Subhankar and Shamanna paper PhL 1.7.12
updated 9.18.16

This was a real find for me, a 2003 paper that calls Goldstein and others on their arm-waving BS vague discussions of "virtual displacements". I am precisely one of those "students" who does not buy into the discussion.

I. Introduction. In their introduction, these two authors quote from several different books on how the virtual displacement is defined and how it is related to constraints. Not only are the quotes imprecise, some even conflict with each other! These guys quote from 11 rather famous sources listed in the references. I have only 2 of the 13 books mentioned.

These guys are very much on my wavelength! The actually quote various vague statements from all the books, a bit of in your face. I could imagine writing this paper, it is a topic I might choose.

They have this paper stored in the Arxiv, and at Researchgate, and at OpenLib which I was not aware of, this seems to involve peer review and typesetting so probably would not work for me.

Paper has this structure (total 8 pages).

I. Introduction
  opening text
  A. Ambiguity in virtual displacement (where all the quotes are listed)
    text then 1,2,3,4 then more text

II. Virtual Displacements and Forces of Constraint
  A. Constraints and Virtual Displacement
  B. Existence of forces of constraint
  C. Solvability and ideal constraints

III. Examples of Virtual Displacements
  A. Simple pendulum with stationary support
  B. Simple pendulum with moving support
  C. Motion along a fixed inclined plane
  D. Motion along a moving inclined plane

IV Lagrange's Method of Undetermined Multipliers

V. Generalized Coordinates and Lagrange's Equations of Motion

VI. Conclusion, then Ack, then References with no heading

In 2003 the two authors were Physics Profs at different Indian universities.

II. Virtual Displacement and Constraint Forces

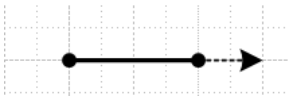
A. They first assume the there are s holonomic constraints $f_i(r_k, t) = 0$, i = 1.2..s. (1). Chain rule then gives their (2). They write then $d\mathbf{r_k} = \mathbf{v_k}dt$ as "an allowed infinitesimal displacement" which is fine by me, and then (3) is obvious for such an allowed displacement. They propose to have a "virtual"

displacement be any difference of such allowed ones, as long as you pick two different allowed ones. So they write $\delta\mathbf{r}_k = d\mathbf{r}_k - d\mathbf{r}'_k$. At least for me this is consistent with motion along a plane and not into the plane which is the constraint.

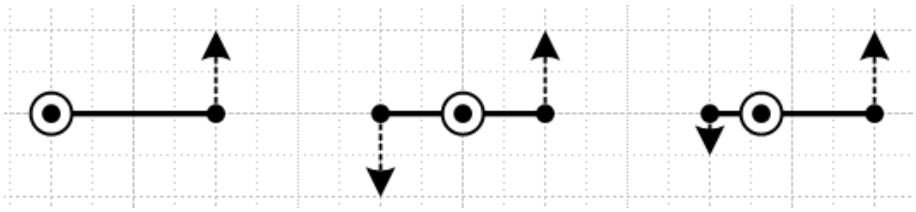
Now what happens when you install such a virtual displacement into the chain rule (4)? You get (7) where the dt term cancels because you took a difference!

Now notice that when they talk about either allowed or virtual, $d\mathbf{r}_k$ or $\delta\mathbf{r}_k$, they are really talking about a vector of items, such as $d\mathbf{r}_1$, $d\mathbf{r}_2$.....$d\mathbf{r}_N$ where each element of the vector is a 3-vector. So perhaps I would write $\{ d\mathbf{r}_k \}$.

Examples: Here are some PL examples. Consider first

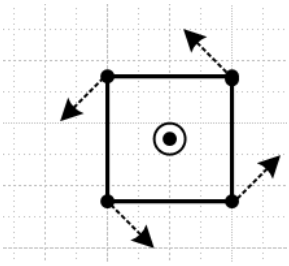


The set here is a null arrow for the left particle and a right arrow for the right particle. This $\{d\mathbf{r}_1, d\mathbf{r}_2\}$ is neither an allowed nor a virtual system displacement because it violates the constraint. On the other hand, these three system displacements would be allowed:



Each one corresponds to a rotation of the rigid object about some origin.

Here is an example of an allowed system displacement for a 4-particle rigid object



Comment: consider the upper right particle. It's two sticks have tensions which, when added, pull the particle toward the rotational origin. This total constraint force is perp to the displacement.

Notes added 9.18.16 : This Section A is completely clear, I have derived all equations. The key idea is to understand the notion of an "allowed set" of $\{d\mathbf{r}_k\}$ or $\{\mathbf{v}_k\}$: the "system of interest" could have these sets without violating any of the constraint equations. Then $\{\delta\mathbf{r}_k\}$ is any set of differences of allowed sets, and this is then their definition of a set of "virtual" displacements: $\delta\mathbf{r}_k \equiv d\mathbf{r}_k - (d\mathbf{r}_k)'$. In exactly the same way, you can define $\boldsymbol{\nu}_k \equiv \mathbf{v}_k - (\mathbf{v}_k)'$ where both $\{\mathbf{v}_k\}$ and $\{\mathbf{v}'_k\}$ are allowed velocity sets. It is quite clear from my examples above that certain $\{\mathbf{v}_k\}$ would NOT be allowed (in fact most choices). Their purpose

in using this definition is merely to cancel out the $\partial_t f_i$ terms on the right side of (2) or (4) so you end up with the simple result (7) for the $\{\delta \mathbf{r}_k\}$ set, and the same equation would also be true fore the $\{\boldsymbol{\nu}_k\}$ set but they don't write this equation.

