

Un nuovo metodo del simplesso per il problema di complementarità lineare mista in sistemi multibody con vincoli unilateri

Alessandro Tasora, Elisabetta Manconi, Marco Silvestri
Università degli Studi di Parma, ITALY
E-mail: tasora@ied.unipr.it

Keywords: Complementarità, MLCP, metodo del simplesso, multibody, vincoli unilateri.

SOMMARIO: In problemi di simulazione multibody, le disequazioni corrispondenti ad eventuali vincoli unilateri comportano la necessità di risolvere un problema di complementarità lineare misto. La soluzione di tale problema implica generalmente una complessità di classe *NP-hard*, con gravi ricadute in termini di scarsa efficienza computazionale. In questo lavoro si propone una variante del metodo del simplesso di Dantzig in grado di sfruttare la sparsità delle matrici di massa e delle matrici jacobiane dei sistemi multibody, al fine di poter calcolare le accelerazioni incognite con poche iterazioni di complessità generalmente $O(n)$, in luogo della complessità $O(n^3)$ solitamente richiesta nelle iterazioni del metodo originale.

1. INTRODUZIONE

Nell'ambito dell'analisi dinamica di sistemi multibody, le procedure d'integrazione numerica richiedono la soluzione delle accelerazioni incognite ad istanti discreti di tempo [Baraff, D., 1993]. Nell'ipotesi in cui tutti i vincoli siano bilateri, come accade ad esempio in caso di coppie rotoidali e prismatiche, è risaputo che il calcolo delle accelerazioni incognite implica la soluzione di un sistema lineare: tale procedimento può avvalersi di procedure note e collaudate quali il metodo di Gauss o la decomposizione di Bunch-Kaufman [Tasora, A., 2001]. Non altrettanto semplice è il calcolo delle accelerazioni qualora siano presenti vincoli unilateri: ciò accade ove sia necessaria la simulazione di contatti e collisioni (meccanismi con scappamenti, palmole, arpionismi, etc.).

In particolare, le disequazioni corrispondenti ai vincoli unilateri implicano la soluzione, ripetuta migliaia di volte nel corso dell'integrazione numerica, di un problema di complementarità lineare misto (MLCP). Quest'ultimo, possedendo una complessità di classe *NP-hard*, comporta gravi conseguenze in termini di scarsa efficienza computazionale quando i vincoli unilateri sono numerosi [Anitescu, M., 1997]. Per tale ragione, in generale, gli attuali codici multibody non sono ancora in grado di trattare efficacemente problemi con centinaia o migliaia di vincoli unilateri.

Alla luce delle precedenti considerazioni si impone la necessità di sviluppare metodi efficienti per la soluzione del problema MLCP associato ai sistemi multibody unilateri.

In questo lavoro si propone una variante del metodo del simplesso di Dantzig in grado di sfruttare la sparsità delle matrici di massa e delle matrici jacobiane dei sistemi multibody, al fine di poter calcolare le accelerazioni incognite con iterazioni di complessità al più $O(n)$, in luogo della complessità $O(n^3)$ solitamente richiesta nelle iterazioni del metodo originale. Inoltre, grazie ad una decomposizione

incrementale della matrice delle basi, si ottiene un ulteriore abbattimento del tempo di calcolo anche in presenza di centinaia di vincoli monolateri.

Il metodo proposto comporta che le matrici vengano memorizzate con un particolare metodo orientato alla massima efficienza computazionale ed alla minore occupazione di memoria. È stato ideato un particolare tipo di decomposizione incrementale, che non altera la simmetria della matrice di stato ogni volta che si aggiorna una base, e che in tal modo ne preserva la sparsità.

Il nuovo algoritmo è stato implementato nel software di simulazione multibody CHRONO, con il quale sono state svolte test che ne dimostrano l'efficienza computazionale e la robustezza.

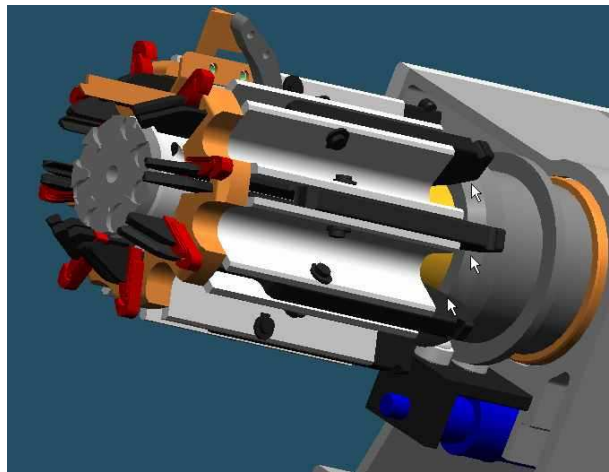


Figura 1: Esempio di sistema multibody con numerosi vincoli monolateri. La camma anulare scorre verso sinistra per chiudere le otto dita della pinza autocentrante.

2. DINAMICA DI SISTEMI MULTIBODY CON VINCOLI UNILATERI

Sia \mathbf{q} il vettore delle coordinate di un sistema di corpi rigidi. Si indichi con \mathbf{C} il vettore delle funzioni olonome-reonomiche dei vincoli bilateri, e con \mathbf{D} il vettore delle funzioni dei vincoli unilateri (tipicamente, contatti fra superfici). I vincoli sono soddisfatti quando:

$$\mathbf{C}(\mathbf{q}, t) = \mathbf{0} \quad \wedge \quad \mathbf{D}(\mathbf{q}) \geq \mathbf{0}$$

