is used to reconstruct the interface with second order accuracy before redistancing is performed. To couple the bulk electrolyte and active surface regions, we developed a semi-implicit coupling method that handles the stiff coupling problem robustly and efficiently.

With numerical experiments, we found that the CPU times scale as a small power of the problem size, approximately 1.15. Direct comparison of our method's efficiency with existing numerical strategies for trench infill revealed that our method can compute in about 30 minutes what previously took 8 hours. In addition, previous methods scaled very poorly with grid refinement, with a power law approximately 2.0, and were not second order accurate near the moving boundary. For the first time to our knowledge, fully three-dimensional time varying simulations of multi-component electrochemical systems are now feasible on a desktop computer, albeit at modest resolutions. We note that higher resolution could be made feasible by parallelizing the algorithm, which we are currently pursuing.

Finally, we applied our coupled bulk-surface algorithm to study the infill of a three dimensional via. We found that our method is able to predict the superfilling phenomenon seen experimentally with a careful balance of solution additives, as well as some subtle details of the infill characteristics.

Keywords: *electrochemical systems, electrodeposition, numerical simulation, coupling algorithm*

Buoni, M., Petzold, L., An algorithm for simulation of electrochemical systems with surface-bulk coupling strategies

1. INTRODUCTION

Electrochemical processes are widely used throughout industry. Applications include batteries, fuel cells, photovoltaics, application of coatings onto metals and fabrication of interconnects in computer processors. Generally such processes involve electrolytic solutions containing various ions and additives interacting with conducting surfaces on which chemical reactions and various physical processes occur. These surface processes either produce a voltage difference or are induced by an applied voltage difference between the surface and electrolyte solution. Although the fundamental equations describing the bulk electrolyte solution are well known, the surface chemistry and dynamics are often less clear and subject to modeling trial and error. We have developed a highly efficient algorithm that solves the governing equations for the bulk while allowing the user to include his or her particular surface model with relative ease.

Here we describe our algorithm for simulation of electrochemical systems on three-dimensional, irregular domains with moving boundaries. We give particular attention to the coupling of the dilute electrolyte model of the bulk to the surface dynamics model on the moving boundary. Our method uses finite volume discretization of the arbitrarily shaped spatial domain. The resulting differential algebraic equation system is solved with a time splitting method that involves a projection step used to satisfy the algebraic constraints. We illustrate the power of our method by applying it to the challenging problem of simulating electrodeposition of a copper via structure. We will show that simple, intuitive strategies for coupling the surface and bulk models fail, resulting in unstable simulations unless the time step is reduced well beyond the point of computational feasibility. The active boundary and surface variables are advected using the level set method, but modified with the closest point algorithm to prevent the degradation of accuracy that can result from the first order accurate fast marching method.

It is important to emphasize that to date very few three-dimensional simulations of realistic models for copper electrodeposition have been performed. Three-dimensional simulations are essential for studying problems such as copper nucleation and growth dynamics on resistive metal substrates, and for studying the process of copper infill of via interconnect structures in computer processors. Figure 1 illustrates such a system. In this figure we see that the via consists of a trench shaped region 20-200 nm in width and around 5:1 to 15:1 aspect ratio. At the bottom of the trench a tapered cylinder is bored out (used to connect to a lower layer of interconnects on a computer processor). The region above the trench consists of dilute electrolyte solution (bulk) and forms a mass flux boundary layer. At the top of the layer (10-100 μm), the solution is well mixed. It has a fixed far-field composition represented by a Dirichlet boundary condition.

This paper is organized as follows. In Section 2 we give a brief description of the governing equations, including the active surface. In Section 3 we discuss the numerical method, with particular attention given to the surface/bulk model coupling. In Section 4 we give the results of a series of numerical experiments that test the algorithm's accuracy and efficiency. Finally, in Section 5 we illustrate the method by simulating the fully three-dimensional copper infill of a via.