B. They write F = ma for the k^{th} particle. Differentiation ∂_t on (2) gives (9). This last is regarded as a constraint on the possible acceleration sets $\{\mathbf{a}_k\}$ for the system. Note that F_k refers to an "external force" and not a constraint force. So then (10) includes external forces F_i and constraint forces R_i, fine. They count then 3N equations of motion, but 6N unknowns if you include all the constraint forces. But the s constraints get you down to 6N unknowns and 3N+s equations.

[Added] The 6N comes from the $\mathbf{r}_k(t)$ and the $\mathbf{R}_k(t)$, which is 3N + 3N functions of time in general. If you know these functions, you can then differentiate to get $\mathbf{v}_k(t)$ and $\mathbf{a}_k(t)$. But Newton only provides 3N equations as in (10) and you have s constraint equations as in (1), so you have 6N - (3N+s) = 3N - s degrees of freedom in the solution. So to have an exact solution, you need to find 3N - s extra scalar equations. Where will these come from?

C. How do you solve the problem? They talk in terms of constraint surface (marble on table) and comment that both dr_k and δr_k are tangential to the allowed motion surface, while the R_k are perp, and this then gives

$$\sum_{k=1}^{N} \mathbf{R}_k \cdot d\mathbf{r}_k = 0, \qquad \sum_{k=1}^{N} \mathbf{R}_k \cdot \delta\mathbf{r}_k = 0$$

based on the fact of perpendicularity.

Question: Is the above true individually for each particle: $\mathbf{R}_k \bullet d\mathbf{r}_k = 0$? The Indians don't comment on this, but the reader would assume that it was true I think. Two marbles rolling on a surface? Their examples only deal with a single particle so this subject does not arise.

Temporary conclusion: $\mathbf{R}_k \bullet \delta\mathbf{r}_k = 0$ for each particle even if the platform is moving. Then it follows that the sum $\Sigma_k \mathbf{R}_k \bullet \delta\mathbf{r}_k = 0$ as well.

They first talk about the constraint surface not moving. But what happens if the table is moving. Then certainly allowed motions are no longer tangential to the surface at some particular time t because the constraint surface has moved "up" say. But if you take their difference idea, the surface motion contribution cancels. So for moving constraints in time we get (11). Now suppose you take the right equation above (or in (11) and replace s of the $d\mathbf{r}_i$ by linear combinations of the other $d\mathbf{r}_i$ (since s are dependent on the others since we have our s holonomic constraints). Assume that these are the higher numbered items. Perhaps you then do some shuffling of the 3N-s variables into some other ones, The claim is that when you are done shuffling, you get something of the form (12) where there are now only n = 3N-s terms in the sum and the $\delta\tilde{x}_j$ are now all independent. When you do the elimination and shuffle, you will end up with some $\tilde{R}_j$ coefficients in (12). I don't think the authors want that dot in (12), we just have this set of n = 3N-s new coordinates $\{\tilde{x}_j\}$ and they no longer form 3-vectors. How many equations

does (12) represent? Well, the equations are really $\widetilde{R}_j = 0$ for j = 1 to n, so these represent 3N-s constraints. So we now have 6N variables and 3N+s equations of motion and constraint equations, and now we have 3N-s new equations $\widetilde{R}_j = 0$ so we then have 6N equations in 6N unknowns, and that is what we want.

So the right equation above will be their "principle of zero virtual work". Problems which fit into this picture have "ideal constraints".

Comments added 9.18.16:

For *certain* constrained problems, like balls rolling on a curved surface, it is obvious that all allowed $d\mathbf{r}_k$ are tangential to the surface, but I would not assume this concept for all holonomic constraint problems! Problems of this type they say have "ideal constraints". In this case it is pretty obvious for a *static* surface that $\Sigma_{k=1}^{N} \mathbf{R}_k \bullet d\mathbf{r}_k = 0$ and therefore also $\Sigma_{k=1}^{N} \mathbf{R}_k \bullet \delta\mathbf{r}_k = 0$. For such problems, the forces of constraint $\mathbf{R}_k$ therefore "do no work", and this is sort of a vague connection to the literature notion that there is "zero virtual work". Since the 3N numbers in $\{d\mathbf{r}_k\}$ or $\{\delta\mathbf{r}_k\}$ are not linearly independent (due to the constraints), you cannot conclude that $(\mathbf{R}_i)_k = 0$, and of course we know that not all constraint forces are NOT zero for a general situation.

The equation $\Sigma_{k=1}^{N} \mathbf{R}_k \bullet \delta\mathbf{r}_k = 0$ has 3N "component" terms on the left side. But in the set $\{ \delta\mathbf{r}_k \}$ there are only 3N-s independent components due to the s constraint equations (4). Rewrite the dependent equations perhaps as

$\Sigma_{j=1}^{3N} R_j (\delta r_j) = 0$ equation has 3N terms

where the (δr_j) are the original 3N components and the R_j are the force components, all now written out in a single linear list.