Un metodo apparentemente semplice per trattare problemi di contatto in sistemi di corpi rigidi è quello dei metodi di penalità, in altre parole i vincoli unilateri potrebbero essere rappresentati da forze di campo non-lineari che intervengono improvvisamente quando due corpi si compenetrano. Dato che ciò corrisponde alla modellazione di molle molto rigide che esercitano repulsione entro zone limitate, quest'approccio non comporterebbe modifiche sostanziali a codici multibody tradizionali, salvo l'aggiunta delle forze di campo \mathbf{F}_D dovute ai contatti unilateri. In pratica, con riferimento alla

formulazione lagrangiana, detta $[\mathbf{M}]$ la matrice di massa, $[\mathbf{C}_q]$ lo jacobiano di \mathbf{C} rispetto a \mathbf{q} , \mathbf{Q}_M il vettore delle forze giroscopiche, \mathbf{Q}_F il vettore delle forze generalizzate lagrangiane, si ha la classica formulazione [Shabana, A., 1989]:

$$\begin{cases} [\mathbf{M}]\ddot{\mathbf{q}} - [\mathbf{C}_q]\boldsymbol{\lambda} = \mathbf{Q}_M + \mathbf{Q}_F + \mathbf{F}_D(\mathbf{q}) \\ \mathbf{C}(\mathbf{q}, t) = \mathbf{0} \\ \ddot{\mathbf{q}} = \frac{d\dot{\mathbf{q}}}{dt}, \quad \dot{\mathbf{q}} = \frac{d\mathbf{q}}{dt} \end{cases} \quad (2.1)$$

A prescindere dalla stabilizzazione dei vincoli, il precedente sistema DAE differenziale-algebrico è sovente trattato come un sistema ODE convenzionale, con incognite $\ddot{\mathbf{q}}$ (accelerazioni) e $\boldsymbol{\lambda}$ (moltiplicatori lagrangiani, che esprimono le reazioni nei vincoli bilateri), risolvendo un sistema lineare del seguente tipo ad ogni intervallo dell'integratore esplicito:

$$\begin{cases} [\mathbf{M}]\ddot{\mathbf{q}} - [\mathbf{C}_q]\boldsymbol{\lambda} = \mathbf{Q}_M + \mathbf{Q}_F + \mathbf{F}_D(\mathbf{q}) \\ \ddot{\mathbf{C}}(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}, t) = 0 \leftrightarrow [\mathbf{C}_q]\ddot{\mathbf{q}} - \mathbf{Q}_C = 0 \end{cases} \quad (2.2)$$

$$\begin{bmatrix} [\mathbf{M}] & -[\mathbf{C}_q] \\ [\mathbf{C}_q] & [0] \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{Q}_M + \mathbf{Q}_F + \mathbf{F}_D(\mathbf{q}) \\ \mathbf{Q}_C \end{Bmatrix} \quad (2.3)$$

Si osserva tuttavia che tale approccio non garantisce l'esatto soddisfacimento della condizione $\mathbf{D} > \mathbf{0}$ (se non con oscillazioni nell'intorno della stessa) e soprattutto comporta inevitabili difficoltà nella procedura d'integrazione numerica per effetto delle brusche variazioni delle forze di penalità nella (2.3). In pratica si ottiene un sistema differenziale fortemente *stiff* che, per evitare il fenomeno della divergenza, richiederebbe metodi d'integrazione impliciti da applicarsi direttamente al sistema DAE del terzo ordine (2.1), di complessa implementazione oltre che di limitata efficienza ove non si operino scelte oculate per il guadagno delle funzioni di penalità \mathbf{F}_D .

Pertanto un approccio più rigoroso consiste nel considerare i vincoli unilateri come condizioni geometriche: in tal caso, l'esplicita introduzione delle disequazioni $\mathbf{D} \geq \mathbf{0}$ comporta la necessità di descrivere il modello dinamico come un sistema di equazioni differenziali-algebriche-complementari del seguente tipo:

$$\begin{cases} [\mathbf{M}]\ddot{\mathbf{q}} - [\mathbf{C}_q]\boldsymbol{\lambda}_C - [\mathbf{D}_q]\boldsymbol{\lambda}_D = \mathbf{Q}_M + \mathbf{Q}_F \\ \mathbf{C}(\mathbf{q}, t) = \mathbf{0} \\ \mathbf{D}(\mathbf{q}, t) \geq \mathbf{0} \perp \boldsymbol{\lambda}_D \geq \mathbf{0} \\ \ddot{\mathbf{q}} = \frac{d\dot{\mathbf{q}}}{dt}, \quad \dot{\mathbf{q}} = \frac{d\mathbf{q}}{dt} \end{cases} \quad (2.4)$$

Nel sistema precedente si è fatto uso della condizione di complementarità: $\mathbf{D}(\mathbf{q}, t) \geq \mathbf{0} \perp \boldsymbol{\lambda}_D \geq \mathbf{0}$. Questa esprime la condizione per la quale, nei vincoli unilateri, o le reazioni sono nulle $\boldsymbol{\lambda}_D = \mathbf{0}$ e il vincolo è ‘aperto’ $\mathbf{D}(\mathbf{q}, t) > \mathbf{0}$, oppure si ha la condizione di ‘contatto’ con reazione necessariamente positiva e residuo nullo per l’equazione di vincolo: $\mathbf{D}(\mathbf{q}, t) = \mathbf{0} \wedge \boldsymbol{\lambda}_D > \mathbf{0}$. Con notazione alternativa, la condizione di complementarità può essere espressa come: $D_i(\mathbf{q}, t) \geq 0 \wedge \lambda_{Di} \geq 0 \wedge \lambda_{Di} D_i(\mathbf{q}, t) = 0$.

Si osservi ora che la presenza di disequazioni nella (2.4) rende inutilizzabili i metodi di integrazione classici, che presuppongono la continuità della $\mathbf{q}(t)$. Anche l’accorgimento di arrestare e far ripartire il solutore ad ogni discontinuità, per un integrale nel senso di Carathéodory, sarebbe praticabile solo in caso di pochi vincoli unilateri, e sicuramente impraticabile qualora si ipotizzino numerosi contatti ad ogni intervallo d’integrazione, dato che potrebbero facilmente innescarsi serie infinite di apertura-chiusura contatti separati da intervalli di tempo nullo o quasi.