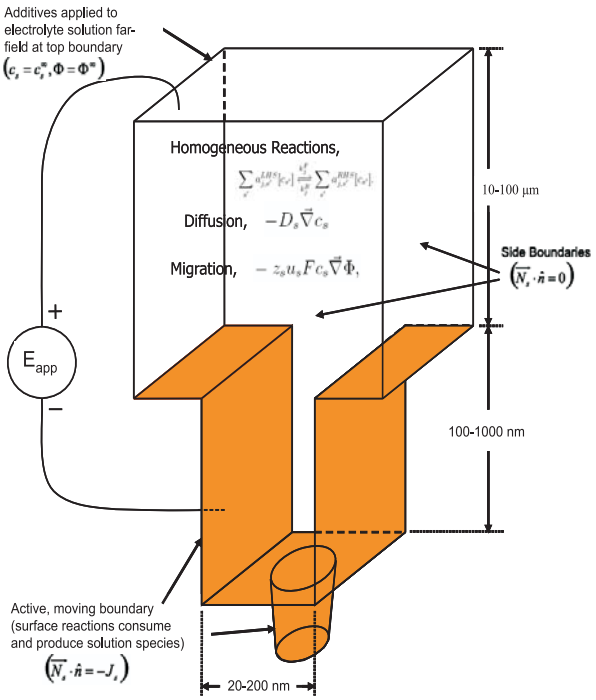

Figure 1. Illustration shows the shape and dimensions of the via physical domain. The governing physics are also indicated for the different regions.

2. GOVERNING EQUATIONS

2.1. Bulk Electrolyte

The dilute electrolyte in the bulk is modeled by the mass flux given by Nernst-Planck Equation with homogeneous reaction source terms plus electroneutrality. These equations describe the time evolution of the concentrations of each chemical species, c_s,

$$\frac{\partial c_s}{\partial t} = R_s\left(\{c_s\}\right) - \vec{\nabla} \cdot \vec{N_s}$$

where R_s is the net rate of production of chemical

species *s* due to chemical reactions, and is a function of all the other chemical species concentrations, c_s. The flux, $\overrightarrow{N_s}$, of chemical species *s* to due to diffusion and electric field migration is given by

$$\overrightarrow{N_s} = -D_s \vec{\nabla} c_s - F z_s u_s c_s \vec{\nabla} \Phi ,$$

where Φ is the electric potential, D_s is the diffusion coefficient for species *s*, z_s is the charge of species *s*, $u_s = D_s / RT$ is the mobility constant for species *s*, and *F* is Faraday's constant. The algebraic constraint enforces zero net charge density for the electrolyte solution.

2.2. Active Surface

Our computational framework is quite general and can accommodate a variety of different types of surface models. Included are fully stochastic Kinetic Monte-Carlo models, fully deterministic mean field models, as well as hybrid approaches that treat important events (such as copper nucleation) stochastically but everything else deterministically. For electrodeposition into a via, we will be using an ordinary differential equation model. In such a mean field model, we have a set of adsorbed surface species, {*ss*}, each having a (constant) molar surface density Π_{ss} and fractional surface coverage θ_{ss}, satisfying $\sum_{ss} \theta_{ss} = 1$.

Mass conservation on the moving, reacting surface is represented by the reaction-advection equation,

$$\frac{\partial \theta_{ss}}{\partial t} = \frac{R_{ss}^{surf}}{\Pi_{ss}} - v_n \kappa \theta_{ss} ,$$

where R_{ss}^{surf} is the net production rate of surface species *ss* due to surface reactions, $\vec{v} = v_n \hat{n}$ is the interface velocity and κ is the interface curvature. Additional details of this surface model can be found in the references.