Now suppose you were to first replace the LAST s of the (δr_j) by linear combinations of the others using the equations (4). You would then have

$\Sigma_{j=1}^{3N-s} R'_j (\delta r_j) = 0$ equation has 3N-s terms which include just the FIRST (δr_j)

There is a prime on R'_j because some or all of the R_j got extra contributions when the linear combinations were inserted and terms collected. Now make a second "transformation" where instead of using the first (δr_j), we use some arbitrary subset of the original 3N (δr_j) as the independents. Call these $\{\delta\xi_j\}$ just to have a name, and then we get

$\Sigma_{j=1}^{3N-s} A_j (\delta\xi_j) = 0$

where now the A_j are whatever they come out being. Now do yet another transformation and define a new set of 3N-s independent variables (call them $d\eta_j$) to be some linear combinations of the $d\xi_j$. Then we get

$\Sigma_{j=1}^{3N-s} B_j (\delta\eta_j) = 0$

or

$$\Sigma_{j=1}^{3N-s} \tilde{R}_j (\delta\tilde{x}_j) = 0 \qquad \tilde{R}_j = B_j \qquad \delta\tilde{x}_j = \delta\eta_j$$

where now the B_j are "whatever they come out being" when you insert the linear combinations and collect terms. This then is the meaning of equation (12). In passing we might refer to the $\delta\tilde{x}_j$ as "generalized coordinates" and the $\tilde{R}_j$ as "generalized forces", but let's hold on that for a while. Now since all the $\delta\tilde{x}_j$ are linearly independent, we have these 3N-s new scalar equations:

$$\tilde{R}_j = 0 \qquad j = 1,2...3N-s$$

and these are the "missing" 3N-s equations that we need to solve the problem, mentioned at the end of B.

Now suppose the surface on which the marbles are rolling is itself "moving". Specifically, assume that for some one of the point particles the surface has some non-zero velocity $\mathbf{w}_k$. I think that point on the surface could be accelerating, and the surface could be changing shape as well, but there is some $\mathbf{w}_k$ no matter what is going on with the surface. Due to this $\mathbf{w}_k$ we no longer have $d\mathbf{r}_k$ = tangent to surface. As a simple example, suppose the surface is just moving downward without acceleration and without changing shape. Then the marble has to move ON the surface, and $d\mathbf{r}_k$ will have a component tangential to the surface and another component $\mathbf{w}_k dt$ due to the downward surface motion. The key point is this: if we were to start with some OTHER set $\{d\mathbf{r}'_k\}$ of allowed displacements at the same time t and at the same locations $\mathbf{r}_k$ on the surface at time t, the downward component of $d\mathbf{r}'_k$ would be the same as that of $d\mathbf{r}_k$. Then when you construct $\delta\mathbf{r}_k \equiv d\mathbf{r}_k - d\mathbf{r}'_k$, the result is tangential to the surface because the vertical components cancel out in the subtraction. You are then back to the situation of the non-moving surface in terms of the $\delta\mathbf{r}_k$, and this is shown in the right equation of (11). The analysis then goes forward exactly as described above and you have a "solvable problem" for these "ideal constraints". We then maintain the idea of "zero virtual work" in the sense of (12) or of (11) right equation.

So these authors start with a clean definition of a "virtual displacement (set)" and end up even for a moving surface with a "zero virtual work" concept.

III. Examples (hurray!!) [N = 1]

A. Pendulum with fixed support. I agree with the picture and opening equations. I guess x is to the right and y is up in their picture. The virtual work rule is simply $\mathbf{T}\bullet\delta\mathbf{r} = 0$ for this 1-particle system, where T is the tension which is perp to the motion and is an example of a constraint force. // Language error noted. This example is entirely kinematic only: what are the allowed velocities and the allowed displacements? The problem is not solved.

B. Now allow the pendulum support to drop in time at speed u. Now we get my expected picture of the various displacements and the virtual ones, being differences, are still perp to the constraint tension. This will be true for vertical or horizontal motion of the support (or any combination).

I went through all the equations on 9.18.16 there is nothing too exciting here. Yes, the displacement d**r** is not along the velocity **v**. Yes the *virtual* displacement δ**r** is collinear with the velocity **v**. And yes we have that **T** • δ**r** = 0 so there is no "virtual work" done by the string on the mass.

C and D. Fixed and then moving inclined plane. He does this in polar coordinates so then the plane constraint is a fix on angle as particle goes down the plane. Again they draw the various allowed and virtual displacements. They then let the plane move to the left at u and redraw the arrows. Whether or not the plane moves, the zero virtual work rule looks the same, in this case it is **N**•δ**r** = 0 where N is the normal force.

So this C,D example is essentially the same as the A,B example. In the AB case string tension is perp to virtual displacements whether the support is moving or not (in any direction, with any speed or acceleration etc etc). In the CD case the normal force is perp to the virtual displacements no matter now the plane is moving.