Una soluzione debole della (2.4), che prevede calcoli ad istanti fissi di tempo e che contempla la possibilità di discontinuità nello stato del sistema, è un’inclusione differenziale di Filippov, la quale fra l’altro si riconduce ad un integrale classico nel caso non vi siano discontinuità [Filippov, 1964]. Con tale approccio si è recentemente sperimentato [Anitescu, M., 2002] che conviene trattare il problema in termini d’impulsi piuttosto che di accelerazioni, in tal modo le incognite da risolvere ad ogni istante dell’integrazione sono le variazioni delle velocità. In definitiva la soluzione al problema (2.4), con metodo d’integrazione del primo ordine, è data ad istanti discreti di tempo dal seguente metodo:

$$\left\{ \begin{array}{l} [\mathbf{M}](\dot{\mathbf{q}}^{l+1} - \dot{\mathbf{q}}^l) - [\mathbf{C}_q] \boldsymbol{\gamma}_C - [\mathbf{D}_q] \boldsymbol{\gamma}_D = h \mathbf{Q}_F \\ [\mathbf{C}_q] \dot{\mathbf{q}}^{l+1} + \frac{K}{h} \mathbf{C}(\mathbf{q}, t) + \frac{\partial \mathbf{C}}{\partial t} = \mathbf{0} \\ [\mathbf{D}_q] \dot{\mathbf{q}}^{l+1} + \frac{K}{h} \mathbf{D}(\mathbf{q}) \geq \mathbf{0} \perp \boldsymbol{\gamma}_D \geq \mathbf{0} \\ \mathbf{q}^{l+1} = \mathbf{q}^l + h \dot{\mathbf{q}}^{l+1} \end{array} \right. \quad (2.5)$$

Nella (2.5) si è introdotto l’intervallo d’integrazione h . L’apice l rappresenta l’indice dello step d’integrazione. Si osservi la presenza dei termini $(K/h)\mathbf{C}$ e $(K/h)\mathbf{D}$ per una semplice ma efficace stabilizzazione dei vincoli qualora vi sia violazione \mathbf{C} non nulla e/o violazione \mathbf{D} negativa.

La fase critica del metodo (2.5) è la soluzione delle velocità incognite $\dot{\mathbf{q}}^{l+1}$ (nonché degli impulsi incogniti $\boldsymbol{\gamma}$ scambiati nei vincoli), infatti le prime tre righe della (2.5) corrispondono ad un problema di complementarità lineare misto (MLCP) riassumibile nella seguente forma:

$$\left[\begin{array}{ccc} [\mathbf{M}] & [\mathbf{C}_q] & [\mathbf{D}_q] \\ [\mathbf{C}_q] & [0] & [0] \\ [\mathbf{D}_q] & [0] & [0] \end{array} \right] \left\{ \begin{array}{l} \dot{\mathbf{q}}^{l+1} \\ -\boldsymbol{\gamma}_C \\ -\boldsymbol{\gamma}_D \end{array} \right\} - \left\{ \begin{array}{l} [\mathbf{M}] \dot{\mathbf{q}}^l + h \mathbf{Q}_F \\ -\frac{K}{h} \mathbf{C}(\mathbf{q}, t) - \frac{\partial \mathbf{C}}{\partial t} \\ -\frac{K}{h} \mathbf{D}(\mathbf{q}) \end{array} \right\} = \left\{ \begin{array}{l} \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{\alpha} \end{array} \right\} \quad (2.6)$$

$$\boldsymbol{\alpha} \geq \mathbf{0} \perp \boldsymbol{\gamma}_D \geq \mathbf{0}$$

TEOREMA 1. *Ogni problema di complementarità lineare misto (MLCP), derivante dal problema multibody espresso nella forma (2.6), può essere ricondotto ad un problema di complementarità lineare in forma canonica (LCP).*

Dimostrazione. Si pongano $[\mathbf{M}]\dot{\mathbf{q}}^l + h\mathbf{Q}_F = \mathbf{b}$, $-\frac{K}{h}\mathbf{C}(\mathbf{q}, t) - \frac{\partial \mathbf{C}}{\partial t} = \mathbf{c}$ e $-\frac{K}{h}\mathbf{D}(\mathbf{q}) = \mathbf{d}$. Dalla prima equazione vettoriale del sistema (2.6) si ha $\dot{\mathbf{q}}^{l+1} = [\mathbf{M}]^{-1}([\mathbf{C}_q]^T \boldsymbol{\gamma}_C + [\mathbf{D}_q]^T \boldsymbol{\gamma}_D) + \mathbf{b}$. Sostituendo quest'ultima nella seconda e terza equazione, si ottiene:

$$\begin{aligned} [\mathbf{C}_q][\mathbf{M}]^{-1}[\mathbf{C}_q]^T \boldsymbol{\gamma}_C + [\mathbf{C}_q][\mathbf{M}]^{-1}[\mathbf{D}_q]^T \boldsymbol{\gamma}_D + [\mathbf{C}_q]\mathbf{b} - \mathbf{c} &= \mathbf{0} \\ [\mathbf{D}_q][\mathbf{M}]^{-1}[\mathbf{C}_q]^T \boldsymbol{\gamma}_C + [\mathbf{D}_q][\mathbf{M}]^{-1}[\mathbf{D}_q]^T \boldsymbol{\gamma}_D + [\mathbf{D}_q]\mathbf{b} - \mathbf{d} - \boldsymbol{\alpha} &= \mathbf{0} \end{aligned}$$