3. NUMERICAL SOLUTION

3.1. Bulk Electrolyte

Temporal discretization is accomplished via a splitting technique that uses the Implicit Euler method combined with a projection step. The basic idea is as follows. First, we perform a finite volume discretization over the spatial domain. For this, we consider the most complex case: a cell that is cut in an arbitrary way by the active (moving) boundary. From this we obtain an equation of the form:

$$V_{i,j,k}^{rel} \frac{d\overline{(c_s)}_{i,j,k}}{dt} = (RHS)^{(rxns)} + (RHS)^{(diff)} + (RHS)^{(mig)} + (RHS)^{(flux,c)} + (RHS)^{(flux,\Phi)} ,$$

where $\overline{(c_s)}_{i,j,k}$ is the concentration of chemical species *s* at the centroid of cell (*i,j,k*), while $(RHS)^{(rxns)}$, $(RHS)^{(diff)}$, $(RHS)^{(mig)}$, $(RHS)^{(flux,c)}$ and $(RHS)^{(flux,\Phi)}$ are the cell-integrated reaction, diffusion, migration and boundary flux (diffusion and migration, respectively) terms.

The right hand side of the species conservation of mass equation is then split into three sets of terms: 1) reaction terms, 2) diffusion terms (plus diffusion boundary flux terms), and 3) migration terms (plus migration boundary flux terms), as indicated by the superscript. To advance the concentration fields, $\overline{(c_s)}_{i,j,k}$, from time t_n to $t_{n+1} = t_n + \Delta t$, two intermediate values, $\overline{(c_s)}_{i,j,k}^{(*,rxns)}$ and $\overline{(c_s)}_{i,j,k}^{(*,diff)}$, are calculated using Backward Euler discetization. Schematically, we do the following:

$$(c_s)_{i,j,k}^{(n)} \rightarrow^{reactions} (c_s)_{i,j,k}^{(*,rxns)} \rightarrow^{diffusion} (c_s)_{i,j,k}^{(*,diff)} \rightarrow^{projection} \Phi_{i,j,k} \rightarrow^{migration} (c_s)_{i,j,k}^{(*,n+1)}$$

By *projection*, what is meant is that $\Phi_{i,j,k}$ is computed such that after migration, the charge neutrality constraint is satisfied at every solution-containing cell center. Solving for $\Phi_{i,j,k}$ requires the solution of an elliptic Poisson-like equation.

The diffusion step and projection step require solution of nearly symmetrical linear equation systems, with asymmetrical terms appearing along the irregular boundary. These equation systems are solved easily and efficiently using Sparsekit2. We note that similar discretizations have been used to solve the heat and Poisson equations on irregular grids.

3.2. Coupling Algorithm

The bulk electrolyte and surface models are coupled by the fluxes, $\{J_{s'}\}$, and bulk electrolyte concentrations and potential at the surface, $\{c_{s'}, \Phi\}$. The fluxes are output from the surface reaction code and provide boundary conditions to the bulk electrolyte code, while the concentrations and potential are output from the bulk electrolyte code and act as parameters in the surface reaction code. In our work, we explored three different approaches for coupling the codes: *explicit coupling*, *fully implicit coupling*, and *semi-implicit coupling*. Explicit coupling and fully implicit coupling were unable to solve our problem efficiently, so we had to resort to a less intuitive semi-implicit coupling method, which we explain in more detail.

To aid in the discussion, we introduce the following notation. Define one iteration of the surface code (reactions only) to be $F = (F_1, F_2)$,

$$\theta^{(n+1)} = F_1\left(\theta^{(n)}, c\right)$$
$$J^{(n+1)} = F_2\left(\theta^{(n)}, c\right)$$

Here, the bulk concentration (and potential) c may be regarded as a constant parameter during the integration window $[t_n, t_n + \Delta t]$. Similarly, we define one iteration of the bulk code to be G,

$$c^{(n+1)} = G\left(c^{(n)}, J\right)$$

This time, the active boundary flux J is regarded as the constant parameter.

3.2.1 Explicit and Fully Implicit Coupling

Using the notation introduced above, explicit coupling is written as

$$\theta^{(n+1)} = F_1\left(\theta^{(n)}, c^{(n)}\right)$$
$$J^{(n+1)} = F_2\left(\theta^{(n)}, c^{(n)}\right)$$
$$c^{(n+1)} = G\left(c^{(n)}, J^{(n+1)}\right)$$

Examining the system Jacobian, we find that explicit coupling will be unstable if the time step size is greater than the fastest surface reaction timescale. This is a *very* severe restriction and renders the method infeasible for all but the most trivial systems.