In this example they don't talk about vertical motion of the plane. I think things work out as long as the vertical acceleration keeps the ball on the plane. The same comment applies to moving the plane to the right too fast. These issues are swept under the rug. You would have a similar problem with the pendulum if the support were to descend faster than the ball drops to keep the string taut .

IV. Those Lagrange Multipliers

Although I have written an entire paper about the Method of Lagrange Multipliers, I am hard pressed to find a connection between that paper and the discussion here. The main issue is that I cannot figure out for what "function" one is trying to find a stationary point. In my paper that function was f but below I will refer to it instead as F since the Indian constraints are called f_i.

Notation

1. So in my paper (N→M, f→F) I have **r** = (x_1, x_2... x_M) being a set of M non-independent coordinates. In the Indian paper we have (**r_1**, **r_2** .... **r_N** , t) being 3N+1 non-independent coordinates. But I think when you are talking about δ type differentials, you can ignore t, so let's now ignore t for the time being. So let's write

r = (**r_1**, **r_2** .... **r_N**) = (r_{1x}, r_{1y}, r_{1z}, r_{2x}, r_{2y}, r_{2z} , ... r_{Nx}, r_{Ny}, r_{Nz}) with M = 3N items

= (r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23} , ... r_{N1}, r_{N2}, r_{N3})

= (x_1, x_2, x_3, x_4, x_4, x_6 ,..... x_{3N}) M = 3N

so that the fancy vector **r** consists of N 3-vectors so the full list has M = 3N components. I will refer to these components as x_i and then we have a clear meaning for $\partial_i = \partial/\partial x_i$ for i = 1,2...3N.

Now suppose you are given one of these M objects x_n . You could say

$x_n = r_{kj}$ where k = 1 + Int(n/3) j = Rem(n/3)

Then we can write

$\partial_n = \partial/\partial x_n = \partial/\partial x_{kj} = "\partial_{kj}"$ again with k and j determined from n as above

Attempt Comparison between Indian paper and my paper

2. In my paper we have a(**r**) = 0, b(**r**) = 0, ... q(**r**) = 0 being S-1 constraints. In the Indian paper we have $f_i = 0$ being the s constraints, so we are pretty close on this idea and s = S-1.

3. In my paper there is a function F we want to extremize so dF = 0, and this leads to

$$\partial_m F(\mathbf{r}) + \lambda_1 \partial_m a(\mathbf{r}) + \lambda_2 \partial_m b(\mathbf{r}) + ... + \lambda_{S-1} \partial_m q(\mathbf{r}) = 0 \qquad m = 1,2...M . \quad (4.1.2)$$

In the Indian paper this would be written

$$\partial_n F(\mathbf{r}) + \lambda_1 \partial_n f_1(\mathbf{r}) + \lambda_2 \partial_n f_2(\mathbf{r}) + ... + \lambda_s \partial_n f_s(\mathbf{r}) = 0 \qquad n = 1,2...3N . \quad (4.1.2)$$

or

$$\partial_{kj} F(\mathbf{r}) + \lambda_1 \partial_{kj} f_1(\mathbf{r}) + \lambda_2 \partial_{kj} f_2(\mathbf{r}) + ... + \lambda_s \partial_{kj} f_s(\mathbf{r}) = 0 \qquad k = 1,2...N,\ j = 1..3$$

or

$$\partial_{kj} F(\mathbf{r}) + \Sigma_{i=1}^{s} \lambda_i \partial_{kj} f_i(\mathbf{r}) = 0 \qquad k = 1,2...N,\ j = 1..3$$

The second term here matches the second term in the Indian paper equation (17). We could think of this as a vector equation like so

$$\nabla_k F(\mathbf{r}) + \Sigma_{i=1}^{s} \lambda_i \nabla_k f_i(\mathbf{r}) = 0 \qquad k = 1,2...N$$

where now $\nabla_k = (\partial_{k1}, \partial_{k2}, \partial_{k3})$.

4. Now what does my paper have to say about "differentials"? I do of course have this,

$0 = dF = \nabla F(\mathbf{r}) \bullet d\mathbf{r} = \Sigma_{m=1}^{M} (\partial_m F)\, dx_k$ // seeking a stationary point
$0 = da = \nabla a(\mathbf{r}) \bullet d\mathbf{r} = \Sigma_{m=1}^{M} (\partial_m a)\, dx_k$ // trivially true for each constraint
$0 = db = \nabla b(\mathbf{r}) \bullet d\mathbf{r} = \Sigma_{m=1}^{M} (\partial_m b)\, dx_k$
...
$0 = dq = \nabla q(\mathbf{r}) \bullet d\mathbf{r} = \Sigma_{m=1}^{M} (\partial_m q)\, dx_k$

In the Indian paper notation this could be written

$0 = dF = \nabla F(\mathbf{r}) \bullet d\mathbf{r} = \Sigma_{n=1}^{3N} (\partial_n F)\, dx_n$
$0 = df_j = \nabla f_j(\mathbf{r}) \bullet d\mathbf{r} = \Sigma_{n=1}^{3N} (\partial_n f_j)\, dx_n$ j = 1,2...s one equation for each constraint