Moltiplicando la prima equazione per $([\mathbf{C}_q][\mathbf{M}]^{-1}[\mathbf{C}_q]^T)^{-1}$, la seconda per $([\mathbf{D}_q][\mathbf{M}]^{-1}[\mathbf{C}_q]^T)^{-1}$, e sottraendo la prima dalla seconda, si ottiene la seguente espressione:

$$\begin{aligned} &\left[([\mathbf{D}_q][\mathbf{M}]^{-1}[\mathbf{C}_q]^T)^{-1}[\mathbf{D}_q][\mathbf{M}]^{-1}[\mathbf{D}_q]^T - ([\mathbf{C}_q][\mathbf{M}]^{-1}[\mathbf{C}_q]^T)^{-1}[\mathbf{C}_q][\mathbf{M}]^{-1}[\mathbf{D}_q]^T\right] \boldsymbol{\gamma}_D + \\ &+ ([\mathbf{C}_q][\mathbf{M}]^{-1}[\mathbf{C}_q]^T)^{-1} \mathbf{c} - ([\mathbf{D}_q][\mathbf{M}]^{-1}[\mathbf{C}_q]^T)^{-1} \boldsymbol{\alpha} - ([\mathbf{D}_q][\mathbf{M}]^{-1}[\mathbf{C}_q]^T)^{-1} \mathbf{d} + \\ &+ \left\{([\mathbf{D}_q][\mathbf{M}]^{-1}[\mathbf{C}_q]^T)^{-1}[\mathbf{D}_q] - ([\mathbf{C}_q][\mathbf{M}]^{-1}[\mathbf{C}_q]^T)^{-1}[\mathbf{C}_q]\right\} \mathbf{b} = \mathbf{0} \end{aligned}$$

Moltiplicando la precedente per $[\mathbf{D}_q][\mathbf{M}]^{-1}[\mathbf{C}_q]^T$ ed operando le opportune semplificazioni, si ottiene un sistema lineare del tipo $\boldsymbol{\alpha} = [\mathbf{A}]\boldsymbol{\gamma}_D + \mathbf{B}$, che unitamente alle condizioni di complementarità, porta al seguente problema di complementarità lineare (LCP) in forma canonica:

$$\begin{cases} \boldsymbol{\alpha} = [\mathbf{A}]\boldsymbol{\gamma}_D + \mathbf{B} \\ \boldsymbol{\alpha} \geq \mathbf{0} \perp \boldsymbol{\gamma}_D \geq \mathbf{0} \end{cases} \quad (2.7)$$

con le seguenti posizioni:

$$\begin{aligned} [\mathbf{A}] &= \left[[\mathbf{D}_q][\mathbf{M}]^{-1}[\mathbf{D}_q]^T - [\mathbf{D}_q][\mathbf{M}]^{-1}[\mathbf{C}_q]^T ([\mathbf{C}_q][\mathbf{M}]^{-1}[\mathbf{C}_q]^T)^{-1} [\mathbf{C}_q][\mathbf{M}]^{-1}[\mathbf{D}_q]^T \right] \\ \mathbf{B} &= [\mathbf{D}_q][\mathbf{M}]^{-1}[\mathbf{C}_q]^T ([\mathbf{C}_q][\mathbf{M}]^{-1}[\mathbf{C}_q]^T)^{-1} \{ \mathbf{c} - [\mathbf{C}_q]\mathbf{b} \} + [\mathbf{D}_q]\mathbf{b} - \mathbf{d} = \mathbf{0} \end{aligned} \quad (2.8)$$

COROLLARIO 1.1. *La matrice [A] in (2.8) è simmetrica e semidefinita positiva.*

Dimostrazione. Si verifica considerando che la somma di matrici semidefinite positive genera sempre una matrice semidefinita positiva e ricordando che, per qualsiasi matrice rettangolare [B], la matrice [P]=[B][B]^T è sempre simmetrica semidefinita positiva. Analoga considerazione vale per prodotti del tipo [P]=[B][Q][B]^T se [Q] è simmetrica definita positiva, come lo è [M] nel nostro caso.

COROLLARIO 1.2. *Il problema di complementarità lineare (2.7) è la soluzione duale-primale di un programma quadratico (QP) convesso del seguente tipo:*

$$\begin{aligned}
& \min \left(\frac{1}{2} \dot{\mathbf{q}}^{l+1T} [\mathbf{M}] \dot{\mathbf{q}}^{l+1} + \dot{\mathbf{q}}^{l+1T} \{ -[\mathbf{M}] \dot{\mathbf{q}}^l - h \mathbf{Q}_F \} \right) \\
& \text{s.t.} \begin{cases} \mathbf{z} = \mathbf{0}; & \mathbf{z} = \frac{d\mathbf{C}}{dt} + \frac{\kappa}{h} \mathbf{C} \\ \mathbf{w} \geq \mathbf{0}; & \mathbf{w} = \frac{d\mathbf{D}}{dt} + \frac{\kappa}{h} \mathbf{D} \end{cases} \quad (2.9)
\end{aligned}$$

Dimostrazione. Detta Ψ la funzione lagrangiana di (2.9), le condizioni di Karush-Kuhn-Tucker del prim'ordine per la soluzione di tale programma quadratico sono: $d\Psi/d\dot{\mathbf{q}}^{l+1} = \mathbf{0}$, $d\Psi/d\boldsymbol{\gamma}_C = \mathbf{0}$, $d\Psi/d\boldsymbol{\gamma}_D \geq \mathbf{0}$, $\boldsymbol{\gamma}_D \geq \mathbf{0}$, $\boldsymbol{\gamma}_{D_i} (d\Psi/d\boldsymbol{\gamma}_{D_i}) = 0$. Le ultime tre corrispondono alla condizione di complementarità $\boldsymbol{\alpha} \geq \mathbf{0} \perp \boldsymbol{\gamma}_D \geq \mathbf{0}$, pertanto svolgendo le derivate si ottiene il problema MLCP (2.6), che per il Teorema 1 si può sempre ricondurre all'LCP canonico (2.7). Infine, dato che $[\mathbf{M}]$ è definita positiva, il problema (2.9) è un QP convesso.