In contrast, fully implicit coupling is given by

$$\theta^{(n+1)} = F_1\left(\theta^{(n)}, c^{(n+1)}\right)$$
$$J^{(n+1)} = F_2\left(\theta^{(n)}, c^{(n+1)}\right)$$
$$c^{(n+1)} = G\left(c^{(n)}, J^{(n+1)}\right)$$

Although the system Jacobian is actually stable, the difficulty lies in solving for $c^{(n+1)}$ in an efficient way. We designed an iterative method that would converge within a few iterations most of the time, but was not robust. Other times the iterations would only converge if the time step size was reduced beyond computational feasibility.

3.2.2 Semi-implicit Coupling

Semi-implicit coupling attempts to capture the best features of explicit coupling and fully implicit coupling. The fact is it does even better than that. It is more stable and robust than fully implicit coupling, but every bit as efficient per time step as explicit coupling. Semi-implicit coupling solves the following equations at each time step:

$$\theta^{(n+1)} = F_1\left(\theta^{(n)}, c^{(n)}\right)$$
$$J^{(n+1)} = F_2\left(\theta^{(n)}, c^{(n)}\right)$$
$$c^{(n+1)} = G\left(c^{(n)}, J^{(*)}\right)$$

Here, $J^{(*)}$ is the semi-implicit flux given by the mixed time step expression

$$J_s^{(*)} = \alpha_s^{(n)} + \beta_s^{(n)} c_s^{(n+1)},$$

where α and β are functions of θ and c.

The expression for $J_s^{(*)}$ uses the fact that J_s is linear in c_s. In other words, both α_s and β_s are independent of c_s. This linearity assumption holds in our model since surface reaction rates are proportional to bulk concentrations with constant rest potentials for Faradaic reactions. Furthermore, they are trivial to compute in terms of the flux function F_2,

$$\alpha_s^{(n)} = F_2\left(\theta^{(n)}, c^{(n)} - c_s^{(n)}\right)$$
$$\beta_s^{(n)} = F_2\left(\theta^{(n)}, c^{(n)} - c_s^{(n)} + 1_s\right) - \alpha_s^{(n)}$$

Thus, $\alpha_s^{(n)}$ is calculated by computing the flux function F_2 with $c_s = 0$, and $\beta_s^{(n)}$ is calculated by computing the flux function with $c_s = 1$ and subtracting $\alpha_s^{(n)}$. Although linearity does not hold for more refined surface models, we can modify this approach by allowing $\beta_s^{(n)}$ to have c_s dependence. We have done this successfully for rest potentials with c dependence (Nernst correction).

The linear decomposition of J_s with respect to c_s is precisely what gives this method its high computational efficiency. Recall from Section 3.1 that the active boundary flux terms are included in the *diffusion step* of the bulk splitting algorithm. In that step, we solve a separate linear system for each bulk species, *s*. Since our expression for $J_s^{(*)}$ is linear in c_s, the implicit part ($\beta_s^{(n)} c_s^{(n+1)}$) may be readily incorporated into the linear system and solved at little or no extra computational expense.

Examining the system Jacobian shows more clearly why this method works. The implicit part of the flux containing the stiffest terms appears in an inverse matrix, which provides stability. This strategy may also be regarded as a highly efficient approximation to the fully implicit Jacobian.

4 NUMERICAL TESTS

4.1 Accuracy

We tested the accuracy of our coupled models using the full additive surface reaction model, described in the references. The domain shape was a nearly fixed trench shape (180nm width, 5:1 aspect ratio and 50 μm boundary layer) and was run for 0.5 seconds (during the initial system transient). A total of four

computational grids were used, with $\Delta t/\Delta x^2$ fixed. The resolutions and fixed time step sizes are given in Table 1.