Now rewrite the above where • has a new meaning. In the above lines • is dot product of two huge vectors, but now we shall have it refer to a regular 3-vector dot product. Then we can write the above as

$$0 = dF = \Sigma_{k=1}^{N} (\nabla_k F) \bullet d\mathbf{r}_k$$

$0 = df_j = \Sigma_{k=1}^{N} (\nabla_k f_j) \bullet d\mathbf{r}_k$ $\quad j = 1,2...s$ $\quad$ one equation for each constraint

$0 = \delta F = \Sigma_{k=1}^{N} (\nabla_k F) \bullet \delta\mathbf{r}_k$ (*)

$0 = \delta f_j = \Sigma_{k=1}^{N} (\nabla_k f_j) \bullet \delta\mathbf{r}_k$ $\quad j = 1,2...s$ $\quad$ one equation for each constraint (**)

where now we have defined,

$\mathbf{r}_k = (r_{k1}, r_{k2}, r_{k3})$
$d\mathbf{r}_k = (dr_{k1}, dr_{k2}, dr_{k3})$
$\delta\mathbf{r}_k = (\delta r_{k1}, \delta r_{k2}, \delta r_{k3})$

Equation (**) above is what the Indians mean by their equation (16) which they write in this more ambiguous notation

$0 = \delta f_j = \Sigma_{k=1}^{N} (\partial f_j/\partial \mathbf{r}_k)\, \delta\mathbf{r}_k$

5. So what is F for the Indian paper? I want to identify my (*) above with their equation (13). So compare

$0 = \delta F = \Sigma_{k=1}^{N}\ \mathbf{R}_k \bullet \delta\mathbf{r}_k$ $\quad$ // their (13)

$0 = \delta F = \Sigma_{k=1}^{N} (\nabla_k F) \bullet \delta\mathbf{r}_k$ $\quad$ // my (*)

To make this connection, we must identify

$\nabla_k F = \mathbf{R}_k$.

So we don't really know F, we only know its gradient for the k^{th} coordinate. But maybe I could construct a possible F. Maybe we can then claim that

$F = \Sigma_{k=1}^{N} \int \mathbf{R}_k \bullet d\mathbf{r}_k$

This tells me that the function F we are trying to make stationary is in fact the sum of all the work done by the constraint forces for a set of displacements $d\mathbf{r}_k$. In a simpler case we would say

$\nabla F = \mathbf{R}$ $\quad \Rightarrow \quad$ $F(\mathbf{r}) - F(\mathbf{r}_0) = \int_{\mathbf{r}_0}^{\mathbf{r}} \mathbf{R} \bullet d\mathbf{r} = \int_{\mathbf{r}_0}^{\mathbf{r}} \nabla F \bullet d\mathbf{r}$ = line integral

For a very short "paths" we could write

$F(\{\mathbf{r}_{k0} + d\mathbf{r}_k\}) - F(\{\mathbf{r}_{k0}\}) = \Sigma_{k=1}^{N} \mathbf{R}_k \bullet d\mathbf{r}_k$

$F(\{\mathbf{r}_{k0} + d'\mathbf{r}_k\}) - F(\{\mathbf{r}_{k0}\}) = \Sigma_{k=1}^{N} \mathbf{R}_k \bullet d'\mathbf{r}_k$

Then subtracting we get

$$F(\{\mathbf{r}_{k0} + d\mathbf{r}_k\}) - F(\{\mathbf{r}_{k0} + d'\mathbf{r}_k\}) = \Sigma_{k=1}^{N} \mathbf{R}_k \bullet \delta\mathbf{r}_k$$

or

$$F(\mathbf{r}_{k0}; \delta\mathbf{r}_k) = \Sigma_{k=1}^{N} \mathbf{R}_k \bullet \delta\mathbf{r}_k = \text{the total virtual work done by the constraint forces } \mathbf{R}_k = 0$$

Let's try this again for finite paths. We have

$$F(\mathbf{r}_k) - F(\mathbf{r}_{k0}) = \Sigma_{k=1}^{N} \int_{\mathbf{r}_{k0}}^{\mathbf{r}_k} \mathbf{R}_k \bullet d\mathbf{r}_k \qquad (***)$$

$$F(\mathbf{r'}_k) - F(\mathbf{r}_{k0}) = \Sigma_{k=1}^{N} \int_{\mathbf{r}_{k0}}^{\mathbf{r'}_k} \mathbf{R}_k \bullet d\mathbf{r}_k$$

So, if I were to define the full function F by (***), that would lead to $\nabla_k F = \mathbf{R}_k$ which in turn would lead to $0 = \delta F = \Sigma_{k=1}^{N} (\nabla_k F) \bullet \delta\mathbf{r}_k$ and this is the $\delta F = 0$ stationary situation of my Lagrange paper. Notice that in the examples with moving platforms, the work done is NOT zero, but probably still $\delta F = 0$.

So I guess I am happy now to have made this Connection between my paper and the Indian paper, though there are still some loose ends.