TEOREMA 2. *La formulazione impulsiva di qualsiasi problema multibody contenente vincoli monolateri e bilateri, come nella forma MLCP (2.6), ha una soluzione unica e calcolabile attraverso un numero finito di passi con un algoritmo del simplesso.*

Dimostrazione. Il Teorema 1 dimostra che la formulazione (2.6) è equivalente al LCP (2.7). Il Corollario 1.1 mostra che la matrice $[\mathbf{A}]$ di tale LCP è sicuramente semidefinita positiva, sotto l'ipotesi che la matrice di massa $[\mathbf{M}]$ sia definita positiva (ciò è sempre verificato nei problemi reali).

Inoltre, se $[\mathbf{A}]$ è semidefinita positiva, è $\mathbf{y}^T [\mathbf{A}] \mathbf{y} \geq 0 \forall \mathbf{y} \in \mathbf{R}^n$, sicuramente è anche copositiva (ovvero $\mathbf{y}^T [\mathbf{A}] \mathbf{y} \geq 0 \forall \mathbf{y} \geq \mathbf{0}$).

Sotto l'ipotesi di $[\mathbf{A}]$ copositiva e di vincoli realizzabili, è noto [Murty 1983] che esiste una singola soluzione al problema LCP, e che questa può essere ottenuta in un numero finito di passi da un algoritmo del simplesso.

3. SOLUZIONE DEL PROBLEMA DI COMPLEMENTARITA'

La soluzione di problemi di complementarità lineare, quale quello implicato da sistemi multibody con molteplici vincoli unilateri, comporta una difficoltà nettamente superiore al caso di soli vincoli bilateri. Infatti, se fossero assenti le disequazioni nel modello (2.6), questo corrisponderebbe ad un semplice sistema lineare, la cui soluzione potrebbe essere ottenuta in tempo polinomiale (ad esempio $O(n^3)$ in caso di decomposizione di Gauss, di classe *P-TIME*). Tuttavia proprio la presenza delle condizioni di complementarità causate dai vincoli unilateri a rigore impone che, con tecnica esaustiva, si debbano provare tutte le possibili combinazioni di vincoli attivi/inattivi, risolvendo il sistema lineare per ogni combinazione e, ogni volta, verificando sia che le reazioni $\boldsymbol{\gamma}_D$ dei vincoli attivi non siano negative, sia che per i vincoli inattivi non sia $\boldsymbol{\alpha}$ minore di zero. Ovviamente tale approccio esaustivo non può essere adottato in pratica, data la complessità esponenziale del metodo (di classe *EXPTIME*), per il quale si avrebbero tempi di calcolo inaccettabili anche con poche decine di vincoli unilateri.

Al fine di ridurre i tempi di calcolo, sono stati studiati diversi metodi, riassumibili in tre categorie.

Nella categoria più ampia si trovano i metodi pivot o *del simplesso*, ed è il caso di quello proposto in queste pagine. Essi percorrono vertici e spigoli del simplesso, attivando/disattivando i relativi vincoli unilateri e risolvendo ogni volta un sistema lineare coi soli vincoli attivi. Così facendo, è possibile raggiungere il vertice di minimo locale dopo varie iterazioni. Tali metodi, sebbene aventi classe di complessità *NP-hard* nel caso generale, si comportano piuttosto bene in casi pratici, richiedendo un numero di iterazioni solitamente accettabile.

Una seconda categoria include i metodi *di punto interno*. Tali algoritmi traggono spunto dalla notevole scoperta [Khachiyan, 1979] che il problema di programmazione lineare a matrici convesse non è di classe *NP-hard*, bensì di classe polinomiale. I recenti metodi di punto interno raggiungono $O(nL)$ [Karmarkar,1984], [Wright,1996] perciò possono risolvere problemi di complementarità ad elevate dimensioni. Tuttavia qui non si è fatto uso di un metodo di punto interno per due ragioni: l'elevata sofisticazione nell'implementazione¹ e la difficoltà nello sfruttare la sparsità delle matrici jacobiane dei vincoli.

Una terza recente categoria comprende i metodi LCP *iterativi*, quali il Gauss-Siedel proiettivo, tuttora poco conosciuti. Questi ultimi promettono di risolvere problemi di elevatissime dimensioni con complessità strettamente lineare, a fronte di una precisione inferiore nel soddisfacimento dei vincoli e delle complementarità. Vista la necessità di precisione nel contesto della presente ricerca, rimandiamo lo studio e l'implementazione dei metodi iterativi a futuri studi.

3.1. Descrizione del metodo del simplesso.

Il metodo del simplesso qui proposto si ispira ai classici algoritmi di Dantzig/Lemke, e ne rende più efficiente l'esecuzione attraverso l'introduzione di alcuni accorgimenti.

Lo stato dell'arte nell'ambito dei metodi del simplesso per problemi multibody è l'algoritmo di Baraff [Baraff,1994]. Tale metodo è una riedizione del metodo di Dantzig, adattato alle esigenze della simulazione multibody, tuttavia il metodo di Baraff prevede la costruzione di una matrice densa, quale la matrice $[A]$ per il problema LCP nella forma (2.7). Nel nostro caso, invece, il sistema LCP non è trattato nella forma canonica: la matrice $[A]$ non viene mai calcolata e si fa riferimento solo ed esclusivamente alla matrice del sistema MLCP (2.6). Quest'ultima matrice è sicuramente sparsa, vista l'usuale sparsità delle matrici jacobiane $[C_q]$ e $[D_q]$ che caratterizzano i problemi multibody.

Il metodo qui proposto consiste nella soluzione ripetuta più volte di un sistema lineare del tipo

$$\begin{bmatrix} [M] & [C_q] & [D_q] \\ [C_q] & [0] & [0] \\ [D_q] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{q}}^{l+1} \\ -\underline{\gamma}_C \\ -\underline{\gamma}_D \end{Bmatrix} = \begin{Bmatrix} [M]\dot{\mathbf{q}}^l + h\mathbf{Q}_F \\ -\frac{K}{h}\mathbf{C}(\mathbf{q},t) - \frac{\partial \mathbf{C}}{\partial t} \\ -\frac{K}{h}\mathbf{D}(\mathbf{q}) \end{Bmatrix} \quad (3.1)$$

in cui compaiono solamente i vincoli unilateri 'attivi', ovvero quelli in cui avviene il contatto fra le parti, e per i quali è $\underline{D}=\mathbf{0}$. In tal caso $[D_q]$ è una porzione dello jacobiano $[D_q]$, in cui sono state rimosse le righe corrispondenti ai vincoli non attivi.