The coarser grid solutions (grids 1, 2 and 3) were compared to grid 4 by averaging the grid 4 numerical solutions over 8 x 8, 4 x 4 and 2 x 2 sub-grid squares, respectively. Errors were computed using the following definition for the error in species *s*,

$$\left(E_s\right)_{L_\infty} = \frac{\max_{i,j}\left|(c_s)_{i,j} - (c_s)_{i,j}^{(exact)}\right|}{\max_{i,j}\left|(c_s)_{i,j}^{(exact)}\right|}$$

The actual data is not included for brevity, but rather is discussed. The first observation is that there is a large difference in the accuracy of the most to least accurate species. This is because some species change very little relative to their far-field concentration and reach a steady-state quickly. In addition, species with small flux to diffusivity ratios tend to be computed more accurately, which is the result of small, well-resolved gradients near the active boundary. On the other hand, species that are not present as additives but are generated by fast surface reactions and consumed equally quickly tend to have more complex spatial concentration distributions and are more difficult to resolve, making them subject to larger numerical errors. We also observe that the numerical errors in some species converge slightly slower than the expected $O\left(\Delta x^2\right)$. The large flux creates a sharp boundary layer in these species near the active boundary during the transient that is difficult to resolve with our uniform grid spacing.

Grid no.	Uniform region res.	Time step, Δt (seconds)
1	20 x 50	4.0×10^{-3}
2	40 x 100	1.0×10^{-3}
3	80 x 200	2.5×10^{-4}
4	160 x 400	6.25×10^{-5}

Table 1. Grid resolutions and time step sizes for coupled models, short timescale tests.

Grid no.	CPU time per time step (seconds)	Total CPU time
1	0.056	37 min.
2	0.198	5.5 hr.
3	1.24	5.7 days
4	6.30	117 days

Table 2. CPU time (P4 - 3.2 GHz) data for 2D trench problem.

Grid no.	CPU time per time step (seconds)	Total CPU time
1	1.28	8.9 hr.
2	13.7	15.9 days
3	150	1.9 years
4	1559	79 years

Table 3. CPU time (P4 - 3.2 GHz) data for 3D via problem.

4.2 Efficiency

Here, we measure the efficiency for both a 2D and 3D problem (using semi-implicit coupling).

First, we consider the trench shaped domain in 2D. CPU time versus problem size is shown in Figure 2. Actual CPU times for entire simulations are shown in Table 2. The scaling complexity of our method is excellent. With optimal complexity scaling being $O\left(N_{eqns}{}^{1.0}\right)$, our coupled method scales as $O\left(N_{eqns}{}^{1.14}\right)$. We note that this is a huge improvement over previous methods, that scaled approximately as $O\left(N_{eqns}{}^{2.0}\right)$. Most of our simulations are run on Grid 2, requiring a total CPU time of about 5.5 hours running on a single core of a 3.2Gz Pentium 4 processor. We note that we can perform entire numerical simulations on Grid 1 in about 37 minutes. This is remarkable since we verified that this grid yields good qualitative accuracy. As a point of reference, previous methods

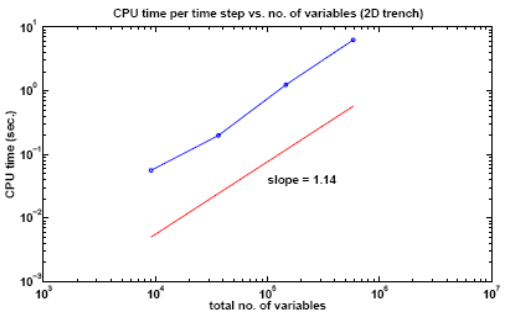

Figure 2. Efficiency of the coupled simulation for 2D trench problem.

using even coarser grids took 7-8 hours. This is a 15-fold increase in computational efficiency that only gets better as the grid is further refined!

Next, we measured the efficiency for the 3D problem. The computational complexity is plotted in Figure 3 and the CPU times for entire simulations are given in Table 3. Note that the last two simulations were not performed in their entirety but instead were solved for a few time steps and the timing results were obviously extrapolated. The complexity scales about the same as for the 2D trench problem, but the overall constant of proportionality is about four times larger for the 3D problem. This can be attributed to the extra couplings of the unknown variables, making the linear solvers less efficient, as is observed by the increased number of iterations required for convergence. It is clear from the total simulation time that highly resolved 3D simulations are out of the question at this point. However, from the results of Section 4.1.3, we don't actually have to use highly resolved grids to make qualitatively accurate predictions. One way we can envision to improve the efficiency is to parallelize the linear solvers. We conclude this section by noting that our semi-implicit algorithm has been stable and robust for all the surface model parameters that we have tested.