Idea. I am looking at the equation set (2.6) in my paper. Suppose I were to sum the first M equations (something I had no reason to do). I would get

$$[F_1 + F_2 + .. + F_M] = +\lambda_1 [a_1 + a_2 + ... + a_M] + \lambda_2 [b_1 + b_2 + ... + b_M] + ... + \lambda_s [q_1 + q_2 + ... + q_M]$$

or

$$\Sigma_{i=1}^{M} [F_i - \lambda_1 a_i - \lambda_2 b_i - ... - \lambda_s q_i] = 0 \qquad // \Sigma_{i=1}^{M} H_i = 0$$

The sum here is zero because each [...] bracket is zero because each of my equations is 0. Recall that you have to FIND a set of λ_j values which make the individual equations be true for some **r**.

Comment: H is a function of M+s *independent* variables, we *seek* a point where dH = 0, and this is why we require that $H_i = \partial_i H = 0$ for each i. I don't state this very clearly in my paper.

Suppose I rename things so that: $a = f^{(1)}$, $b = f^{(2)}$, ...$q = f^{(s)}$. Then I could write the above as

$$\Sigma_i [F_i - \lambda_1 \partial_i f^{(1)} - \lambda_2 \partial_i f^{(2)} - ... - \lambda_s \partial_i f^{(s)}] = 0$$

or

$$\Sigma_{i=1}^{M} [F_i - \Sigma_{j=1}^{s} \lambda_j \partial_i f^{(j)}] = 0 \qquad // \Sigma_{i=1}^{M} H_i = 0$$

Another option would be to multiply each of my original equations by some arbitrary function Q_i and THEN add them, and the result would be

$$\Sigma_i [F_i - \lambda_1 a_i - \lambda_2 b_i - ... - \lambda_s q_i] Q_i = 0$$

or

$$\Sigma_{i=1}^{M}\,[\;F_i - \Sigma_{j=1}^{s}\lambda_j\,\partial_i f^{(j)}]Q_i\;=0 \qquad\qquad (*) \qquad\qquad //\;\Sigma_{i=1}^{M}\,H_iQ_i = 0$$

The coordinates are enumerated by index i. I could regard the last s coordinates as being the dependent ones, and the first M-s coordinates as being the independent ones. Is it possible to arrange to have

$$[F_i - \Sigma_{j=1}^{s}\lambda_j\,\partial_i f^{(j)}]\;=0 \quad \text{for i = the last s values of i ending in M ?} \qquad //\;H_i = 0\;?$$

This would be s equations (linear in the λ_i) enumerated by index i, and there are of course s values λ_j. So yes, you could select the λ_j to be just those which cause all these last s brackets to be zero. Call this solution set $\boldsymbol{\lambda}$ with elements λ_j.

This is NOT something I do in my paper! But suppose we do this here. Then we get:

$$\Sigma_{i=1}^{M-s}\,[\;F_i - \Sigma_{j=1}^{s}\lambda_j\,\partial_i f^{(j)}]\;Q_i\;=0 \qquad\qquad //\;\Sigma_{i=1}^{M-s}\,H_i(\boldsymbol{\lambda})\;Q_i\;=0$$

because the last s brackets of (*) are all zero.

Comment: This set $\boldsymbol{\lambda}$ is surely *not* a set which causes dH = 0 and for this λ we do *not* expect $H_i = 0$ for all the i. So each bracket $[\;F_i - \Sigma_{j=1}^{s}\lambda_j\,\partial_i f^{(j)}]$ in the above equation is not a priori 0. So going backwards now, we have no justification for our derivation of (*) ! **So this current line is illogical.** I leave the text below nevertheless.

Suppose now that the set of $\{Q_i\}$ objects is linearly independent, none can be written as a sum of the others. This means that the Q_i can have arbitrary values. In this situation, the only way the above sum can vanish is if the brackets all vanish, that is to say,

$$F_i - \Sigma_{j=1}^{s}\lambda_j\,\partial_i f^{(j)}\;=0 \qquad\qquad i = 1,2....M\text{-}s$$

Using this very particular set of λ_j constants, we end up with this fact:

$$F_i\;=\;\Sigma_{j=1}^{s}\lambda_j\,\partial_i f^{(j)} \qquad\qquad i = 1,2....M\text{-}s$$

This is very different from what I do in my paper. In that paper, F is a given function, and thus F_i are all predetermined. You cannot just set F_i as in the last equation above! You have some given F(**r**) function.

In particular, you see above that the λ_j are determined entirely by just the last s values of F_i and $\partial_i f^{(j)}$ for just the last s partials of all the $f^{(j)}$. This $\boldsymbol{\lambda}$ set *knows nothing* about F_1 and F_2 for example. In my paper, F_1 and F_2 could be anything at all. So consider the equation resulting from above

$$F_1\;=\;\Sigma_{j=1}^{s}\lambda_j\,\partial_1 f^{(j)}\;.$$

This equation gives some F_1, but that is not likely to be the same F_1 that comes from your arbitrarily selected function F(**r**).

So :

• The Indian application of Lagrange multipliers is not compatible with my paper's idea of having all the F_i being determined ahead of time since you have specified F(**r**).

• The Indian solution **λ** is not associated with any stationary point of the function F(**r**).

Nevertheless, if you select the special set **λ**, the above WILL be true regardless of the F(**r**) you start with.