¹ I metodi di punto interno dipendono da un numero elevato di tarature e regolazioni, la cui accorta (ma non semplice) scelta condiziona fortemente l'efficienza e la precisione.

La prima iterazione viene eseguita senza considerare i vincoli unilateri. Nelle iterazioni successive, tali vincoli vengono attivati uno dopo l'altro nel caso in cui vi sia una violazione (ovvero $\alpha_i < 0$), aumentando gradualmente il numero di righe dello jacobiano $[\mathbf{D}_q]$ ed il numero di moltiplicatori lagrangiani corrispondenti $\boldsymbol{\gamma}_b$. Contemporaneamente, iterazione dopo iterazione, possono essere rimossi vincoli unilateri per i quali si verifichi una reazione diretta nel verso non consentito (ovvero $\gamma_i < 0$); tale evento riduce il numero di righe dello jacobiano.

Al termine di ogni iterazione può accadere che si trovi più di un vincolo con violazione ($\alpha_i < 0$): nell'iterazione successiva verrà attivato solo il vincolo che, lungo la direzione di esplorazione del semplice, manifesta *per primo* il fenomeno della violazione (l'iterazione successiva parte con lo stato del sistema corrispondente a tale evento). Analogo criterio di precedenza viene impiegato per la disattivazione dei vincoli unilateri, se al termine di un'iterazione si trovano più vincoli con reazioni non consentite $\gamma_i < 0$.

Il metodo termina se, ad una determinata iterazione, nessun vincolo unilatero ha violazione o reazione negativa. L'algoritmo termina sempre in un numero finito di passi, per effetto del Teorema 2.

3.2. Implementazione.

Un primo accorgimento utile consiste nell'evitare di costruire matrici $[\mathbf{D}_q]$ ad ogni iterazione, poiché per disattivare un vincolo non è necessario rimuoverlo dalla matrice del sistema lineare: si è sperimentato che è sufficiente inserire un valore molto elevato (es. 10^{20}) sulla diagonale del sistema lineare completo, in modo che tale pivot comporti un valore nullo per il moltiplicatore lagrangiano corrispondente (mentre l'effetto dell'equazione così disattivata sul restante sistema è impercettibile). In pratica, ad ogni iterazione, si risolve il seguente sistema lineare, con $[\mathbf{P}]$ matrice avente sulla diagonale numeri elevati (es. 10^{20}) in corrispondenza dei vincoli disattivati, e 0 in corrispondenza dei vincoli attivi:

$$\begin{bmatrix} [\mathbf{M}] & [\mathbf{C}_q] & [\mathbf{D}_q] \\ [\mathbf{C}_q] & [0] & [0] \\ [\mathbf{D}_q] & [0] & [\mathbf{P}] \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}^{l+1} \\ -\boldsymbol{\gamma}_C \\ -\boldsymbol{\gamma}_D \end{bmatrix} = \begin{bmatrix} [\mathbf{M}]\dot{\mathbf{q}}^l + h\mathbf{Q}_F \\ -\frac{K}{h}\mathbf{C}(\mathbf{q}, t) - \frac{\partial \mathbf{C}}{\partial t} \\ -\frac{K}{h}\mathbf{D}(\mathbf{q}) \end{bmatrix} \quad (3.2)$$

Si osservi inoltre che il sistema lineare (3.2), del tipo $[\mathbf{S}]\mathbf{x}=\mathbf{r}$, è simmetrico: per la risoluzione conviene pertanto adottare un procedimento che, a differenza del metodo di Gauss, possa sfruttare questa proprietà. In particolare si è ritenuto utile usare una variante della decomposizione di Bunch-Kaufman descritta in [Tasora, 2001]. Tale decomposizione fattorizza la matrice dei coefficienti $[\mathbf{S}]$ nella forma $[\mathbf{S}]=[\mathbf{L}][\mathbf{D}][\mathbf{L}]^T$, con $[\mathbf{L}]$ triangolare. Per sfruttare la sparsità, le matrici vengono memorizzate con un particolare metodo orientato alla massima efficienza computazionale ed alla minore occupazione di memoria. Data la sparsità di $[\mathbf{S}]$, anche $[\mathbf{L}]$ contiene numerosi zeri strutturali. Per meccanismi con topologia aciclica, si è dimostrato [Tasora, 2001] che tale decomposizione ha complessità lineare ed è calcolata in $O(n)$ per n vincoli, pertanto è molto più efficiente rispetto alla decomposizione di Gauss $O(n^3)$.

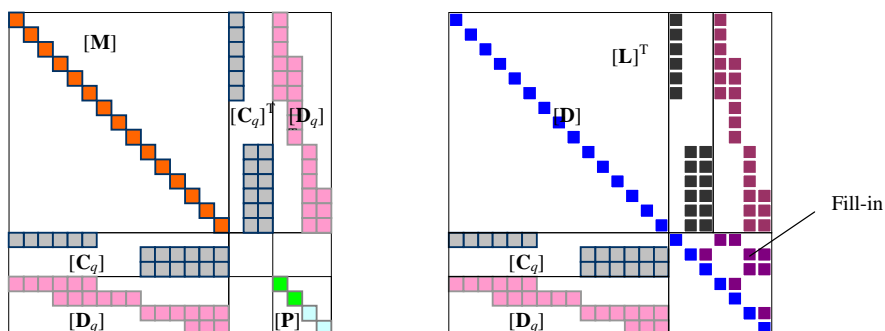


Figura 2: Esempio di matrice dei coefficienti, a sinistra prima della fattorizzazione, e a destra dopo la fattorizzazione LDL in-situ. Si noti l'esiguo numero di fill-in ed il mantenimento della sparsità.