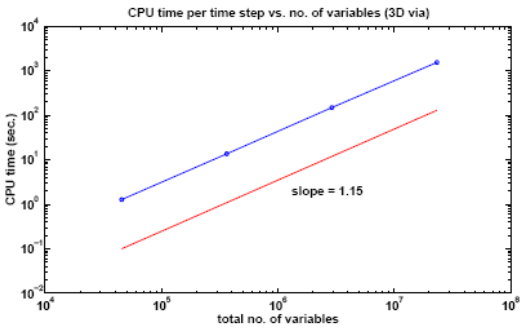

Figure 3. Efficiency of the coupled simulation for 3D via problem.

5. APPLICATION TO 3D VIA INFILL

In this section, we consider the infill of a three-dimensional via consisting of a 2D trench with a tapered cylinder extending down from the trench bottom. The trench width is 180nm with 5:1 aspect ratio, and the cylinder diameter is 90nm with 1:1 aspect ratio. These geometries are often used in computer processor chips to connect one level of interconnects to the next lower level. The surface chemistry mechanism used is described in detail in the references, but may be understood as follows. Two suppressor additives (Cl^- and PEG) and one accelerator additive (SPS) are included in solution in addition to the basic $CuSO_4$ salt plus acid H_2SO_4 cocktail required for electrodeposition.

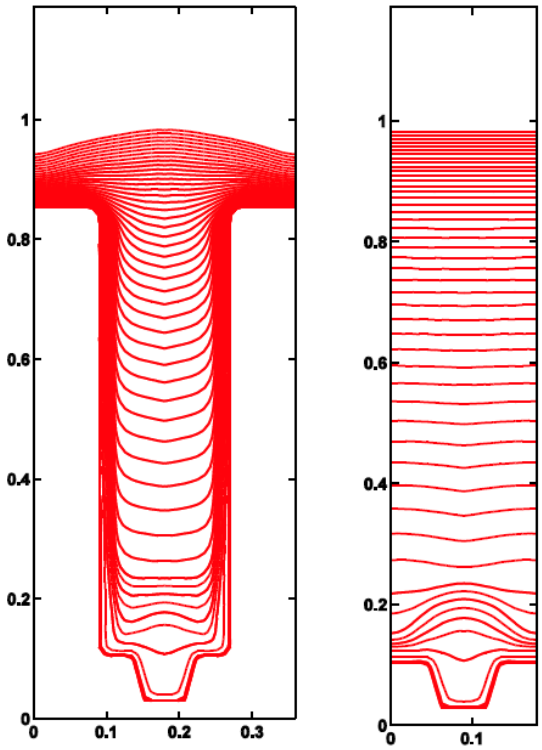

Figure 4. Via infill contours perpendicular (left) and parallel (right) to the trench axis are plotted at 1 sec. intervals for first 10 sec. followed by every 2 sec.

The infill profiles are plotted along the two planes of symmetry. Figure 4 shows infill contours along the plane perpendicular and parallel to the trench and through the cylinder's diameter. The simulations reveal that the small cylinder initially begins to fill conformally. As it fills, the deposition rate near the cylinder axis rapidly increases, leading to the formation of a large hump at the trench bottom. The hump then spreads out (more so along the trench axis) as the trench section starts to fill. The trench then continues to fill from the bottom up, a phenomenon referred to as superfilling, which is caused by the adsorbed layer of accelerator species that promotes copper deposition and preferentially adsorbs at bottom.

REFERENCES:

M. Buoni (2008), Multiscale modeling and simulation of copper electrodeposition, Ph.D. thesis, University of California, Santa Barbara, Mechanical Engineering, available for download at: http://www.cs.ucsb.edu/~buoni/papers/buoni_phd_thesis.pdf (all other references contained therein)