Now OK, suppose I go back and replace Q_i by dx_i so I then have

$$\Sigma_{i=1}^{M} [\ F_i - \Sigma_{j=1}^{s} \lambda_j \partial_i f^{(j)}] dx_i = 0$$

where $\{ dx_i \}$ = d**r** = some variation of the composite coordinate **r** which is allowed by the constraints. It is just some differential displacement, but due to the constraints, all M of the dx_i are not linearly independent. But now if I select the special set $\{\lambda_j\}$ just outlined above, I then get the last s brackets vanishing and am left with

$$\Sigma_{i=1}^{M-s} [\ F_i - \Sigma_{j=1}^{s} \lambda_j \partial_i f^{(j)}]\ dx_i = 0$$

Now all the dx_i ARE independent, and I then conclude that

$$F_i = \Sigma_{j=1}^{s} \lambda_j \partial_i f^{(j)} \qquad i = 1,2....M\text{-}s$$

FOR THIS particular $\{\lambda_j\}$ set. The set is computed exactly to make this M-s equations be true!

I agree with (13) through (16). Notice that (16) has no time derivative. In SS's version of things we have a reason for this -- $\delta \mathbf{r}_k$ is a difference of allowed displacements, this is just (7) again with definition of δf_i. I agree with (17) as a sum of equations each of which is 0. You choose the λ_i so that each coefficient in (17) vanishes and this gives (19). Then the entire problem becomes 3N + s equations in 3N+s unknowns, where the λ_i are s of those unknowns.

Another shot at this Section IV.

Let's now (for the moment) forget the PL paper and just look at what is happening here. I will do this one equation at a time:

equation A: this is just Newton's law for particle k. F_k would be $F_k^{(a)}$ in Goldstein, the externally applied force. And R_k is the total internal force due to all the other particles acting on particle k, which Goldstein calls f_k.

(13) $\Sigma_k \mathbf{R}_k \bullet \delta\mathbf{r}_k = 0$. I think each term in this sum is zero, and the idea is that the constraint force acting on particle k is perp to the virtual displacement of that particle, and this is valid even if the platform is moving. So here each term is 0, and hence the sum is also 0.

(14) Here we just replace $\mathbf{R}_k$ by $m\mathbf{a}_k - \mathbf{F}_k$ which again is just Newton's law. Each term in (14) is separately 0. (14) says this: $\Sigma_k [m\mathbf{a}_k - \mathbf{F}_k] \bullet \delta\mathbf{r}_k = 0$.

(15) Is just a statement of the s constraint equations, $f_i = 0$ for i = 1..s.

(16) This is the statement that $df_i = 0$ for each constraint. It says $\Sigma_k (\nabla^{(k)} f_i) \bullet d\mathbf{r}_k = 0$ where $\nabla^{(k)}$ is gradient wrt the 3 coordinates of particle k which has position $\mathbf{r}_k$. For *this* equation, we do *not* claim that the individual terms in the sum are 0!! In general they are not individually zero. If the $d\mathbf{r}_k$ displacements were linearly independent, we could conclude that the terms are each 0, but the $d\mathbf{r}_k$ for an "allowed" displacement does not form an independent set of course due to the constraints.

(17) which I will now gradually derive. First construct this scalar object:

$$Q_k \equiv \{ \Sigma_{i=1}^{s} \lambda_i (\nabla^{(k)} f_i) \bullet \delta\mathbf{r}_k \}$$

There is no reason to think this is 0, it is just something we construct. The λ_i at this point are arbitrary constants, so no way that all the Q_k are 0 for arbitrary λ_i. Notice now that:

$$\Sigma_k Q_k = \Sigma_k \{ \Sigma_{i=1}^{s} \lambda_i (\nabla^{(k)} f_i) \bullet \delta\mathbf{r}_k \} = \Sigma_{i=1}^{s} \lambda_i [\Sigma_k (\nabla^{(k)} f_i) \bullet \delta\mathbf{r}_k]$$

But the square bracket vanishes by (16). It does not vanish term by term, but the sum vanishes. Thus:

$$\Sigma_k Q_k = \Sigma_{i=1}^{s} \lambda_i [0] = 0$$

Again, only the sum $\Sigma_k Q_k = 0$, regardless of the choice of the λ_i , whereas $Q_k \neq 0$.

Now combine $\Sigma_k Q_k = 0$ with (13) which says $\Sigma_k \mathbf{R}_k \bullet \delta\mathbf{r}_k = 0$. As a reminder, $\Sigma_k Q_k = 0$ is not true term by term, whereas $\Sigma_k \mathbf{R}_k \bullet \delta\mathbf{r}_k = 0$ is true term by term. Subtracting we get

$$0 = 0 - 0 = \Sigma_k \mathbf{R}_k \bullet \delta\mathbf{r}_k - \Sigma_k Q_k = \Sigma_k [\mathbf{R}_k \bullet \delta\mathbf{r}_k - Q_k]$$

$$= \Sigma_k [\mathbf{R}_k \bullet \delta\mathbf{r}_k - \Sigma_{i=1}^{s} \lambda_i (\nabla^{(k)} f_i) \bullet \delta\mathbf{r}_k]$$

$$= \Sigma_{k=1}^{N} [\mathbf{R}_k - \Sigma_{i=1}^{s} \lambda_i (\nabla^{(k)} f_i)] \bullet \delta\mathbf{r}_k \qquad (17)$$

I have now derived equation (17). It is NOT true term by term for the simple reason that $\Sigma_k Q_k = 0$ is not true term by term. This is further supported by the idea that the allowed $\delta\mathbf{r}_k$ set is not independent.