La decomposizione avviene *in-situ*: la matrice $[\mathbf{L}]^T$ è sovrascritta alla metà superiore della matrice $[\mathbf{S}]$, assumendo $L_{j,j}=1$, mentre $[\mathbf{D}]$ viene sovrascritta alla diagonale (Figura 2). La metà inferiore di $[\mathbf{S}]$ non viene alterata: serve principalmente per avere a disposizione lo jacobiano $[\mathbf{D}_q]$ quando, ad ogni iterazione del metodo del sempliceo, si calcola la violazione dei vincoli unilateri, ovvero dalla (2.6),

$$\boldsymbol{\alpha} = [\mathbf{D}_q] \dot{\mathbf{q}}^{l+1} + (K/h) \mathbf{D}(\mathbf{q}) \quad (3.3)$$

Sebbene il metodo di fattorizzazione per matrici sparse qui introdotto presenti una complessità che nella maggior parte dei casi si avvicina al caso ideale $O(n)$, è pur vero che l'algoritmo del sempliceo richiede la soluzione del sistema per un numero elevato di volte: nel caso più fortunato il sistema (3.2) deve essere risolto almeno m volte, detto m il numero di vincoli unilateri, ma in casi critici il tempo di calcolo può salire a valori inaccettabili.

Si è pertanto introdotta un'ulteriore ottimizzazione, con notevole abbattimento del numero di operazioni. Il principio ispiratore è il seguente: ad ogni iterazione del metodo del sempliceo, nel sistema (3.2) cambia solo l'attivazione/disattivazione di una singola equazione di vincolo, pertanto la

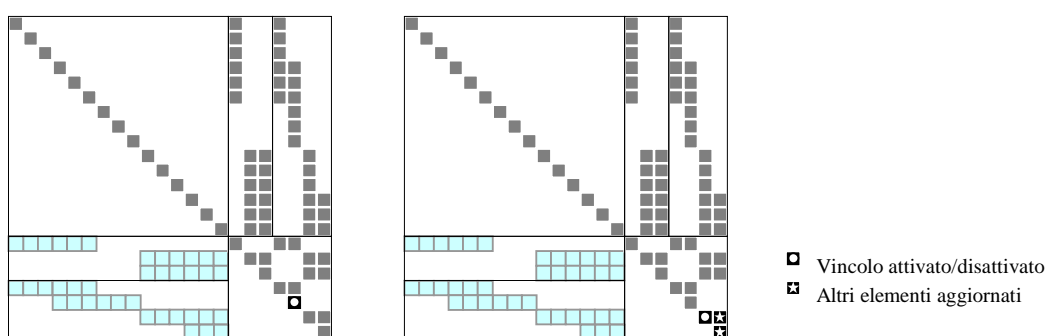


Figura 3: Decomposizione incrementale: attivazione/disattivazione di equazioni di vincolo. Nell'esempio di sinistra viene cambiato il secondo dei vincoli unilateri, mentre nell'esempio di destra viene cambiato il terzo.

decomposizione della matrice dei coefficienti $[S]$ può essere calcolata semplicemente aggiornando la precedente decomposizione, coinvolgendo il minor numero di elementi (v. Figura 3).

Nella maggior parte dei casi, ogni qualvolta si disattivi un vincolo ponendo il rispettivo pivot a 10^{20} (o qualora si riattivi, ripristinando il pivot originale), è necessario aggiornare un numero abbastanza esiguo di elementi. Detto m il numero dei vincoli unilateri, si noti come gli elementi ricalcolati siano in ogni caso confinati entro la porzione $m \times m$, in basso a destra nella matrice dei coefficienti.

In letteratura è già noto un tipo di decomposizione LU incrementale (fattorizzazione di Golub-Bartels) che tuttavia presenta il notevole inconveniente, rispetto a quella descritta nel presente lavoro, di alterare la simmetria e la sparsità della matrice dei coefficienti.

3.2. Pseudocodice.

Si riporta, in forma di pseudocodice volutamente semplificato, lo schema dell'algoritmo. Il compito più oneroso è svolto dalla funzione che calcola la soluzione del sistema (3.2) attraverso la decomposizione LDL incrementale.

Risolvi_LCP ($[S]$, B , x)

Disattiva tutti i vincoli,

Calcola \dot{q} , γ_C , γ_D risolvendo il sistema (3.2),

while

 Calcola i residui α usando la (3.3).

 Cerca un qualsiasi k tale che il vincolo k sia disattivato e che sia $\alpha_k < 0$

if tutti i vincoli hanno $\alpha_k \geq 0$

break

 Attiva il vincolo k -esimo

while

 Calcola \dot{q}^{NEW} , γ_C^{NEW} , γ_D^{NEW} risolvendo il sistema (3.2) con solutore LDL incrementale

 Calcola i residui α^{NEW} usando la (3.3).

$[s_{step}, i_{changed}] = \text{Calcola_step_max}(\gamma, \alpha, \gamma^{NEW}, \alpha^{NEW})$

$\alpha := (1 - s_{step}) \alpha + (s_{step}) \alpha^{NEW}$

$\dot{q} := (1 - s_{step}) \dot{q} + (s_{step}) \dot{q}^{NEW}$

$\gamma_C := (1 - s_{step}) \gamma_C + (s_{step}) \gamma_C^{NEW}$

$\gamma_D := (1 - s_{step}) \gamma_D + (s_{step}) \gamma_D^{NEW}$

if $s_{step} = 1$

break

if vincolo $i_{changed}$ -esimo attualmente disattivato

 attiva vincolo $i_{changed}$ -esimo,

else if vincolo $i_{changed}$ -esimo attualmente attivato

 disattiva vincolo $i_{changed}$ -esimo,

if troppe iterazioni

break

if troppe iterazioni

break

L'algoritmo impiega la funzione **Calcola_step_max** per ottenere l'entità dell'avanzamento lungo i lati del semplice, nonché per valutare quale vincolo debba essere attivato (o disattivato):

```
[  $s_{step}$ ,  $i_{changed}$  ] = Calcola_step_max (  $\gamma$ ,  $\alpha$ ,  $\gamma^{NEW}$ ,  $\alpha^{NEW}$  )
```

```

 $s_{step}$  = 1 ;       $i_{changed}$  = -1 ;
for  $i = 1 \dots m$ 
  if vincolo  $i$ -esimo è attivo
    if  $\gamma_i^{NEW} < 0$ 
       $s_{test} = - \gamma_i / (\gamma_i^{NEW} - \gamma_i)$ 
      if (  $s_{test} < s_{step}$  )
         $s_{step} = s_{test}$  ;
         $i_{changed} = i$  ;
  for  $i = 1 \dots m$ 
    if vincolo  $i$ -esimo è disattivato
      if  $\alpha_i^{NEW} < 0$ 
         $s_{test} = - \alpha_i / (\alpha_i^{NEW} - \alpha_i)$ 
        if (  $s_{test} < s_{step}$  )
           $s_{step} = s_{test}$  ;
           $i_{changed} = i$  ;
if  $s_{step} \leq 0$  oppure  $s_{step} \geq 1$ 
  errore, vincoli mal condizionati o LCP non risolvibile

```

L'introduzione dell'attrito statico nei problemi 3D comporta un passaggio al problema MNLCP (complementarità mista non-lineare) che tuttavia può essere risolto dall'algoritmo con alcune modifiche, ad esempio linearizzando i coni d'attrito ed approssimandoli a piramidi [Anitescu, 2002].

4. ESEMPI

Il metodo del semplice descritto in queste pagine è stato implementato nel software multibody CHRONO ed è stato testato con vari problemi pratici, tipicamente simulazioni di contatti simultanei fra numerosi corpi rigidi. Le ottimizzazioni introdotte hanno permesso un guadagno, in termini di efficienza computazionale, di almeno un ordine di grandezza. L'esempio di Figura 4 mostra una simulazione in cui i numerosi contatti fra le sfere comportano la presenza di centinaia di vincoli unilateri. L'esempio di Figura 5 mostra la simulazione di una macchina per il packaging: numerosi vincoli unilateri sono presenti per effetto dei contatti fra le superfici delle bottiglie.

5. CONCLUSIONE

L'analisi di problemi multibody con vincoli unilateri comporta la soluzione di problemi di complementarità non lineare mista. Nella presente memoria si è descritta una variante del metodo del semplice di Dantzig in grado di sfruttare la sparsità delle matrici di massa e delle matrici jacobiane. Il nuovo metodo è in grado di calcolare le accelerazioni o gli impulsi incogniti con poche iterazioni di complessità al più $O(n)$, in luogo della complessità $O(n^3)$ solitamente richiesta nelle iterazioni del metodo originale.

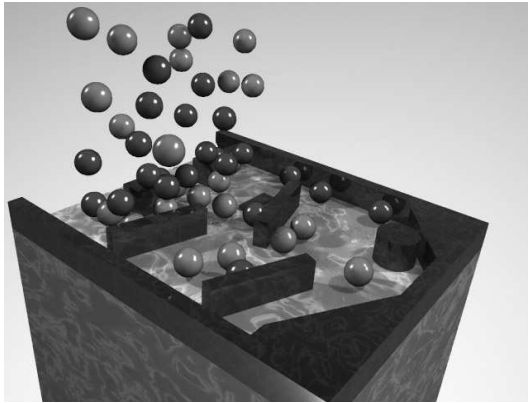


Figura 4: cascata di corpi rigidi



Figura 5: simulazione dinamica con numerosi contatti

Il nuovo algoritmo è stato implementato nel codice multibody CHRONO ove ha esibito doti di efficienza (ovvero elevata velocità di calcolo) e di robustezza, mostrandosi in grado di fornire risultati attendibili anche in presenza di problemi mal condizionati o patologici.

References:

- Karmarkar , N., (1984), *A new polynomial time algorithm for linear programming*, *Combinatorica* 4, N.4, 373-395, 1984
- Wright,S.J, (1996), *Primal-Dual Interior-Point Methods*, SIAM, 1996.
- Anitescu, M., Potra, F.A., (2002), *Time-stepping schemes for stiff multi-rigid-body dynamics with contact and friction*. *International Journal for Numerical Methods in Engineering*, 55(7):753–784
- Anitescu, M., Potra, F.A., (1997), *Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems*. *Nonlinear Dynamics*, 14:231–247, 1997.
- Baraff, D., (1993), *Issues in computing contact forces for non-penetrating rigid bodies*, *Algorithmica*, 10:292–352, 1993.
- Baraff, D., (1994), *Fast contact force computation for nonpenetrating rigid bodies*. *Computer Graphics (Proc. SIGGRAPH)*, 28:23–34, 1994.
- Tasora, A., (2001) *An optimized lagrangian-multiplier approach for interactive multibody simulation in kinematic and dynamical digital prototyping*, VIII International Symposium on Computer Simulation in Biomechanics, 4-6 July 2001, Politecnico di Milano, Italy.
- Filippov, A. F., (1964) *Differential equations with discontinuous righthand side*, *Amer. Math. Soc. Translations*, vol. 42, ser. 2, pp.199-231,1964.
- Shabana, A., (1989), *Multibody systems*, John Wiley & Sons, New York 1989.
- Murty, K.G., (1983), *Linear programming*, John Wiley & Sons, New York, 1983.
- Todd M.J., (1976) *Extensions of Lemke’s Algorithm for the Linear Complementarity Problem*, *Journal of Optimization Theory and Applications*, vol. 20, 1976, pp. 397-416.