Derive (18) : Now rewrite (17) using fancier notation so you can see all 3N individual terms:

$$0 = \Sigma_{k=1}^{N} \Sigma_{j=1}^{3} [(\mathbf{R}_k)_j - \Sigma_{i=1}^{s} \lambda_i (\partial_{kj} f_i)] (\delta \mathbf{r}_k)_j \qquad (17)$$

Using notation of above, rewrite this again using generic coordinates called dx_i

$$\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2 \ldots \mathbf{r}_N) = (r_{1x}, r_{1y}, r_{1z}, r_{2x}, r_{2y}, r_{2z}, \ldots r_{Nx}, r_{Ny}, r_{Nz}) \text{ with M = 3N items}$$
$$= (r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, \ldots r_{N1}, r_{N2}, r_{N3})$$
$$= (x_1, x_2, x_3, x_4, x_4, x_6, \ldots x_{3N}) \qquad M = 3N$$

to get

$$0 = \Sigma_{n=1}^{3N} [(\mathbf{R}_k)_j - \Sigma_{i=1}^{s} \lambda_i (\partial_{kj} f_i)] (\delta \mathbf{r}_k)_j \qquad (17)$$

$$0 = \Sigma_{n=1}^{3N} [R_n - \Sigma_{i=1}^{s} \lambda_i (\partial_n f_i)] \delta x_n \qquad (18)$$

This is what the Indians mean by their equation (18).

Comments: In the above (18) there are 3N terms added together. The set $\{dx_n\}$ is not linearly independent due to the constraints, just as the set $\{d\mathbf{r}_k\}$ was not independent. The terms in the sum are generally not 0. The λ_i are arbitrary at the moment, so terms cannot be 0.

Derive (19) :Now to make it simple, suppose we select the vector λ which makes the *last* s brackets vanish. This is just a set of s linear (in λ_i) equations in s unknowns λ_i, a Cramer's Rule problem. Then since the last s terms of (18) vanish, we rewrite (18) as follows:

$$0 = \Sigma_{n=1}^{3N-s} [R_n - \Sigma_{i=1}^{s} \lambda_i (\partial_n f_i)] \delta x_n \qquad (18)$$

where λ_i are the special solution values. Now there are only 3N-s displacements δx_n so they ARE independent. This then means that all the lower brackets just be 0 ! So we get these equations

$$[R_n - \Sigma_{i=1}^{s} \lambda_i (\partial_n f_i)] = 0 \qquad n = 1, 2, \ldots 3N\text{-s} \qquad 3N\text{-s equations}$$

But now the lower 3N-s brackets AND the upper s brackets are all zero, so combine to get

$$[R_n - \Sigma_{i=1}^{s} \lambda_i (\partial_n f_i)] = 0 \qquad n = 1, 2, \ldots 3N \qquad 3N \text{ equations}$$

We can then express the forces of constraint (in our n component notation) as

$$R_n = \Sigma_{i=1}^{s} \lambda_i (\partial_n f_i) \qquad n = 1, 2, \ldots 3N \qquad (19)$$

and thus is derived equation (19). Restoring the vector notation, we get

$$\mathbf{R}_k = \Sigma_{i=1}^{s} \lambda_i (\nabla^{(k)} f_i) \qquad k = 1, 2, \ldots N \qquad (19)$$

Now we only know that the λ_i exist such that you can write the constraint forces this way. In order to actually compute the λ_i you would need to know the R_n for the highest s of the constraint forces, and we really don't have this information, so this is not really a *solution* to the problem. But we are allowed to restate the problem as in (20) and (21):

Solve these equations: $m_k\mathbf{a}_k = \mathbf{F}_k - \mathbf{R}_k = \mathbf{F}_k - \Sigma_{i=1}^{s} \lambda_i (\nabla^{(k)} f_i)$ $k = 1..N$ (20)

subject to the constraints $f_i = 0$ (21)

the first line is 3N equations, the second is s equations, so there are 3N+s equations. The number of unknowns are the 3N coordinate components for $\mathbf{r}_k(t)$ and the s values of λ_i . So you have 3N+s equations in 3N+s unknowns. If you can solve this big problem, then (19) gives you the $\mathbf{R}_k$.

Conclusion: I think I follow every single detail of Section IV. It involves s constraints, and there is one Lagrange multiplier λ_i for each constraint. In this last section I make no connection to *my* paper, but there certainly are the two common elements just stated! I want to think about the connection a little more.

V. Generalized coordinates.

I agree with (22). They then do Goldstein's derivation of the Euler-Lagrange equations. They just do this for the sake of "completeness".

So I think this was an interesting paper. Their main interest I think was the time aspect, whereas mine was just the understanding of what is meant by a system displacement as $\{d\mathbf{r}_k\}$ of either allowed or virtual kinds.